

# A2 Coursework Report

Keying Song

4th April

**Word Count:** 2,997 words

## Abstract

This coursework presents the implementation and analysis of medical image processing across three modules: PET-CT reconstruction, MRI denoising, and CT-based lung nodule segmentation and classification. For PET-CT, several reconstruction algorithms (e.g., FBP, OS-SART, SIRT, OSEM, MLEM) and a bilinear HU transformation for attenuation correction [1] were evaluated. In MRI denoising, 3D knee images acquired from six coils were processed; two coils were found faulty. Both image-space and k-space filtering methods were applied to enhance image quality. For CT data from 40 lung nodules [2], an adaptive threshold segmentation algorithm with optional boundary refinement was developed, improving the average Dice score from 0.05 (Otsu) to 0.55. Finally, three radiomic features were extracted and analysed for benign versus malignant classification.

## Contents

<b>1 Module 1: PET-CT Image Reconstruction</b>	ii
1.1 Exercise 1.1 . . . . .	ii
1.2 Exercise 1.2 . . . . .	ii
1.3 Exercise 1.3 . . . . .	iii
1.4 Exercise 1.4 . . . . .	iv
1.5 Exercise 1.5 . . . . .	vi
<b>2 Module 2: MRI Image Denoising</b>	vii
2.1 Exercise 2.1: Visualisation and identifying noise . . . . .	vii
2.2 Exercise 2.2: Removing noise . . . . .	ix
<b>3 Module 3: CT Image segmentation and classification</b>	xiii
3.1 Exercise 3.1: Image segmentation . . . . .	xiii
3.2 Exercise 3.2: Image feature extraction and classification . . . . .	xv
<b>References</b>	xviii

# 1) Module 1: PET-CT Image Reconstruction

## 1.1 Exercise 1.1

### Problem Statement

Clean up the sinograms ("ct\_sinogram.npy" and "pet\_sinogram.npy") with given measurements ("ct\_dark.npy", "ct\_flat.npy" and "pet\_calibration.npy") before reconstruction [3]

For CT sinogram, the intensity of pixels are the negative log-ratio of the detected intensities and emitted intensities, which can be obtained from the dark field correction and flat field correction:

$$p = -\log \frac{I}{I_0} = -\log \frac{sino - dark}{flat - dark} \quad (1.1)$$

Where *sino*, *flat* and *dark* above represents the given data "ct\_sinogram.npy", "ct\_flat.npy" and "ct\_dark.npy" respectively.

For PET initial correction, "pet\_calibration.npy" was used to compensate for detector gain calibration, containing sensitivity factors for each detector element, accounting for variations in detector efficiency. The PET sinogram was simply corrected by:

$$PET = \frac{PET_{sino}}{calibration} \quad (1.2)$$

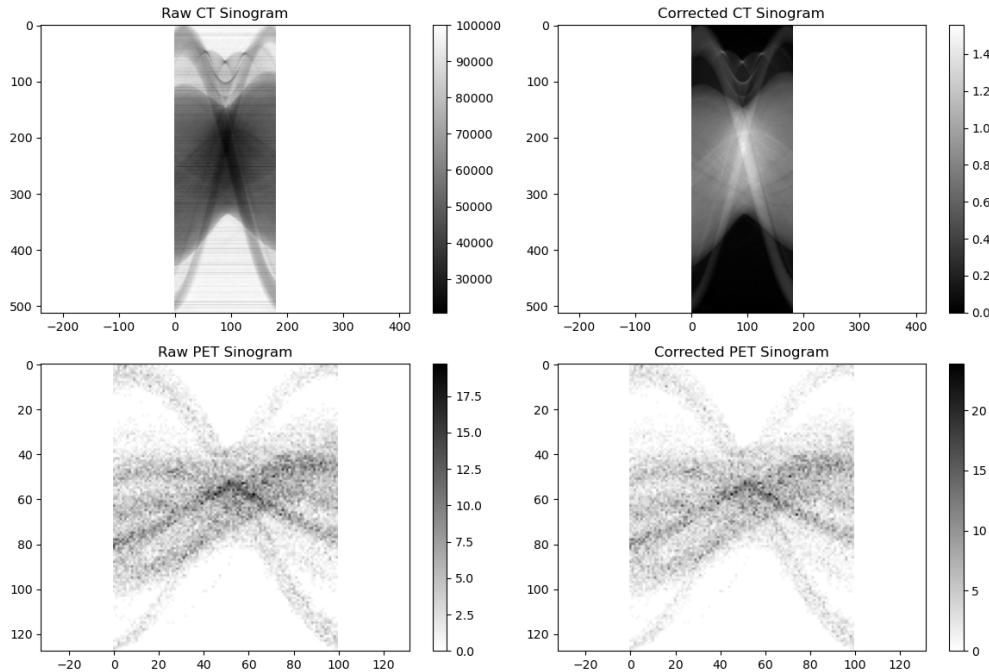


Figure 1: The initial corrected PET and CT sinograms

## 1.2 Exercise 1.2

### Problem Statement

Reconstruct the CT image with FBP and OS-SART. Find the appropriate values of  $\gamma$  and  $K = \max(k)$  that produces a better image than FBP. [5] Extra: How does it compare to SIRT? Does it reach a good result faster or slower? More or less iterations?

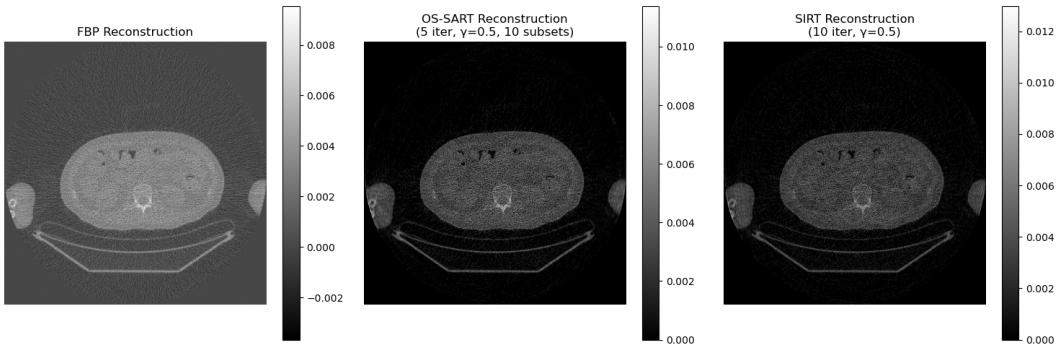


Figure 2: The CT reconstruction with FBP, OS-SART and SIRT.

CT reconstruction using FBP offers a fast baseline but suffers from noise amplification. OS-SART ( $\gamma = 0.5$ ,  $K = 10$ ) yields smoother image with reduced noise while preserving anatomical details. By processing data in subsets, it achieves faster convergence than standard SART.

Compared to SIRT, OS-SART reaches comparable image quality with fewer iterations-requiring only 5 iterations versus SIRT's 10-thus offering greater computational efficiency. (See `run_module1.ipynb` for timing and metric details.)

### 1.3 Exercise 1.3

#### Problem Statement

Resize the CT image to the PET size, produce a sinogram, and attenuation correct the PET sinogram. [3]

To perform PET attenuation correction using the CT data, the CT image reconstructed from the previous steps was first resized to match the dimensions of the PET image (from  $512 \times 512$  pixels to  $128 \times 128$  pixels). This resizing was performed using cubic interpolation (order=3) inside the `resize()` function of `skimage` library. The main implementations are displayed in the flow chart 3:

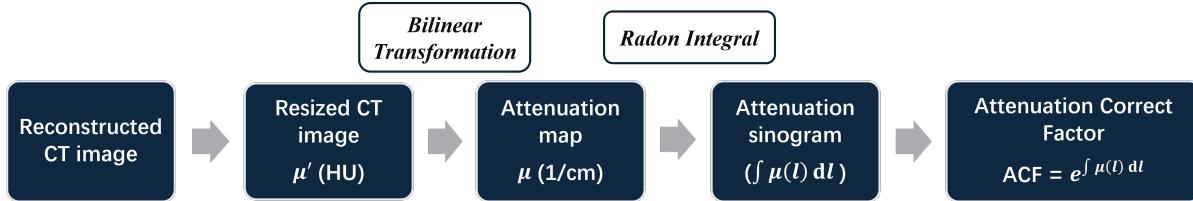


Figure 3: The flow chart of PET attenuation correction using CT image

However, the pixel values  $\mu'$  in CT images are in Hounsfield units (HU), defined as [3]:

$$HU = 1000 \times \frac{\mu' - \mu'_{water}}{\mu'_{water} - \mu'_{air}} \quad (1.3)$$

which differ from the attenuation coefficient  $\mu$  at photon energy 511 keV (PET rays), since  $\mu$  are energy dependent. To handle this, a bilinear transformation method described in [1] were employed. I assume that the CT data provided is under X-rays energy around 80 keV, so that the bilinear fitting results shown in Table 3 in [1] can be used with equation 2 in [1]:

$$\begin{aligned} &\text{Air/lung } (HU < -950): \mu = 0 \\ &\text{Soft tissue } (-950 \leq HU \leq 50): \mu = 9.6 \times 10^{-5} \cdot (HU + 1000) \text{ cm}^{-1} \\ &\text{Bone } (HU > 50): \mu = 3.64 \times 10^{-5} \cdot (HU + 1000) + 6.26 \times 10^{-2} \text{ cm}^{-1} \end{aligned} \quad (1.4)$$

Subsequently, an attenuation sinogram was generated by forward-projecting the attenuation map, which is equivalent to the radon transform (actually a line integral at each angle) described in E.q. 2.5 in [4]. Finally, The PET sinogram was then corrected by multiplying with the exponential of the attenuation sinogram:

$$PET_{\text{atte\_corrected}} = PET_{\text{sinogram}} \times e^{\int \mu(l)dl} \quad (1.5)$$

This correction compensates for the varying levels of photon attenuation in different tissues, particularly important for the PET reconstruction in the next step. Figure 4 shows the original CT image, the resized CT image, the derived attenuation map, and both the uncorrected and attenuation-corrected PET sinograms.

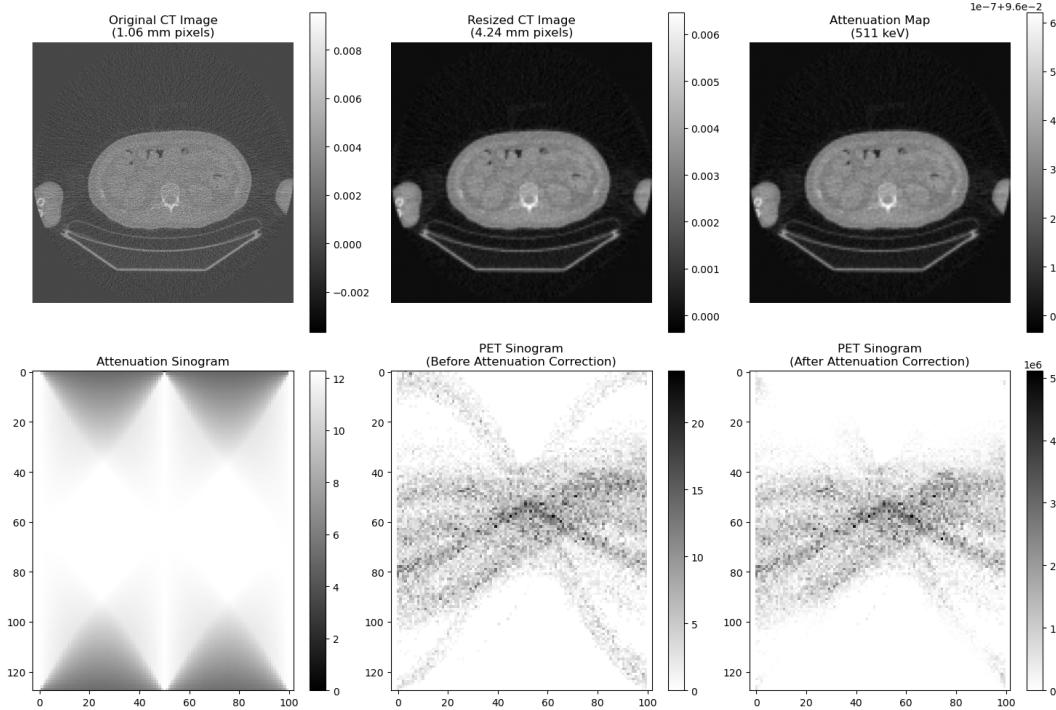


Figure 4: PET Attenuation correction using resized CT image

## 1.4 Exercise 1.4

### Problem Statement

Reconstruct the PET image with FBP and with OSEM [5]

Extra: How does OSEM compare to MLEM? Does it reach a good result faster or slower? Does it need more or less iterations?

The reconstructions using FBP, OSEM and MLEM are presented in Figure 5. The FBP reconstruction of the PET data shows significant noise and streak artifacts radiating from the high-uptake regions. While FBP is computationally efficient, it performs poorly with the relatively noisy nature of PET data. OSEM with 4 iterations and 5 subsets provides a dramatically improved image quality compared to FBP. The algorithm converges quickly while producing clear tracer regions with good contrast and significantly reduced noise.

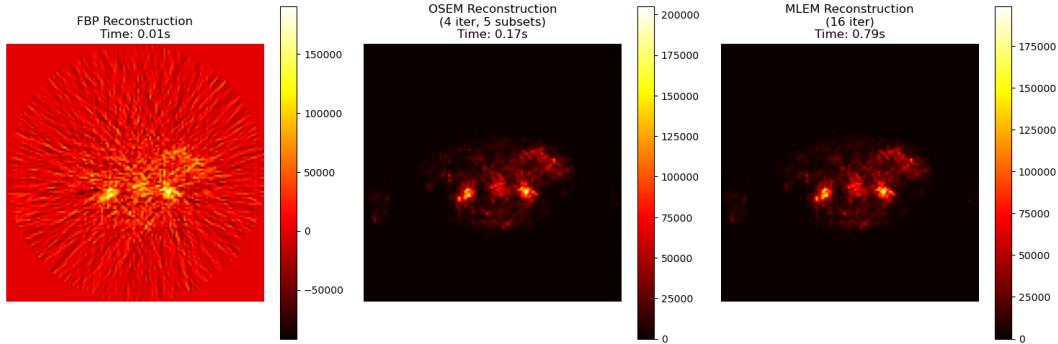


Figure 5: PET reconstructions with FBP, OSEM and MLEM.

In comparison to MLEM, OSEM achieves similar image quality with fewer iterations and time. The converge processes of OSEM and MLEM are shown in Figure 6 and 7. It can be seen that the OSEM converges approximately 4 times faster than MLEM, reaching a stable solution by iteration 4, while MLEM requires about 16 iterations to achieve comparable results.

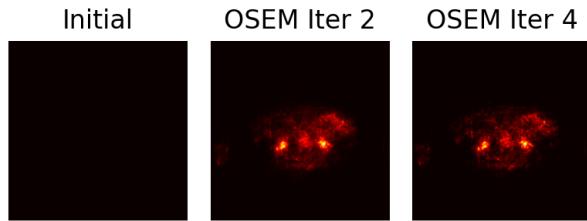


Figure 6: The converge process of OSEM.

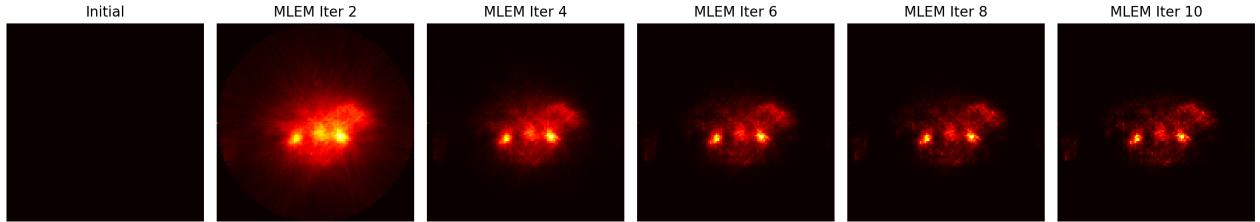


Figure 7: The converge process of MLEM.

Calculating the relative difference between intermediate results of both algorithms and their own final results, the convergence plot in Figure 8 clearly demonstrates this performance difference:

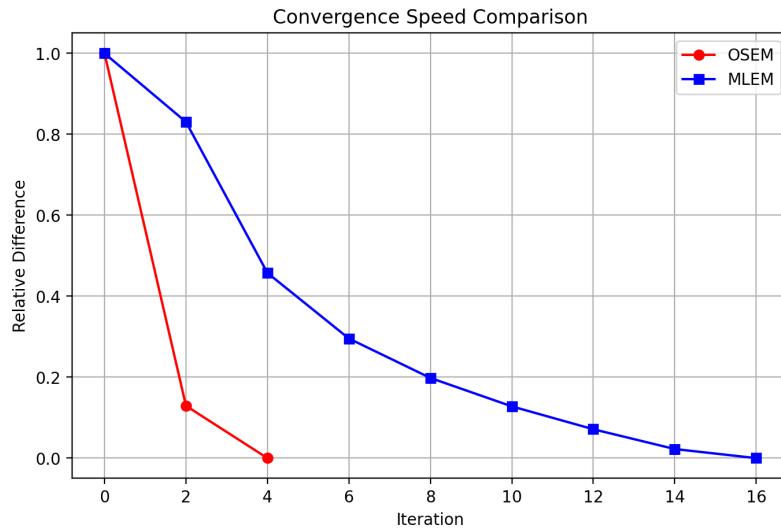


Figure 8: PET Attenuation correction using resized CT image

## 1.5 Exercise 1.5

- Display the PET and CT scans overlaid.

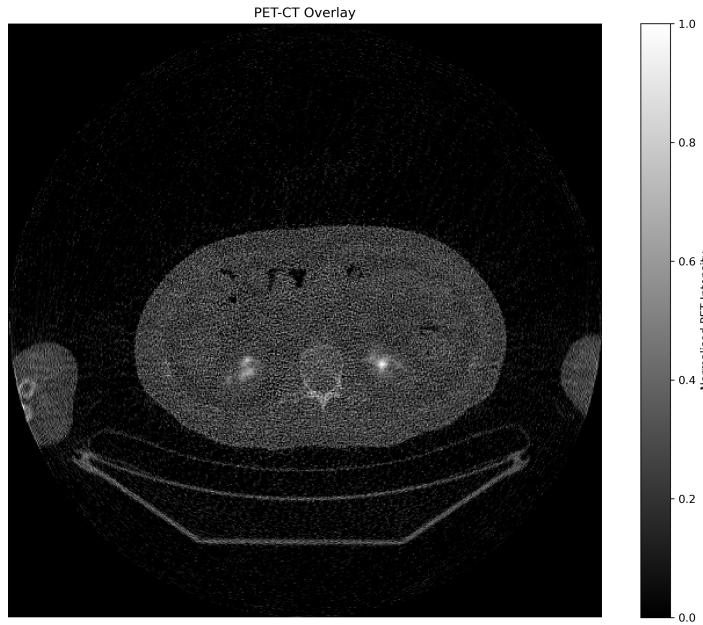


Figure 9: The overlaid PET-CT image

- If had a TOF-PET scanner, how accurate needed to be at measuring time to know the location of the emission so precisely that reconstruction would not be needed?

For TOF-PET to achieve such precise localisation, we may need timing resolution on the order of 30 picoseconds or better. The timing precision would need to match the desired spatial resolution (typically 2-3 mm), requiring approximately  $c\Delta t/2 = 3\text{mm}$ , or  $t \approx 20\text{ps}$ .

- Why is OSEM the most used algorithm in PET, but not in CT? What makes OSEM different (mathematically) to gradient descent?

PET data follows a Poisson distribution, which OSEM explicitly models, unlike log-transformed CT data. Unlike gradient descent, OSEM uses multiplicative updates, making it well-suited for non-negative PET emission data.

- PET-MR scanners exist. What is the extra data processing you need to take in a PET-MR that does not exist in a PET-CT? How is it done?

In PET-MR, attenuation correction is challenging because MRI does not directly reflect electron density, unlike the linear CT HU-attenuation relationship. Thus, tissue segmentation (e.g., air/lung, soft tissue, bone) is needed to assign attenuation coefficients for  $\mu$  map. A suitable model can be constructed, with parameters inferred via statistical methods such as Bayesian inference.

## 2) Module 2: MRI Image Denoising

### 2.1 Exercise 2.1: Visualisation and identifying noise

**Ex 2.1.1 - 2:** Load the complex data "kspace.npy" using the `np.load()` routine. Determine which array dimension is the coil dimension. [1] Create an image of the magnitude of k-space for each coil. [1]

The k-space data and its shape was loaded and printed. The array has dimensions (6, 280, 280), where the first dimension (size 6) is the coil dimension.

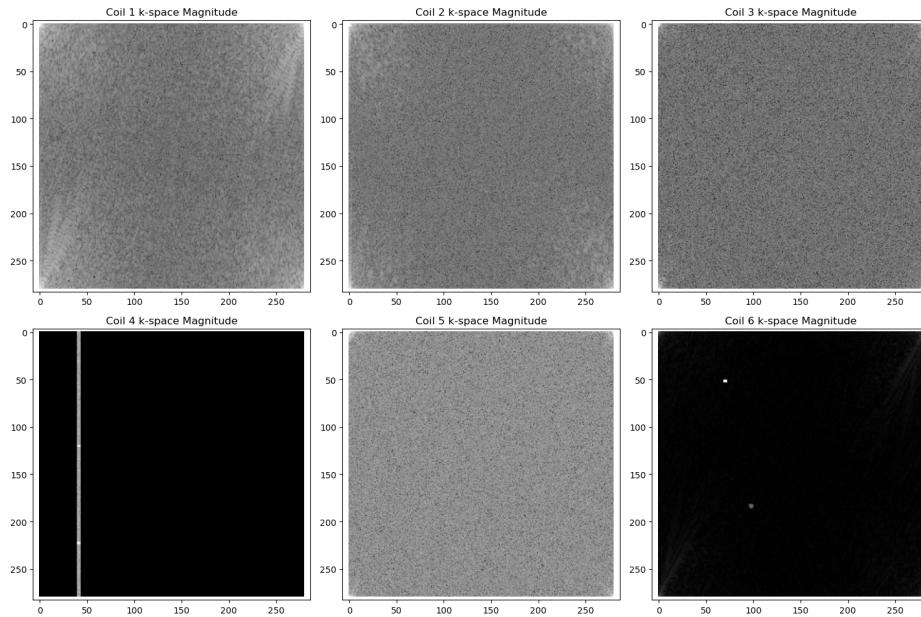


Figure 10: The magnitude images of six coils

Figure 10 displays the magnitude of k-space data for all six coils after applying `np.log1p()`. From these images, coils 1, 2, and 3 show well-structured k-space data with the expected central low brightness representing low spatial frequencies. Coils 4 and 6 appear predominantly dark, while coil 5 shows some signal but with an unusual distribution pattern. This initial presentation suggests that coils 4 and 6 may not have been functioning correctly during the scan or have extremely low sensitivity to signal emitting areas.

**Ex 2.1.3 - 4:** Transpose the data into image space using the Fourier transform. Create and show a magnitude and phase image from one coil. [1] Show the magnitude images from all coils. [1]

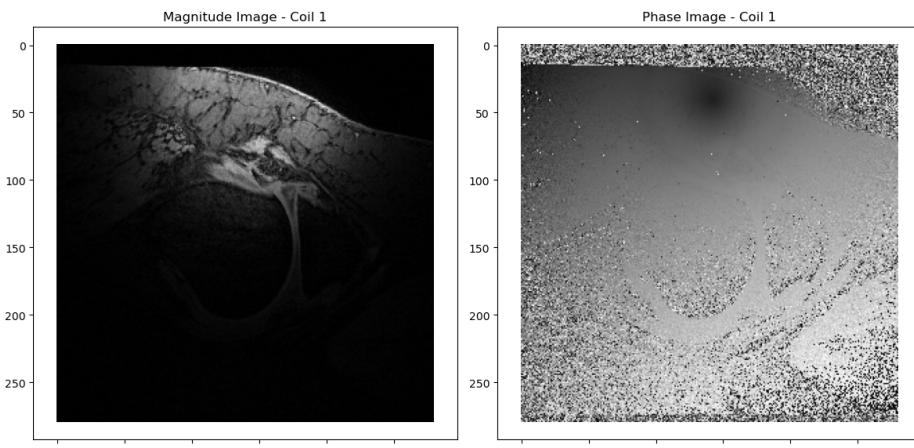


Figure 11: Magnitude and phase images from coil 1.

The data was transformed from k-space to image space using the 2D inverse Fourier transform (`np.fft.ifft2`). Figure 11 shows both the magnitude and phase images from coil 1, which has captured the data effectively.

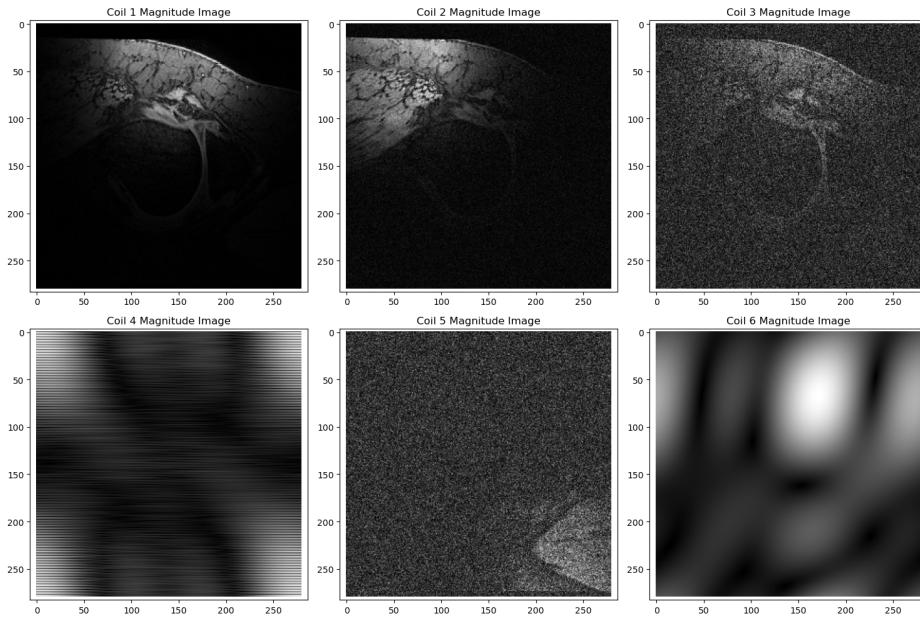


Figure 12: Magnitude of all coils in image space.

Figure 12 shows the magnitude images from all six coils after Fourier transformation. Coils 1, 2, and 3 provide good quality images with different sensitivity profiles typical of a phased array configuration, where each coil has maximum sensitivity to tissues closest to its position. In contrast, coils 4 and 6 produce extremely low signal images, so that they should be filtered out when combining.

**Ex 2.1.5 - 6:** Choose an appropriate way to combine the data from all of the coils into a single image. Show this image. [2] Write about your observations. [2]

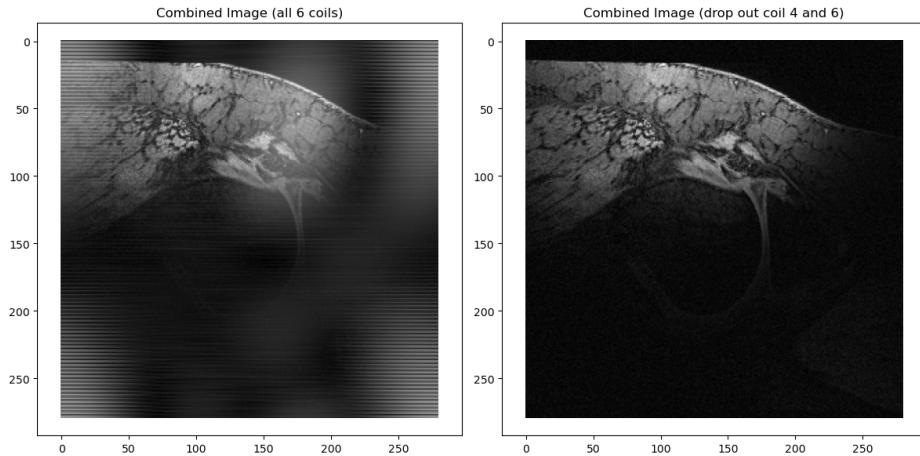


Figure 13: The combined image from all coils (left) and the combined image from all coils except for coils 4 and 6.

To combine the images, the squared magnitude of each coil image was calculated and summed across all coils, and then took the square root of the result, also followed by dividing by the number of coils used (for normalisation). Figure 13 shows two versions of the combined image: one using all coils and another excluding coils 4 and 6.

Coils 1, 2, and 3 produce good quality images with complementary sensitivity profiles, while coils 4 and 6 may fail almost completely, contributing only noise to the observation area. Coil 5 captures some signal but with poor SNR and distortion. The knee image is best visualised in the combined image excluding coils 4 and 6.

## 2.2 Exercise 2.2: Removing noise

**Ex 2.2.1:** Choose three denoising methods using the image space data, and show the effects of the denoising for all coils. Comment on the different denoising methods. [10]

- **Gaussian Filtering:** A Gaussian kernel ( $\sigma = 1.0$ , kernel size= $9 \times 9$ ) was applied to smooth the image data. As shown in Figures 15, Gaussian filtering effectively reduced the background noise in all coils but also blurred some anatomical boundaries, particularly noticeable in the well-functioning coils (1-3).
- **Bilateral Filtering:** This edge-preserving smoothing filter (spatial  $\sigma = 1$ , intensity  $\sigma = 0.1$ ) was applied to maintain sharp transitions at tissue boundaries while reducing noise in homogeneous regions. The bilateral filter performed particularly well on coils 1 and 2, where it preserved anatomical details better than Gaussian filtering while still effectively suppressing noise.
- **Wavelet Filtering:** BayesShrink, an adaptive approach of wavelet decomposition with soft thresholding was implemented. This technique performed well in preserving structural details while removing noise. As shown in the figures, wavelet denoising maintained sharper edges than Gaussian filtering and performed particularly well in areas with textural detail. However, it introduced more noise in regions with very low SNR (in coils 3 and 5).

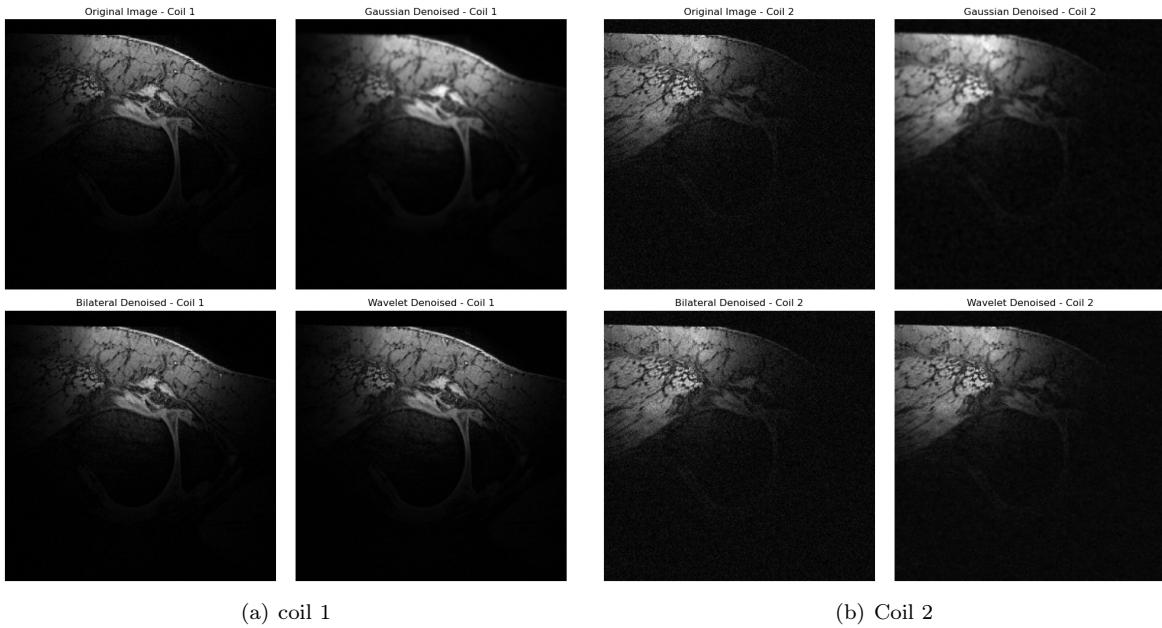


Figure 14: Filtering results of coil 1, 2. Each sub figure, from left to right and top to bottom are the original spatial magnitude, Gaussian denoised image, Bilateral denoised image, and Wavelet denoised image, respectively.

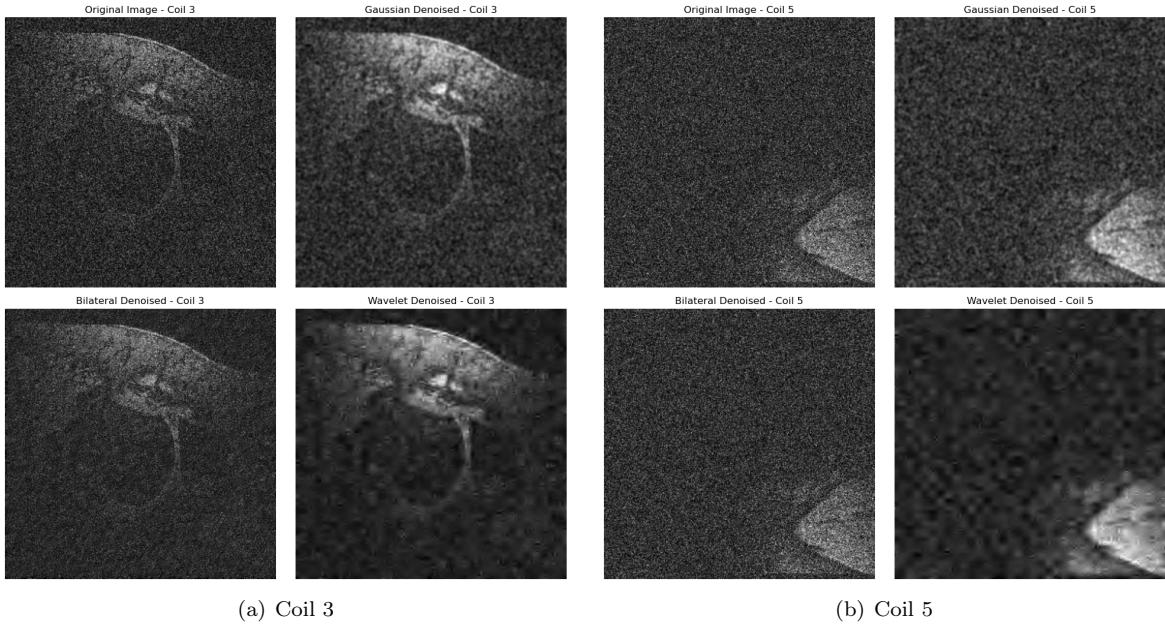


Figure 15: Filtering results of coil 3, 5. Each sub figure, from left to right and top to bottom are the original spatial magnitude, Gaussian denoised image, Bilateral denoised image, and Wavelet denoised image, respectively.

### Quantitative Filters Comparison

In order to quantitatively calculate the SNR for filters evaluation and comparison, the signal and noise mask was firstly drawn as shown in Figure 16.

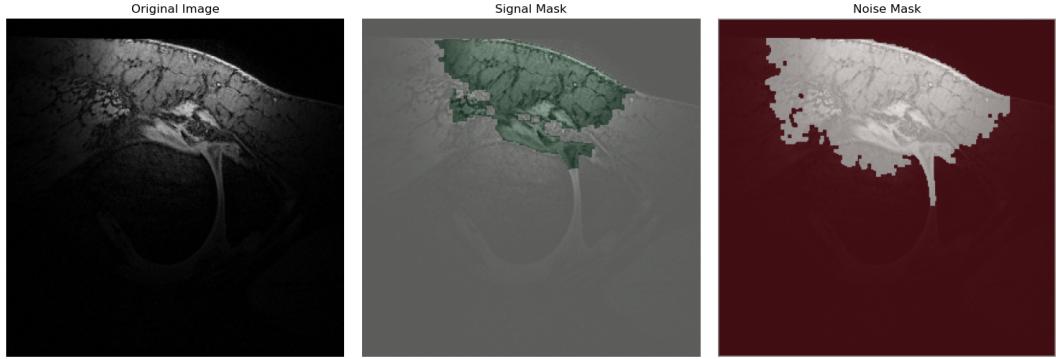


Figure 16: Signal (centre) and noise (right) mask of magnitude image in coil 1 (left).

Choosing coil 1 for metrics calculation, the quantitative denoising performance of each algorithm (including the Butterworth method) is listed in Table 1.

Method	MSE	PSNR	SSIM	Original_SNR	Denoised_SNR	SNR_Improve_ratio
Gaussian	2.97e-03	25.274857	0.916051	19.868162	20.256682	1.019253
Bilateral	3.94e-07	64.043229	0.999699	19.868162	19.896668	1.001435
Wavelet	3.38e-06	54.704843	0.996776	19.868162	20.324122	1.022949
Butterworth	7.42e-04	31.295146	0.982388	19.868162	19.595252	0.986264

Table 1: Denoising performance comparison of different filters.

It shows that **bilateral filtering** achieved the highest PSNR and SSIM, indicating superior preservation of image structure. However, **wavelet denoising** provided the best SNR improvement ratio, suggesting better noise reduction in terms of overall signal enhancement. Figure 17 provides a visual SNR comparison, the visualisation of PSNR and SSIM are placed in Figures 28 and 29 in the Appendix.

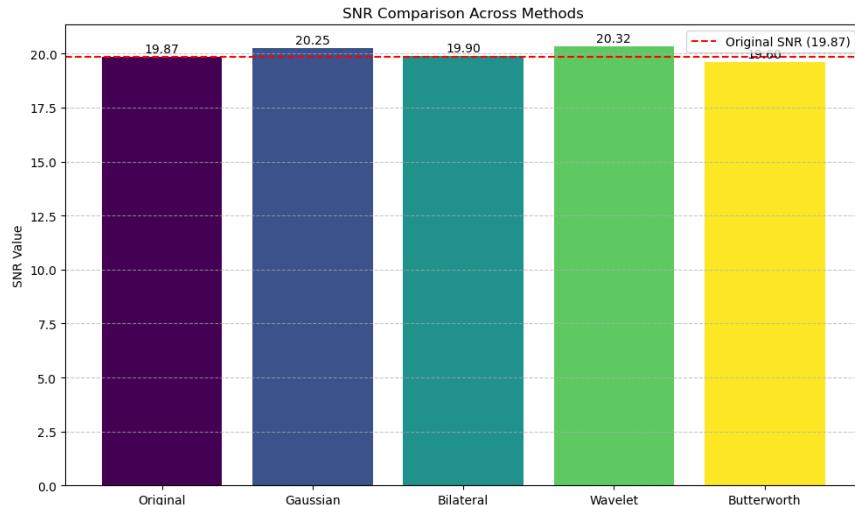


Figure 17: SNR comparison of four filtering results

**Ex 2.2.2:** For the first coil, apply a low-pass Butterworth filter within k-space and show both magnitude and phase new images. How does this method compare to the image-based methods? [6]

A low-pass Butterworth filter was applied directly in k-space. The filter with cutoff frequency at 220 and order  $n=2$  provided a smooth transition band rather than an abrupt cutoff. This approach targets high spatial frequencies containing noise while preserving the lower frequencies that contain most of the structure information. Figure 18 shows the results for coil 1, comparing the original and Butterworth-filtered images in both magnitude and phase.

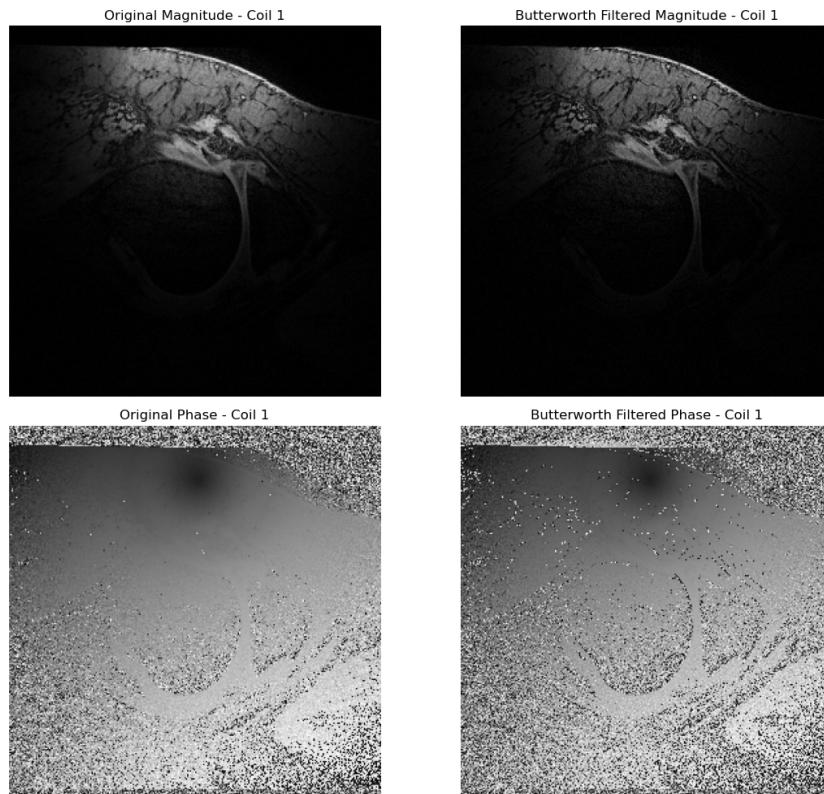


Figure 18: Low-pass Butterworth filtering results of coil 1

Compared to image-space methods, k-space filtering offers more direct control over spatial frequency, as well as preservation of the complex nature of the MRI signal. Besides, it allows more efficient computation for large datasets but less effective at addressing spatially varying noise.

**Ex 2.2.3:** Using one of denoising methods on the images, recreate a new combined image. [4]

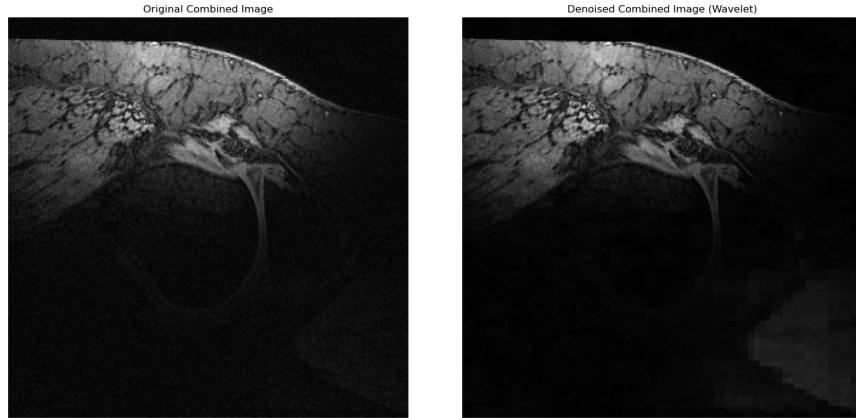


Figure 19: Denoised combination image.

Based on the performance metrics, the wavelet denoising method was selected to process all coil images before combination. After denoising, the images were combined using the modified sum-of-squares approach:

1. Apply a coil-specific weighting factor based on estimated SNR;
2. Square the product of magnitude and its normalised SNR for each coil;
3. Sum the weighted squared magnitudes;
4. Take the square root of the result.

In addition, the coefficients for coils 4 and 6 were set to zero due to their obvious incorrectness.

Magnitude	Original	Denoised	Improvement
SNR	16.03	22.17	38.31 %

Table 2: Comparison between the SNR of original and denoised image.

Figure 19 and Table 2 compare the original combined image with the denoised combined image. The denoised version achieved an around 1.38 times improvement (38%) in SNR compared to the original.

**Ex 2.2.4:** Comment on methods that might be able to further improve the combined image quality. [2]

Several methods might further improve the combined image quality:

- **Adaptive coil combination:** Rather than global weighted sum-of-squares, implementing a spatially adaptive combination based on local SNR would optimise the contribution of each coil.
- **Deep learning-based denoising:** CNNs trained on similar MRI data may provide better noise reduction while preserving anatomical details.
- **Non-local means filtering:** This approach would leverage redundancy in the image by identifying similar patches throughout the volume, potentially preserving fine details better than wavelet or bilateral filtering.

### 3) Module 3: CT Image segmentation and classification

#### 3.1 Exercise 3.1: Image segmentation

**Ex 3.1.1:** Open the NIfTI files and create one Numpy array per patient scan and one per segmentation mask. Find the range of voxels in which the segmentation exists for each patient. Create a numpy array with a subvolume by increasing 30 voxels in the x and y directions in both senses and 5 in the z direction. [7]

Firstly, the CT scans and segmentation masks were loaded from NIfTI files using the `NiBabel` library, I implemented it specifically in a function named `load_nii` in `segmentation.py` module. Subsequently, I analysed the mask (non-zero area) for each patient and then expand the region as required to obtain the subvolume. (See `segmentation.py` and `run_module3.ipynb` for details.)

```
For patient 31:  
Segmentation range (without padding): x=(365, 396), y=(295, 321), z=(168, 178)  
Subvolume shape: (91, 86, 20)
```

One example of subvolume was printed above, confirming the correctness of the shape of subvolume.

**Ex 3.1.2:** Create your own processing-based segmentation function using thresholding algorithm. Apply your algorithm to the subvolumes. [4]

**Ex 3.1.3:** Compare your segmentations to the ground truth ones. Is your method working as expected? Describe how you could improve it. [4]

Initially, I used the Otsu method from `scikit-image` as a baseline approach. As shown in Figures 20 and 21, this approach yielded poor results.

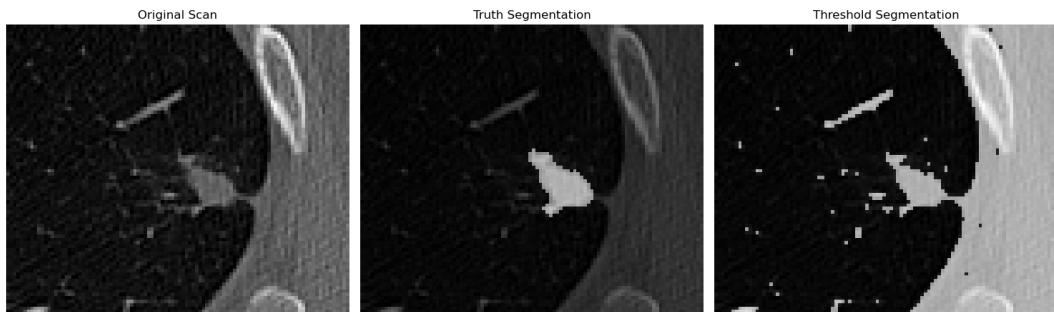


Figure 20: Poor OTSU segmentation result of case 22.

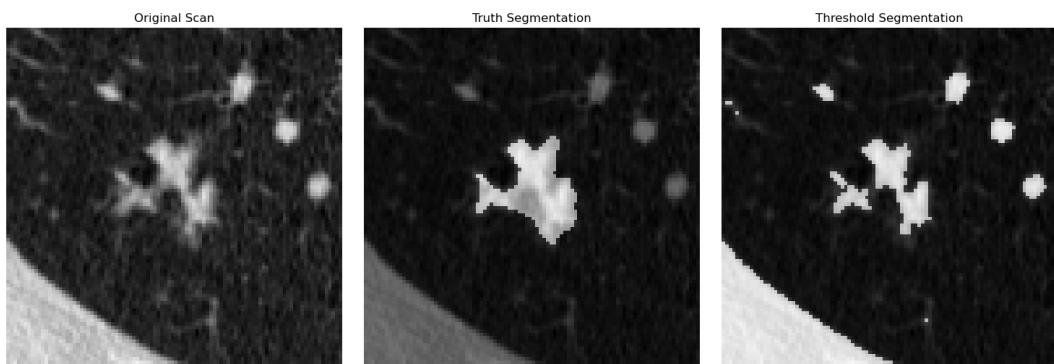


Figure 21: Poor OTSU segmentation result of case 38.

To improve the quality, I developed a function `adapt_thd_segm()` incorporating three main steps:

- Firstly, the initial segmentation was obtained by setting the max/min thresholds according to the min/max of the voxel intensities inside the segmentation. Considering the variations, the threshold range expansion with a tunable margin factor was also be set:

```

mask_indices = np.where(mask > 0)
voxel_values = subv[mask_indices]

thrh_min = np.percentile(voxel_values, 10)
thrh_max = np.percentile(voxel_values, 90)

range_width = thrh_max - thrh_min
thrh_min -= range_width * margin_factor
thrh_max += range_width * margin_factor

```

- Secondly, morphological operations (implemented with `scipy.ndimage`) were exploited to clean up the small noise spots in the initial segmentation.
- Most importantly, to handle the large false-positive segmentations near the border of subvolume, I proposed a problem-targeted boundary-aware algorithm to meticulously determine connected regions on the boundary. This method is based on the basic logical judgement:

For each connected fragment which has been initially segmented to foreground, let the boundary (with tunable width) split it into 2 parts: A (outside part) and B (inside part). Following the judgement workflow shown in the pseudocode, the false-positive pieces were reclassified correctly and effectively.

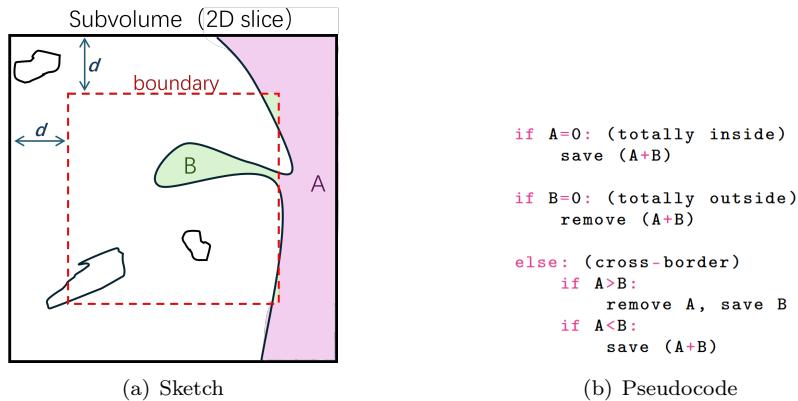


Figure 22: The sketch and pseudocode of boundary aware algorithm

Besides, the boundary region thickness  $d$  is a tunable parameter adaptive to the specific number of voxels for each subvolume, implemented by a scaling factor with value between 0 and 1.

Applying these to the subvolumes and extracted same two cases above, the results shows great improvement:

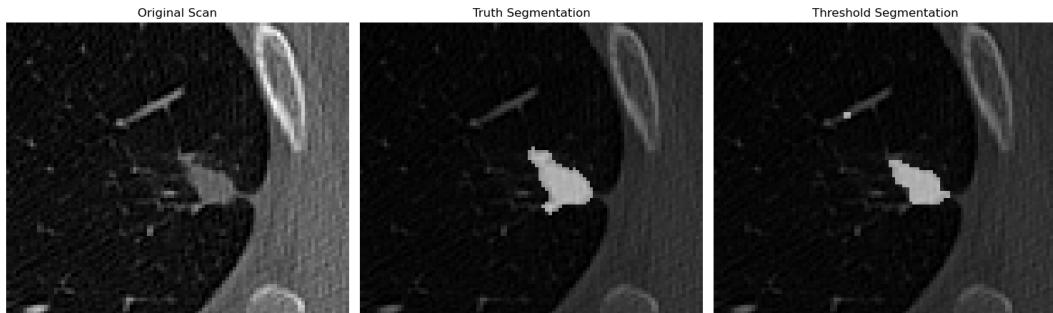


Figure 23: Improved result of case 22, using adaptive threshold segmentation algorithm

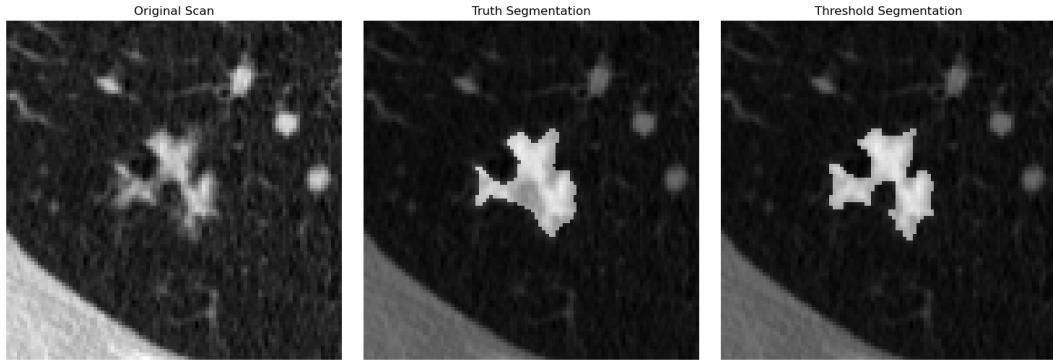


Figure 24: Improved result of case 38, using adaptive threshold segmentation algorithm

It can be seen that the large false-positive area near the boundary shown in Figure 20 and 21 are perfectly dropped out as expected.

Quantitive metrics were also calculated, the average Dice Score across all 40 patients was highly improved from  $0.0515 \pm 0.0517$  to  $0.5376 \pm 0.2276$ .

### Further improvements

However, this algorithm only take effect with target located in the central region of the ROI. The boundary region is defined as a hollow "shell" around the subvolume, where the thickness varies by direction, but maintains a consistent proportion relative to the lengths in each respective direction. This fixed scaling can be problematic when nodules are positioned asymmetrically within the extracted subvolume.

To solve it, implementing an adaptive scaling approach would be beneficial. This requires additional nodule localisation techniques, potentially informed by physical constraints when scanning. These could then be integrated with the boundary-aware algorithm to enable automated scaling factor selection, thereby achieving high-quality segmentation with low computational cost.

### 3.2 Exercise 3.2: Image feature extraction and classification

**Ex 3.2.1 - 2:** Write code to calculate the three features. Find the ranges of min-max voxels and decide a sensible number of bins for Uniformity. [6] Compute the value of each feature on each patient's nodule. [3]

Three features were calculated in the `extract_fea()` function in the `classification.py` module, and the results were presented in `run_mpdule3.ipynb`. For the histogram binning required to calculate uniformity, I used 100 equally-spaced bins spanning the minimum to maximum intensities found in each nodule, which sufficient to capture the intensity distribution while avoiding overly sparse histograms. The features were computed for all 40 cases using the ground truths. Boxplots 25 compares the distribution of each feature across different diagnostic categories: 1 (benign), 2 (primary malignant), and 3 (metastatic malignant). (Label description are obtained from the whole `Patient Diagnoses` dataset from [2]).

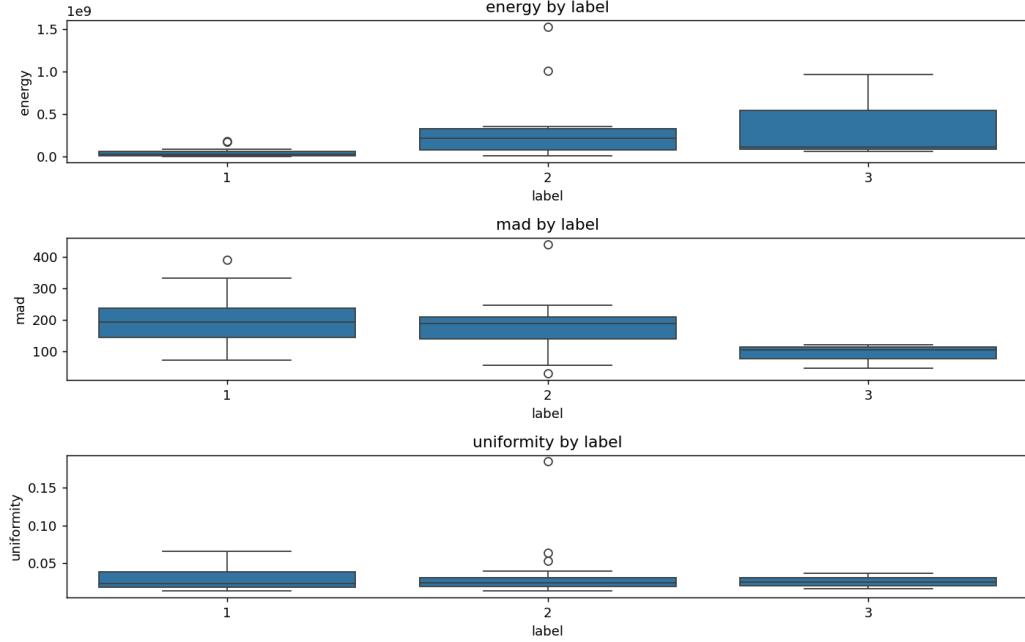


Figure 25: Boxplots of each feature across three diagnostic groups.

It can be seen that energy values are substantially higher for malignant (both primary and metastatic) compared to benign; MAD are higher in benign and primary malignant than in metastatic nodules; Uniformity decreases progressively from benign, to primary malignant, then to metastatic malignant. But appears less discriminative than the other features.

Figure 26 displays the correlation matrix between features, showing weak negative correlations between uniformity and the other features (uniformity-MAD: -0.37, uniformity-energy: -0.11), and a tiny correlation between energy and MAD (-0.033). This suggests the features capture complementary aspects of the nodules with small redundancy.

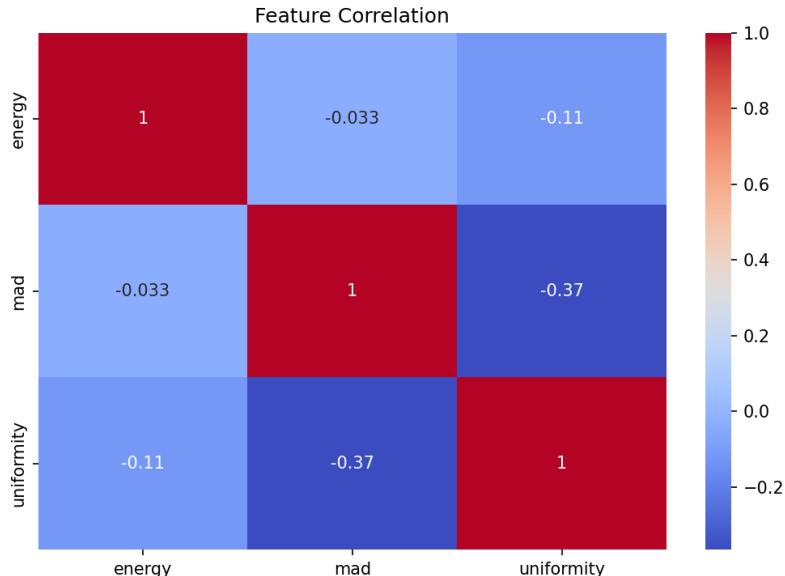


Figure 26: The correlation matrix between the three features.

**Ex 3.2.3:** Which of those features would you use to classify between the benign and malignant lesions, why? [6]

Table 3: t-test results for distinguishing between benign and malignant lesions

Feature	t-statistic	p-value	Mean (Benign)	Mean (Malignant)	Std (Benign)	Std (Malignant)
<b>Energy</b>	-3.052541	0.006369	4.174371e+07	3.137498e+08	5.127233e+07	3.850141e+08
<b>MAD</b>	1.005067	0.321331	1.967732e+02	1.693786e+02	7.829315e+01	8.936224e+01
<b>Uniformity</b>	-0.712795	0.482593	2.894423e-02	3.535466e-02	1.455553e-02	3.639880e-02

Based on the statistical analysis presented above, the **Energy** feature would be the best choice for these classification. The reason includes:

- **Statistical significance:** Energy shows the strongest statistical difference between benign and malignant groups with a t-statistic of -3.052541 and p-value of 0.006369, well below the conventional threshold of 0.05. In contrast, MAD ( $p=0.321331$ ) and Uniformity ( $p=0.482593$ ) show no statistically significant differences.
- **Biological rationale:** Energy reflects nodule intensity and density, which are typically higher in malignant lesions due to higher metabolic activity, consistent with radiomic features of malignancy.
- **Technical robustness:** Energy calculation is less susceptible to segmentation errors and binning choices compared to Uniformity, making it more robust across different preprocessing pipelines.

Figure 27 also demonstrates that **Energy** provides better separation power.

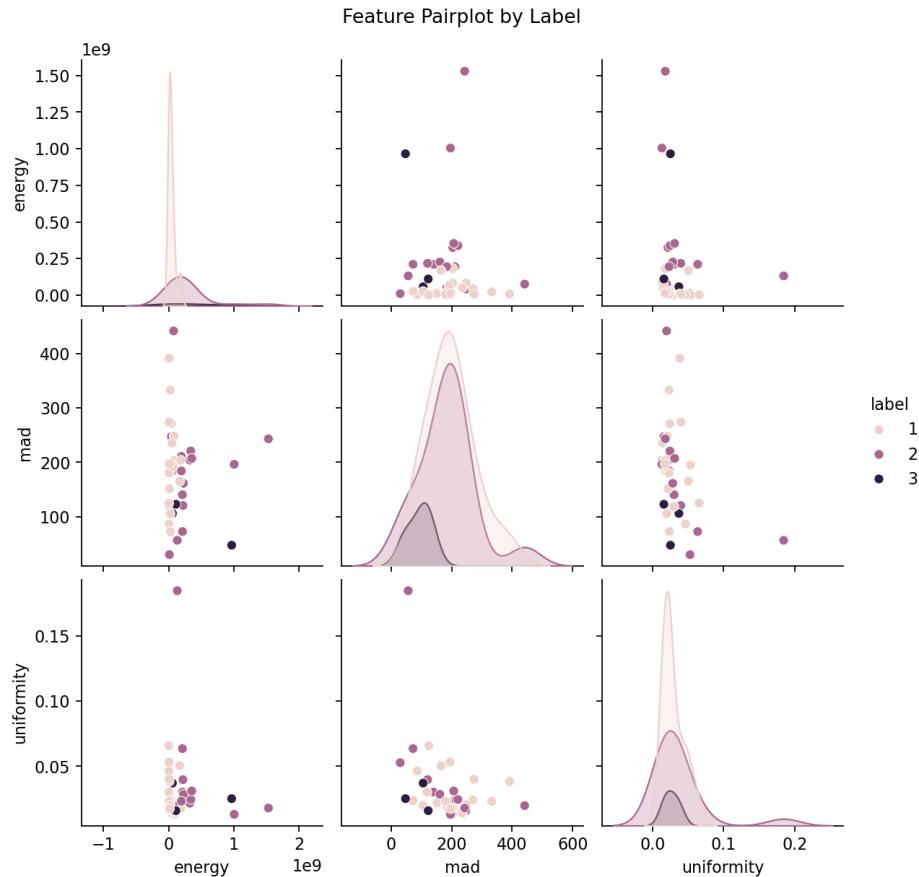


Figure 27: The pair plot of the three features by labels.

## References

- [1] J. P. J. Carney, D. W. Townsend, V. Rappoport, and B. Bendriem, *Method for transforming ct images for attenuation correction in pet/ct imaging*, *Medical Physics* **33** (2006), no. 4 976–983, [<https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1118/1.2174132>].
- [2] Armato III, Samuel G., 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, E. A. Kazerooni, H. MacMahon, E. J. Van Beek, D. Yankelevitz, A. M. Biancardi, P. H. Bland, M. S. Brown, R. M. Engelmann, G. E. Laderach, D. Max, R. C. Pais, D. P. Qing, R. Y. Roberts, A. R. Smith, A. Starkey, P. Batra, P. Caligiuri, A. Farooqi, G. W. Gladish, C. M. Jude, R. F. Munden, I. Petkovska, L. E. Quint, L. H. Schwartz, B. Sundaram, L. E. Dodd, C. Fenimore, D. Gur, N. Petrick, J. Freymann, J. Kirby, B. Hughes, A. V. Casteele, S. Gupte, M. Sallam, M. D. Heath, M. H. Kuhn, E. Dharaiya, R. Burns, D. S. Fryd, M. Salganicoff, V. Anand, U. Shreter, S. Vastagh, B. Y. Croft, and L. P. Clarke, *Data from lidc-idri*, 2015.
- [3] Z. H. Levine, A. P. Peskin, A. D. Holmgren, and E. J. Garboczi, *Preliminary x-ray ct investigation to link hounsfield unit measurements with the international system of units (si)*, *PLOS ONE* **13** (12, 2018) 1–18.
- [4] P. A. Toft, *The radon transform-theory and implementation*, .

## Appendix

**Declaration:** No auto-generation tools were used in this report except for generation of BibTeX references.

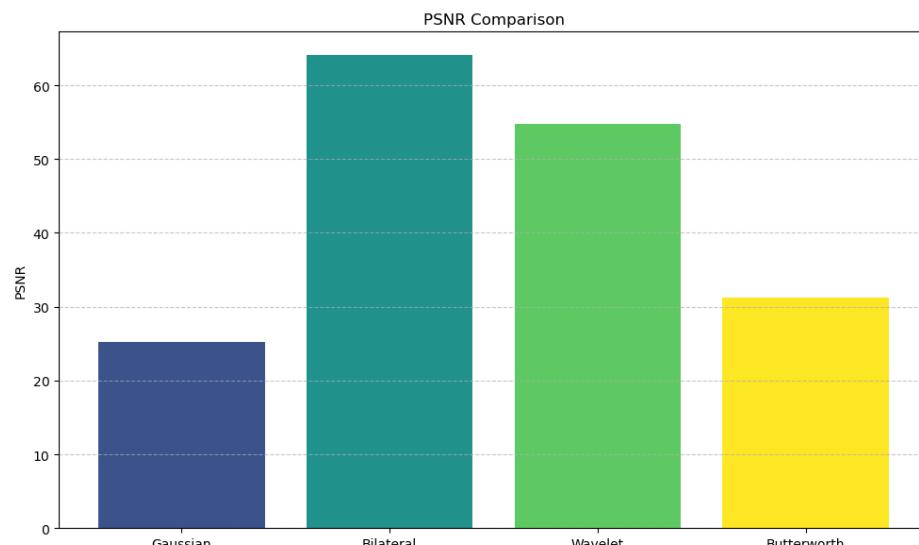


Figure 28: PSNR comparison of four filtering results

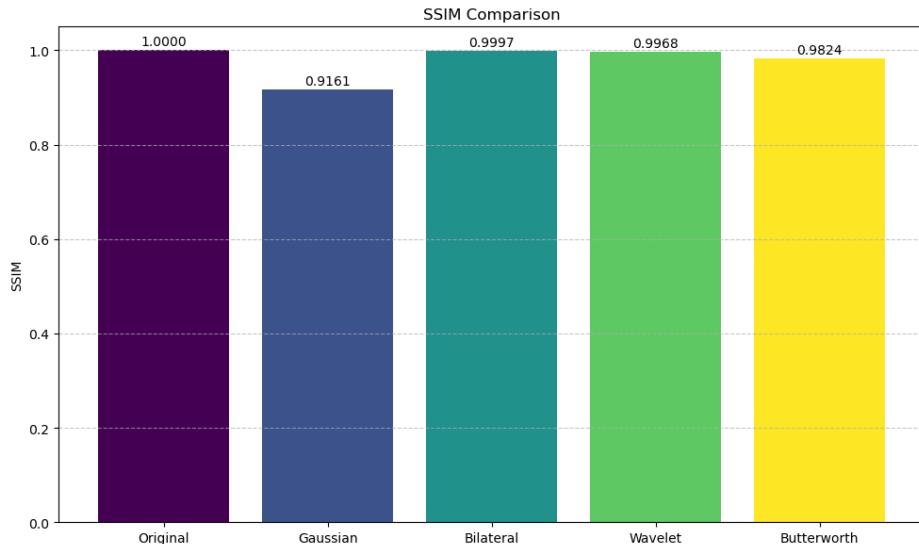


Figure 29: SSIM comparison of four filtering results