

A2 Medical Imaging Coursework

Zihan Xu (zx282)

Lent Term 2025

Word Count: 2999

1 Module 1: PET-CT Image Reconstruction

1.1 Clean up Sinograms

To clean up the sinograms for CT scans in "ct.sinogram.npy" before the reconstruction, note that the dark field data, "ct_dark.npy", provides a baseline for noise level in the system when no object is in the field of view and X-ray beams turned off. On the other hand, "ct_flat.npy" gives information about what detector measures when X-ray beams are turned on while no sample is present in the field. Together, they provide information on detector sensitivity and potential X-ray beam inhomogeneity. To correct CT sinograms with these two pieces of data, the following formula is used:

$$ct_{corrected} = \frac{ct_{original} - ct_{dark}}{ct_{flat} - ct_{dark}}$$

Subtracting ct_{dark} from $ct_{original}$ and ct_{flat} removes system noise and the final division corrects differences in the final pixel-to-pixel sensitivity.

Similarly, for PET sinograms, the detector gain calibration sinogram shows variation in detector sensitivity. To correct for the disparity in sensitivity, the correction is performed with the euqation below:

$$pet_{corrected} = \frac{pet_{corrected}}{pet_{calibration}}$$

The corrected sinograms are plotted in comparison with the original sinogram:

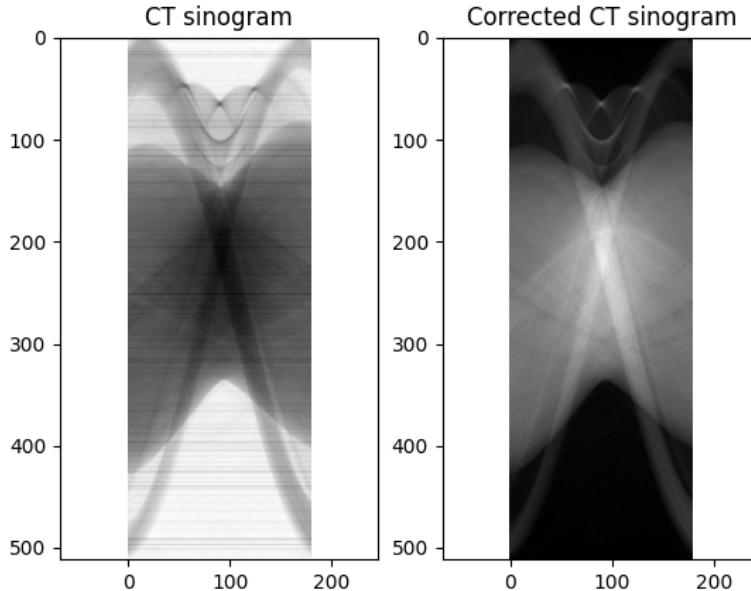


Figure 1: Dark-Flat Field Corrected CT sinogram

After the correction, the blurry horizontal lines present in the original CT sinograms are eliminated, and there is a better contrast between the curves.

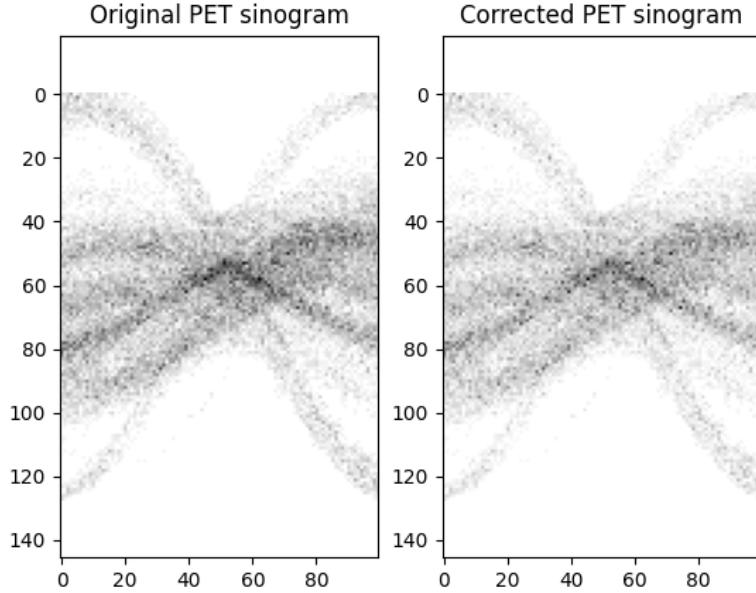


Figure 2: Calibration Corrected PET sinogram

The extreme contrast in the original PET sinogram has been smoothed out after applying the calibration correction.

1.2 Reconstruct CT Image

From the cleaned CT sinogram, the CT image could be constructed. The FBP and OS-SART methods are used for reconstruction, and their reconstruction qualities are examined. The FBP reconstruction uses the power of inverse radon transformation, which is only valid when X-ray beams are parallel, which in this case it is. The OS-SART algorithm, in particular, requires investigation of its parameter choices. The OS-SART is the following:

$$x^{k+1} = x^k + \gamma * A_i^T (A_i x^k - b)$$

Here, x^0 is chosen to be the zero matrix, with dimension equal to the final output image, 512×512 . A_i corresponds to the radon transformation on subset i out of the 10 subsets while A_i^T corresponds to the inverse of radon transformation. b is the sinogram. Parameters γ and $K = \max(k)$ is searched for the best reconstruction quality:

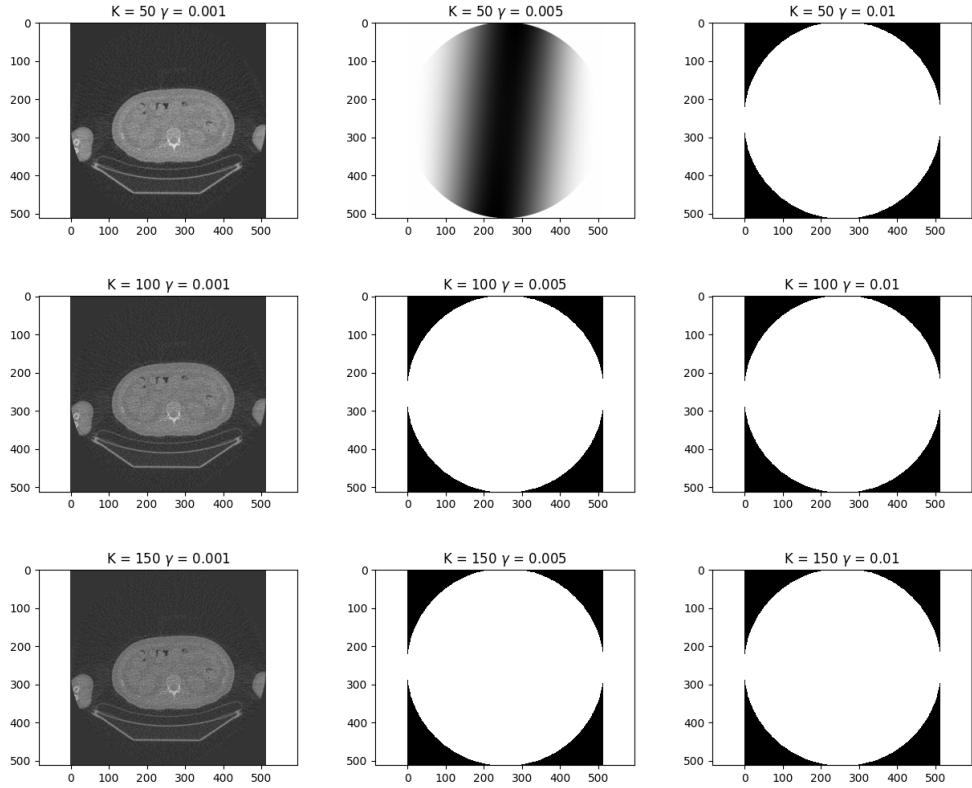


Figure 3: OS-SART: Parameter search

From the figure it could be seen that when γ is set to be 0.01, the reconstruction gives a clear image. When $K = 150$, it gives the best quality image among this grid-wise search. This image is taken to be compared with the FBP reconstruction below:

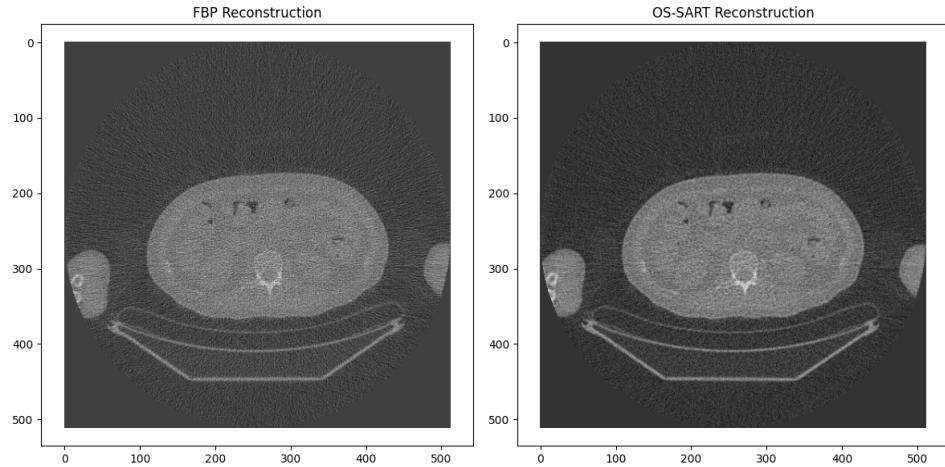


Figure 4: CT reconstruction: FBP v.s OS-SART

It could be seen that the OS-SART algorithm gives a slightly better image, with the

boundaries having higher contrast and different regions clearly divided. However, it is worth noting that the OS-SART algorithm has a significantly higher computational cost than FBP, due to the fact that radon and inverse radon transformation is applied once every iteration versus once through the entire reconstruction with the FBP method.

OS-SART algorithm is also compared with the SIRT algorithm, where the sinogram is passed into the transformation as a whole rather than subsets. Both algorithms are executed with parameters $\gamma = 0.001$ and $K = 150$:

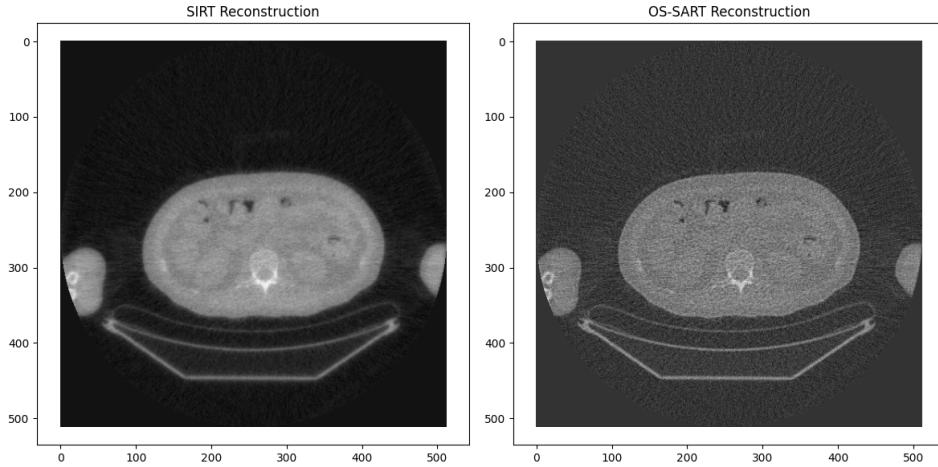


Figure 5: CT reconstruction: SIRT v.s OS-SART

Although the regions are more clearly divided for SIRT reconstruction, with different tissues having high contrast against each other, the boundaries are highly blurry. The OS-SART reconstruction gives better boundaries for tissues. However, the distinction in tissues is less emphasized.

1.3 Rescaling CT

Given that the CT image and PET image have distinct size, a rescaling factor is calculated to enable attenuation correction for the PET image. The factor is calculated using the pixel sizes:

$$factor = \frac{pixel_{CT}}{pixel_{PET}} = \frac{1.06}{4.24}$$

The target resized CT image hence has size:

$$CT_{resized} = CT_{original} * factor$$

The new size is passed into the *resize* function from *skimage* package to produce the new image, which is subsequently gone through radon transformation to produce CT absorption sinogram.

The attenuation correction for the PET sinogram can then be produced using Beer-Lambert rule:

$$I_{attenuation\ corrected} = I_{original}/e^{-\mu_i l_i} = I_{original} * e^{-\mu_i l_i}$$

where

$$sinogram_{CT} = p = \sum \mu_i l_i$$

The resulting corrected PET sinogram is the following:

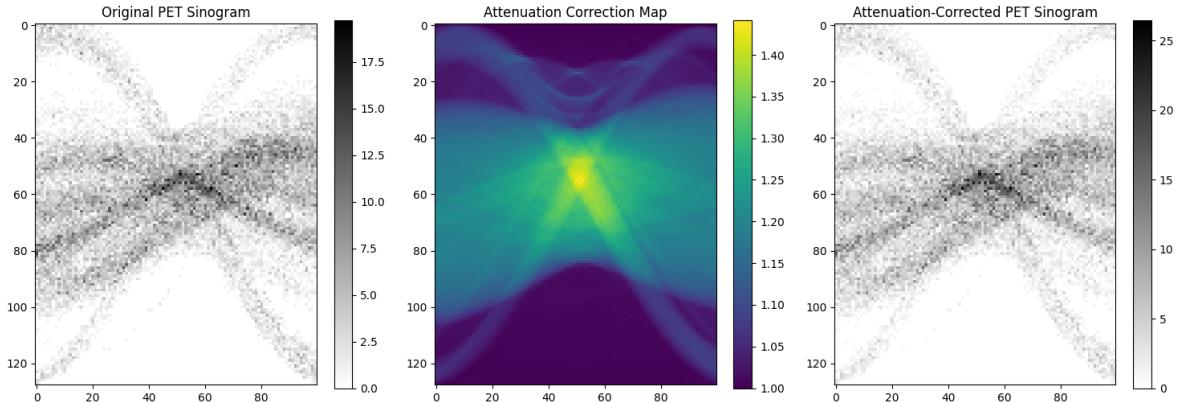


Figure 6: Attenuation Corrected PET Sinogram

The curve with large amplitude is lighter after undergone attenuation correction. In the middle, where most detected points coincide, the intensity is slightly increased. This behavior is expected, as the attenuation correction map has the highest intensity in the middle, around coordinate (50,55).

1.4 Reconstruct PET

Similar to the CT reconstruction, PET reconstruction methods, FBP and OSEM, are implemented and compared. OSEM is defined as, as mentioned in the coursework question sheet:

$$x^{k+1} = x^k A_i^T \left(\frac{b}{A_i x^k} \right)$$

b corresponds to the sinogram, A_i corresponds to the radon transformation matrix on the i^{th} subset out of the 10 subsets of the detector angles. x^0 is initialized to be the zero matrix, with dimension equal to the final reconstructed image.

FBP method uses only one instance of inverse radon transformation. Same as the case with CT reconstruction, the computation cost is significantly higher with OSEM reconstruction. Below are the reconstruction results:

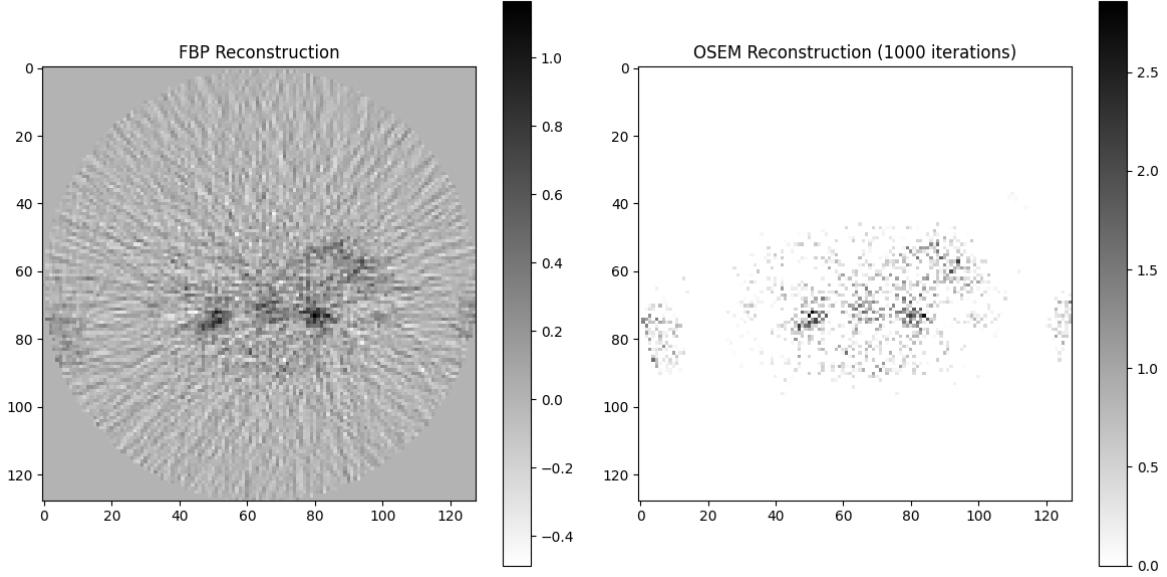


Figure 7: PET reconstruction: FBP v.s OSEM

The OSEM reconstruction is implemented with 1000 iterations. Due to the initialization with zero matrix, the OSEM PET image has no background noises, which creates a clear figure. The FBP reconstruction, on the other hand, is impacted hugely by noises. The margins for the tissues are ambiguous.

The capability of the OSEM algorithm is further compared with MLEM, defined as the following:

$$x^{k+1} = x^k A^T \left(\frac{b}{Ax^k} \right)$$

The results are the following:

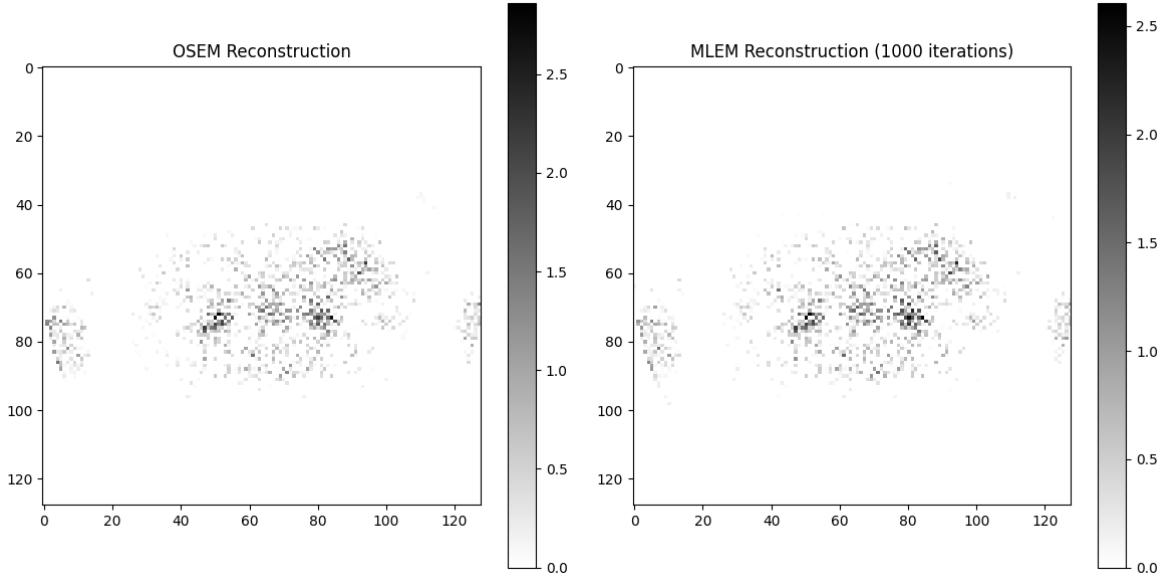


Figure 8: PET reconstruction: OSEM v.s MLEM

Both figures are produced with x^0 equaling to the zero matrix of the same size with final image and iterations up to 1000.

There is no significant disparity between the construction results. The MLEM figure has slightly more condensed points distributed in the left side of the image (x-coordinate below 40).

Below is the reconstruction using 100 iterations:

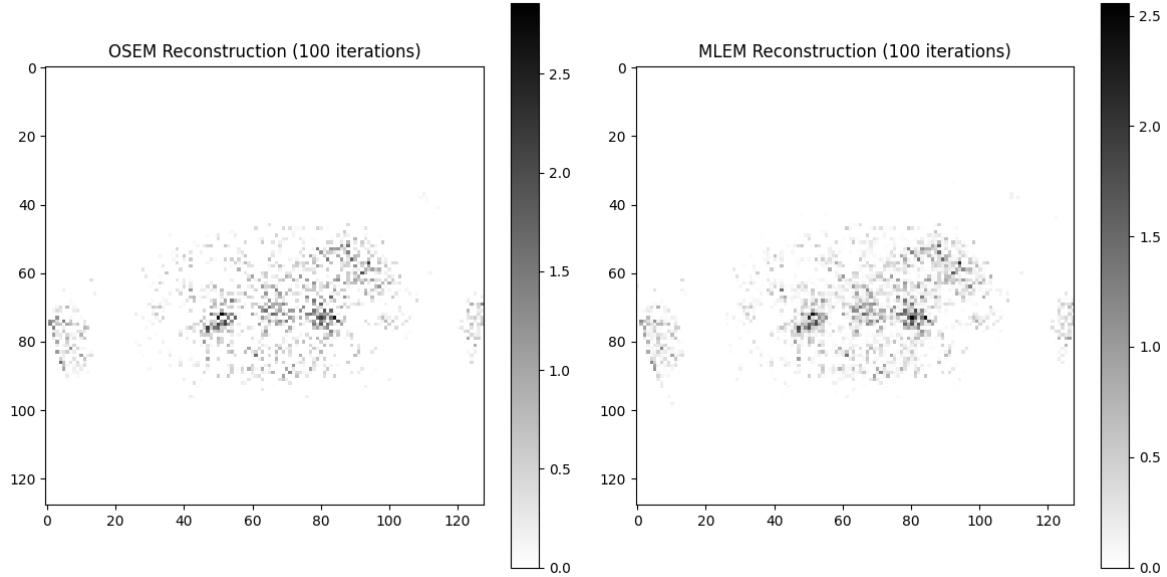


Figure 9: Caption

1.5 Theory

(1) Overlay of PET and CT scan is the following:

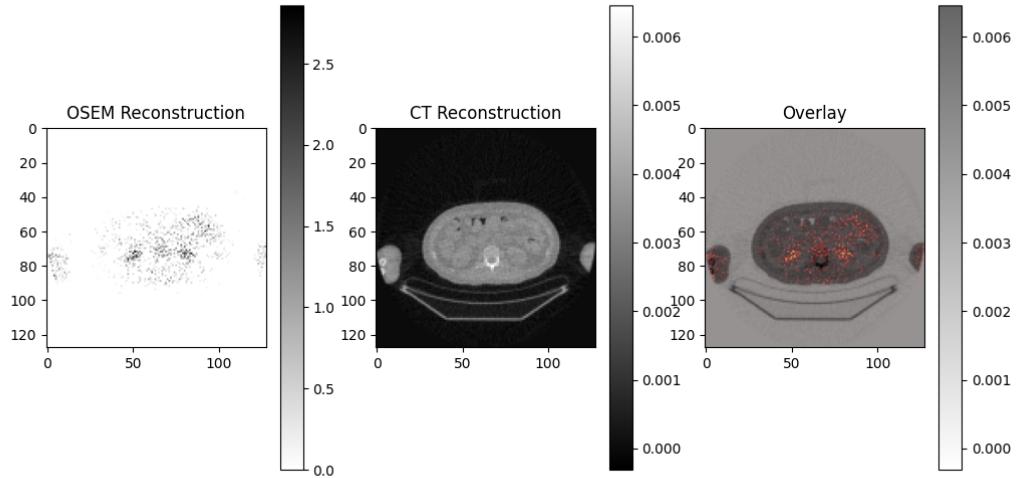


Figure 10: PET-CT Overlay

The regions with high PET measurements corresponds to some of the soft tissues region in the CT reconstruction, showing the success in reconstructing both images. The

OSEM reconstruction is produced with OSEM after 1000 iterations while CT reconstruction is implemented with 150 iterations using the OS-SART algorithm.

(2) According to the time of flight equation, the resolution in time and space follows the following relationship [1]:

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

where Δx , Δt corresponds to space and time resolution and c being the speed of light. Given the pixel size is $4.24mm$:

$$\Delta t = \frac{2 \times 0.00424m}{3 \times 10^8 m/s} = 2.82 \times 10^{11}s$$

(3) PET scans measures instances of positron collision with electrons, which follows Poisson distribution. OSEM is designed with multiplicative structure, distinctive from the additive updates of gradient descent. OSEM's structure captures the non-linear relationship between measured counts and activity distribution. CT measurements, however, has linear relationship with attenuation values, which makes additive updates appropriate.

(4) MRI does not provide attenuation coefficients for PET to perform attenuation correction. Hence, additional data processing is required to correct PET data.

2 MRI Image Denoising

2.1 Visualisation and identifying noise

(1) The complex data "kspace.npy" is loaded using Numpy, resulting in an array with shape (6, 280, 280). Thus, it can be inferred that the first dimension (size 6) corresponds to the coil dimension.

(2) Below is an image of the magnitude of k-space for each coil. The data is scaled using `np.log1p()`:

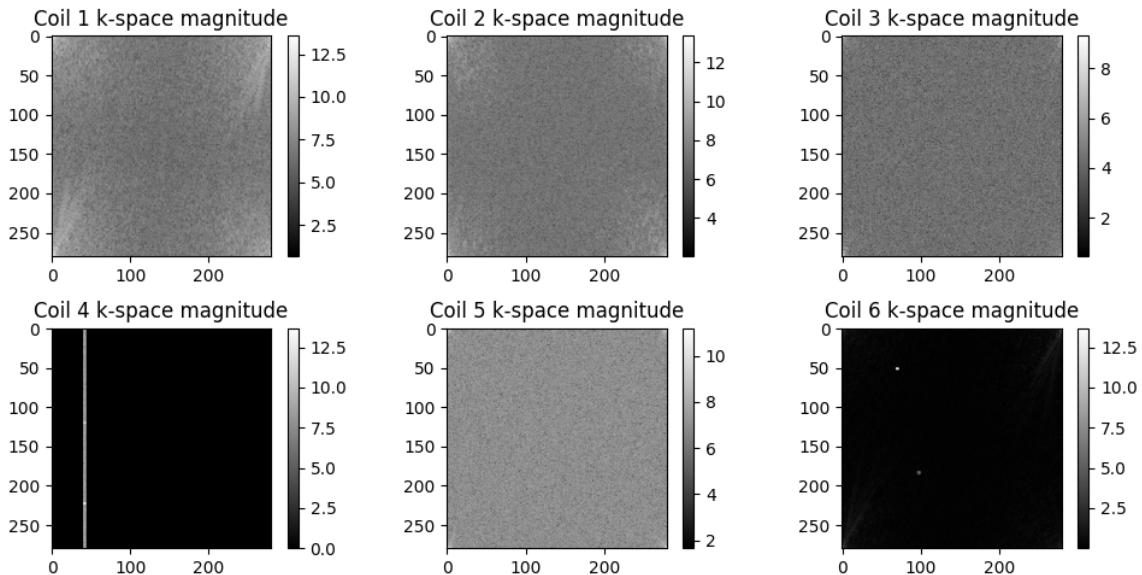


Figure 11: k-space Data Magnitude

The magnitude of data in coils 2, 3, and 5 is relatively homogeneous. Coil 1 exhibits minor peaks near the bottom-left and top-right corners. In coil 4, there is a distinct line of spikes around x-coordinate 40, while coil 6 shows an isolated peak at coordinate (60,50).

(3) Fourier transform is applied to the data using the `fft` module from Numpy. A `np.fft.fftshift` function is applied first to shift the zero frequency component to the center, and then the Fourier transform is applied. This measure ensured alignment of data from all coils. The magnitude and phase image from coil 1 is visualised:

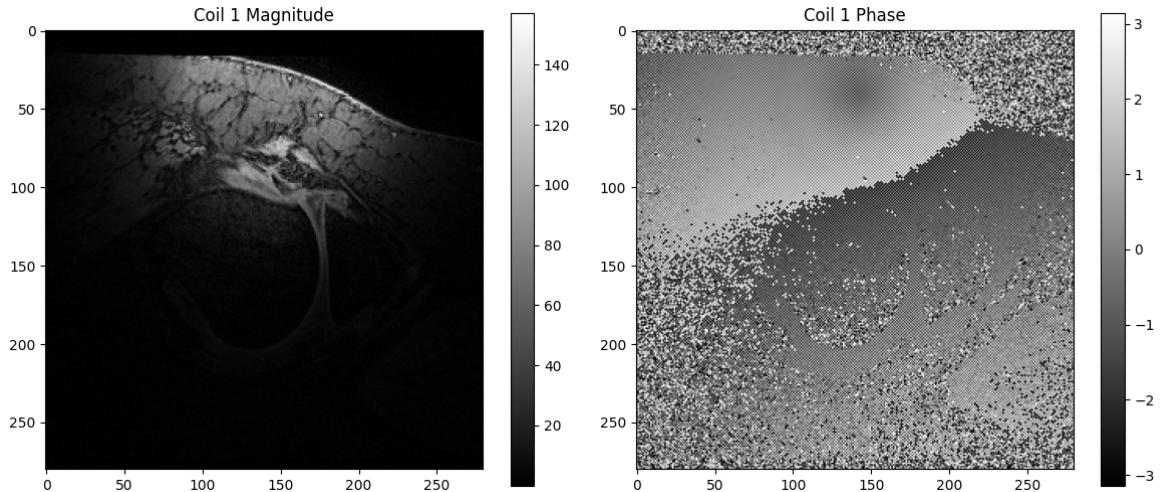


Figure 12: Magnitude & Phase Image from Coil 1

The magnitude image reveals the overall shape of the scanned knee. In the phase image, two distinct regions emerge, each corresponding to the areas where the knee is identified in the magnitude image.

(4) The magnitude image from all coils is later on computed:

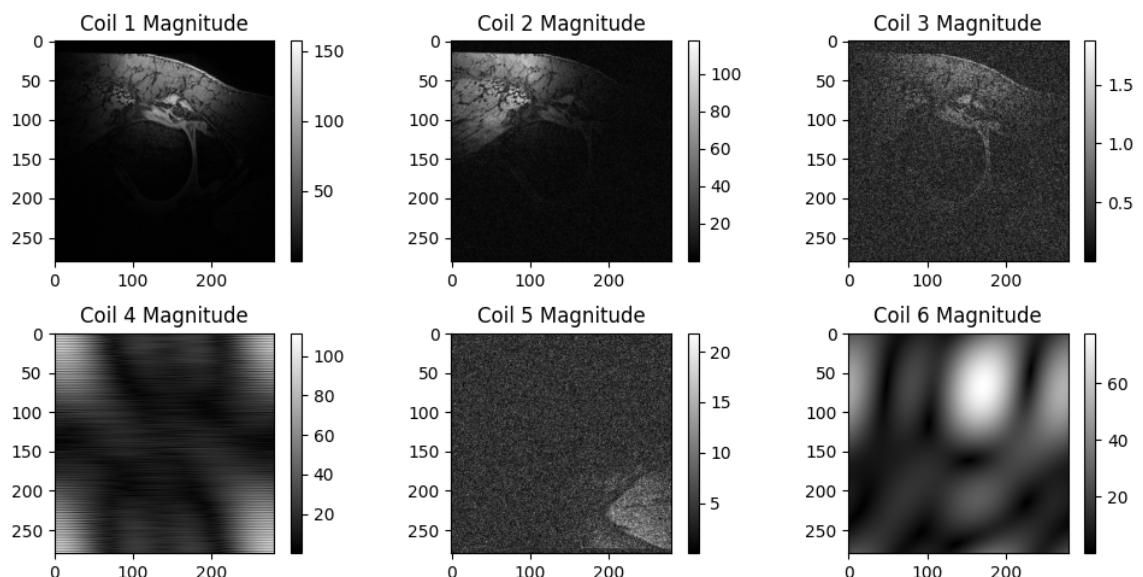


Figure 13: Magnitude Image: All Coils

Each image appears to contain part of the inspected knee.

- (5) The combined image, following procedure outlined in the question sheet, is presented below:

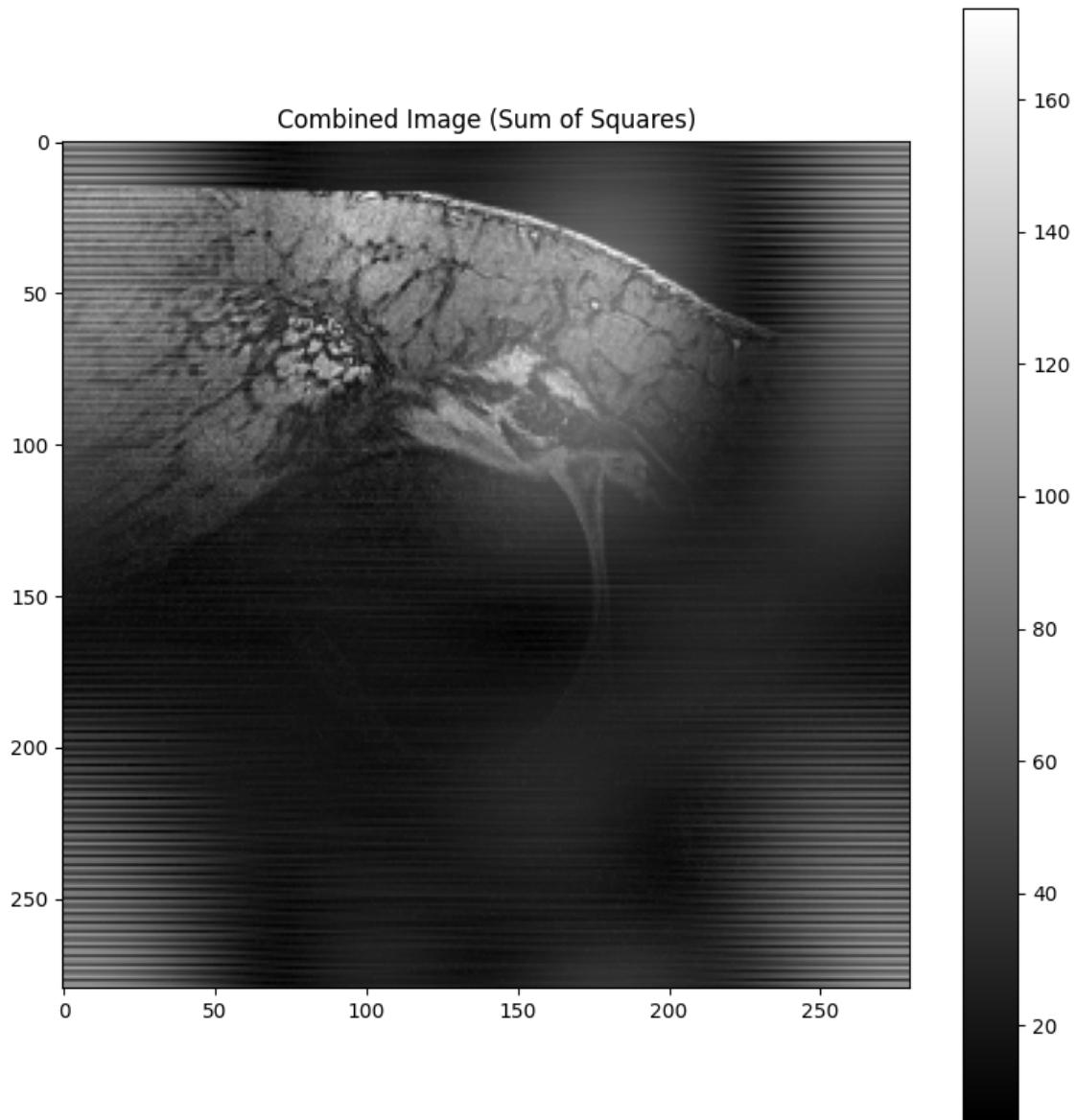


Figure 14: Combined Image

- (6) Numerous horizontal lines are visible in the background, indicating sudden jumps in measurements, affecting the clarity of the image. Near the center of the knee, the horizontal lines gradually disappeared.

Back to the k-space data and magnitude data from all coils, the data from coils 4 and 6 exhibit distinct patterns compared to the other coils. This anomaly may contribute to the noise present in the image.

2.2 Removing noise

- (a) Gaussian, bilateral, and wavelet filters are chosen to be the 3 denoising methods.

The Gaussian filter is defined as below:

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

The Gaussian filter is computed using `scipy.ndimage`. The standard deviation, σ is chosen to be 1. This choice offers a balance between smoothing and preservation of detail for the image.

The bilateral filter is implemented using the cv2 library. It is configured such that for each pixel in the image, the filter considers a 9×9 neighborhood surrounding it. The filter combines spatial proximity and color similarity to preserve edges while reducing noise. Pixels that are both close in spatial distance and similar in color to the target pixel have a larger influence on the filtered value, ensuring that smoothing occurs primarily within regions of similar colors and preserving sharp boundaries between different color regions.

The skimage.restoration module is used to apply the wavelet filter. After transforming the data into the wavelet domain using a wavelet transform, the BayesShrink thresholding method is applied. BayesShrink is an empirically effective method for wavelet filter denoising, as demonstrated in previous studies [2].

The images denoised with various methods are plotted below in comparison with the original image:

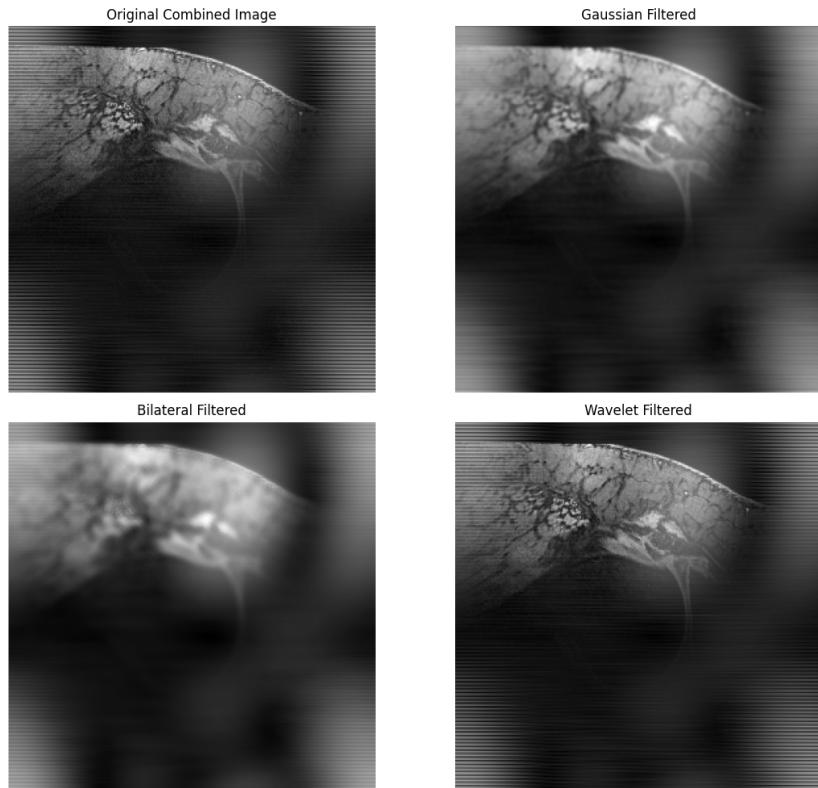


Figure 15: Denoising Method Comparison

The effect of these denoising methods is demonstrated above. In the Gaussian Filtered image (top-right), the Gaussian filter smooths the image, reducing the noise. However,

this method also blurs the edges between tissues, resulting in a loss of details and sharpness. It fails to preserve the finer structures of tissues, leading to an overly smoothed appearance.

The Bilateral filter is less effective at preserving edges and reduce noises compared to the Gaussian filter. This method reduces noise and smoothed the images even further. The boundaries between tissues are almost indistinguishable. Although bilateral filter gives a more blurry picture, it is performing worse in blurring the noisy lines in comparison with Gaussian filter.

The Wavelet filter excels in removing noise and preserving both edges and details. This method provides a cleaner output, particularly in areas with sharp transitions. The wavelet filter effectively maintains the structure of the image, making it the most effective method among the three for preserving overall image quality. However, the noises are the less reduced among all three methods. The horizontal linear noises are the sharpest compared to the others.

The behavior of all three methods signified the trade-off between detail preservation and noise reduction. Overall, Gaussian filter strike the best balance between the two.

(b) The magnitude and phase images for the first coil after applying a low-pass Butterworth filter within k-space is the following:

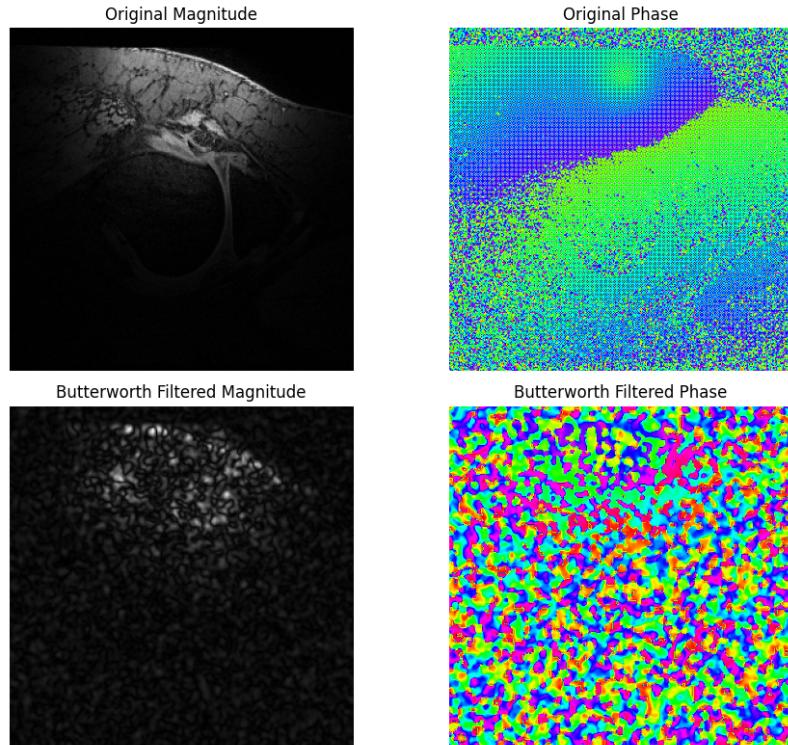


Figure 16: Butterworth Filter Magnitude and Phase

It is evident from both the magnitude and phase images that the Butterworth filter significantly distorted the original patterns. The edges became overly coarse, and only the contrast between bright and dark regions was faintly preserved. It makes Butterworth filter less favorable in medical imaging, as details are significant for medical images.

(c) To further examine the effect of low-pass Butterworth filter, a new combined image is created:

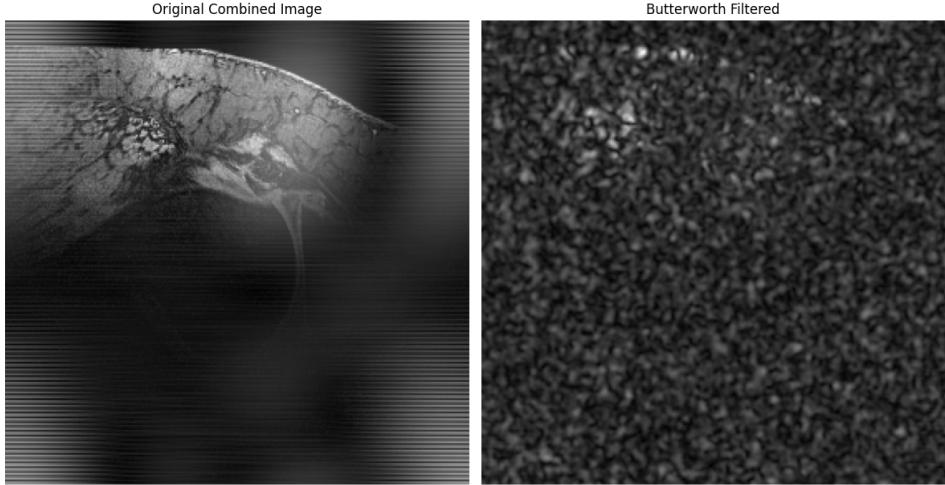


Figure 17: Combined image: Low-pass Butterworth Filter

As expected, the resulting image by passing through a low-pass Butterworth filter completely lost the details. The filter has removed much of the high-frequency noise, resulting in a smoother, more uniform appearance across the entire image. However, this smoothing comes at the cost of edges and details, which are notably lost. Additionally, the image exhibits grainy noise patterns, especially in regions where the filter was less effective at completely removing noise. This indicates that the Butterworth filter might not be ideal for preserving image features and structures.

(d) Among all the methods explored in this section, the Wavelet filter is likely to further improve the quality of the combined image. It is able to preserve the details and structure of the image, which is highly significant for medical images. Varying the thresholding function could potentially elevate its denoising ability, hence producing a clear and noise-free image.

3 CT Image segmentation and classification

3.1 Image segmentation

(a) The `nibabel` library is used to open the NIfTI files and process the CT scans to Numpy arrays. Redistricted by the RAM of my local machine, the code `CT_classification.ipynb` is ran on Google Colab in order to store all the arrays into one array. Each scan consisted of different number of slices, with each slice having size 512×512 .

Segmentation mask exists at a specific voxel if the masks has positive value on that index. Using this information, the range of coordinates in which segmentation exist for each patients is: $x : (35, 432)$, $y : (166, 414)$ and $z : (83, 448)$.

The subvolume of the scans are created by determining the range of values in which segmentation exist for each scan individually, then adding 30 voxels in x and y directions (min and max), followed by another 5 voxels in the z direction. The function that execute the above operations is called `create_subvolume` in the python script.

(b) A simple threshold algorithm is created by finding the min and max of the intensities inside the subvolume segmentation mask, then apply the threshold to the subvolume of the scan. The voxels that satisfy the threshold will be marked. The functions performs the algorithm are called `apply_thresholding` and `find_intensity_range`.

(c) Below is an example of the segmentation obtained using the threshold function in comparison with the ground truth. This example is created by viewing the scans of patient "case_0" and taking the middle slice.



Figure 18: Threshold Segmentation Comparison

The threshold mask has a partial agreement with the ground truth, which is expected as the threshold is obtained with knowledge of the ground truth. However, the calculated mask has a significantly larger marked area than the ground truth mask. Despite having a wider region in the middle of this slice, there are additional marks beyond the area of interest.

This result is expected as some regions in the brain is likely to have similar presentation under X-ray as the tumor. To improve it, the function can combine with edge detections. There will be drastic changes in intensity on the edge of tissues, which can serve as an indicator to where the segmentation mask should end. This can help better delineate boundaries between objects, hence more likely to agree with the ground mask.

3.2 Image feature extraction and classification

(a) The code that calculates energy, mean absolute deviation and uniformity are present under the CT folder. In order to determine a sensible number of equally spaced bins, the min and max intensities of the voxels are calculated. To avoid disruption of data, only patients that are diagnosed with tumor is included in this calculation. The number of bins for each image is determined using Scott's rule, by dividing the range of intensities in the segmentation mask by the optimal band width as derived from Scott's rule:

$$num_{bins} = \frac{intensity_{max} - intensity_{min}}{3.5\sigma/n^{1/3}}$$

Here, n is the number of voxels with intensities and σ is the standard deviation of the intensities.

By iterating the equation above through all the patients with tumor, a number of suitable bin numbers is obtained. Taking the mean over al bin numbers give the ultimate bin number that ought to work for all cases.

(b) To avoid attaching an overly long table here, the table that contains values for energy, MAD and uniformity for all the patients are attached in the appendix, labeled table 1. The values are rounded to two decimal places and is calculated using the program mentioned in (a).

(c) Uniformity would be the most effective feature among the three features for classifying lesions because it captures the homogeneity of the scan. Benign lesions typically present more uniform characteristics, as they are often well-defined with consistent intensity values throughout.

In contrast, malignant lesions tend to show greater heterogeneity due to their irregular structure, leading to more diverse intensity values within the lesion. By measuring how evenly distributed the intensity values are within the region of interest, Uniformity can effectively differentiate between the more homogeneous nature of benign lesions and the complex patterns associated with malignant ones. Therefore, Uniformity provides a straightforward and reliable way to identify malignancy, as malignant lesions are generally characterized by higher heterogeneity and lower Uniformity scores.

4 Acknowledgment

This report was developed with the assistance from Anthropic’s Claude 3.5 Sonnet. Specifically, Claude was used to improve the clarity of written content and assist with debugging Python code implementations. All AI-generated suggestions were manually reviewed.

References

- [1] Kadrmas DJ, Casey ME, Conti M, Jakoby BW, Lois C, Townsend DW. Impact of time-of-flight on PET tumor detection. *J Nucl Med.* 2009 Aug;50(8):1315-23. doi: 10.2967/jnumed.109.063016. Epub 2009 Jul 17. PMID: 19617317; PMCID: PMC2786272.
- [2] S. G. Chang, Bin Yu and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," in *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532-1546, Sept. 2000, doi: 10.1109/83.862633. keywords: Image denoising;Image coding;Noise reduction;Quantization;Bayesian methods;Wavelet coefficients;Gaussian distribution;Image processing;Parameter estimation;Bit rate,

5 Appendix

Below are the value of Energy, MAD and Uniformity for each patient:

Table 1: Case Data Summary with Diagnosis

Case ID	Energy	MAD	Uniformity	Diagnosis
case_0	27673733.0	189.50	0.04	[2]
case_1	57513797.0	106.04	0.13	[3]
case_10	6673689.0	274.03	0.06	[1]
case_11	965718972.0	47.67	0.09	[3]
case_12	2368756.0	118.83	0.05	[1]
case_13	182834811.0	203.77	0.06	[1]
case_14	168770785.0	165.20	0.17	[1]
case_15	226613543.0	161.47	0.10	[2]
case_16	210292123.0	140.45	0.10	[2]
case_17	8671432.0	391.40	0.06	[1]
case_18	22688285.0	332.64	0.05	[1]
case_19	325362110.0	204.06	0.08	[2]
case_2	10559604.0	30.29	0.18	[2]
case_20	43936602.0	271.02	0.08	[1]
case_21	1609759.0	194.67	0.09	[1]
case_22	194329843.0	210.90	0.08	[2]
case_23	5397690.0	180.29	0.06	[1]
case_24	67220097.0	192.58	0.05	[1]
case_25	1690860.0	124.76	0.09	[1]
case_26	26732676.0	72.91	0.06	[1]
case_27	25624693.0	185.28	0.05	[1]
case_28	5168077.0	86.61	0.07	[1]
case_29	12438594.0	151.48	0.07	[1]
case_3	13421304.0	197.12	0.05	[1]
case_30	83087566.0	248.21	0.07	[1]
case_31	194505593.0	184.18	0.08	[2]
case_32	217564273.0	120.70	0.13	[2]
case_33	211094910.0	72.76	0.21	[2]
case_34	75503835.0	441.40	0.06	[2]
case_35	41967170.0	247.32	0.04	[2]
case_36	47480282.0	183.57	0.06	[2]
case_37	353427813.0	207.22	0.11	[2]
case_38	1004737259.0	196.41	0.05	[2]
case_39	1529487855.0	243.22	0.06	[2]
case_4	110762266.0	122.90	0.05	[3]
case_5	81139875.0	203.62	0.04	[1]
case_6	338651050.0	220.79	0.09	[2]
case_7	131749294.0	56.72	0.57	[2]
case_8	24864639.0	105.55	0.05	[1]
case_9	50534100.0	235.49	0.04	[1]