

A2 Coursework

jn492

April 11, 2025

Word count: 2982 words

1 Introduction

This report is organized into three sections, each corresponding to a module, and supported by its respective Jupyter notebook. The notebooks can be executed using the provided Docker image by following the instructions in the accompanying `README.md` file.

2 PET-CT Image Reconstruction

Positron Emission Tomography (PET) and Computed Tomography (CT) are often used in combination, as they provide complementary information. While CT offers detailed anatomical structure by measuring attenuation coefficients $\mu(s)$, PET reveals regions of abnormal metabolic activity, such as tumors. Both techniques are governed by Equation 1, where CT reconstructs $\mu(s)$, and PET uses these coefficients to estimate the original emitted intensity I_0 from the tumor.

$$I = I_0 \cdot e^{-\int \mu(s) ds} \quad (1)$$

2.1 Ex.1: Clean up sinogram

The PET and CT sinograms are shown in Figure 1. Horizontal stripes are visible in the CT sinogram, indicating that certain projection angles are affected by noise across all detectors, likely due to beam-related errors.

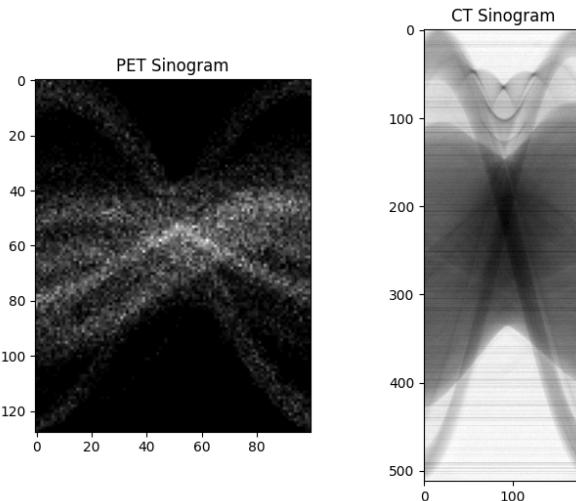


Figure 1: PET and CT sinograms

To correct for the noise, dark-field and flat-field corrections are applied. The dark field image accounts for detector noise with no incoming signal, while the flat field image captures systematic noise like uneven lighting [1]. Both corrections are applied using Equation 2.

$$\text{Corrected sinogram} = M((\text{sinogram} - \text{dark}) / (\text{flat} - \text{dark})). \quad (2)$$

The corrected sinograms, especially the CT sinogram, displayed in Figure 2 are visibly less noisy. This is likely due to dark field correction, which compensates for baseline detector offsets that affect all detectors at specific projection angles.

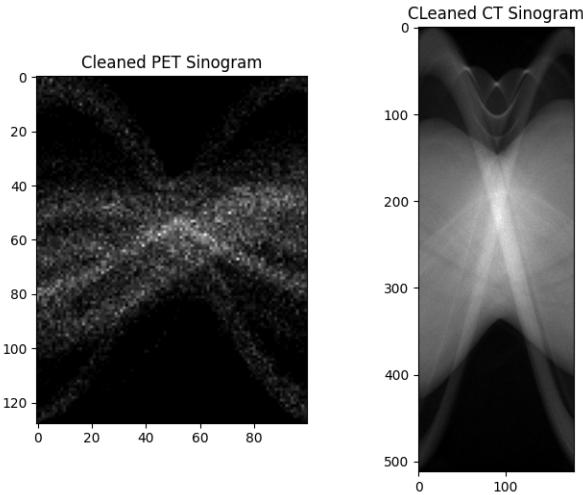


Figure 2: Cleaned PET and CT sinograms

2.2 Ex.2: Reconstruction of CT with FBP, OS-SART and SIRT

This section compares three CT reconstruction methods: Filtered Back Projection (FBP), Ordered Subsets Simultaneous Algebraic Reconstruction Technique (OS-SART) and Simultaneous Iterative Reconstruction Technique (SIRT).

FBP applies a high-pass filter to the sinogram and backprojects it via Fourier transform, distributing projection data across the image. OS-SART and SIRT are both iterative methods, such that the reconstruction is based on solving the linear system in Equation 3, where p is the sinogram, e is noise, μ is the target attenuation map, and A represents the projection geometry. OS-SART differs in that it computes each update with a batch of angles as stated in Equation 4, while SIRT processes all 180 angular measurements collectively before each update, as stated in Equation 5.

$$A\mu = p + \hat{e} \quad (3)$$

$$x^{k+1} = x^k + \gamma \cdot A_i^T (A_i x^k - b) \quad (4)$$

$$x^{k+1} = x^k + \gamma \cdot A^T (Ax^k - p) \quad (5)$$

Figure 3 presents a comparison between FBP and OS-SART using a fixed learning batch size of 10 projection angles and a constant iteration count of 5, while varying the learning rate (γ). The FBP reconstruction successfully produced sharp edges and well-defined details, but also introduces significant graininess. This graininess is likely a result from the high-pass filter used in FBP, which enhances high-frequency noise. In comparison, OS-SART reconstruction reveals blurry edges and poorly defined anatomical structures at $\gamma = 0.0001$ and 0.0005 , indicating insufficient convergence at this low learning rate. The best results obtained was from $\gamma = 0.001$ as larger steps are needed to overcome the noise and local minimums.

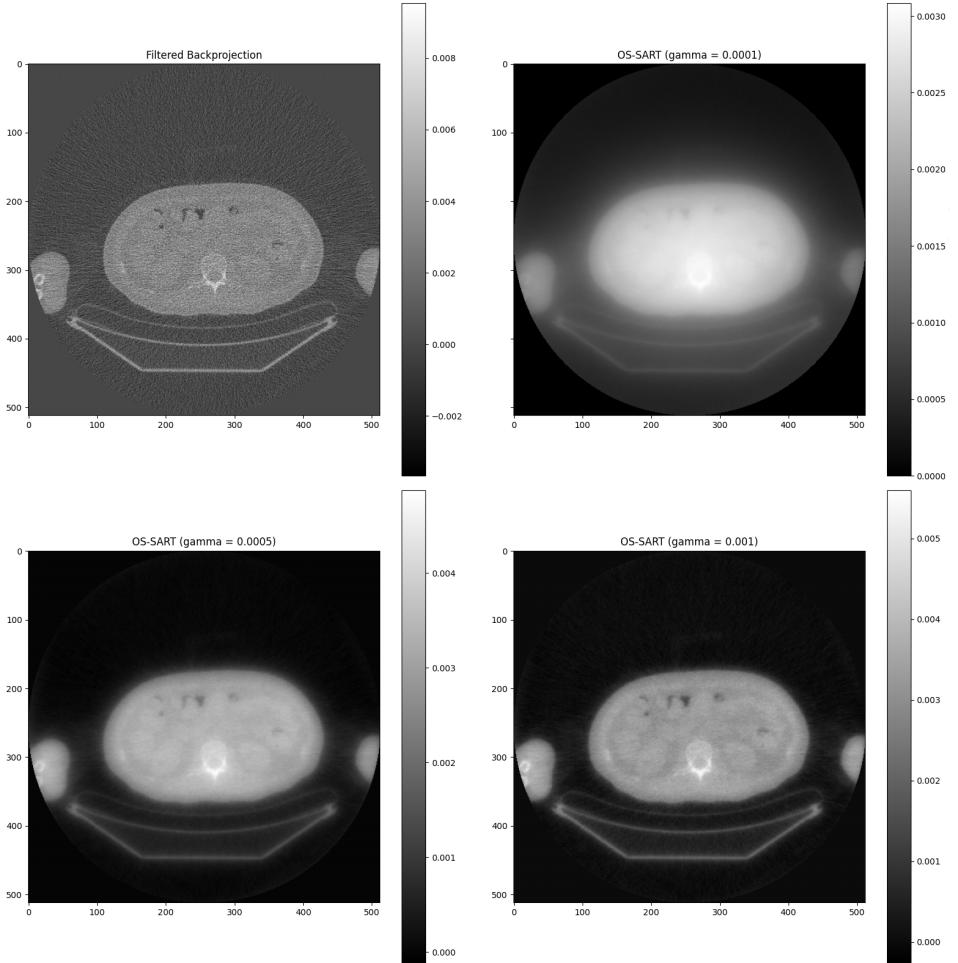


Figure 3: FBP vs OS-SART for differing learning rates

Using the optimal γ value of 0.001, I increased the number of iterations to allow the algorithm to converge to a better image. The results in Figure 4 show that as the number of iterations increase from 10 to 30, the boundaries between the pelvis and background become significantly more defined as the algorithm converges. The OS-SART reconstruction performs better than FBP at 30 iterations as it is able to smooth out the noise while retaining anatomical details.

However, at 50 iterations, image graininess increases as more noise is introduced. This phenomenon aligns with findings from Ghetti et al. [2] and Zeraatkar et al. [14], both who observed that excessive iterations can introduce noise accumulation. Without regularization, the algorithm likely begins overfitting to the noise present in sinograms, thus attempting to reconstruct measurement errors rather than the target anatomical features.

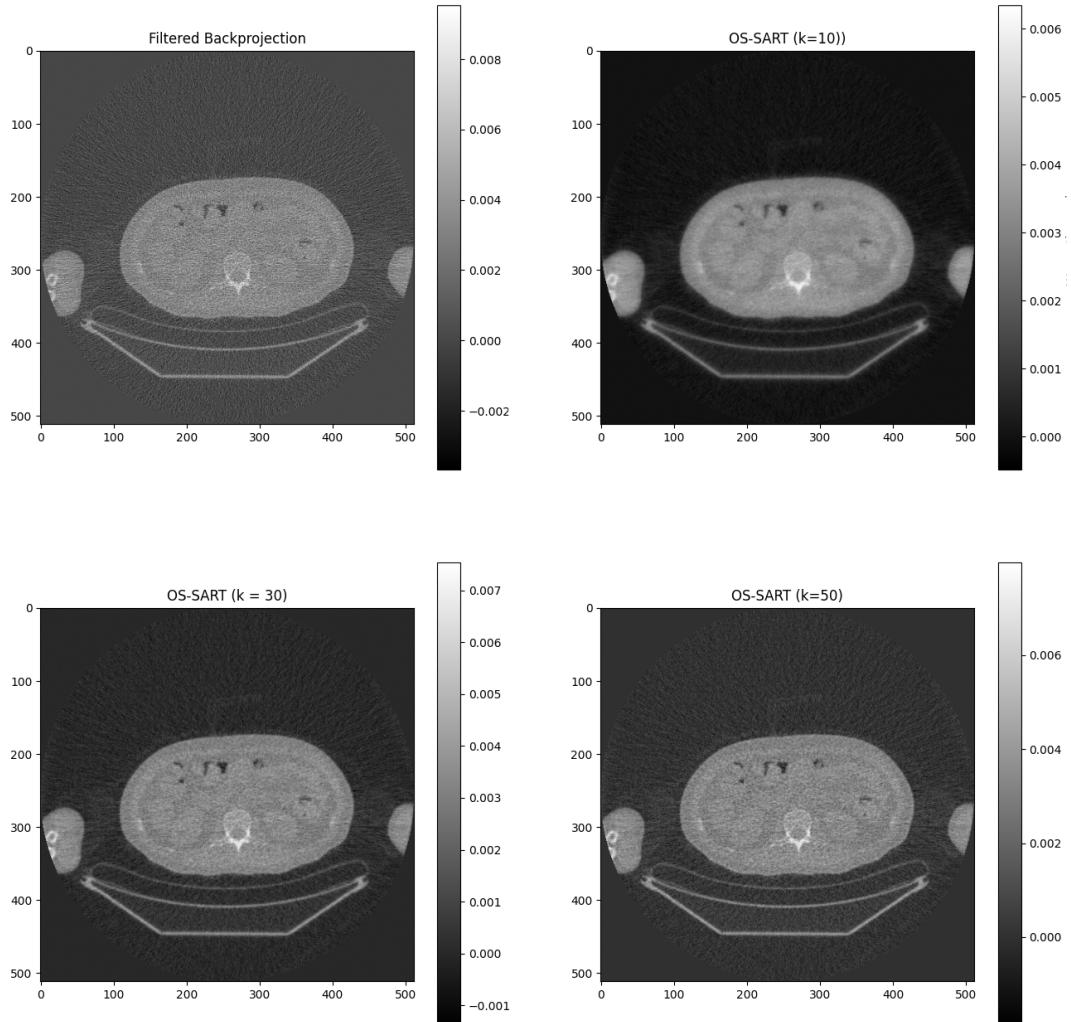


Figure 4: FBP vs OS-SART with different number of iterations

Comparing the OS-SART reconstructed image ($\gamma = 0.001$, `iterations` = 30) to a SIRT image using the same γ value and number of iterations, SIRT produced a significantly smoother but blurrier image, as demonstrated in Figure 5.

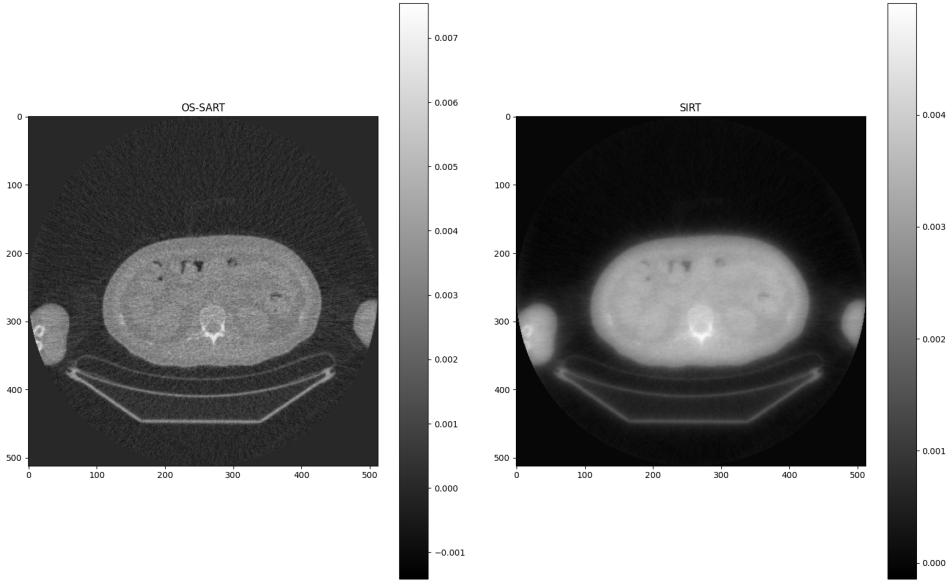


Figure 5: OS-SART and SIRT ($\gamma = 0.001$ and iteration = 30)

When more iterations were tested, SIRT only obtained a comparable outcome to OS-SART with $\gamma = 0.001$ and iteration = 100, as shown in figure Figure 6.

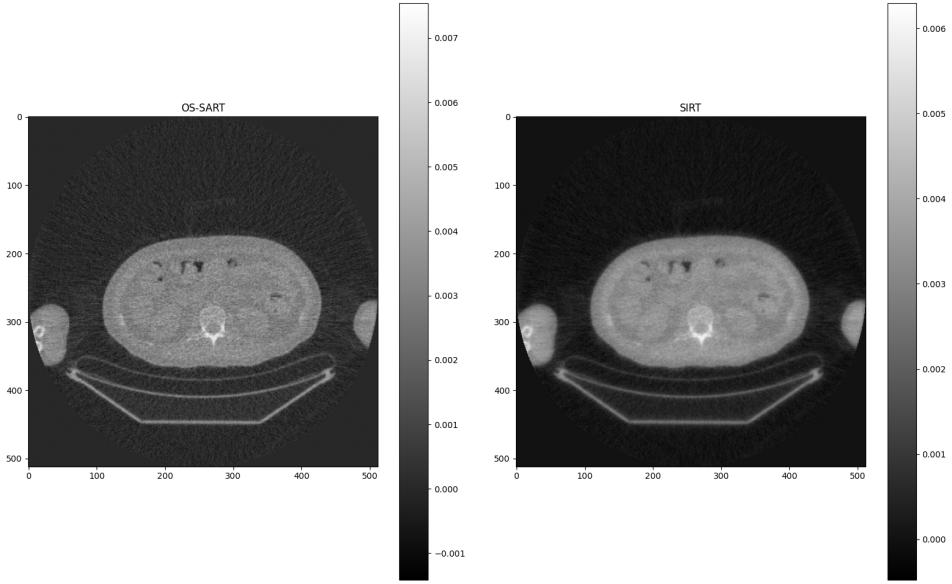


Figure 6: OS-SART (Iteration = 30) vs SIRT (iteration = 100)

OS-SART converged 3.5 times faster than SIRT (90.5s vs. 308s) while achieving comparable image quality. This aligns with Shi et al. [10], who attribute OS-SART's improved performance to its use of projection subsets, enabling faster updates and reduced noise propagation.

2.3 Ex.3: Attenuation correction of PET

PET Attenuation correction is essential because gamma photons travel through various tissues with different densities before reaching the detector, thus affecting our ability to accurately locate the photon source.

Due to resolution differences between imaging modalities, the CT image dimensions is downsized by a factor of 4.24/1.06 to match PET's lower resolution before forward projecting into the new sinogram.

The attenuation corrected PET sinogram is shown in Figure 7.

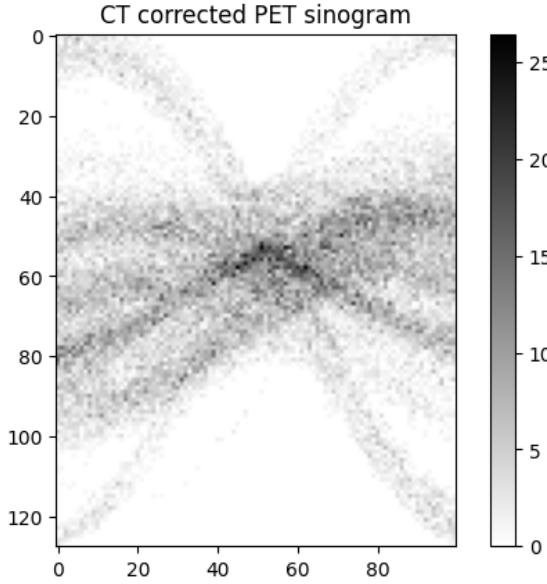


Figure 7: Attenuation corrected PET

2.4 Ex.4: Reconstruction of PET with FBP, OSEM and MLEM

The attenuation-corrected PET data can be reconstructed using methods like FBP, Ordered Subset Expectation Maximization (OSEM), and Maximum Likelihood Expectation Maximization (MLEM). Both OSEM and MLEM operate by maximizing the likelihood of observing the measured sinogram given the estimate of the image, but while OSEM updates after processing a subset of projection angles, MLEM uses all projection data at once. For instance, if OSEM was ran without subsets, then it will produce the same image as MLEM as shown in Figure 12.

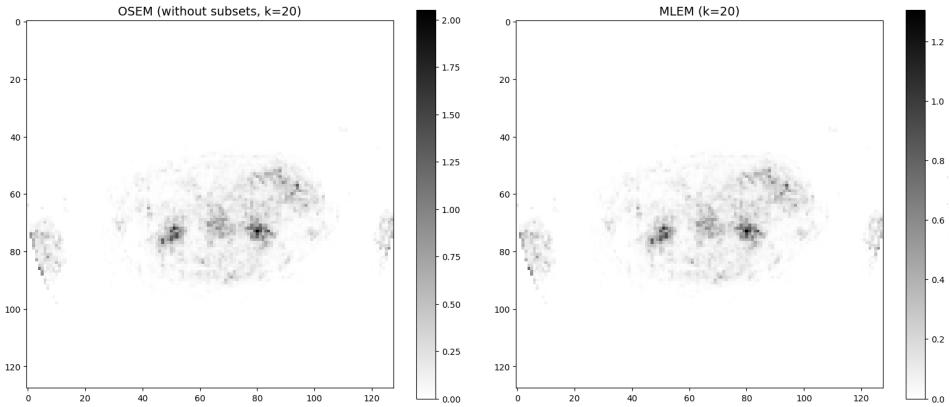


Figure 8: OSEM (no subsets) and MLEM reconstructions

To compare FBP and OSEM, I used 10 subsets, $\gamma = 0.001$, and 20 iterations for the reconstruction in Figure 9. It is clear that OSEM greatly outperforms FBP, as it contains much less noise and streaking artefacts.

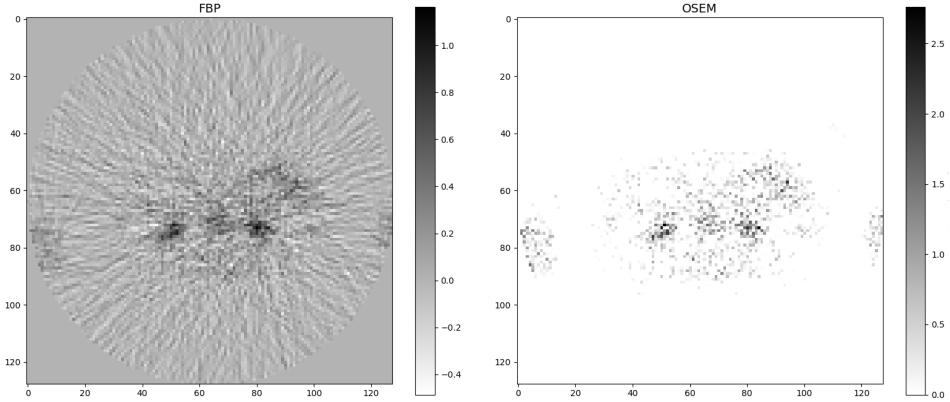


Figure 9: FBP and OSEM reconstructions

Using the same number of iterations and γ value, I reconstructed the PET image with both MLEM and OSEM, which lead to OSEM producing noticeably better results, as shown in Figure 11. This is likely because OSEM is less susceptible to noise from the full projection data.

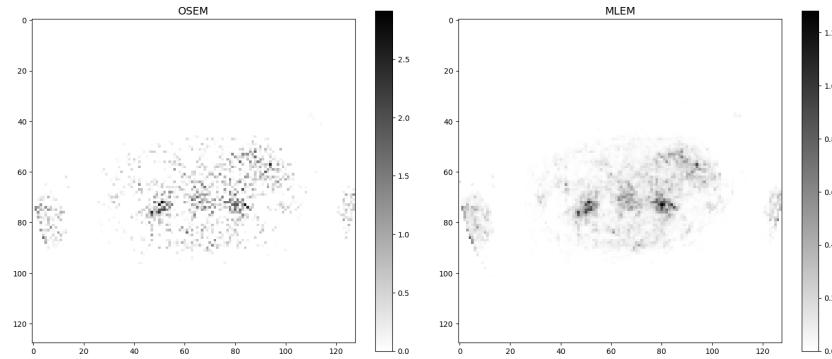


Figure 10: OSEM and MLEM reconstructions

The reconstruction using MLEM gradually improved at 50 iterations, as shown in Figure 11. However, little improvement can be seen with further increasing the number iterations to 200. In terms of performance, OSEM (1.90 seconds) converged to the same image quality 3 times faster than MLEM (5.72 seconds).

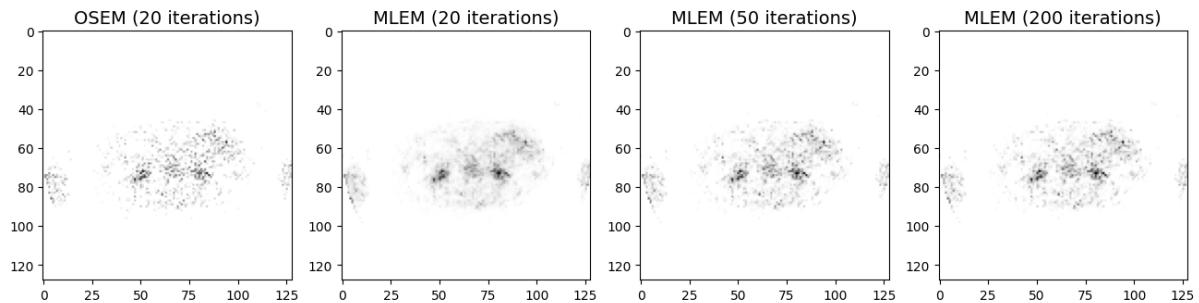


Figure 11: MLEM reconstructions with increasing number of iterations

2.5 Ex.5: Theory

1. The overlaid PET and CT images are shown in Figure 12. The PET reconstruction is displayed in color, with lighter yellow indicating higher activity levels, overlaid on the anatomical structure provided by the CT scan.

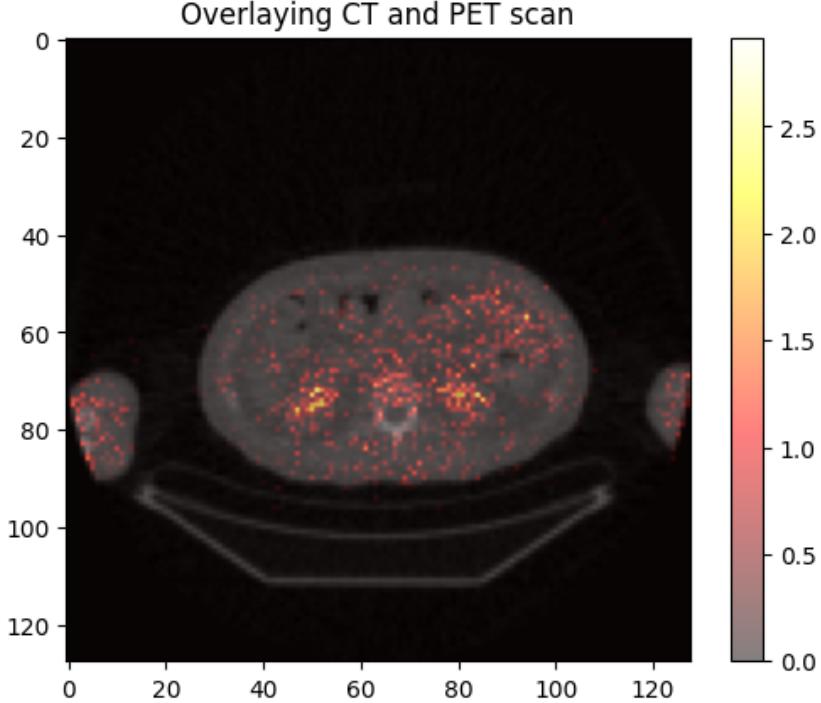


Figure 12: PET and CT overlaid

2. In theory, if technology advances to the point where we can precisely measure the time difference between the two gamma photons produced during annihilation along the line of response (LOR), we could directly infer the annihilation point without needing iterative reconstruction [7]. According to Lecoq et al. [7], achieving accurate 3D localization would require a time resolution of at least 10 picoseconds. Only then could reconstruction become unnecessary.

3. OSEM's Poisson likelihood model aligns well with PET data's statistical nature [11]. While both PET and CT data follow Poisson distributions, CT employs higher-intensity externally controlled X-ray beams, yielding higher count rates and resolution. PET relies on limited radiotracer quantities with inevitable detection losses, resulting in sparser, noisier data that benefits significantly from OSEM's statistical approach. As CT has higher signal-to-noise ratio, simpler analytical methods like FBP is sufficient [4] for reconstruction.

Mathematically, OSEM is derived from a Poisson likelihood model, aiming to maximize the probability of observing the measured sinogram b , given the current image estimate x^k , as seen in Equation 6 [3]. The algorithm adjusts the image so that the sinogram estimate $A_i x^k$ becomes more consistent with the actual PET data. The equation shows that OSEM does a multiplicative update, and operates in batches (A_i).

$$x^{k+1} = x^k A_i^T \left(\frac{b}{A_i x^k} \right) \quad (6)$$

In contrast, the update step in gradient descent shown in Equation 7 subtracts the gradient of a loss function to iteratively minimize it. Traditional gradient descent uses subtraction, and does not operate in batches.

$$x^{k+1} = x^k - \gamma \nabla \mathcal{L}(x^k) \quad (7)$$

4. PET-MR produces images with anatomical and functional features, similar to a PET-CT scan. PET sinograms require attenuation correction to produce accurate images. However, MRI intensities are not directly related to tissue density or atomic number, as they reflect magnetic signals primarily from water molecules [6]. This makes it difficult to derive attenuation information directly from MRI data. An additional step is to segment the MR image into different tissue and bone types, then assign predefined attenuation coefficients to each region. Alternatively, deep learning approaches could be used to convert the MRI image into a CT-like image [9].

3 MRI denoising

3.1 Ex 2.1: Visualisation and identifying noise

The k-space data was loaded, and the first dimension (index 0) was found to represents the 6 coils. The k-space of each coil and image space of coil 1 is then plotted in Figure 13 and Figure 14.

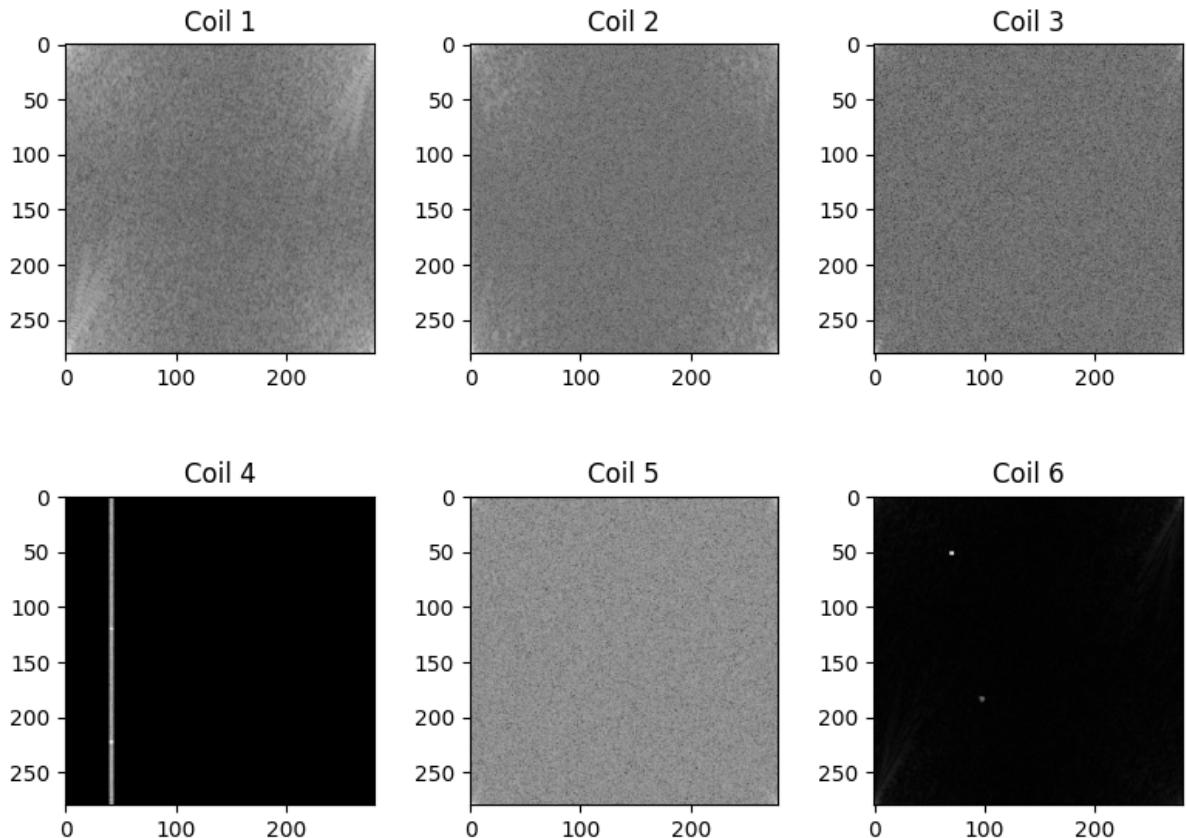


Figure 13: Magnitude of k-space for all coils

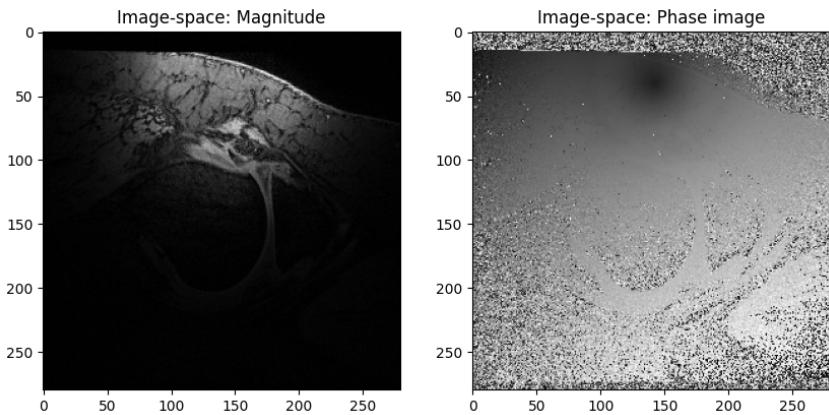


Figure 14: Magnitude and Phase plotted for coil 1

The magnitude is then plotted for all the coils, as shown in Figure 15.

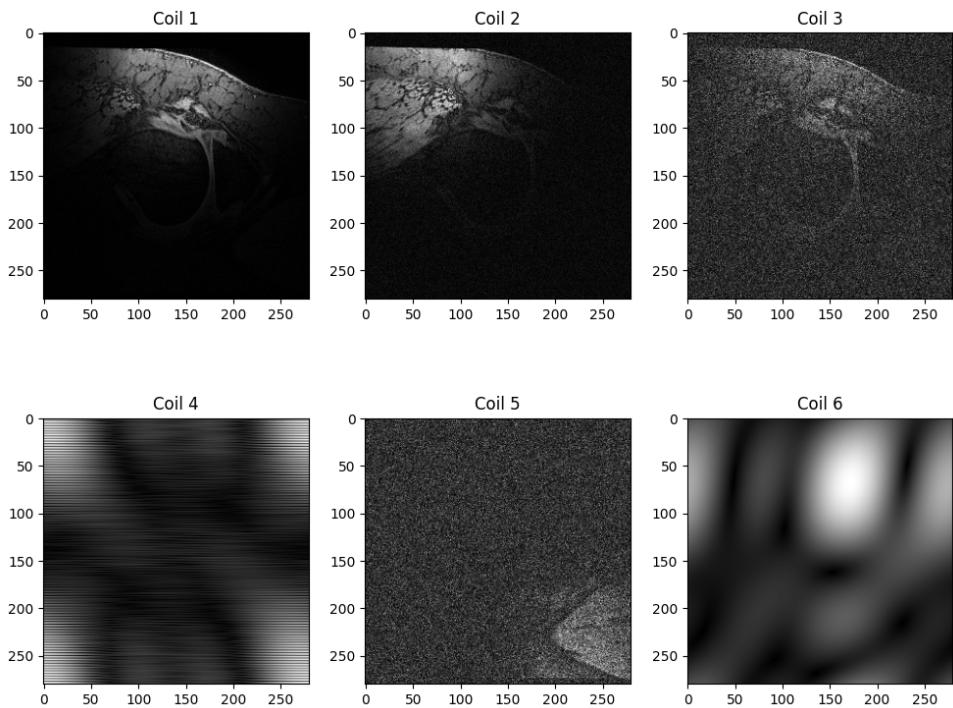


Figure 15: Magnitude plotted for all coils

All the coils were then combined into a single image by squaring the image from each coil, summing the magnitudes, then square rooting the final result, as shown in Figure 16.

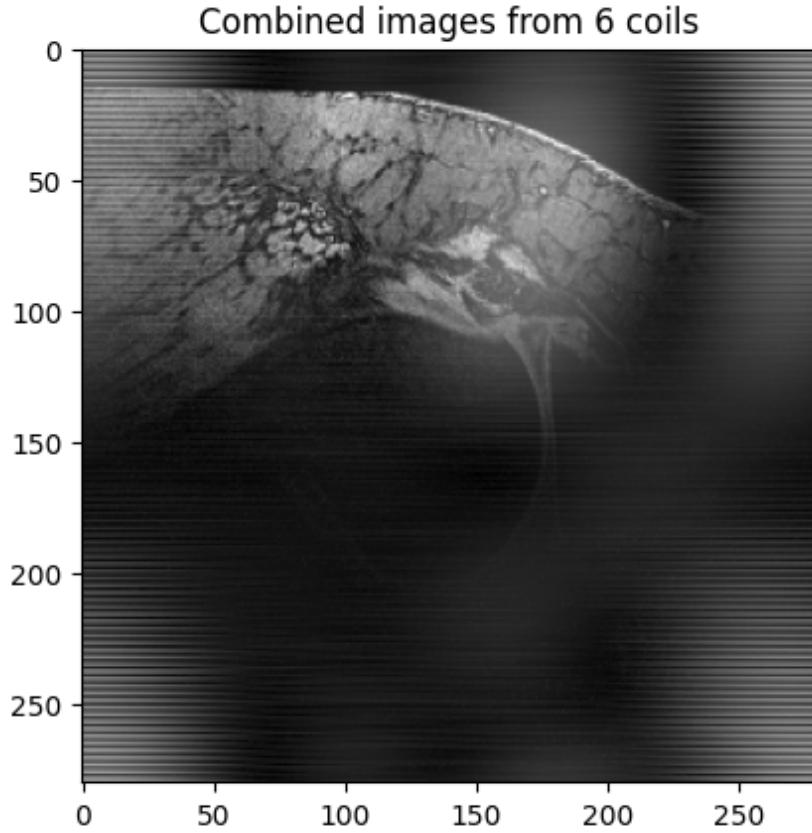


Figure 16: Final result from combining the images from call coils

The combined image reveals some knee joint details but suffers from significant artifacts. These include striping, non-uniform brightness, and graininess, which is likely a result of the variations in coil reception quality. The blur and graininess likely originate from coils 3 and 5, while the horizontal striping appears to stem from coil 4, potentially indicating gradient non-linearity, RF coil malfunction, or magnetic field inhomogeneity [8]. Coil 6 appears particularly problematic, capturing only shadows and highlights, suggesting possible malfunction.

3.2 Ex 2.2: Removing noise

I have chosen to evaluate three image denoising techniques, namely Mean, Gaussian and Bilateral filters, which are applied after the images are fourier transformed into the image space.

The mean filter is a low-pass filter defined by averaging the pixel values within a defined kernel window and assigning the result to the center pixel. Figure 17 demonstrates the results of the Mean filter with a kernel size of 3. This filter produces a noticeable blurring effect, creating smoother tissue structures but reducing edge definition throughout the knee image.

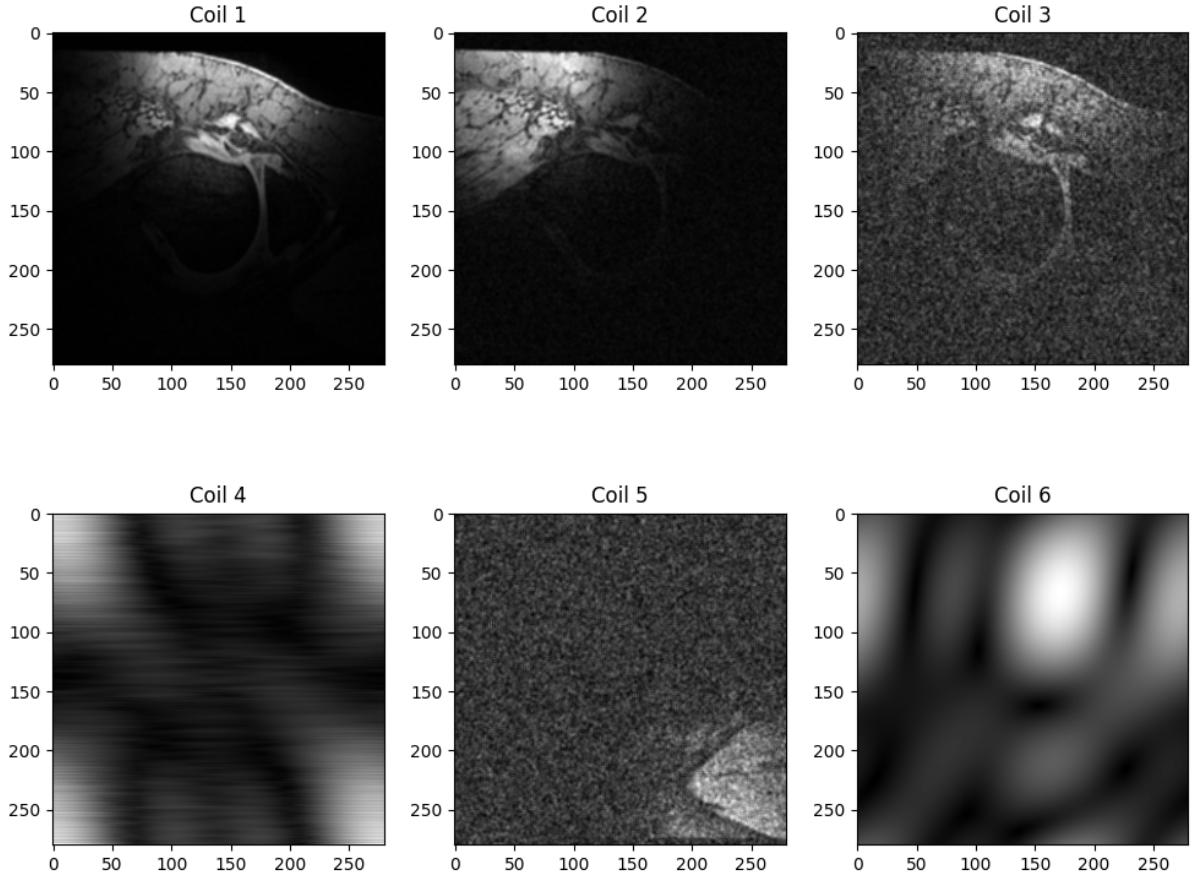


Figure 17: Mean Filter

The Gaussian filter is also a type of low-pass filter, but rather than applying a uniform blur to neighboring pixels, it assigns greater weight to pixels closer to the center. Using $\sigma = 1$, I implemented a 9×9 kernel where weights follow the Gaussian function in Equation 8.

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (8)$$

x and y represent coordinates relative to the kernel's center. The filtered value is thus calculated as the weighted sum of neighborhood pixel values. As shown in Figure 18, this filter produced a smoothed image with blurred edges, yielding results visually comparable to the Mean filter.

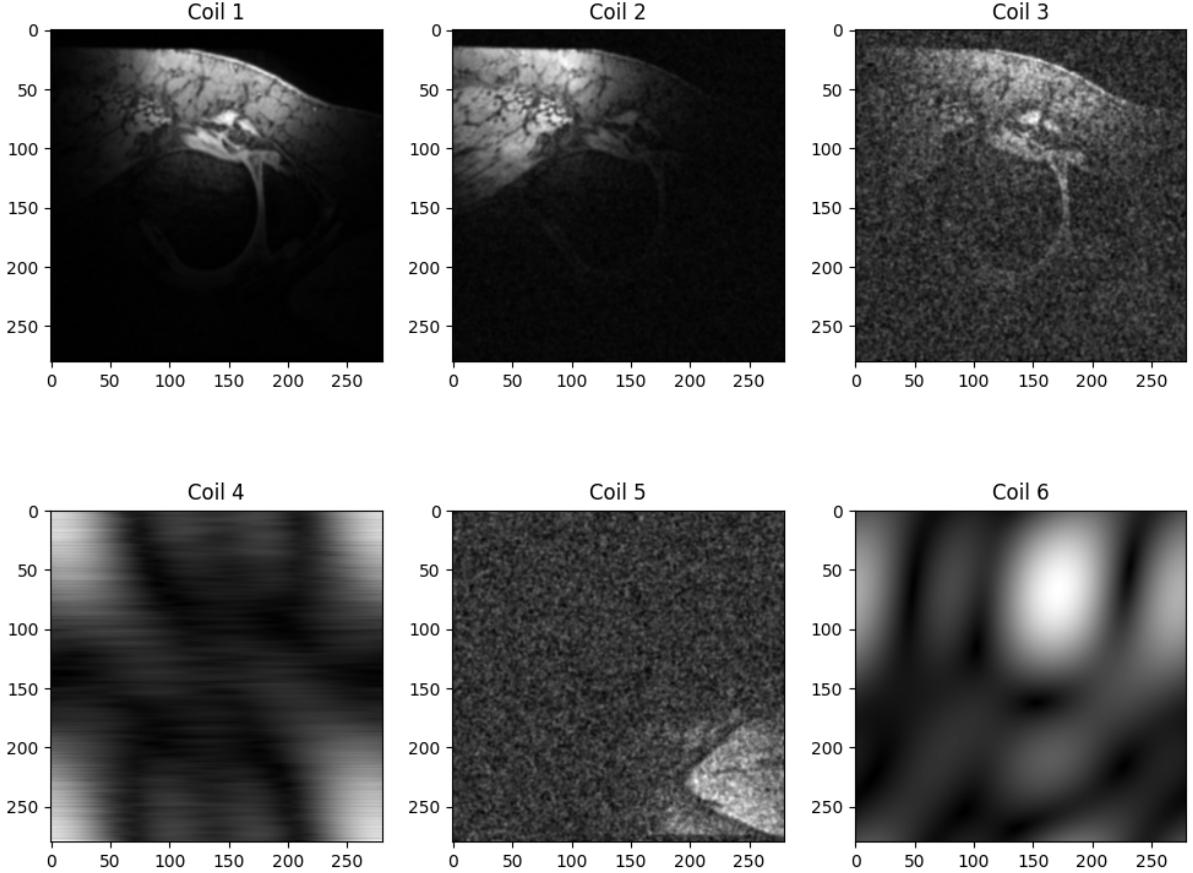


Figure 18: Gaussian Filter

Finally, I applied bilateral filtering, which follows Equation 9. I used a 5×5 filter window with parameters $d = 5$, $\sigma_{colour} = 10$ and $\sigma_{space} = 5$. d controls the size of the square neighborhood considered, σ_{colour} determines intensity similarity (lower values average only similar intensities, higher values average across different intensities), while σ_{space} governs spatial influence (lower values prioritize pixels near the center).

$$I_{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega(d \times d)} I(x_i) \cdot \exp \left(-\frac{\|x_i - x\|^2}{2\sigma_{\text{space}}^2} \right) \cdot \exp \left(-\frac{\|I(x_i) - I(x)\|^2}{2\sigma_{\text{colour}}^2} \right) \quad (9)$$

This technique produced the best results as shown in Figure 19. This technique effectively preserves edges while smoothing noise. The moderate σ_{space} and relatively low σ_{colour} combination I selected preserves high-contrast features like tissue boundaries while allowing moderate smoothing within the 5×5 window, creating an optimal balance between noise reduction and structural preservation.

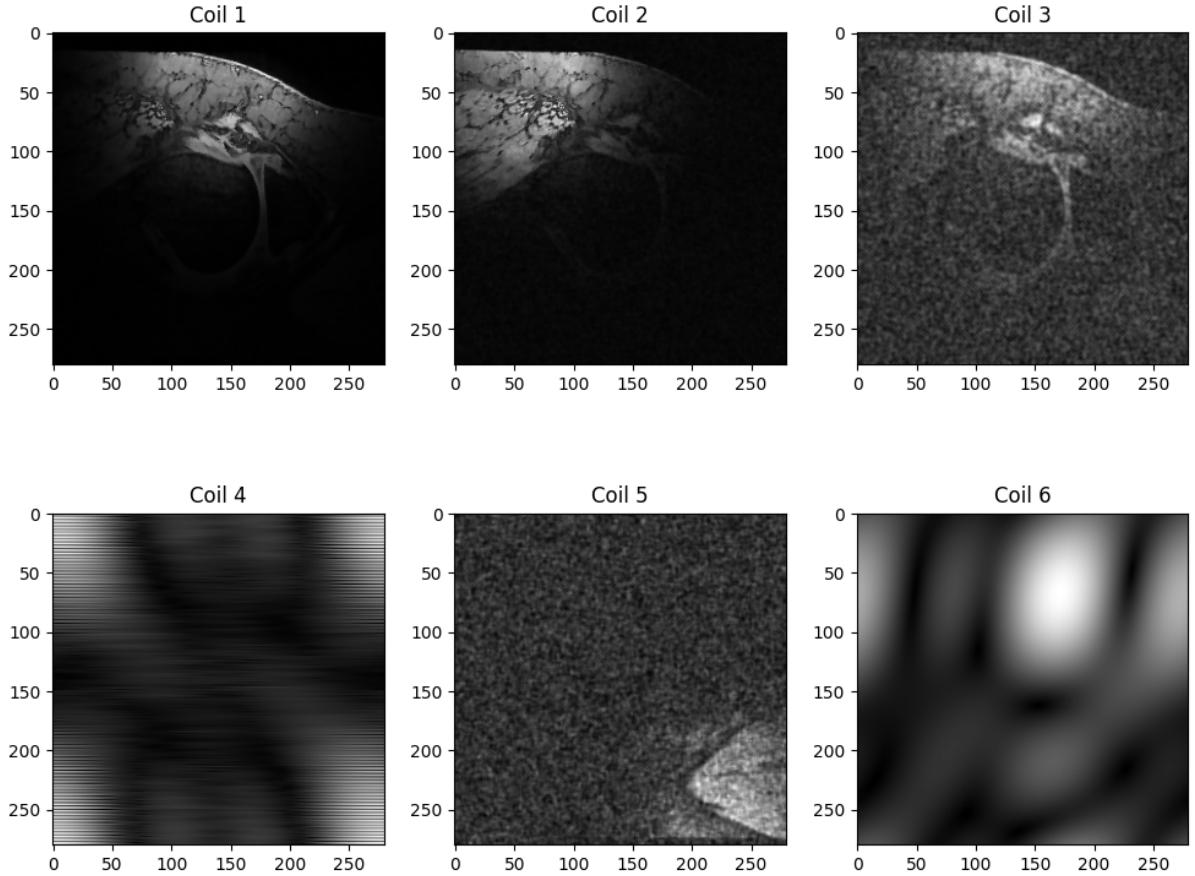


Figure 19: Bilateral Filter

The Butterworth filter offers optimal balance between attenuation and phase response by operating directly in k-space [5]. Its key parameters are D_0 (cutoff frequency) and n (transition steepness), which maintains smooth transitions to avoid the ringing artifacts common with other frequency-domain filters. Figure 20 displays a low pass butterworth filter with $D_0 = 30$ and $n = 2$.

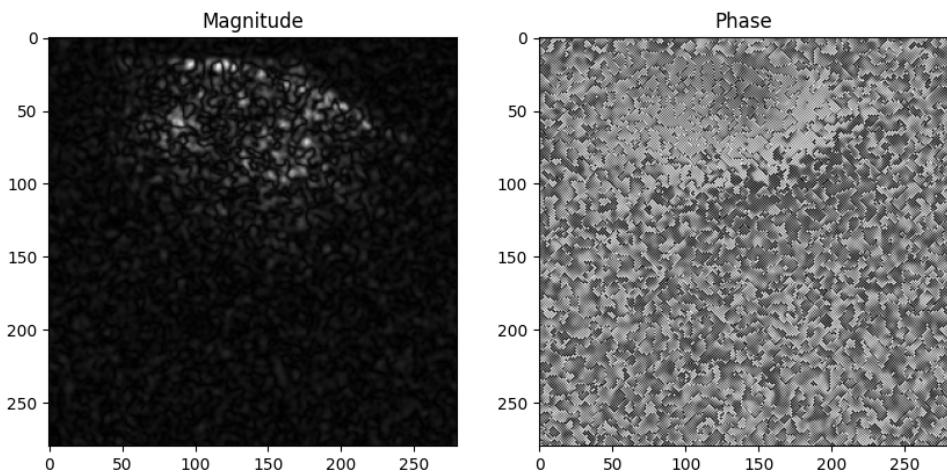


Figure 20: Low Pass Butterworth filter ($D = 30$, $n = 2$) acting on the k-space and fourier transformed into an image

The image quality with $D_0 = 30$ and $n = 2$ is poor. This is likely due to the low cutoff frequency eliminating essential high-frequency components responsible for image detail and contrast. Thus only

dark, low frequency information is preserved and shown in the image. Compared to the image based methods that caused mostly image blurring and smoothing, this frequency domain approach completely removed critical image components. To determine optimal parameters, I systematically tested various D_0 and n combinations, as shown in Figure 21. The $D_0 = 120$ and $n = 2$ obtained a good result in reducing the noise in the image while keeping the edges and details. It achieved better denoising results than the image-based techniques, as the high-frequency noise is removed.

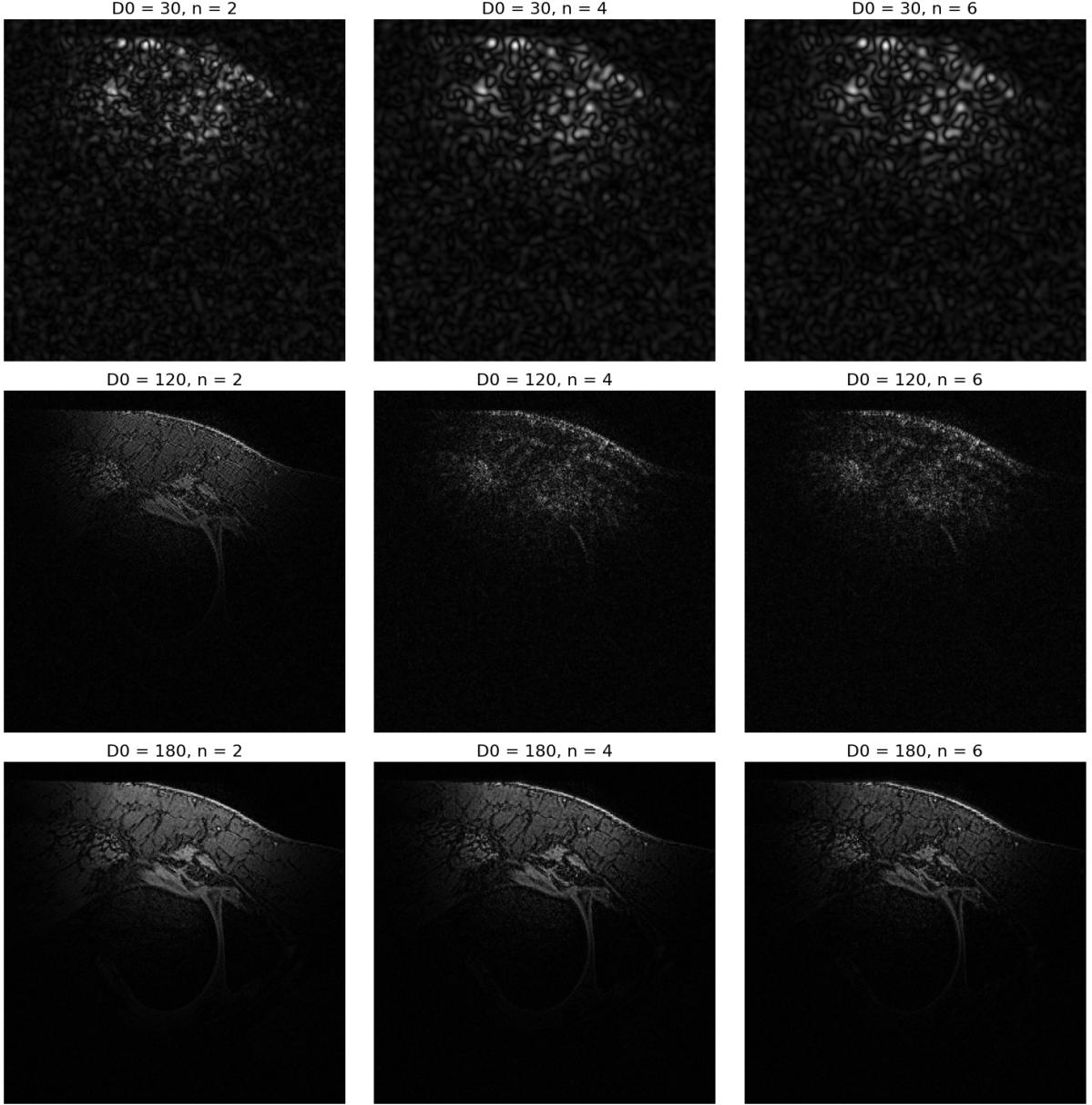


Figure 21: Butterworth filters with different D_0 and n values

Using the mean filter, I generated a new combined image of all the coils, as shown in Figure 22. However, this approach did not effectively reduce noise in certain coils, resulting in persistent uneven shading and visible zebra-like lines. This limitation arises because the mean filter simply performs spatial averaging, which smooths the image uniformly without specifically targeting high-frequency noise or very low-frequency artifacts such as shadows. In this context, a frequency-domain filters like the Butterworth filter may be more appropriate, as it allows selective attenuation of undesirable frequency components while preserving important structural details.

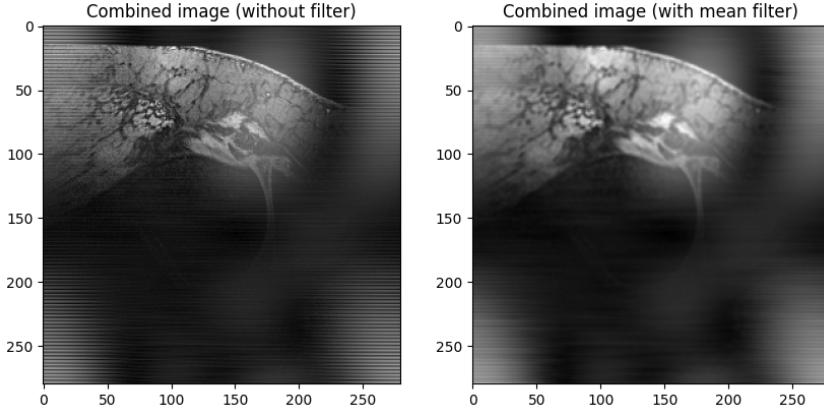


Figure 22: Combined mean filter images

To improve the quality of the combined image, I would selectively exclude faulty coils 4 and 6. Coil 4 exhibits clear zipper artifacts indicative of RF interference, possibly from an open door during acquisition, resulting in dark shadows and zebra-line patterns. Coil 6 shows a predominantly black k-space with scattered white dots, suggesting RF coil malfunction. This may have stemmed from decoupling failures, creating the black-white blob patterns.

Secondly, I could use a different coil combination method to obtain better resolution. For example, implementing coil-sensitivity weighting would prioritize signals from better-performing coils while minimizing contributions from noisy ones like coil 3, effectively reducing graininess and improving signal-to-noise ratio.

Thirdly, I could explore more advanced denoising methods such as Principal Component Analysis (PCA) and deep learning-based approaches. PCA works by decomposing the image data into orthogonal components, ranked by variance. Noise, which tends to contribute low variance, can be suppressed by discarding or zeroing out the less significant components.

An alternative would be deep learning models, where synthetic noise is added to images, and algorithms like U-Net learn to remove the noise. This data-driven approach allows the model to generalize well to real-world noise patterns and artefacts not easily handled by classical filters.

4 CT Image segmentation and Classification

4.1 Image segmentation

To begin segmentation, I extracted a subvolume from the full 3D CT image, focusing on the tumor region indicated by the mask. I organized the CT scans and masks by patient ID, and for each case, computed the tumor's bounding box with added padding (30, 30, 5 voxels in x, y, z) to provide spatial context. These coordinates were used to slice the original scan and isolate the subvolume.

I then applied a simple intensity-based thresholding algorithm to identify potential tumor regions based on characteristic intensity values. This method can also be used to detect additional suspicious areas within the same scan, as illustrated in Figure 23.

```

def threshold_segmentation(image_to_be_thresholded, scan, mask):
    """
    Applies a simple intensity-based thresholding algorithm to segment a region of interest.

    The method uses the intensity values from within the tumor region (as defined by the ground truth mask)
    to establish a minimum and maximum intensity range. It then applies this threshold to the
    provided image (typically a subvolume) to generate a binary segmentation mask.

    Args:
        image_to_be_thresholded (numpy array): The image on which thresholding is applied.
        | This is typically a subvolume of the full scan.
        scan (numpy array): The subvolume CT image from which intensity thresholds are derived.
        mask (numpy array): The binary ground truth mask corresponding to the same subvolume.

    Returns:
        segmented_image (numpy array): A binary mask where voxels within the tumor intensity
        range are labeled as 1, and all others as 0.
    """

    # Obtaining all the voxels in the scan where the mask is non-zero (tumor is present)
    inside_values = scan[mask > 0]

    #Find the min and max intensities of these voxels
    min_val = inside_values.min()
    max_val = inside_values.max()

    #If voxel in image is within range of tumor values, then set to 1.
    segmented_image = ((image_to_be_thresholded >= min_val) &
                       (image_to_be_thresholded <= max_val)).astype(np.uint8)

    return segmented_image

```

Figure 23: Simple thresholding function

The performance of the threshold-based segmentation algorithm varied significantly across different CT images. In cases where there is substantial overlap in voxel intensity values between the tumor and surrounding non-tumorous tissues, the algorithm struggles to differentiate between them. As a result, the segmented output poorly aligns with the ground truth, as illustrated in Figure 24.

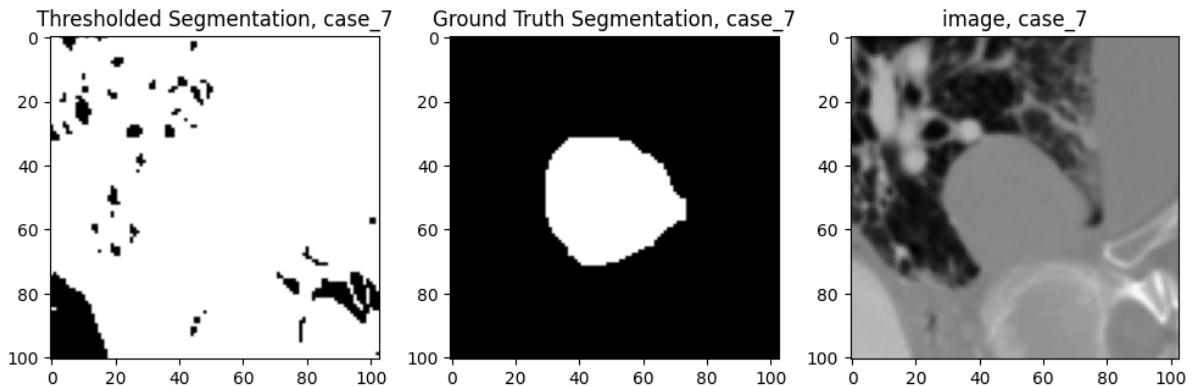


Figure 24: Poor segmentation performance for case 7

In contrast, when the tumor exhibits strong intensity contrast against the background, such as in cases with clearly distinct voxel values, the segmentation performs better, producing a more accurate outline of the tumor, as shown in Figure 25. However, even in these cases, the method is prone to false positives, as other anatomical structures or tissues with voxel values similar to the tumor are also incorrectly classified as tumor regions.

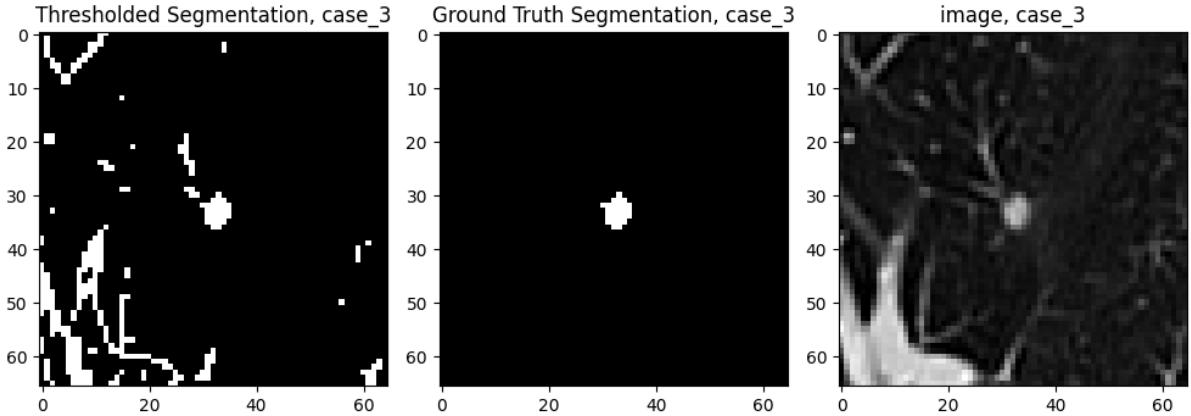


Figure 25: Better segmentation performance for case 3

The thresholding algorithm has several limitations. One key issue is the presence of noise within the tumor's intensity range, which makes the minimum and maximum thresholds overly sensitive to high-frequency outliers. This can lead to misclassification, particularly where bright, high-attenuation structures like bone are mistakenly identified as tumor regions. To improve, a percentile-based thresholding could be used to exclude extreme outliers, given an appropriate percentile is chosen to reflect typical tumor intensity.

Another challenge is that the intensity values of lung tumors can vary significantly depending on their type and stage. For instance, Yang et. al reported that pure ground-glass nodules occurring in lungs often show brighter intensity if malignant, and weaker intensity if benign [12]. As a result, using a single threshold range for all tumor types can lead to inaccurate segmentation. To address this, perhaps it is beneficial to separate the training data by class, then training a deep learning model to recognise the underlying patterns of each class.

4.2 Exercise 3.2: Image feature extraction and classification

Energy, Mean Absolute Deviation (MAD), and Uniformity are commonly used metrics to quantify tumor intensity characteristics. These are computed using the functions shown in Figures 26, 27, and 28, respectively.

```
# ENERGY FUNCTION
def energyfunc(list_of_intensities):
    """
    This function calculates energy,
    which is computed as the sum of squared intensities,
    such that higher energy means more intense or prominent features in the image region.

    Args:
        list_of_intensities (numpy array): The list of intensity values from the region of interest (tumor).

    Returns:
        energy: a metric called energy to quantify the intensity of the voxels.
    """
    energy = np.sum(list_of_intensities**2)
    return energy
```

Figure 26: Energy equation

```

# MAD FUNCTION
def madfunc(list_of_intensities):
    """
    This function calculates the mean difference between all intensity values and the Mean Value of the set of voxel intensities

    Args:
        list_of_intensities (numpy array): The list of intensity values from the region of interest (tumor).

    Returns:
        mad: a metric called mean absolute deviation to quantify the spread of the intensity values.
    """
    mean_value = np.mean(list_of_intensities)
    mad = np.mean(np.abs(list_of_intensities - mean_value))
    return mad

```

Figure 27: MAD equation

```

# UNIFORMITY FUNCTION
def uniformityfunc(image, bins=64):
    """
    Calculates the uniformity of an image based on its intensity histogram.

    Uniformity is a measure of how concentrated or spread out the intensity values are.
    It is computed as the sum of squared values from the normalized histogram.
    A lower score means intensities are more evenly distributed and vice versa.

    Example:
        - If 90% of pixels fall into bin 1 and 10% into bin 2:
            p1 = 0.9, p2 = 0.1
            Uniformity = p1^2 + p2^2 = 0.81 + 0.01 = 0.82 (high uniformity)

        - If pixels are evenly distributed across bins, low uniformity.

    Args:
        image (ndarray): The input image or region of interest.
        bins (int): Number of histogram bins to use.

    Returns:
        float: The uniformity score (sum of squared normalized histogram values).
    """
    hist, _ = np.histogram(image, bins=bins, range=(-1024, 1733), density=False)
    probs = hist / np.sum(hist) # p_i = P(i)/N
    uniformity = np.sum(probs ** 2) # sum

    return uniformity

```

Figure 28: Uniformity equation

To ensure meaningful uniformity values I first assessed the intensity range to select a meaningful bin count that balances resolution and noise. While too many bins would make the histogram overly sensitive to noise, too few could obscure important variations. Figure 29 shows that intensity ranges vary significantly, with some spanning the full range (-1024.0 to 1733) while others occupied narrow bands (-743 to -306). Based on this analysis, I selected 64 bins to balance sensitivity and noise resilience, which corresponded to approximately 43 intensity units per bin.

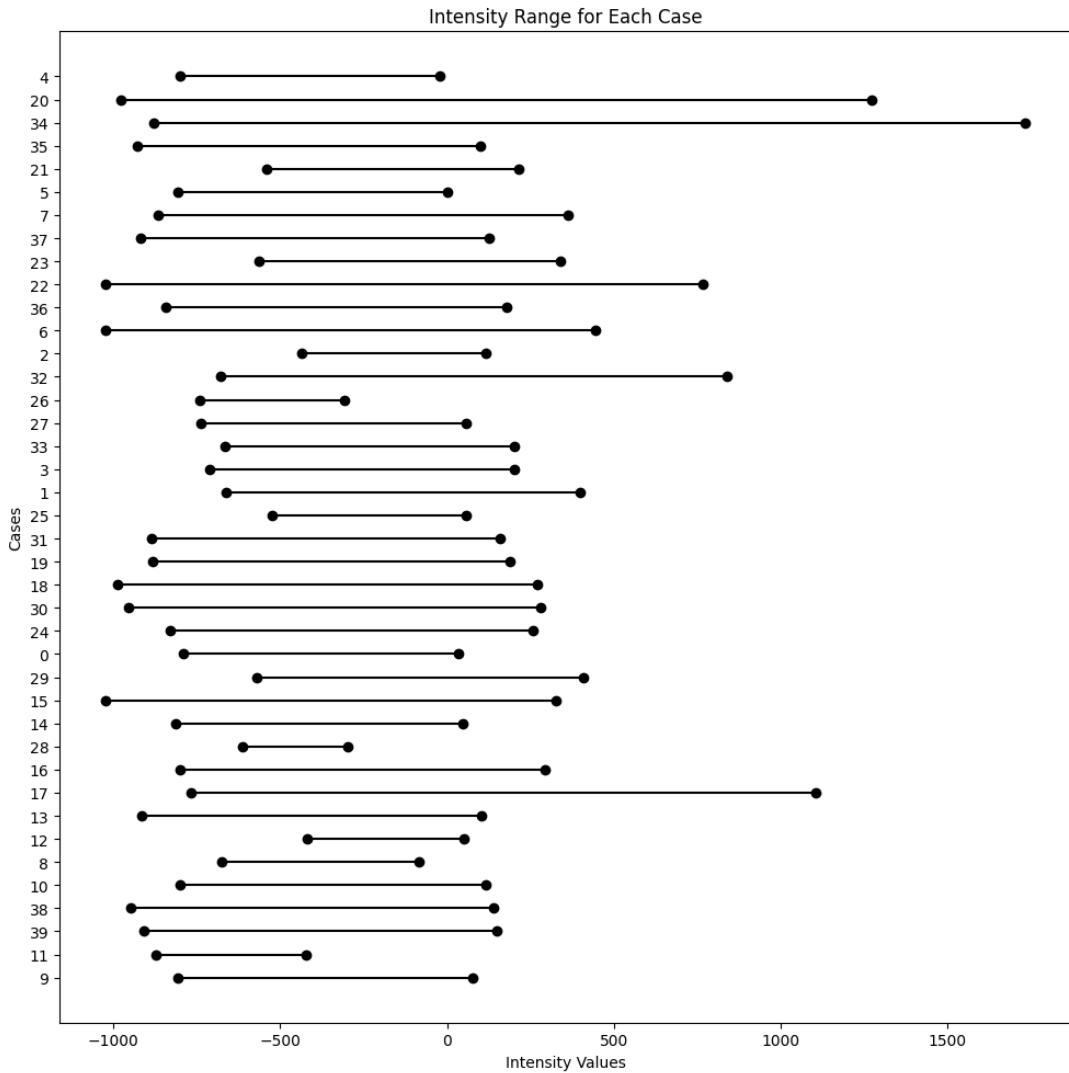


Figure 29: Range of intensity values for all cases in a number line

Figure 30, 31, and 32 show the energy, mad and uniformity values and plotted per case. The blue dotted line is the 50th percentile for MAD and Uniformity values, which will be used and discussed in later paragraphs.

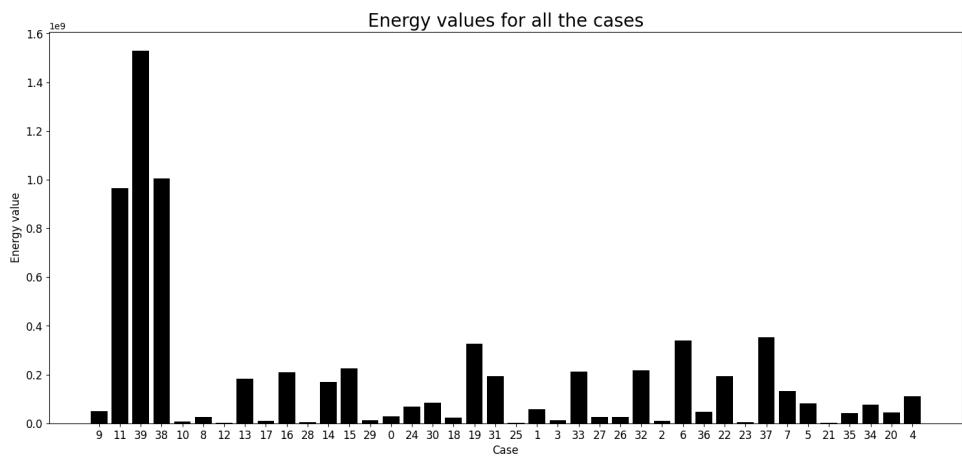


Figure 30: Energy values for all cases

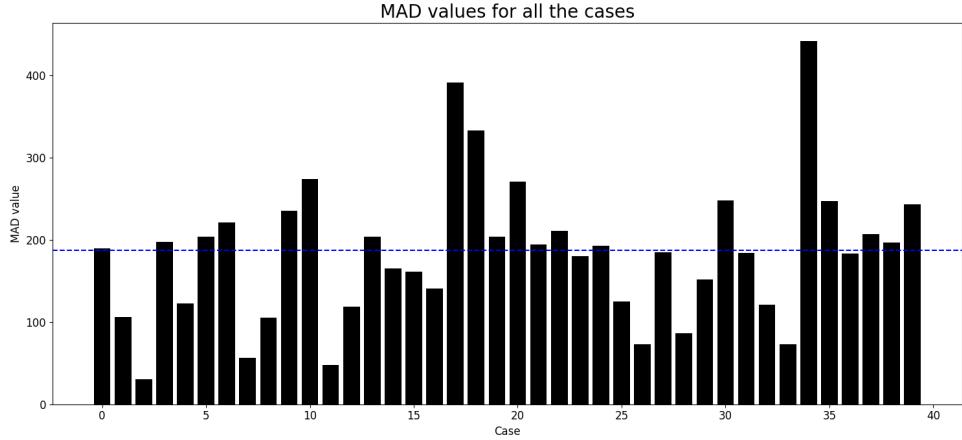


Figure 31: MAD values for all cases

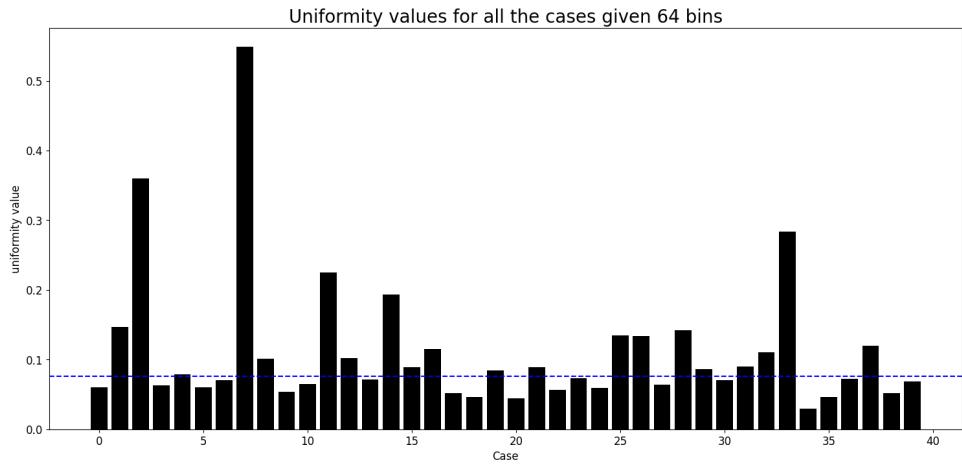
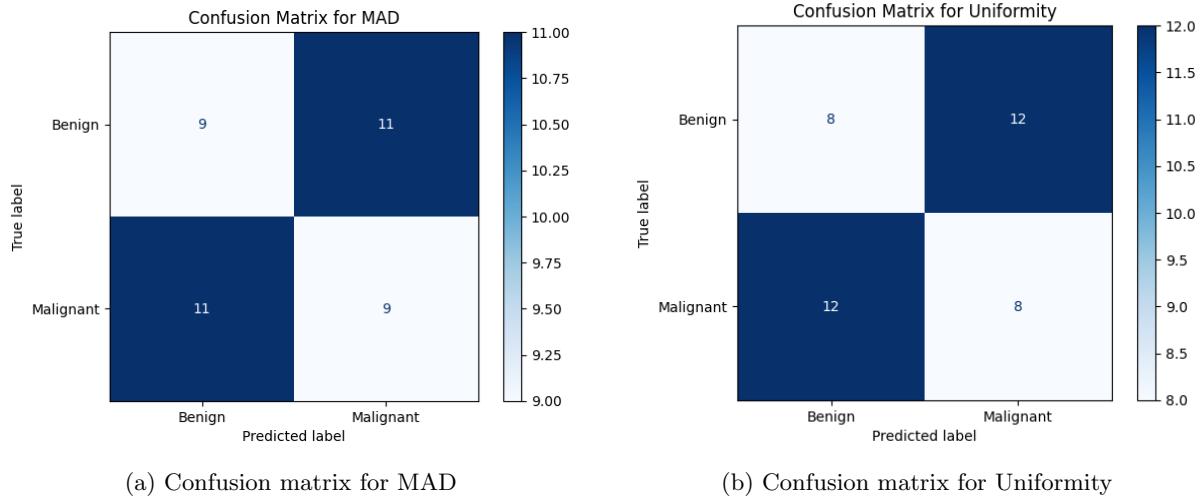


Figure 32: Uniformity values for all cases

Energy represents the intensity values, MAD reflects how far voxel values deviate from the mean, and uniformity measures how frequently specific intensity levels occur. According to You et al. [13], malignant tumors typically exhibit inhomogeneous intensity patterns. This inhomogeneity can be better captured by metrics like MAD and uniformity than energy.

Thus, I evaluated the performance of both Uniformity and MAD for identifying malignant tumors. Based on the ground truth labels, there are 20 malignant cases out of 40 in total. To reflect this in evaluation, I used the 50th percentile of each metric as the threshold: cases in the upper 50% of MAD values and lower 50% of Uniformity values are classified as malignant. This is because malignant tumors are often less homogeneous, which corresponds to higher MAD and lower Uniformity. The resulting confusion matrices are shown in Figure 33.



(a) Confusion matrix for MAD

(b) Confusion matrix for Uniformity

Figure 33: Comparison of classification performance using MAD and Uniformity

Figure 33 shows that the number of false positives (cases predicted as malignant but actually benign) and false negatives (cases predicted as benign but actually malignant) outweigh the true positives and true negatives for both metrics. While MAD slightly outperforms Uniformity, given the small sample size, these results are likely inconclusive.

In theory, uniformity is likely the more robust metric, as MAD is sensitive to outliers and may falsely indicate heterogeneity in otherwise homogeneous benign tumors. For example, in a benign tumor with mostly homogeneous intensities and a few noisy high-intensity outliers, MAD may incorrectly report high heterogeneity, while uniformity correctly emphasizes the dominant intensity group and avoids false positives

5 Appendix

Generation tools such as ChatGPT and Claude.ai are used to debug and refine the python code associated to this report. The tools were used to facilitate LaTeX formatting and spellcheck for the report.

References

- [1] Nancy Adam, Erin Degelman, Sophie Briggs, Rima-Marie Wazen, Pina Colarusso, Karl Riabowol, and Tara Beattie. Telomere analysis using 3d fluorescence microscopy suggests mammalian telomere clustering in hert-immortalized hs68 fibroblasts. *Communications Biology*, 2:451, 2019. eCollection 2019.
- [2] Claudio Ghetti, Marco Ortenzia, and Paolo Serrel. Which dose for what image? iterative reconstruction for ct scan. *Imaging in Medicine*, 5(5):1–10, 2013.
- [3] H. M. Hudson and R. S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *IEEE Transactions on Medical Imaging*, 13(4):601–609, 1994.
- [4] Brian F. Hutton. Recent advances in iterative reconstruction for clinical spect/pet and ct. *Acta Oncologica*, 50(6):851–858, 2011.
- [5] Walter G. Jung. *Standard Responses*, chapter 5-4, pages 325–348. Newnes, Burlington, 2005.
- [6] Georgios Krokos, Jane MacKewn, Joel Dunn, and Paul Marsden. A review of pet attenuation correction methods for pet-mr. *EJNMMI Physics*, 10:52, September 2023.
- [7] Paul Lecoq, Christian Morel, John O. Prior, Dimitris Visvikis, Stefan Gundacker, Etienne Auffray, Peter Križan, Rosana Martinez Turtos, Dominique Thers, Edoardo Charbon, Joao Varela, Christophe de La Taille, Angelo Rivetti, and Dominique Breton. Roadmap toward the 10 ps time-of-flight pet challenge. *Physics in Medicine Biology*, 65(21):21RM01, 2020.

- [8] MRI Master. Zebra artifact (zebra stripes). <https://mrимaster.com/zebra-artifact-zebra-stripes/>, 2022. Accessed: 2025-04-08.
- [9] Dong Nie, Roger Trullo, Jun Lian, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen. Medical image synthesis with context-aware generative adversarial networks. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2017*, volume 10435 of *Lecture Notes in Computer Science*, pages 417–425, Cham, 2017. Springer.
- [10] Hongli Shi, Shuqian Luo, Zhi Yang, and Geming Wu. A novel iterative ct reconstruction approach based on fbp algorithm. *PLOS ONE*, 10(9):e0138498, 2015.
- [11] Ewa Witkowska-Patena, Anna Budzyńska, Agnieszka Giżewska, Mirosław Dziuk, and Agata Walecka-Mazur. Ordered subset expectation maximisation vs bayesian penalised likelihood reconstruction algorithm in 18f-psma-1007 pet/ct. *Annals of Nuclear Medicine*, 34(3):192–199, 2020.
- [12] Wenjia Yang, Yifeng Sun, Wentao Fang, Fangfei Qian, Jianding Ye, Qunhui Chen, Yifeng Jiang, Keke Yu, and Baohui Han. High-resolution computed tomography features distinguishing benign and malignant lesions manifesting as persistent solitary subsolid nodules. *Clinical Lung Cancer*, 19(1):e75–e83, 2018.
- [13] Xiaofang You, Xiwen Sun, Chunyan Yang, and Yong Fang. Ct diagnosis and differentiation of benign and malignant varieties of solitary fibrous tumor of the pleura. *Medicine*, 96(49):e9058, 2017.
- [14] Navid Zeraatkar, Arman Rahmim, Saeed Sarkar, and Mohammad Reza Ay. Development and evaluation of image reconstruction algorithms for a novel desktop spect system. *Asia Oceania Journal of Nuclear Medicine and Biology*, 5(2):120–133, 2017.