CS 736

Assignment - 2

By: Ank Kumar Gupta (22B0623) & Veenus (22B0704)

Overview

- **Objective:** To implement and analyze image reconstruction techniques in X-ray CT using Radon Transform, Filtered Backprojection, and Algebraic Reconstruction Technique.

- **Tasks:**
  1. X-Ray Computed Tomography: Radon Transform.
  2. X-Ray Computed Tomography: Reconstruction by Filtered Backprojection (FBP).
  3. X-Ray Computed Tomography: Incomplete Data.
  4. Algebraic Reconstruction Technique (ART).

Task 1

Objective

X-Ray Computed Tomography: Radon Transform

Description

- Implement integration along parameterized lines to compute the Radon transform of the Shepp-Logan phantom.
- Compare Radon transforms for different step sizes (Δs) and analyze smoothness in 1D function plots for $\theta = 0°$ and $90°$.
- Discuss optimal parameter settings (Δt, $\Delta\theta$) for X-ray CT scanner design, balancing resolution and computational efficiency.

Task 1

Approach

- Used skimage.data.shepp_logan_phantom() to generate the Shepp-Logan Phantom.
- Applied skimage.transform.resize() function to resize the phantom to 128x128 pixels, ensuring consistency for further processing.
- Changed the coordinate system, making the center of the image as origin.
- Defined function myXrayIntegration() to perform line integration of image intensities and chose $\Delta s = 0.5$ and performed bilinear interpolation as image interpolation scheme.

Task 1

Approach

- Defined function myXrayCTRadonTransform() computing the Radon transform Rf(t) for the discrete set of values of t and θ.
- The resized image can be further **transformed using FFT** for frequency analysis.
- Plotted and compared radon transform for different values of $\Delta s = 0.5$.
- Brainstormed about various parameter setting of designing a CT reconstruction software relying on ART.

Result

1(a) **Justification for choosing the value of $\Delta S = 0.5$:**

As the image has a pixel spacing of 1 unit, choosing A s = 0.5 (that can be said as sampling twice per pixel) seems a good choice managing both accuracy and computational cost.

On increasing the A s, it may fail to capture variations in image intensity, while decreasing it can burden the computation.

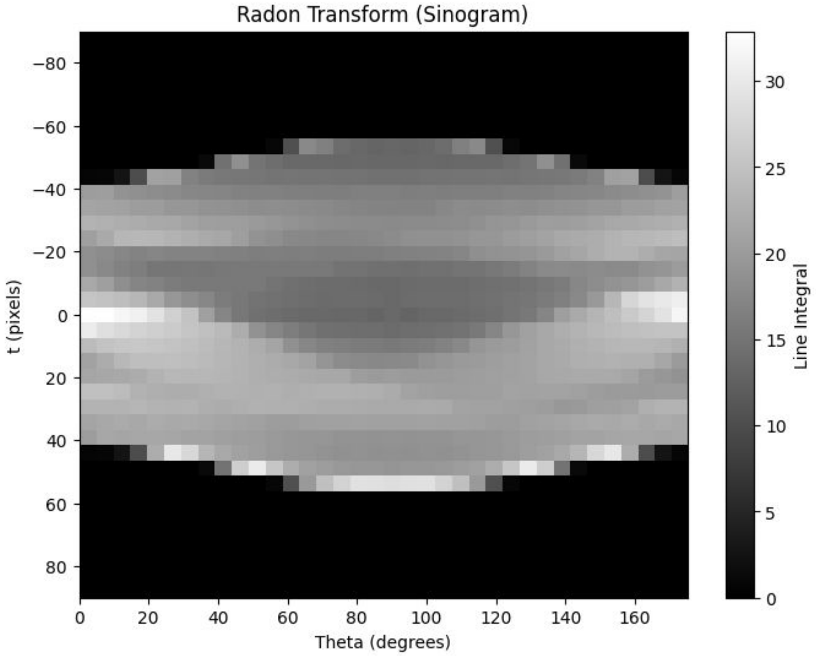
Justification for choosing Bilinear Interpolation as image interpretation scheme:

Bilinear Interpolation is used because it is easy to implement and also gives a good balance between computational efficiency and accuracy. It also provides a continuous, smooth approximation of the image intensity between pixel centers by taking weighted average of neighbouring pixels.

Result

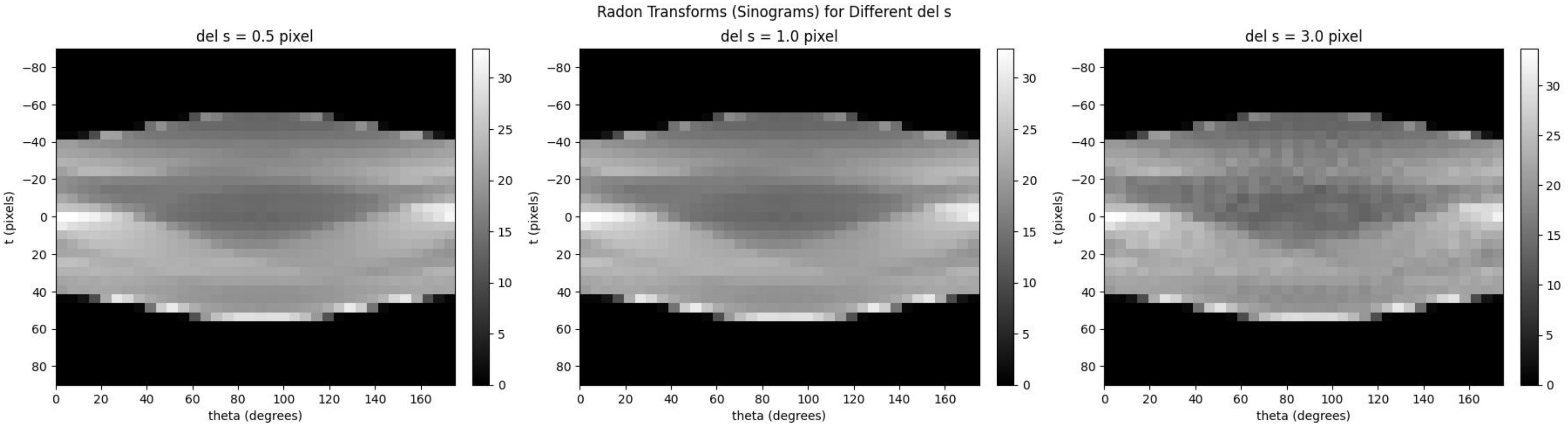
1(b)

Obtained Sinogram image after executing function myXrayCTRadonTransform()



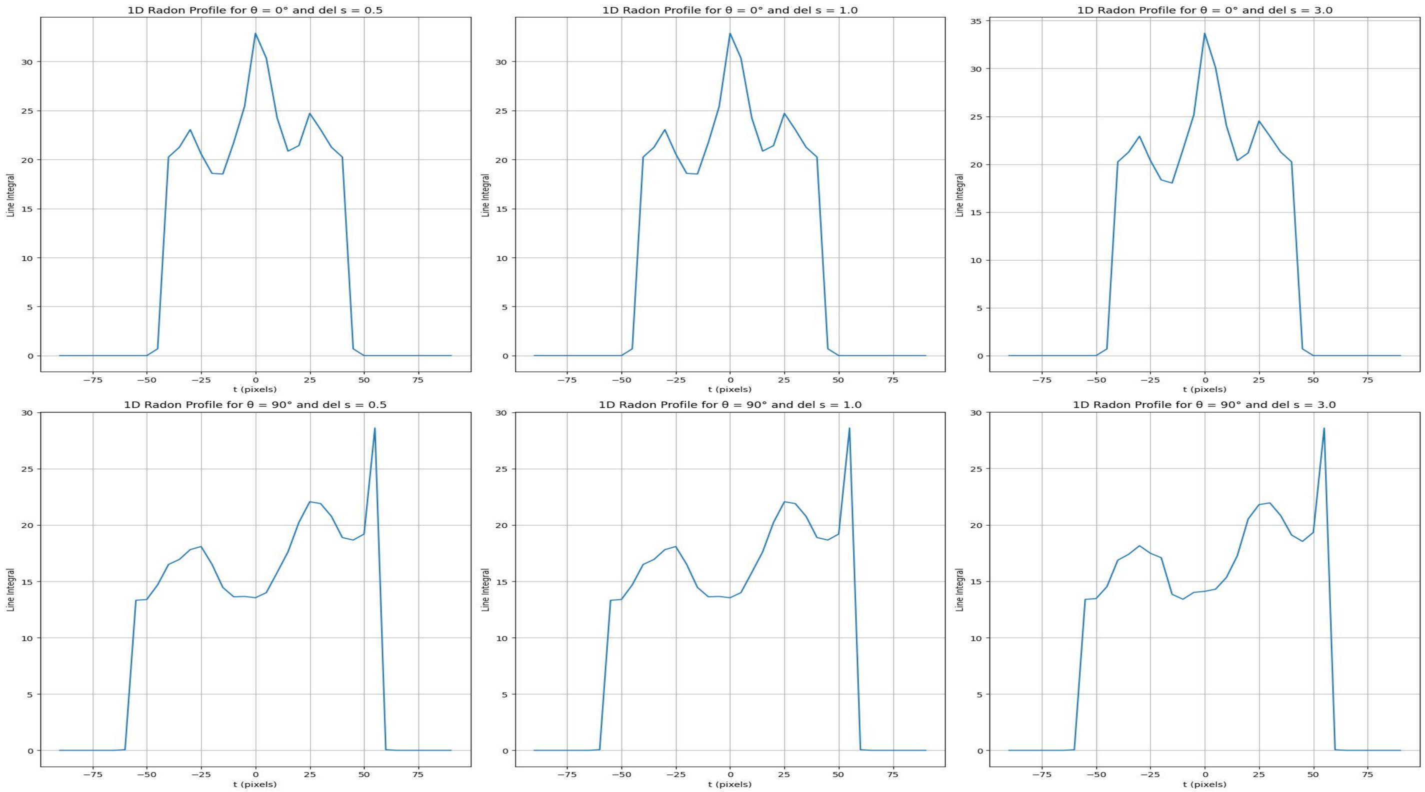
Result

1(c) Comparison of Radon-transform image (for each s)



Result

1(c) 1D function plots for the Radon Transform Images



Result

1(c) Comparison between Radon Transform (Sinogram) Images for different Δ s values:

- Smoothest Sinogram:
  The Radon transform computed with A s = 0.5 appears the smoothest which is because every projection is integrated with high resolution along the line, thus producing a consistent and smooth sinogram.
- Roughest Sinogram:
  The Radon transform computed with A s = 3 is the roughest which is because of the coarse sampling, which resulted in significant discretization error, which can be observed as irregularities and noise in the sinogram.

Result

1(c)

The comparison between 1D function plots for the Radon-transform images at different θ and Δs values: The plots computed with $\Delta s=0.5$ appear the smoothest because the integration along each line is sampled very finely, yielding a good approximation of the continuous integral and resulting in smooth 1D functions. In contrast, the plots with $\Delta s=3$ appear the roughest due to the coarser sampling of integration along each line, making the approximation of the line integral less accurate and more susceptible to discretization noise effects. Additionally, the plot corresponding to $\theta=90^\circ$ appears comparatively smoother, which may be attributed to differences in image variance along different axes. This suggests that intensity variations in the image along the $\theta=90^\circ$ direction are smaller.

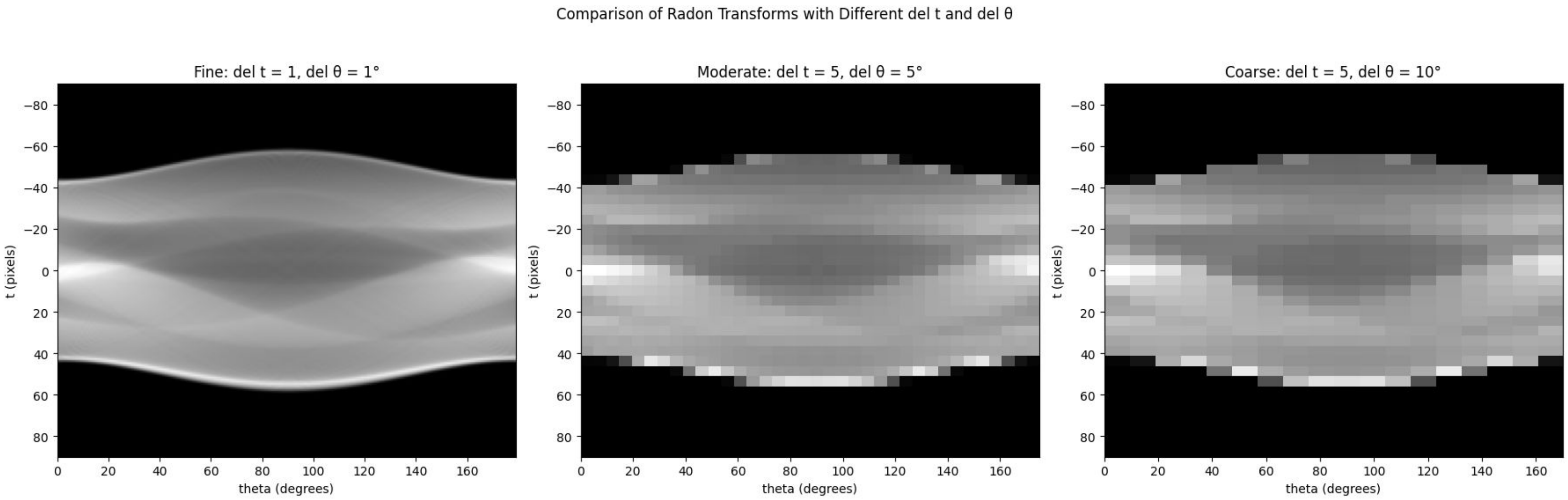
Result

1(d) Design of an Xray CT scanner

The key parameters for designing an X-ray CT scanner are $\Delta\theta$ and Δt. Smaller values ensure high-resolution images but increase computation time, while larger values reduce quality. I will use moderate-to-fine sampling ($\Delta t \approx 1$ pixel, $\Delta\theta = 1^\circ$ to 5°) to balance quality and efficiency. Coarse sampling ($\Delta\theta \geq 10^\circ$) can cause noise and artifacts, so it should be avoided.

Result

1(d) design of an Xray CT scanner



Result

1(e) Design a CT reconstruction software

For CT reconstruction using ART, I will select a pixel size matching the detector resolution to capture all features while ensuring the scene grid fully covers the object with a small margin. If $\Delta s>>1$ pixel, coarse sampling speeds up computation but loses intensity variations, causing inaccuracies. If $\Delta s<<1$ pixel, oversampling improves precision but increases computational load without much gain beyond a certain point. Balancing Δs optimally ensures accurate reconstruction without excessive processing time.

Task 2

Objective

Reconstruction by Filtered Backprojection (FBP).

Description

- Generate a 128×128 Shepp-Logan phantom with the origin at the center and compute its Radon transform for angles [0°, 3°, 6°, …, 177°].
- Implement Ram-Lak, Shepp-Logan, and Cosine filters in the Fourier domain via a function myFilter() using parameter L, filtering Radon data at L=wmax and L=wmax/2.
- Compute the unfiltered backprojection using iradon(), then compare and justify differences between the various filtered images.
- Blur the phantom with Gaussian filters, compute their Radon transforms and Ram-Lak backprojections, and plot RRMSE versus L.

Task 2

Approach

- Generate a 128×128 Shepp–Logan phantom with the center as the origin as previous task.
- Computed its Radon transform for angles 0°–177° in the steps of 3°.
- Implemented myFilter() applying Ram–Lak, Shepp–Logan, and Cosine filters in Fourier domain.
- Filtered the sinogram with L = 0.5 and L = 0.25 to control the frequency cutoff.
- Reconstructed images using iradon() and compare them to unfiltered backprojection.

Task 2

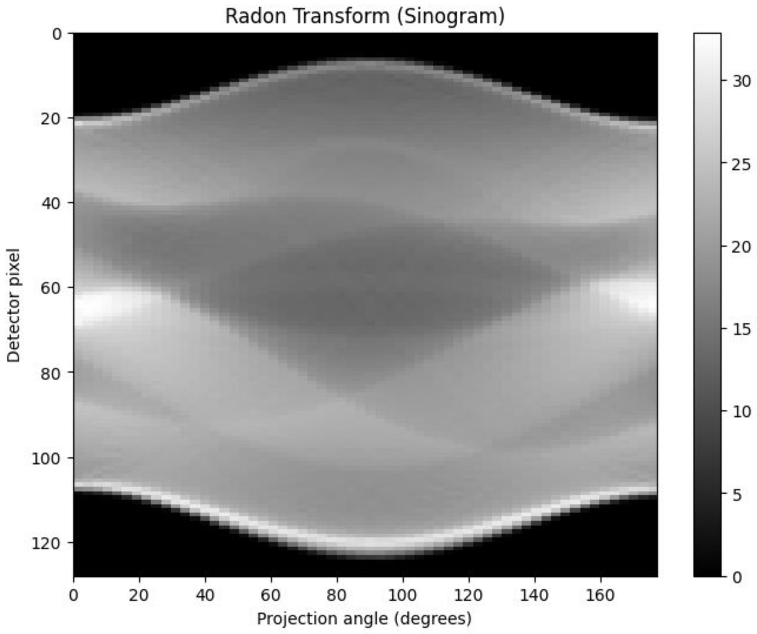
Approach

- Blurred the phantom with Gaussian masks (11, σ=1 and 51, σ=5) to create S1 and S5.
- Computed Radon transforms and applied Ram−Lak filtering on S0, S1, and S5, thus yielding R0, R1, and R5.
- Calculated RRMSE between original and reconstructed images
- Plotted RRMSE versus L curve to analyze the trade-offs.

Result

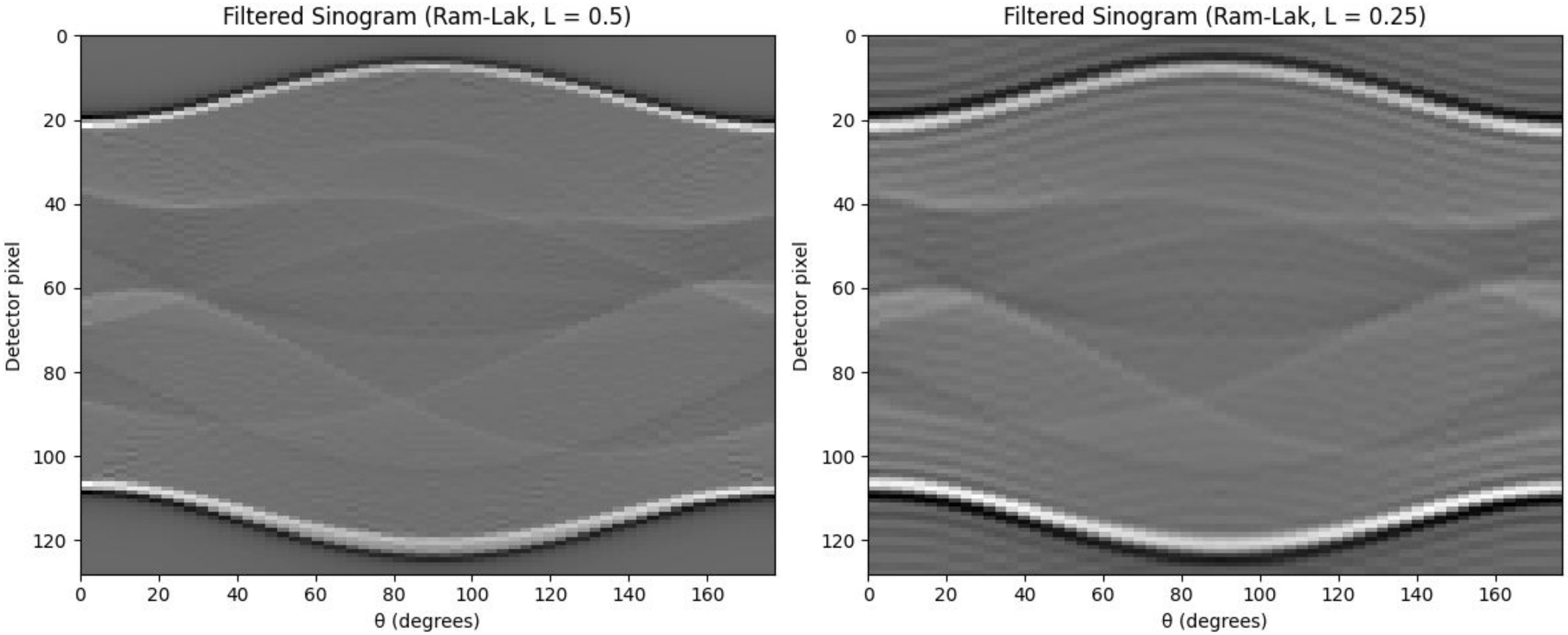
2(a)

Radon transform



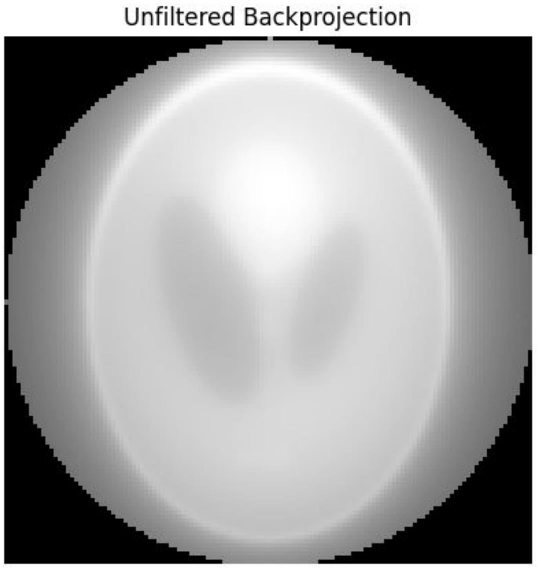
Result

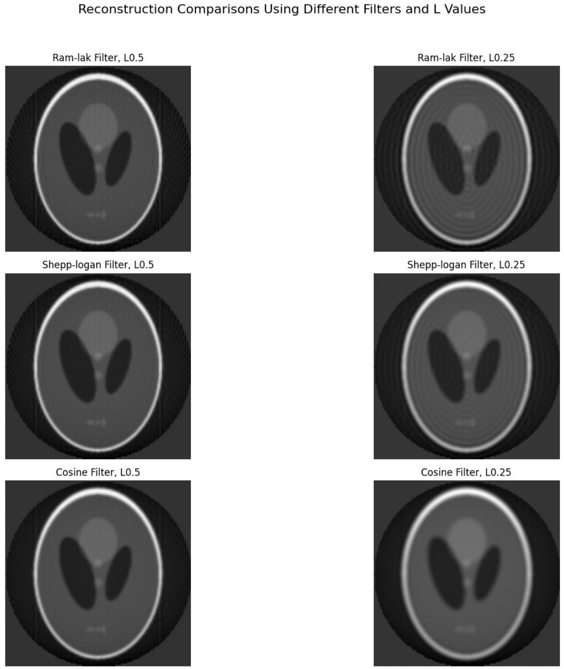
2(a)


Filtered Sinogram (Ram-Lak, L = 0.5)
Detector pixel
θ (degrees)
Filtered Sinogram (Ram-Lak, L = 0.25)
Detector pixel
θ (degrees)

Result

2(a) Implementation of the Ram-Lak filter, Shepp-Logan filter, and Cosine filter





Result

2(a) Justifying the similarities and the differences observed between the different combinations of filters and parameter values

- **Unfiltered Backprojection:** The reconstruction image obtained by simply backprojecting the raw sinogram appears very blurred. This is also expected because unfiltered backprojection simply smears the projection data back over the image domain, which leads to a loss of high-frequency information and significant artifacts.

Result

2. Effect of the different Filter Type used:

a. **Ram–Lak Filter:**
The Ram–Lak filter implements the ideal ramp (i.e. multiplication by |w|). When used with L = 0.5 (i.e. including all available frequencies), the reconstruction is sharp and shows high spatial resolution. But it may also amplify high-frequency noise. In contrast, when L is reduced to 0.25, the high-frequency components are comparatively strongly suppressed, which results in a smoother (less noisy) reconstruction, but some fine details are lost, leading to a slight blurring.

b. **Shepp–Logan Filter:**
The Shepp–Logan filter modifies the ramp by multiplying it with a sinc function. It reduces the amplification of high-frequency noise compared to the pure Ram–Lak filter. With L = 0.5, the image is reasonably sharp with less noise, and with L = 0.25, the reconstruction becomes even smoother (but also come with some loss in resolution).

c. **Cosine Filter:**
The Cosine filter multiplies the ramp by a cosine function, again attenuating the highest frequencies. Its behavior is similar to the Shepp–Logan filter in that it yields smoother reconstructions than the pure Ram–Lak filter. The effect of reducing L (i.e. L = 0.25) is similar – the image becomes smoother but with a slight loss of edge definition.

Result

3. **Effect of the Parameter L:**

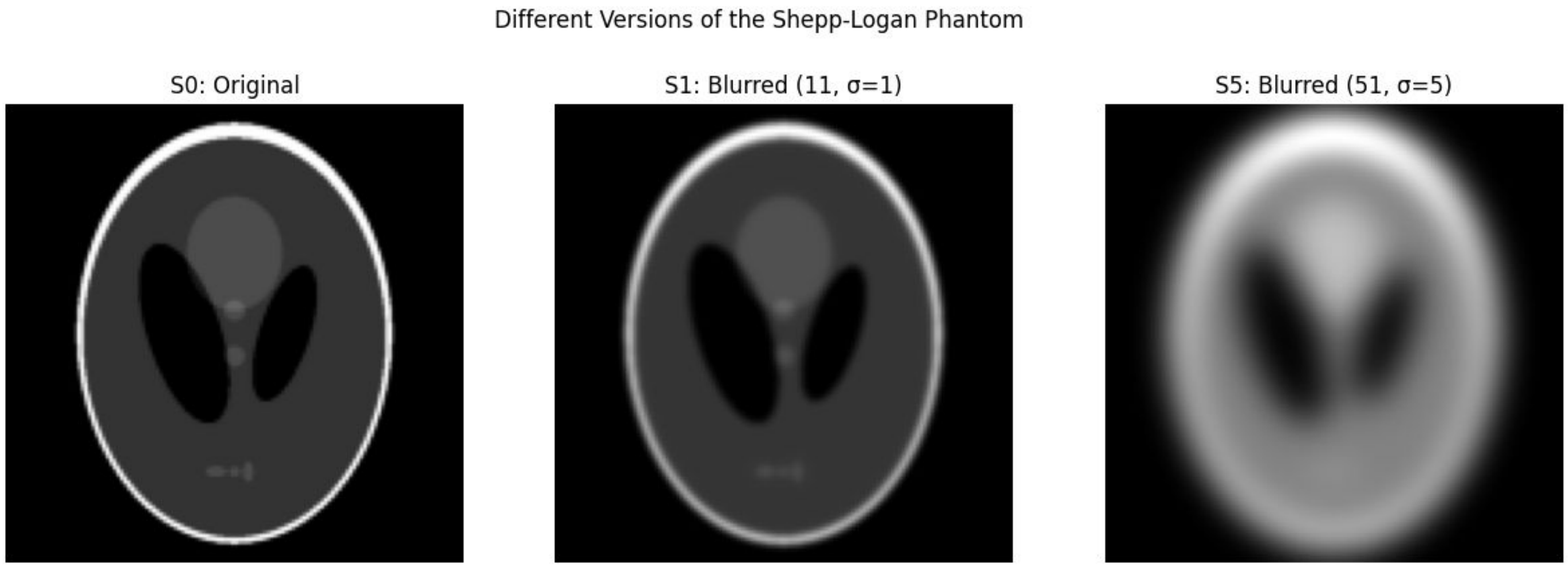
Here, the parameter L acts as a cutoff frequency:

- **L = w_max (0.5 in our discrete setup):**
  Nearly all frequencies up to the maximum are passed. This leads to a reconstruction with higher spatial resolution (i.e. sharper edges) but at the expense of increased high-frequency noise and potential aliasing artifacts.
- **L = w_max/2 (0.25):**
  A lower cutoff frequency means that high-frequency components are suppressed. The resulting image is smoother and has less noise, but some fine details and sharp edges are blurred.

In conclusion, we can infer that the pure Ram–Lak filter (with L = 0.5) gives the sharpest image but can be noisier. The Shepp–Logan and Cosine filters, by attenuating high frequencies, yield reconstructions that are smoother. Lowering L (to 0.25) in any filter further reduces noise and artifacts but causes a loss in resolution (blurring of edges).

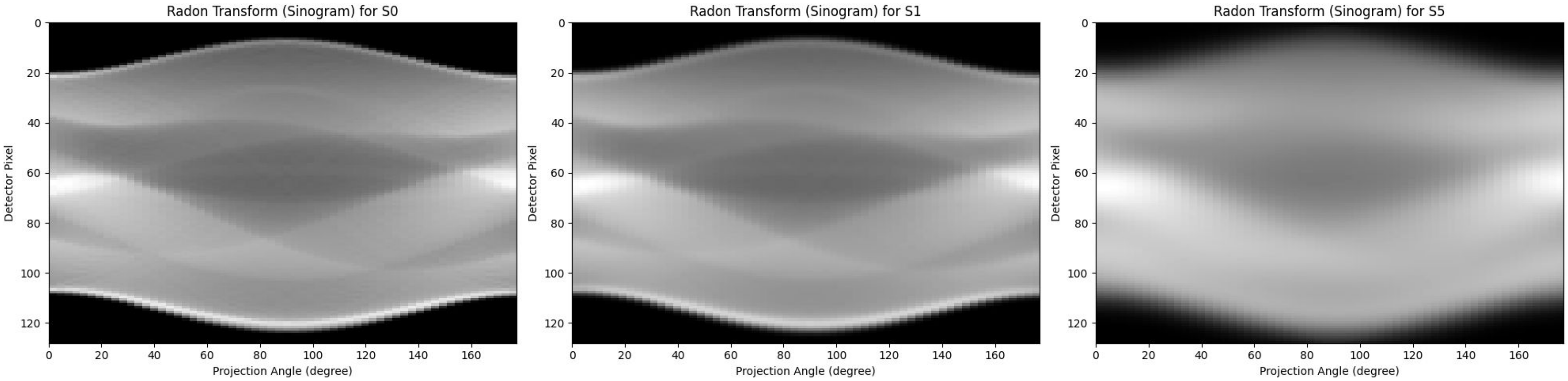
Result

2(b) Blurred versions of the Shepp-Logan image



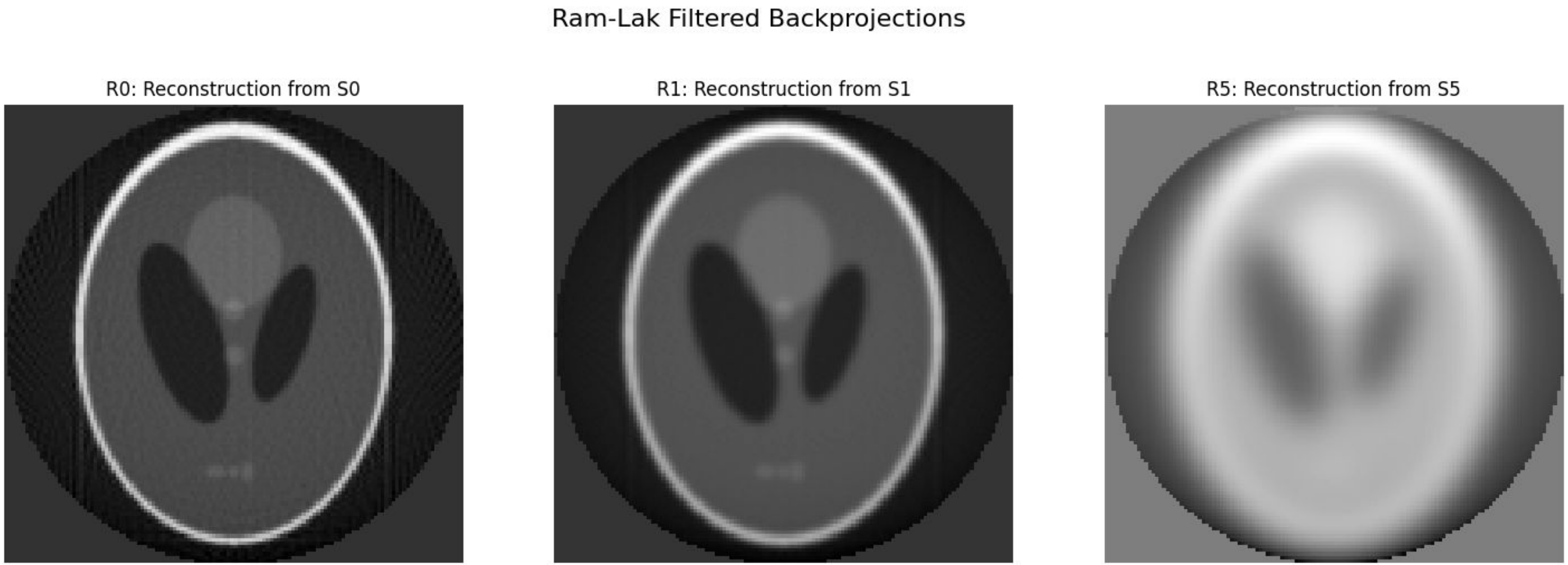
Result

2(b) 3 different versions of the Sinograms obtained after radon transformation



Result

2(b) 3 Ram-Lak-filtered backprojections



Result

2(b) **relative root-mean-squared errors (RRMSE)**

RRMSE(S0, R0) = 0.6529 RRMSE(S1, R1) = 0.6612 RRMSE(S5, R5) = 0.7383

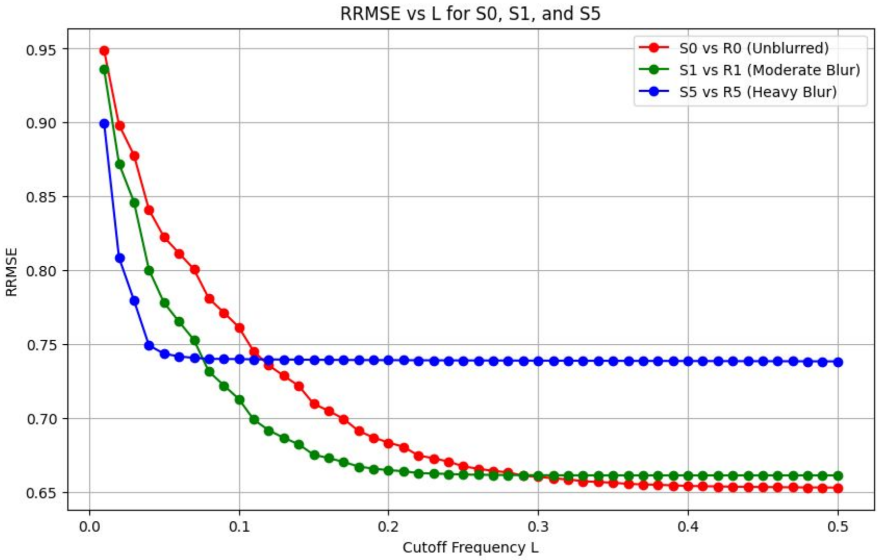
Lowest RRMSE (S0, R0): The original phantom (S0) retains full high-frequency detail, which the Ram-Lak filter reconstructs effectively, yielding an RRMSE of 0.6529 and excellent detail preservation.

Intermediate RRMSE (S1, R1): The moderately blurred phantom (S1) loses some high-frequency content from an 11×11, σ=1 Gaussian filter, resulting in a slight mismatch with the Ram-Lak filter and an RRMSE of 0.6612.

Highest RRMSE (S5, R5): The heavily blurred phantom (S5) from a 51×51, σ=5 Gaussian loses significant high frequencies, causing the Ram-Lak filter to amplify artifacts and sparse details, and yielding an RRMSE of 0.7383.

Result

2(c) Plotting the RRMSE values as a function of L with L = [Wmax/50, 2Wmax/50, … , Wmax].



Result

2(b) Explaination of Plot obtained:

1. **Reconstruction from Unblurred Phantom (S0, R0):**
   - The RRMSE for (S0) tends to decrease as L increases and ultimately converges but it can be observed that it initially drops slowly with increasing L, but later converses to lowest RRMSE, in this case, we can infer that as L approaches w_{max}, most of the significant frequency components are retained, leading to the lowest achievable RRMSE for (S0).
2. **Reconstruction from Moderately Blurred Phantom (S1, R1):**
   - The RRMSE for the moderately blurred phantom (S1) also decreases with increasing L. Here, the decrease is steeper compared to the S0 case and ultimately, the RRMSE converges to a value slightly higher than that of the unblurred case. Here also, we can infer that increasing L still improves reconstruction by allowing more frequencies, but because (S1) inherently lacks some high-frequency details, it leads to slightly higher RRMSE.
3. **Reconstruction from Heavily Blurred Phantom (S5, R5):**
   - The RRMSE for the heavily blurred phantom (S5) decreases very sharply as L increases but despite the sharp decrease, it converges to a higher RRMSE compared to (S0) and (S1). Here, heavy blurring has significantly removed high-frequency content. Thus, even with a high L, the reconstruction (R5) cannot fully restore (S5), leading to lowest RRMSE.

Task 3

Objective

X-Ray Computed Tomography: Incomplete Data

Description

- Load ground-truth images from **chestCT.mat** and **myPhantom.mat**.
- Simulate CT data acquisition over 151 angles spanning 150° in 1° increments.
- Plot RRMSE values between ground-truth and reconstructed images.
- Display the reconstruction with lowest RRMSE and identify the optimal contiguous 151-angle set.

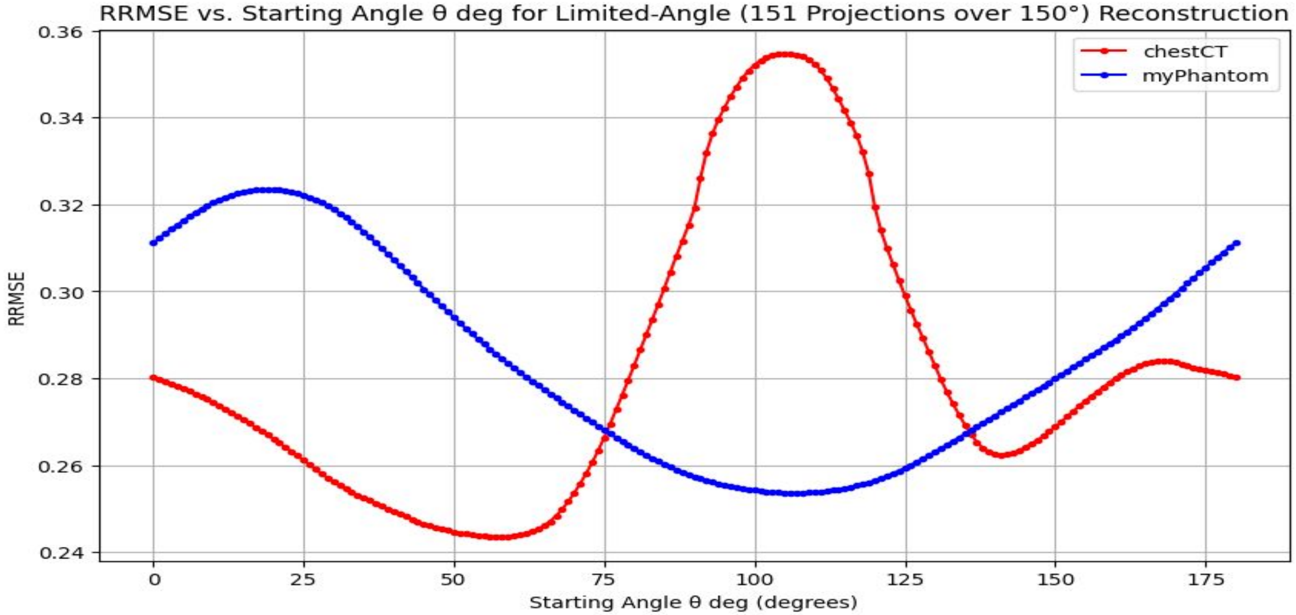
Task 3

Approach

- Loaded chestCT and myPhantom images from the given .mat files and displayed them.
- Simulated CT data using 151 projections spanning 150° for each starting angle $\theta \in [0°, 180°]$.
- Then, Computed the Radon transform and perform filtered backprojection (using ramp filter) for each θ.
- Calculated RRMSE between the ground-truth and reconstructed images, and then plotted RRMSE versus θ.
- Identified the optimal θ (minimum RRMSE) and displayed the corresponding reconstructions.

Results

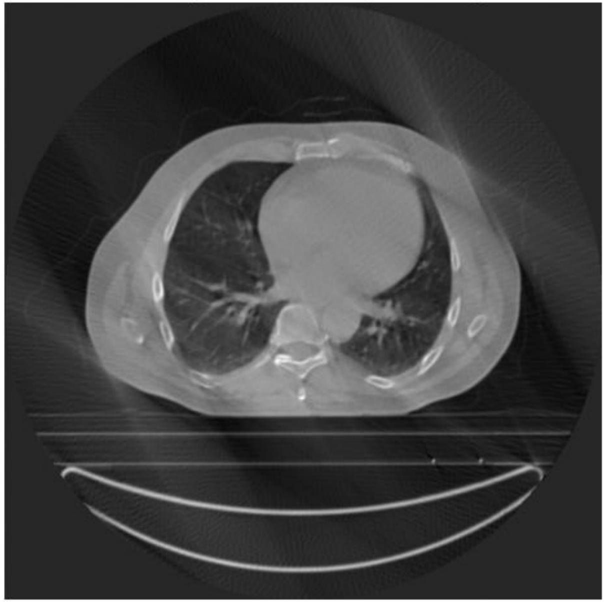
3(a) RRMSE values between the ground-truth images and the reconstructed images for [0, 1, 2, …., 180]



Results

3(b) Reconstructed image for optimal θ

Optimal Reconstruction for chestCT
(151 angles from 57 deg to 27 deg)



Optimal Reconstruction for myPhantom
(151 angles from 107° to 77°)

Task 4

Objective

Algebraic Reconstruction Technique (ART)

Description

- Load ground-truth from chestCT.mat and simulate CT data over 180 angles (0–179°) with Gaussian noise (5% of data range).
- Implement an ART algorithm (myART()) with zero initialization and flexible projection ordering, constructing the radon matrix from scratch.
- For fixed ordering, plot RRMSE versus iterations for step-lengths from 0.1 to 1 in increments of 0.1.

Task 4

Approach

- Loaded chestCT.mat with size (512px * 512px).
- Used 180 angles (0°–179°) and 512 detector bins spanning the image's half-diagonal.
- Numerically computed each ray's weight matrix via integration with step size s (0.5).
- For every ray, computed the projection $b = \Sigma(W \cdot chestCT)$ from the weight matrix.
- Added i.i.d. zero-mean Gaussian noise with σ = 5% of the data's intensity range to each projection.
- Update f by $f \leftarrow f + \lambda \cdot (b - p)/||W||^2 \cdot W$, where p is the current projection of f.

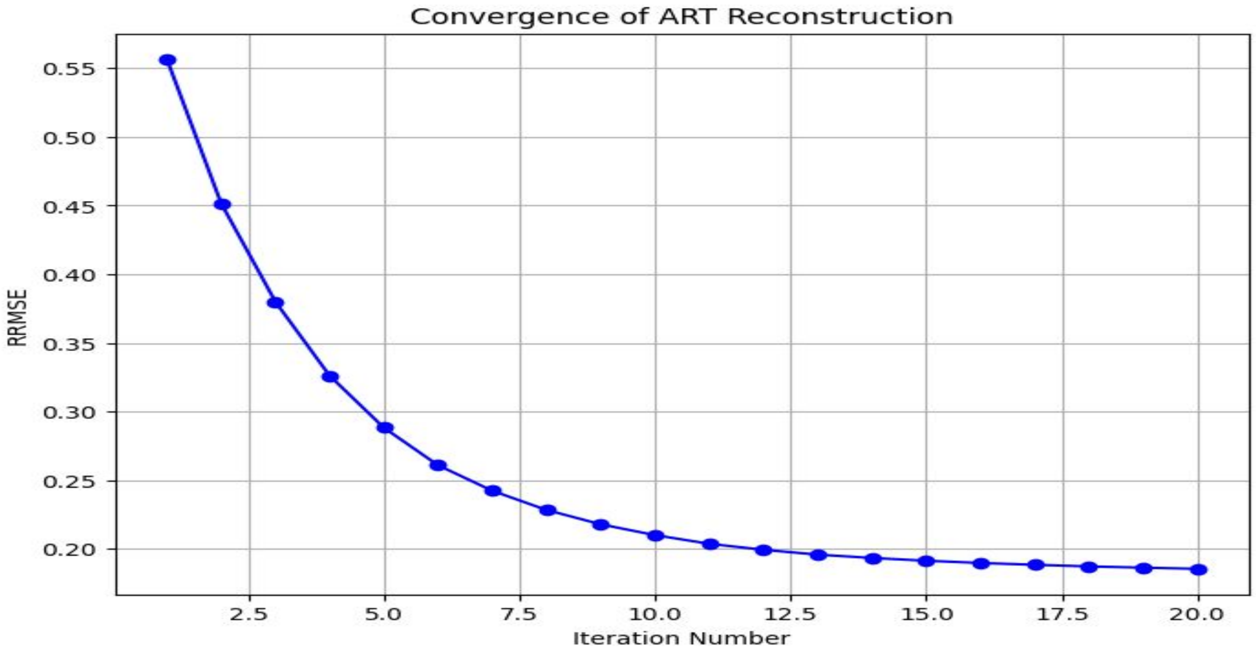
Task 4

Approach

- Used Numba JIT compiler for fast execution of code.
- After each full iteration, calculate RRMSE between the reconstruction and ground truth.
- Run ART for λ values from 0.1 to 1.0 (step 0.1) using sequential ordering.
- Executed independent ART runs for each λ concurrently via Python's multiprocessing.
- Plot RRMSE versus iteration number for each λ to analyze convergence behavior.

Results

Graph of RRMSE v/s Iteration (Used λ = 0.5)



Results

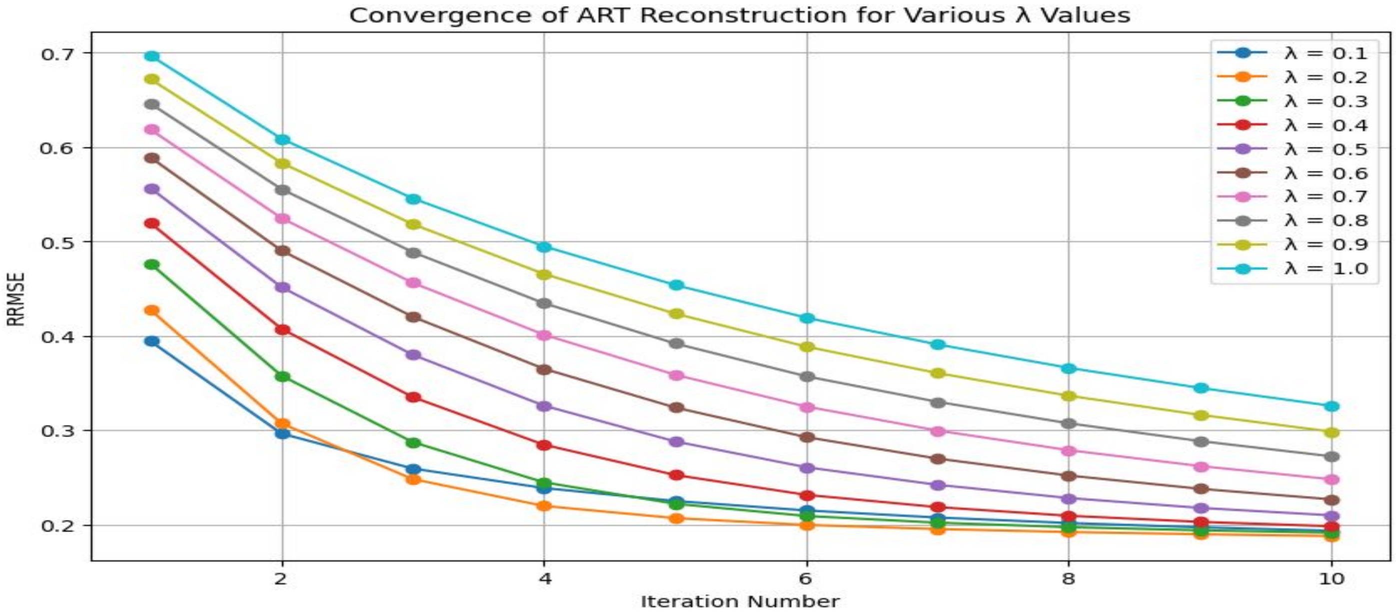
Reconstructed Image

RRMSE value obtained at the convergence (at the end of 20th iteration): **0.1854**

ART Reconstruction after 20 Iterations (λ = 0.5)

Results

Plot of RRMSE v/s λ, for λ in [0.1, 0.2, …., 1]



Thank You!