

BOSTON UNIVERSITY

Faculty of Computing & Data Science

Dissertation

**IMAGE RECONSTRUCTION THROUGH MULTIPLE 1D
APPROXIMATIONS**

by

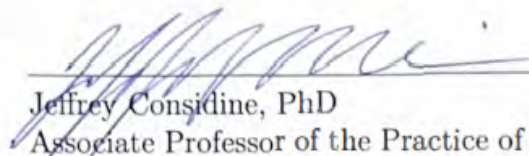
BOHAN WANG

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science

2024

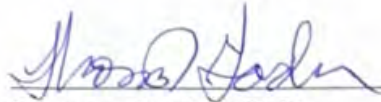
Approved by

First Reader

A handwritten signature in blue ink, appearing to read 'Jeff Considine', written over a horizontal line.

Jeffrey Considine, PhD
Associate Professor of the Practice of Computing and Data Sci-
ences + Computer Science

Second Reader

A handwritten signature in blue ink, appearing to read 'Thomas Gardos', written over a horizontal line.

Thomas Gardos, PhD
Associate Professor of the Practice of Computing

IMAGE RECONSTRUCTION THROUGH MULTIPLE 1D APPROXIMATIONS

BOHAN WANG

Boston University, Faculty of Computing & Data Science, 2024

Major Professors: Jeffrey Considine, PhD
Associate Professor of the Practice of Computing &
Data Sciences + Computer Science

ABSTRACT

Function approximation is a fundamental aspect of computational models and machine learning, often relying on neural networks due to their ability to effectively model complex functions and relationships. However, neural networks can be computationally intensive and lack interpretability. In this thesis, we explore an alternative approach to approximating two-dimensional (2D) functions by decomposing them into multiple one-dimensional (1D) approximations. Our method aims to enhance computational efficiency and interpretability while maintaining high approximation quality.

We propose a framework that projects to approximate 2D functions through a series of 1D interpolations and also uses greedy sampling. By generating uniformly distributed projections and projecting pixel coordinates onto these projections, we form 1D curves and use interpolation to predict the values of the original function. Linear interpolation is employed for its simplicity and speed in estimating values between sampled points. A greedy algorithm is used to select sampling points that significantly reduce approximation error, optimizing the sampling strategy.

We conducted extensive experiments on some images to evaluate the performance of our method. Metrics such as Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) were used to assess reconstruction quality. Additionally, we ran neural network model and some other traditional models for comparison. Our results demonstrate that the proposed method provides a different focus compared to other methods, especially excelling in the restoration of high-contrast details in images.

The findings suggest that multiple 1D approximations can reconstruct 2D functions with efficiency. Contrary to our initial intuition, the results reveal that increasing the number of sample points has a more significant impact on reconstruction quality than increasing the number of projections. Specifically, we observed that under the same parameter count, using as many sample points as possible led to better reconstruction results. Increasing the number of projections, while beneficial for reducing artifacts, has a less pronounced effect compared to increasing sample points. However, adding more projections can improve edge clarity and enhance the accuracy of each step in the greedy selection process, which helps in achieving better sample point locations during reconstruction.

Additionally, we tested various sampling methods, such as uniform sampling and greedy MSE selection, and found that greedy selection of sample points based on MSE yielded significantly improved clarity, particularly around key features of the image. The experiments also showed that incorporating spatial diversity and edge information into the selection process did not always yield better results, highlighting the importance of selecting sample points that balance both edge and surrounding details.

This work contributes to the field by providing an alternative method for function approximation that addresses some limitations of neural networks, particularly in terms of computational efficiency. Future work includes extending the approach

to higher-dimensional data, exploring advanced interpolation techniques, and integrating the method with machine learning models to balance performance and transparency. Additionally, further research is needed to optimize the balance between projections and sample points to achieve the best reconstruction quality under different parameter constraints.

Keywords: Function Approximation, One-Dimensional Approximations, Greedy Sampling, Image Reconstruction, Neural Networks, Computational Efficiency

Contents

1	Introduction	1
1.1	Background	1
1.2	Motivation	2
1.3	Methods	3
2	Related Work	4
2.1	The Universal Approximation Theorem	4
2.2	Neural Fields	4
2.3	AAA Approximation Methods	5
2.4	Other Methods	6
2.4.1	Neural Network	6
2.4.2	Fourier Features	6
2.4.3	Kernel Regression	6
2.4.4	Piecewise Linear Interpolation	7
3	Methodology	8
3.1	Problem Definition	8
3.2	Coordinate-Based Function Approximation	8
3.2.1	Projection of Pixel Coordinates and Values	8
3.3	Generating Uniformly Distributed Projections	10
3.3.1	Linear Interpolation of Curve Values	11
3.4	Greedy Algorithm for Sampling Point Selection	13
3.4.1	Diversity-enhanced Greedy Selections	13

3.5	Image Reconstruction	14
3.5.1	Average Method	14
3.5.2	Optimization Using Least Squares	14
3.6	Evaluation Metrics	15
3.6.1	Mean Squared Error (MSE)	15
3.6.2	Peak Signal-to-Noise Ratio (PSNR)	15
4	Experimental Design	16
4.1	Objective	16
4.2	Experiment Setup	16
4.3	Sampling and Interpolation Strategies	16
4.3.1	Sampling Methods	16
4.3.2	MSE Calculation Methods	17
4.3.3	Interpolation Methods	17
4.3.4	Optimization of Sampling Points and Projections	17
4.3.5	Parameter Count	18
4.4	Comparative Models	18
4.5	Optimized Computational Techniques	18
4.5.1	Pixel Downsampling	19
4.5.2	Memory Optimization	19
4.6	Evaluation Metrics	19
4.7	Analysis of Parameter Impact	20
5	Results and Analysis	21
5.1	Overview	21
5.2	Sampling Methods	21
5.2.1	Uniform Sampling	22
5.2.2	Greedy MSE Selection	23

5.2.3	Advanced Greedy Selection	24
5.3	MSE Calculation Methods	27
5.4	Interpolation Methods	27
5.5	Optimization of Sampling Points and Projections	30
5.5.1	Practical Thinking on Sampling and Projection Balance	34
5.6	Comparison of Different Projections in Selection and Reconstruction .	36
5.7	Comparison with Other Models	39
5.7.1	Parameter Configuration	39
5.7.2	Experimental Setup	40
5.7.3	Results	40
5.8	Analysis	43
6	Conclusion	48
6.1	Summary of Findings	48
6.2	Contributions	49
6.3	Limitations	49
6.4	Future Work	50
A	Github	51
	References	52

List of Tables

5.1	Comparison of Sampling Methods	25
5.2	Comparison of MSE Calculation Methods	27
5.3	Comparison of Interpolation Methods	28
5.4	Number of projections and sample points used	31
5.5	PSNR and MSE Comparison Across Scenarios	38
5.6	Performance Comparison at 4400 Parameters	40
5.7	Performance Comparison at 17000 Parameters	41
5.8	Performance Comparison at 66800 Parameters	42

List of Figures

3·1	A 256×256 image with origin shifted to $(128, 128)$	9
3·2	Comparison of original image (left) and reconstruction using a single projection (right)	12
5·1	Original image of a small cat used for analysis	22
5·2	From left to right: Average reconstruction, weighted least squares reconstruction, and difference image with sampled points in red (Number of Projections: 128, Number of sample points: 10240)	23
5·3	Improved image reconstruction using greedy MSE selection, with sample points concentrating on key features (Number of Projections: 128, Number of sample points: 10240)	24
5·4	Advanced greedy selection results (Number of Projections: 128, Number of sample points: 10240)	25
5·5	Sample points concentrated on the complex subject with a simple background (Number of Projections: 128, Number of sample points: 10240)	26
5·6	Sample points distributed on the simple subject with a complex background (Number of Projections: 128, Number of sample points: 10240)	26
5·7	Compare image reconstruction using spline interpolation and linear interpolation (Number of Projections: 128, Number of sample points: 10240)	28

5-8	Reconstruction image using spline interpolation with a small number of sample points, showing the sample point distribution (Number of Projections: 16, Number of sample points: 160)	29
5-9	Reconstruction image using linear interpolation with a small number of sample points, showing the sample point distribution (Number of Projections: 16, Number of sample points: 160)	29
5-10	Reconstruction of linear vs spline with same sample points chosen by linear (Number of Projections: 16, Number of sample points: 160) . .	30
5-11	Plot of MSE and PSNR across different sampling points and parameters (Pre-Computed 1-d Curves model)	31
5-12	Plot of MSE across different compressed parameter counts. The curves are categorized by the number of projections (Compressed model) . .	33
5-13	Comparison of reconstruction quality with 4 projections (left) and 64 projections (right) with same number of sample points.	34
5-14	Reconstruction results using the same number of projections for greedy selection and reconstruction (Number of Projections: 3, Number of sample points: 20001)	37
5-15	Reconstruction results using a larger number of projections for greedy selection and a newly generated smaller number of projections for reconstruction (Number of Projections: 3, Number of sample points: 20001)	37
5-16	Reconstruction results using a larger number of projections for greedy selection and selecting a subset of those projections for reconstruction (Number of Projections: 3, Number of sample points: 20001)	38
5-17	Visual comparison of reconstruction results at 4400 parameters (Number of Projections: 64, Number of sample points: 880)	41

5.18	Visual comparison of reconstruction results at 17000 parameters (Number of Projections: 64, Number of sample points: 3400)	42
5.19	Visual comparison of reconstruction results at 66800 parameters (Number of Projections: 64, Number of sample points: 13360	43
5.20	Detailed visual comparison of reconstructions with 4400 parameters .	44
5.21	Detailed visual comparison of reconstructions with 17000 parameters	45
5.22	Detailed visual comparison of reconstructions with 66800 parameters	46

List of Abbreviations

PSNR	Peak Signal-to-Noise Ratio
MSE	Mean Squared Error
UAT	Universal Approximation Theorem
AAA	Adaptive Antoulas-Anderson Algorithm
NeRF	Neural Radiance Fields
K	Number Of Projections
N	Number Of Sample Points
C	Number Of Input's Channel
T	Total Number Of Pixels

Chapter 1

Introduction

1.1 Background

Function approximation is a fundamental concept in computational models and machine learning, playing a crucial role in modeling and predicting complex real-world phenomena. The goal of function approximation is to find a representation that captures the behavior of a target function as accurately as possible while balancing computational feasibility. In machine learning, neural networks have emerged as powerful tools for function approximation, capable of learning intricate patterns in data and representing complex functions with high fidelity.

The Universal Approximation Theorem (UAT) is a key theoretical result that underpins the power of neural networks. It states that a feedforward neural network with a single hidden layer, given sufficient neurons and an appropriate activation function, can approximate any continuous function on a compact domain to an arbitrary degree of accuracy. Despite their success, neural networks are not without challenges. One significant issue is the computational cost associated with training and inference, which often requires substantial resources and specialized hardware. Additionally, the interpretability of neural networks remains a concern, as the internal representations learned by these models are often difficult to understand. These challenges have motivated the exploration of alternative methods that may offer greater speed and interpretability while retaining the expressive power of neural networks.

1.2 Motivation

The motivation for this work arises from the desire to explore simpler, faster, and more interpretable alternatives to traditional neural network-based function approximation. Specifically, we investigate whether multiple one-dimensional (1D) approximations can be combined to approximate two-dimensional (2D) functions in a practical field. By breaking down the 2D function approximation problem into multiple 1D approximations, we aim to explore the advantages and limitations of this method and determine whether it can serve as a viable alternative.

Neural fields are continuous functions, typically parameterized by neural networks, that map coordinates to values, such as color or density. However, neural network implementations like Neural Radiance Fields (NeRF) ([Mildenhall et al., 2020](#)) often face challenges due to their high computational cost and limited interpretability. To address these challenges, alternative methods such as PlenOctrees ([Yu et al., 2021](#)) have been proposed. PlenOctrees avoid the use of neural networks, enabling faster and more efficient computation while still achieving high-quality results. In certain scenarios, such representations may offer situational advantages over neural-network-based approaches, providing a more practical solution for specific use cases.

We believe that the process of using neural fields to reconstruct images is particularly suitable for visualization, which is why we chose image reconstruction as our focus. Images can intuitively demonstrate the process of this method, helping us analyze it more effectively.

We adopted a greedy algorithm to prioritize the selection of the most influential pixels. This method prioritizes selecting some sample pixels with the highest error to interpolate, helping to reconstruct the image with fewer samples while retaining key features. By focusing on these impactful pixels, we aim to balance resources with quality, ensuring that critical characteristics of the image are preserved.

Using multiple 1D approximations offers several potential advantages. First, the computational complexity of approximating a 2D function can be reduced by decomposing it into a set of 1D interpolations, each of which is individually faster and more interpretable. Linear interpolation between samples gives an efficient means to estimate values. By using multiple projections for linear interpolation, we can compensate for the information lost during the reduction from 2D to 1D, potentially resulting in a more accurate overall approximation. These 1D problems are inherently simpler and less resource-intensive.

1.3 Methods

The primary objective of this work is to evaluate the accuracy of using multiple 1D approximations as an alternative to traditional methods for approximating 2D functions. Specifically, we aim to:

- Evaluate the accuracy of greedy sampling and projections in approximating 2D functions via multiple 1D approximations.
- Empirically validate the approach through experiments on some images, using metrics such as Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR).
- Investigate the trade-offs between computational resources and approximation quality in comparison to traditional neural network-based methods.
- Optimize sampling strategies and interpolation methods to improve overall performance.
- Discuss the potential of this method to complement existing deep learning techniques, particularly in scenarios with limited computational resources.

Chapter 2

Related Work

2.1 The Universal Approximation Theorem

The Universal Approximation Theorem (UAT) is a foundational result in neural network theory that guarantees the representational power of neural networks. It asserts that a feedforward neural network with a single hidden layer containing a finite number of neurons can approximate any continuous function on a compact subset of \mathbb{R}^n , given that the activation function is non-linear and not polynomial ([Hornik, 1989](#)). While UAT provides a theoretical guarantee of representation capability, it does not address practical aspects such as training speed or complexity.

Our approach diverges from the universal approximations guaranteed by UAT by focusing on simpler, more computationally efficient approximations. By using one-dimensional (1D) function approximations combined in an ensemble-like manner, we aim to strike a balance between computational speed and functional accuracy.

2.2 Neural Fields

In recent years, neural networks have revolutionized the field of computer vision, achieving impressive results in tasks such as image classification, object detection, and generative modeling. Neural fields, a class of continuous neural network representations, have emerged as a powerful tool for modeling complex signals such as 3D shapes, light fields, and physical quantities. One prominent area of research that has

benefited from these advancements is novel view synthesis. A notable method in this field is Neural Radiance Fields (NeRF), which uses neural networks to represent 3D scenes for high-quality novel view rendering.

NeRF models a 3D scene by encoding volumetric density and color at every point in space, using a neural network that takes 3D coordinates and a viewing projection as inputs. By leveraging volumetric rendering techniques, NeRF produces realistic images from novel viewpoints, making it a significant breakthrough in view synthesis. However, despite its success, NeRF has limitations, particularly in terms of computational speed and the need for extensive training data. (Mildenhall et al., 2020).

2.3 AAA Approximation Methods

The Adaptive Antoulas-Anderson (AAA) algorithm is widely used for rational function approximation. The AAA method approximates functions by selecting support points and constructing rational representations, offering a balance between accuracy and computational complexity (Nakatsukasa et al., 2018).

Our understanding of the efficiency of this method lies in its ability to adaptively select the most important support points, rather than simply using all data points. The AAA algorithm not only interpolates the selected support points but also optimizes the fit to other points, ensuring a robust and accurate approximation. It achieves this by greedily selecting support points, which are the positions with the worst error in the previous iteration, and iteratively constructing a rational function that minimizes the overall approximation error (Nakatsukasa et al., 2018). This approach is well-suited for handling complex approximation problems.

However, the AAA algorithm is limited to functions of a single variable.

2.4 Other Methods

In this section, we introduce several existing methods that are used for function approximation, highlighting their unique characteristics and approaches. These methods will be compared with ours.

2.4.1 Neural Network

Neural networks (NNs) are a widely-used class of models for function approximation. The network operates directly on the input space, learning patterns through compositions of linear transformations and nonlinear or non-polynomial activation functions. While powerful, such networks may struggle to approximate functions with high-frequency components due to their limited ability to capture fine-grained oscillatory behavior. This is often referred to as the “spectral bias” ([Rahaman et al., 2019](#)) where NNs tend to first learn low-frequency features before adapting to high-frequency variations.

2.4.2 Fourier Features

Fourier features augment neural networks by mapping inputs into a high-dimensional space using sinusoidal functions, specifically sin and cos transformations. This allows the network to better approximate high-frequency components by leveraging the periodic nature of Fourier basis functions ([Tancik et al., 2020](#)). The addition of these features enables a richer representation of the input space, effectively alleviating the spectral bias issue. The approach is particularly useful in applications such as image reconstruction and modeling functions with sharp transitions or periodic patterns.

2.4.3 Kernel Regression

Kernel regression is a non-parametric method for function approximation that estimates function values based on weighted averages of nearby data points. A kernel

function, such as the Gaussian or polynomial kernel, is used to compute weights based on the distance between the query point and training samples. Kernel regression is versatile and performs well for smooth functions (Silverman, 1986).

2.4.4 Piecewise Linear Interpolation

`LinearNDInterpolator` is a method specifically designed for interpolating scattered data in multiple dimensions. It relies on *Delaunay triangulation* to partition the input domain into simplices (e.g., triangles in 2D, tetrahedra in 3D) and then performs linear interpolation within each simplex (Lee and Schachter, 1980; Virtanen et al., 2020). The algorithm works as follows:

1. **Delaunay Triangulation:** The input points are organized into a triangulated mesh, ensuring that no point lies inside the circumcircle (2D) or circumsphere (3D) of any simplex. This step ensures stability (Lee and Schachter, 1980).
2. **Linear Interpolation:** Within each simplex, the interpolated value is computed as a weighted average of the values at the simplex's vertices. The weights are derived from the barycentric coordinates of the query point relative to the simplex.

Chapter 3

Methodology

3.1 Problem Definition

The goal of this research is to reconstruct a 2D image using a set of sampling points and interpolation methods. The challenge lies in capturing the image's structure using sampling and then performing accurate interpolation to achieve high-fidelity reconstruction.

3.2 Coordinate-Based Function Approximation

3.2.1 Projection of Pixel Coordinates and Values

Our method involves projecting the 2D coordinates and pixel values onto each projection to form 1D curves. The process is outlined as follows:

Step 1: Calculate Projection Coordinates

We'll define our terminology and methodology with an example of calculating projections of pixel values for a 256×256 image.

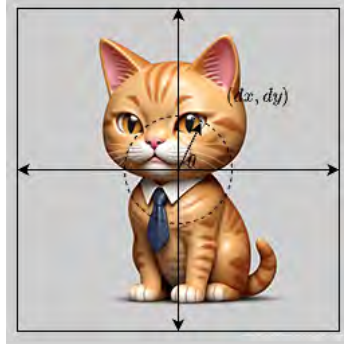


Figure 3.1: A 256×256 image with origin shifted to $(128, 128)$

For the purposes of illustration, we have shifted the origin point to $(128, 128)$ in our 256×256 image.

Now imagine there is a unit circle centered at the origin.

Our projection vectors will be unit vectors terminating somewhere on the circle with an angle θ from the positive x axis. The coordinates of the projection vectors are given by

$$\begin{pmatrix} dx = \cos \theta \\ dy = \sin \theta \end{pmatrix}.$$

We will be projecting the coordinates (x_i, y_i) of a pixel in the image onto the projection vectors.

Recall that the formula for projecting a point \vec{u} onto a vector \vec{v} is given by

$$\text{proj}_{\vec{v}} \vec{u} = \frac{\vec{u} \cdot \vec{v}}{\vec{v} \cdot \vec{v}} \vec{v}.$$

Since $\cos^2 \theta + \sin^2 \theta = 1$, we have

$$\begin{aligned}
\text{proj}_{\vec{v}} \vec{u} &= \frac{\vec{u} \cdot \vec{v}}{\vec{v} \cdot \vec{v}} \vec{v} \\
&= \frac{x_i dx + y_i dy}{1} \vec{v} \\
&= (x_i dx + y_i dy) \vec{v} \\
&= c_i \vec{v}
\end{aligned}$$

where $c_i = x_i dx + y_i dy$ is the scalar projection of the pixel coordinates onto the projection vector.

Step 2: Sort the Projection Coordinates and Values

We sort the projection coordinates c_i in ascending order, along with the corresponding pixel intensity values, to prepare for interpolation.

Step 3: Handle Duplicate Coordinates

Since multiple sampled points may project to the same coordinate, we compute the average of their pixel values to ensure a unique mapping for interpolation.

Step 5: Construct 1D Curves

For each channel, the 1D curve consists of points (c_j, value) .

3.3 Generating Uniformly Distributed Projections

Instead of relying on purely random projections, which may not provide representative coverage of the image space, we generate projections that are uniformly distributed starting from a random initial angle. This approach ensures consistent sampling and better approximation results while introducing randomness to avoid alignment with any specific image features.

Method:

- We first generate a random initial angle θ_0 in the interval $[0, 2\pi)$:

$$\theta_0 = \text{rand}(0, 2\pi)$$

- We divide the unit circle into K equal segments, where K is the number of desired projections.

- The angle increment is calculated as:

$$\Delta\theta = \frac{2\pi}{K}$$

- The angles θ_k are then calculated starting from θ_0 as:

$$\theta_k = (\theta_0 + k\Delta\theta) \bmod 2\pi, \quad k = 0, 1, \dots, K - 1$$

- The corresponding projection vectors are given by:

$$(dx_k, dy_k) = (\cos \theta_k, \sin \theta_k)$$

This method provides a set of evenly spaced projections starting from a random initial angle, ensuring consistent sampling while avoiding potential biases from fixed starting angles.

3.3.1 Linear Interpolation of Curve Values

We use linear interpolation to estimate pixel values between sampled points. This method is chosen for several reasons:

Fast Computation: Linear interpolation involves simple arithmetic operations to calculate the interpolated values, specifically finding the weighted average between two neighboring points. To further optimize the process, binary search can be employed to quickly locate the pair of neighboring points where the query lies, significantly enhancing efficiency for large datasets.

It avoids the need for more complex techniques like higher-order polynomial fitting or spline interpolation, which are computationally expensive and can involve solving larger systems of equations.

Interpretability: The method’s mechanism is straightforward: the interpolated value is a linear combination of two known data points. This simplicity allows practitioners to easily understand and explain how the interpolated value is predicted.

The local nature of linear interpolation ensures that the contribution of each neighboring data point is explicit, making it intuitive to diagnose interpolation behavior in specific regions.

Summary of Key Properties:

- **Continuity:** The interpolated curve is continuous, with no sudden jumps or discontinuities.
- **Locality:** Each interpolated value depends only on its two nearest data points, ensuring that changes in one part of the curve do not affect distant regions.
- **Smoothness:** The interpolated curve is smooth except at the sample points.

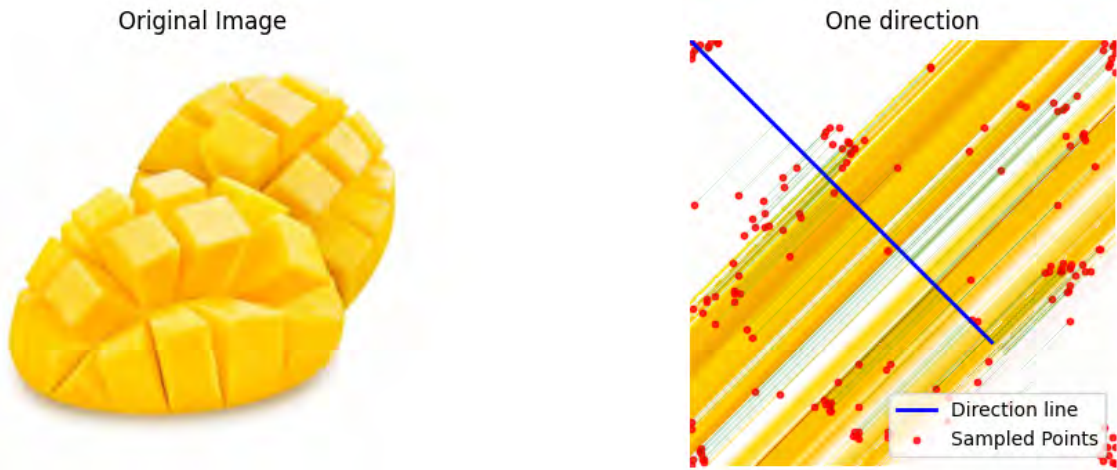


Figure 3.2: Comparison of original image (left) and reconstruction using a single projection (right)

As shown in Figure 3.2, the image on the right is the reconstruction using a single projection, which is consistent with our expectations, demonstrating the simplified feature representation when only one projection is used.

3.4 Greedy Algorithm for Sampling Point Selection

To select sampling points, we use a greedy algorithm that iteratively chooses the points with the highest error. The process is described in the following pseudocode:

Pseudocode for Greedy Sampling Algorithm:

Input: Image tensor, Number of sample points N

Output: Selected sampling points (Px, Py)

1. Initialize with a random pixel point (Px, Py)
2. For each iteration t from 1 to N do:
 3. Project all sampled pixel points onto each projection to generate curves
 4. Predict pixel values using linear interpolation
 5. Calculate Mean Squared Error (MSE) for all points
 6. Select the point with the highest MSE and add to (Px, Py)
7. Return (Px, Py)

This approach greedily reduces the approximation error.

3.4.1 Diversity-enhanced Greedy Selections

This method combines MSE-based selection with strategies to maximize spatial diversity and edge coverage.

Spatial Diversity: Spatial diversity is promoted by selecting points that maximize the minimum distance to already selected points:

$$\text{Diversity Score} = \min_{p \in P_s} \|p_c - p\|, \quad \forall p_c \in P_c$$

where P_s is the set of already selected points, P_c is the set of candidate points, and $\|\cdot\|$ represents the Euclidean distance.

Edge Coverage: Edge coverage is enhanced by assigning higher priority to points

located in regions with high gradient magnitude:

$$\text{Edge Score} = \|\nabla I(p_c)\|, \quad \forall p_c \in P_c$$

where $I(p_c)$ represents the intensity value at point p_c , and $\nabla I(p_c)$ is the gradient at that point.

The final selection criterion combines these scores:

$$\text{Selection Score} = \alpha \cdot \text{MSE Reduction} + \beta \cdot \text{Diversity Score} + \gamma \cdot \text{Edge Score}$$

where α, β, γ are weights balancing the contributions of MSE reduction, spatial diversity, and edge coverage.

3.5 Image Reconstruction

3.5.1 Average Method

Using all sampled points and projections, we predict pixel values across the entire image and compute the average to generate a reconstructed image. This method provides a baseline reconstruction by equally weighting all projections.

3.5.2 Optimization Using Least Squares

To further improve reconstruction accuracy, we assign weights to each projection and use a least-squares optimization to determine the optimal set of weights. We construct a linear system:

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2,$$

where \mathbf{X} is the matrix of predicted values from each projection, and \mathbf{y} is the vector of original pixel intensities. The optimized weights \mathbf{w} are used to compute a weighted reconstruction of the image.

3.6 Evaluation Metrics

To assess the quality of the reconstructed images and quantify the performance of different sampling and interpolation methods, we employ the following metrics:

3.6.1 Mean Squared Error (MSE)

The Mean Squared Error (MSE) is a standard metric used to measure the average squared difference between the original and reconstructed images, providing a quantitative measure of image error:

$$\text{MSE} = \frac{1}{HW} \sum_{x=1}^H \sum_{y=1}^W (I_{\text{original}}(x, y) - I_{\text{reconstructed}}(x, y))^2$$

where H and W denote the height and width of the image, respectively. Lower MSE values indicate a closer match to the original image, signifying better reconstruction quality.

3.6.2 Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) is another critical measure used to evaluate the quality of the reconstructed image relative to the original. It is defined as:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right)$$

where MAX_I represents the maximum possible pixel value of the image (typically 255 for 8-bit images). A higher PSNR value typically correlates with a higher quality of the reconstructed image as perceived by human observers.

Chapter 4

Experimental Design

4.1 Objective

The primary goal of our experimental design is to evaluate different sampling strategies and interpolation methods for optimal image reconstruction. Our aim is to determine the best approach by systematically comparing various sampling techniques and interpolation methods, and then to identify the optimal combination of projections and sampling points. Additionally, we provide a streamlined comparison with other established models.

4.2 Experiment Setup

We use RGB images with 256×256 pixels, totaling 65,536 pixels per image. This resolution provides a balance between maintaining meaningful detail and ensuring computational feasibility. Our sampling strategies and interpolation techniques are compared to evaluate their impact on reconstruction quality. We run each model with various combinations of different numbers of projections and sampled points on some images to assess performance comprehensively.

4.3 Sampling and Interpolation Strategies

4.3.1 Sampling Methods

We explore several sampling strategies to select points for image reconstruction:

- **Uniform Sampling:** Randomly selects points uniformly across the image.
- **MSE Greedy Selection:** Iteratively selects points based on the highest mean squared error (MSE) across all channels.
- **Diversity-enhanced Greedy Selections:** Combines MSE-based selection with a strategy to maximize spatial diversity and edge covering among sampled points.

4.3.2 MSE Calculation Methods

Different methods for handling MSE calculations for channel data during greedy selection:

- **Average MSE across Channels:** Computes the MSE for all channels and averages the result.
- **Max MSE per Channel:** Selects the maximum MSE among all channels for each pixel.
- **MSE of Channel Averages:** First calculates the average value across all channels for each pixel, then computes the MSE of these averages.

4.3.3 Interpolation Methods

We employ two interpolation techniques for reconstructing the image from the sampled points:

- **Linear Interpolation:** Provides a simple and efficient way to estimate values between sampled points but may lack smoothness.
- **Spline Interpolation:** Creates smoother curves by fitting spline functions to the sampled data, improving the visual quality of the reconstruction.

4.3.4 Optimization of Sampling Points and Projections

After determining the most effective sampling method, we explore various combinations of sampling points (N) and projections (K) to find the optimal configuration

for image reconstruction.

4.3.5 Parameter Count

The total number of parameters used by our Pre-Computed 1-d Curves model is calculated as:

$$\text{Total Parameters} = K \times N \times (C + 1) + 2$$

where K is the number of projections, C is the number of channels (3 for RGB images), N is the number of sampling points, and 2 is the initial projection. This formula allows comparison to other models.

Alternatively, we can use a compressed version that only stores the coordinates and values of the sampling points without calculating the projection:

$$\text{Total Parameters (Compressed)} = (2 + C) \times N + 2$$

4.4 Comparative Models

For comparative analysis, we benchmark against several well-known models:

- **Neural Network**
- **Neural Network (with Fourier Features)**
- **Kernel Regression**
- **LinearNDInterpolator**

4.5 Optimized Computational Techniques

To improve the computational efficiency of our image reconstruction process, we have implemented several key techniques

4.5.1 Pixel Downsampling

To manage computational resources more efficiently, we employ a pixel downsampling technique during the sampling point selection process. Instead of evaluating the entire set of 65,536 pixels in each iteration, we focus on a randomly selected subset of 5,000 pixels. This significant reduction in the number of pixels being processed allows for faster computations and less memory usage, without a substantial loss in image quality.

Greedy Sampling Enhancement: Within this downsampling framework, we enhance our sampling efficiency using a greedy algorithm:

- Each iteration of the sampling process begins by randomly selecting 5,000 pixels from the entire image.
- From this subset, we identify and select the 10 pixels with the highest mean squared error (MSE) between the reconstructed and original pixel values.
- These 10 pixels are then added to our set of sampling points, selecting the pixels with the highest error in our model.

4.5.2 Memory Optimization

We optimize memory usage by avoiding the storage of intermediate predicted images during the reconstruction process.

4.6 Evaluation Metrics

We assess the quality of reconstructed images using Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR), crucial metrics for evaluating image fidelity and quality. In addition to numerical metrics, we utilize various graphical tools to provide visual insights into the reconstruction quality:

- **Difference Images and Sampling Point Maps:** These visualizations com-

plement the numerical analysis by showing where sample points most frequently at and how well the sampling points represent the image structure.

- **PSNR and MSE Plots:** Graphs that plot PSNR and MSE values across different configurations of sampling methods and projections, offering a visual comparison of performance across different setups.

The results from both quantitative metrics and visual evaluations are analyzed to determine the efficacy of different sampling strategies and interpolation methods. This analysis helps in identifying the best-performing approaches and guiding future improvements in the algorithm.

4.7 Analysis of Parameter Impact

This section focuses on the impact of varying sampling points and projections, exploring how each factor influences the overall effectiveness of the image reconstruction process and its computational efficiency. We also examine how our approach compares in performance to other models under similar parameter constraints.

Chapter 5

Results and Analysis

5.1 Overview

This section presents our experimental results, focusing on the performance of different sampling methods, interpolation techniques, and MSE calculation strategies. We analyze the impact of these variables on image reconstruction quality through both quantitative measures and visual inspections.

5.2 Sampling Methods

The initial image of a small cat serves as our baseline for comparison (Figure [5.1](#)).

Original Image



Figure 5.1: Original image of a small cat used for analysis

5.2.1 Uniform Sampling

Using a uniform distribution to sample 10,240 points across 128 projections yielded suboptimal results. Figure 5.2 illustrates the outcomes, including the reconstructed images from averaging all projections, weighted reconstruction via least squares, and a difference image showing discrepancies. The PSNR is 19.96 dB and MSE is 0.010084.



Figure 5.2: From left to right: Average reconstruction, weighted least squares reconstruction, and difference image with sampled points in red (Number of Projections: 128, Number of sample points: 10240)

The reconstructed images using uniform sampling capture the basic color information of the original image; however, due to the insufficient number of sample points, the resulting image appears blurrier and lacks detail, particularly along the edges.

5.2.2 Greedy MSE Selection

Using a greedy approach that selects points with the highest MSE improved clarity (Figure 5.3). The PSNR is 25.72 dB and MSE is 0.002680, showing enhanced detail around major features.



Figure 5-3: Improved image reconstruction using greedy MSE selection, with sample points concentrating on key features (Number of Projections: 128, Number of sample points: 10240)

Compared to Uniform Sampling, the MSE and PSNR of Greedy MSE Selection are better than Uniform Sampling. The sample points selected by Greedy MSE Selection are more concentrated around the edges and details of the cat, while fewer points are distributed in the background. Consequently, the reconstructed image using Greedy MSE Selection exhibits higher quality with more pronounced edge details and patterns.

5.2.3 Advanced Greedy Selection

Incorporating spatial diversity and edge information into the selection process did not yield better results. The sample points were primarily concentrated along the edges of the cat, which did not necessarily improve the image quality (Figure 5-4). The PSNR is 18.54 dB and MSE is 0.013996.



Figure 5-4: Advanced greedy selection results (Number of Projections: 128, Number of sample points: 10240)

Due to the inclusion of edge information, the sample points are primarily concentrated along the edges of the cat. However, this distribution of sample points did not lead to a better result. The reconstructed image still lacks clear edges. This suggests that when edge points are projected onto the projection map, they do not contain enough background information or subject details near the edges.

Table 5.1: Comparison of Sampling Methods

Sampling Method	PSNR (dB)	MSE
Uniform Sampling	19.96	0.010084
Greedy MSE Selection	25.72	0.002680
Advanced Greedy Selection	18.54	0.013996

10,240 points, 128 projections

Using only the greedy MSE-selected sample points tends to cover areas that may require both edge points and surrounding regions to achieve better reconstruction. When using Uniform Sampling, too many sample points falling on the background can cause the subject to appear unclear. We hypothesize that including all detailed regions of the image, along with their surroundings, as sample points could lead to better reconstruction. Greedy MSE selection tends to focus sample points on areas with more image details.

To verify this hypothesis, we chose two images for comparison: one with a complex subject and a simple background (Figure 5.5)(PSNR: 19.00 dB, MSE: 0.012580), and another with a simple subject but a complex background (Figure 5.6)(PSNR: 23.47 dB, MSE: 0.004499), using greedy MSE selection in both cases.

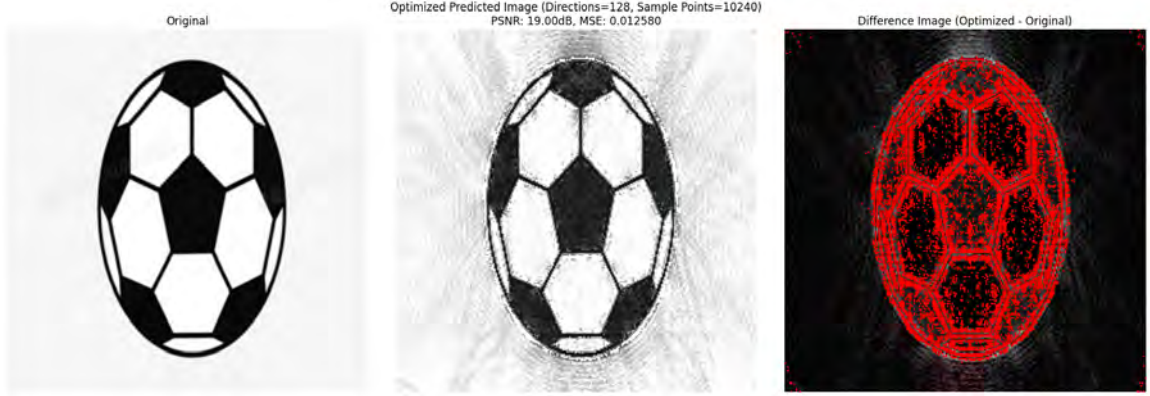


Figure 5.5: Sample points concentrated on the complex subject with a simple background (Number of Projections: 128, Number of sample points: 10240)

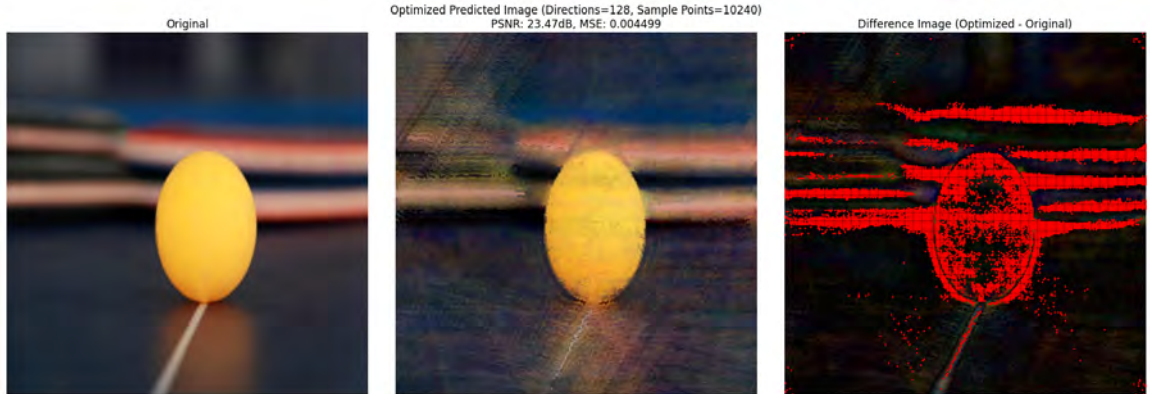


Figure 5.6: Sample points distributed on the simple subject with a complex background (Number of Projections: 128, Number of sample points: 10240)

Due to the characteristics of projection and linear interpolation, more sample points are needed to capture regions with high pixel value differences, such as around

color block edges, to capture image details. Greedy MSE selection targets such regions, which helps achieve better reconstruction results when the number of sample points is fixed.

5.3 MSE Calculation Methods

We compared three approaches to calculating MSE using the same number of projections and sample points. The differences among the methods were minimal; however, the **Max MSE per Channel** method yielded the best reconstruction results. This is because a large error in any single channel can cause significant color space shifts, leading to noticeable color distortions. By selecting the maximum MSE, we select these color distortions. This method better maintains the visual quality of the image, especially in scenarios where color information is critical. Therefore, we chose the Max MSE per Channel method for calculating MSE at each step.

Table 5.2: Comparison of MSE Calculation Methods

Method	PSNR (dB)	MSE
Average MSE Across Channels	24.50	0.003200
MSE of Channel Averages	25.00	0.002800
Max MSE Per Channel	25.72	0.002680

Number of Projections: 128, Number of Sample Points: 10240

5.4 Interpolation Methods

We compared linear and spline interpolation methods with 10,240 sample points and 128 projections to assess their impact on image reconstruction quality. The visual results are shown in Figures 5-3 and 5-7, respectively. The linear interpolation achieved a PSNR of 23.72 dB and an MSE of 0.004248. The sample point distribution for spline interpolation was more dispersed.

Table 5.3: Comparison of Interpolation Methods

Method	PSNR (dB)	MSE
Linear Interpolation	25.72	0.002680
Spline Interpolation	23.72	0.004248

Number of Projections: 128, Number of Sample Points: 10240



Figure 5-7: Compare image reconstruction using spline interpolation and linear interpolation (Number of Projections: 128, Number of sample points: 10240)

Spline interpolation produces smoother curves compared to linear interpolation. However, these smooth curves failed to reconstruct the correct image in the early stages and could not accurately select the sample points falling on the subject (as shown in Figure 5-8 and 5-9, 5-10). This is the result of using 16 sample points and 16 projections for comparison. Additionally, spline interpolation increases computational complexity. Therefore, we opted for linear interpolation due to its better

computational speed and reconstruction accuracy.

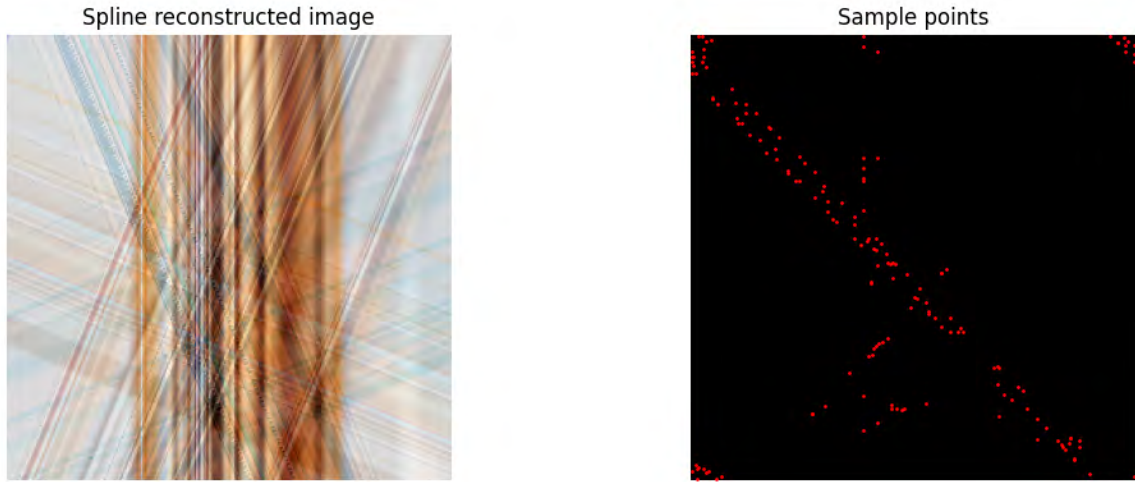


Figure 5-8: Reconstruction image using spline interpolation with a small number of sample points, showing the sample point distribution (Number of Projections: 16, Number of sample points: 160)

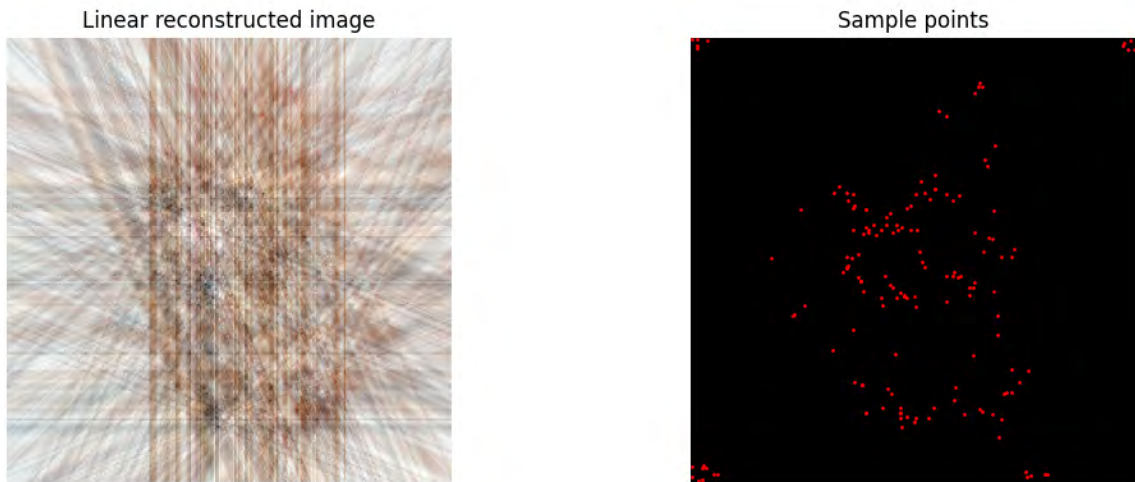


Figure 5-9: Reconstruction image using linear interpolation with a small number of sample points, showing the sample point distribution (Number of Projections: 16, Number of sample points: 160)

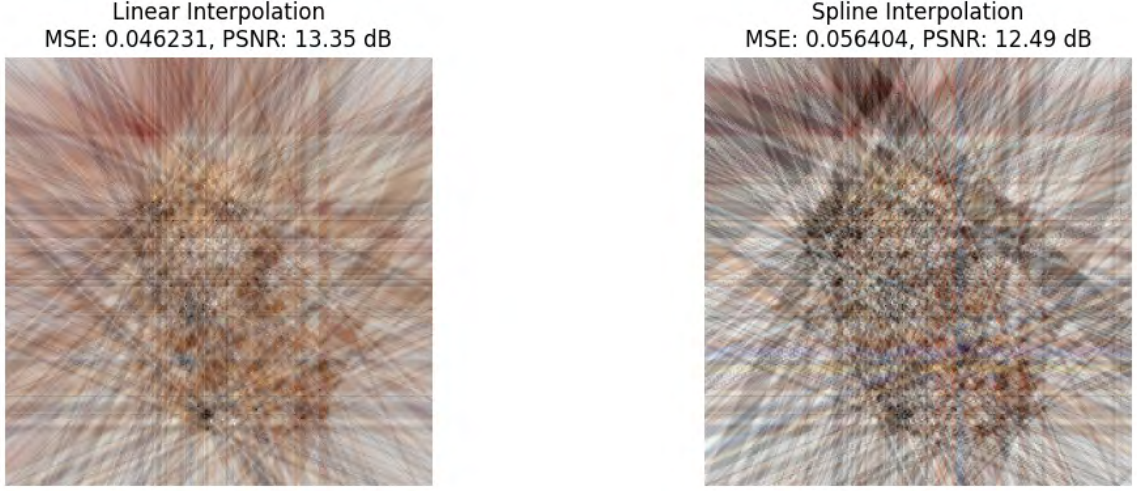


Figure 5.10: Reconstruction of linear vs spline with same sample points chosen by linear (Number of Projections: 16, Number of sample points: 160)

Comparing the difference in the images of the two interpolation methods, it can be seen that the reconstructed image using spline interpolation deviates more significantly from the original image during the early stages of reconstruction compared to the linear interpolation. Additionally, spline interpolation tends to select sample points less accurately on key features of the subject, which results in lower reconstruction quality. This suggests that linear interpolation, though simpler, may provide comparable image quality under these conditions.

5.5 Optimization of Sampling Points and Projections

In this section, we test the hypothesis that balancing the number of projections (to capture positional information) with the number of sample points (to capture value information) yields effective image reconstruction. This hypothesis posits that an optimal combination of these two parameters in fixed number of total parameters is key to achieving high reconstruction quality, with sample points ideally distributed sparsely across the image to avoid unnecessary density.

We conducted a comprehensive analysis to evaluate how varying the number of sampling points and projections affects image reconstruction quality. The results are summarized in the parameters plot (Figure 5.11).

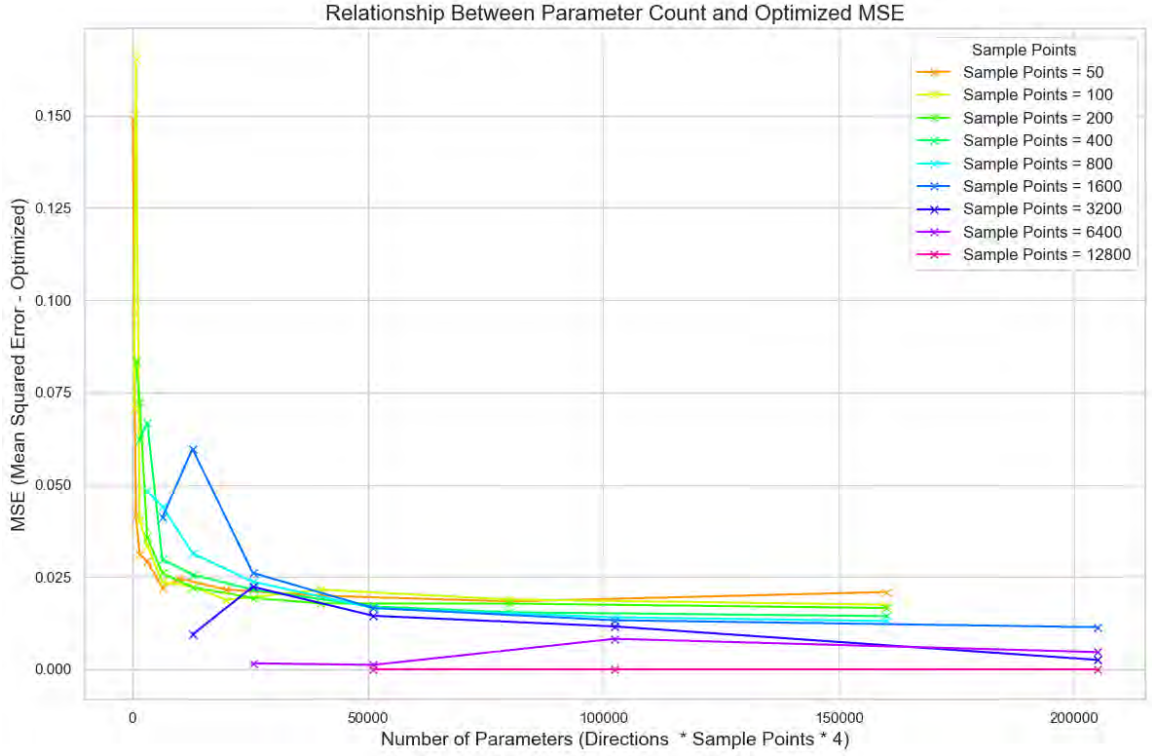


Figure 5.11: Plot of MSE and PSNR across different sampling points and parameters (Pre-Computed 1-d Curves model)

Because of the resource limitation, a 128x128 image was used, and the following parameters were considered:

num_projections_list
1, 2, 4, 8, 16, 32, 50, 100, 200, 400, 800, 1600, 3200, 6400, 12800
num_samples_list
50, 100, 200, 400, 800, 1600, 3200, 6400, 12800

Table 5.4: Number of projections and sample points used

Combinations with total parameters of Pre-Computed 1-d Curves model exceeding 300,000 were excluded. As shown in Figure 5.11, the minimum error occurs when

nearly all image pixels are used as sample points. The second-best result arises when approximately half the pixels are used as sample points. Notably, increasing the number of projections does not significantly reduce MSE, even when the parameter count is higher.

Theoretically, increasing sampling points allows for more accurate prediction of pixel values, while increasing projections helps localize details more precisely. Projecting 2D coordinates into 1D inevitably loses some information. Striking the right balance between projections and sample points enables sufficient retention of both positional and value information, leading to effective reconstruction.

From the results, we observe that under the same parameter count, increasing the number of sample points directly reduces MSE more significantly compared to increasing the number of projections. While additional projections do contribute to improved reconstruction, their effect is far less pronounced than that of increasing the sample points.

In addition to the parameter analysis above, we evaluated a compressed model where parameters are calculated before creating the interpolation curves. In this scenario, the parameter count depends solely on the number of sample points and is defined as:

$$\text{Total Parameters (Compressed)} = (2 + C) \times N + 2$$

The impact of varying the number of projections on reconstruction quality under this compressed parameter regime is shown in [Figure 5.12](#).

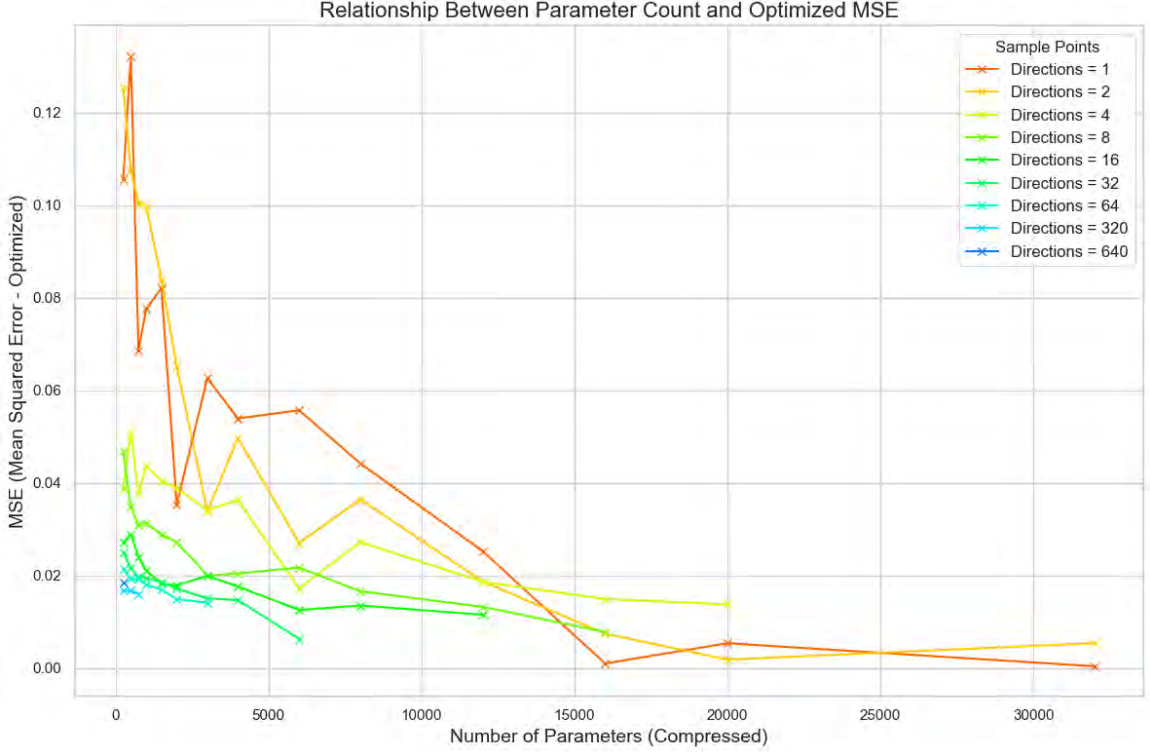


Figure 5.12: Plot of MSE across different compressed parameter counts. The curves are categorized by the number of projections (Compressed model)

From Figure 5.12, we observe that increasing the number of projections allows for achieving lower MSE at the same number of sample points. However, this improvement comes at the cost of increased computation time. The interpolation complexity of the pre-computed 1D curves model grows as $O(K \cdot T \cdot \log N)$, whereas the compressed model grows as $O(T \cdot K \cdot N \cdot \log N) + O(K \cdot T \cdot \log N)$. Here, K represents the number of projections, N is the number of sample points, T is the total number of pixels. This highlights a trade-off between reconstruction fidelity and computational efficiency, requiring careful consideration for real-time or resource-constrained applications.

To illustrate the effect of projection count on reconstruction, we compare images reconstructed using 4 projections and 64 projections (Figure 5.13). When fewer

projections are used, such as 4, ghosting artifacts appear around the background. Increasing the projection count to 64 significantly reduces these artifacts and results in a clearer reconstruction.

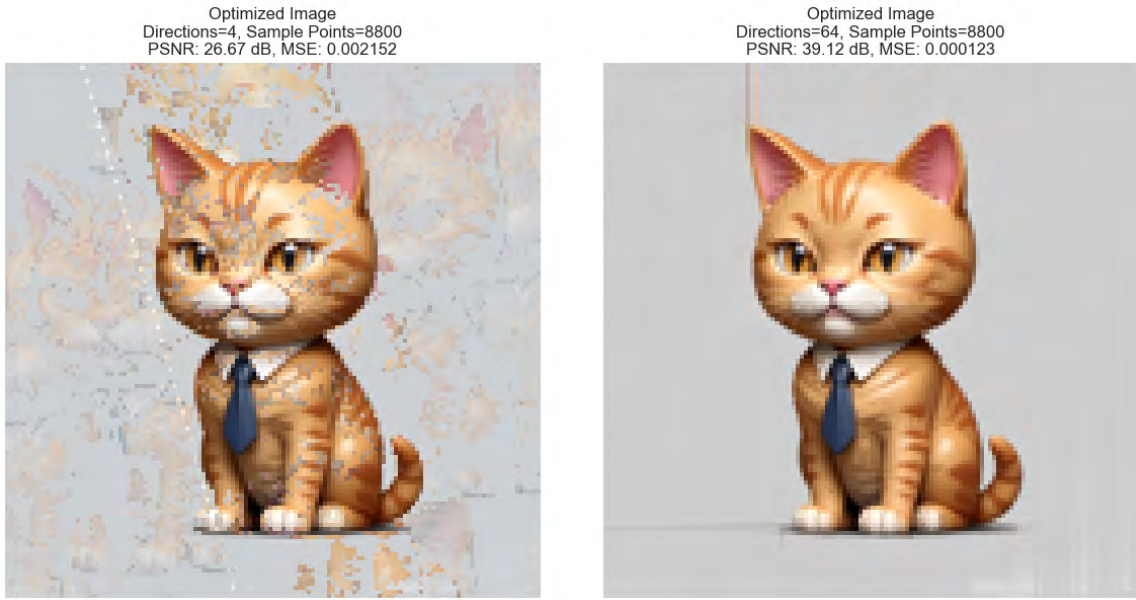


Figure 5-13: Comparison of reconstruction quality with 4 projections (left) and 64 projections (right) with same number of sample points.

5.5.1 Practical Thinking on Sampling and Projection Balance

For Pre-Computed 1-d Curves, the results reveal an interesting insight: contrary to our initial intuition, the reconstruction quality did not improve as expected by using a balanced number of projections and sample points. Instead, under the same parameter count, using as many sample points as possible led to better reconstruction results. The reason for this is that when each sample point has a unique value in the projection, we can increase the number of sample points to make the reconstructed image closer to the original image.

However, adding more projections has several benefits. First, it helps reconcile the quality of reconstructed images across different projection directions, resulting in a clearer background and sharper edges for the main subject. Second, increasing the

number of projections improves the accuracy of each step in the greedy selection of MSE because of the clearer image, which makes the selection of sample points more precise. In other words, more projections can be used to obtain better sample point locations during the selection process, and then, in the final step, a minimal number of sample points and a few projections can be used to achieve a good reconstruction result. When the number of sample points is close to the number of pixels in the image, the best reconstruction results are achieved.

For the compressed model, the trade-off between reconstruction quality and computational efficiency becomes particularly evident. As the parameter count in the compressed model depends solely on the number of sample points, adding more projections does not directly increase the parameter count but does increase computational complexity during inference. This is because additional projections require projecting all sample points and performing sorting operations for each projection, which grows linearly with the number of projections.

Despite this computational cost, the compressed model demonstrates its advantages in scenarios with limited storage, as it only requires storing a minimal number of parameters. Moreover, the ability to dynamically adjust the number of projections during inference provides flexibility for applications where computational resources vary.

In conclusion, for the Pre-Computed 1-d Curves model, increasing the number of sample points remains the most effective way to improve reconstruction quality under fixed parameter constraints. This approach provides consistent improvements in reconstruction fidelity, particularly when the number of sample points approaches the total number of pixels in the image.

For the compressed model, while reconstruction quality is also improved by increasing sample points, its main advantage lies in balancing storage and computa-

tional flexibility. Since the parameter count depends only on the number of sample points, this model is well-suited for scenarios where storage is limited, but computational resources are available for on-the-fly inference. Moreover, increasing the number of projections can help reconstruct better images without increasing the parameter count. The choice between these two approaches ultimately depends on application requirements, such as the trade-off between inference speed, storage constraints, and reconstruction fidelity.

5.6 Comparison of Different Projections in Selection and Reconstruction

In this section, we compare the qualities of using a larger number of projections during greedy selection followed by reconstruction with a smaller number of projections. Shrinking the number of projections allows for better reconstruction with fewer sample points, reducing the parameter count while maintaining quality. We consider three different scenarios:

1. Using the same number of projections for both greedy selection and reconstruction.
2. Using a larger number of projections for greedy selection, followed by reconstruction using a newly generated smaller number of projections.
3. Using a larger number of projections for greedy selection, followed by selecting a subset of those projections for reconstruction.

Figure 5.14 shows the results of using the same number of projections for both greedy selection and reconstruction.



Figure 5-14: Reconstruction results using the same number of projections for greedy selection and reconstruction (Number of Projections: 3, Number of sample points: 20001)

Figure 5-15 presents the results of using a larger number of projections for greedy selection, followed by reconstruction using a newly generated smaller number of projections.



Figure 5-15: Reconstruction results using a larger number of projections for greedy selection and a newly generated smaller number of projections for reconstruction (Number of Projections: 3, Number of sample points: 20001)

Figure 5-16 shows the results of using a larger number of projections for greedy selection, followed by selecting a subset of those projections for reconstruction.



Figure 5-16: Reconstruction results using a larger number of projections for greedy selection and selecting a subset of those projections for reconstruction (Number of Projections: 3, Number of sample points: 20001)

Table 5.5: PSNR and MSE Comparison Across Scenarios

Scenario	PSNR (dB)	MSE
Same Number of Projections (Scenario 1)	22.22	0.00593
Larger to Newly Generated Projections (Scenario 2)	19.63	0.010886
Larger to Subset of Projections (Scenario 3)	26.78	0.002100

Number of Projections: 3, Number of Sample Points: 20001

We conducted five trials for each scenario and averaged the results. The results indicate that scenario 3, where a larger number of projections are initially used for greedy selection, followed by selecting a subset of those projections for reconstruction, yields the best reconstruction quality. This observation can be attributed to the following reasons:

Enhanced Accuracy in Sampling Points When a larger number of projections are used during greedy selection, it allows the algorithm to identify the highest MSE sample points with higher precision in each greedy selection. By leveraging more projections, the algorithm can reconstruct a better image during every step of the greedy selection process. This ensures that more sample points are accurately placed along high MSE regions, such as the main body in the data.

Better Alignment Between Selection and Reconstruction Selecting a subset of the projections used during greedy selection ensures intrinsic consistency between the selection and reconstruction phases. By reusing projections from the greedy selection process, the algorithm avoids potential differences that could occur if an entirely new set of projections were introduced during reconstruction. This alignment ensures that the reconstruction has the same structure gained during the selection phase, leading to improved overall quality.

5.7 Comparison with Other Models

This section presents a comparison of our proposed method with other traditional and modern approaches in image reconstruction, including neural network models with and without Fourier features, Kernel Regression, and LinearNDInterpolator. We compare these models under different parameter configurations to assess their performance in terms of Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR).

5.7.1 Parameter Configuration

Unlike parametric models like neural networks, non-parametric methods such as Kernel Regression and LinearNDInterpolator do not have a fixed number of parameters. Instead, their “parameter count” often depends on the number of data points and the specifics of the implementation. So we choose our compressed model to compare. We define both these methods and our proposed method as having parameters approximately equal to $5N + c$, where N is the number of sample points, and c is a constant, here simplified to $5N$ for ease of computation.

5.7.2 Experimental Setup

For our experiments, we used images of size 256x256 pixels. Our method utilized 64 projections, though adding more projections could potentially enhance performance. We conducted comparisons at different parameter levels: 4400, 17000, and 66800.

5.7.3 Results

Comparison at 4400 Parameters

Table 5.6: Performance Comparison at 4400 Parameters

Model	MSE	PSNR (dB)	Parameters
Neural Network (with Fourier features)	0.000164	37.86	4419
Neural Network (without Fourier features)	0.004850	23.14	4419
Our Proposed Method	0.015405	18.12	4400
Kernel Regression	0.057674	12.39	4400
LinearNDInterpolator	0.026366	15.79	4400

Number of Projections: 64, Number of Sample Points: 880

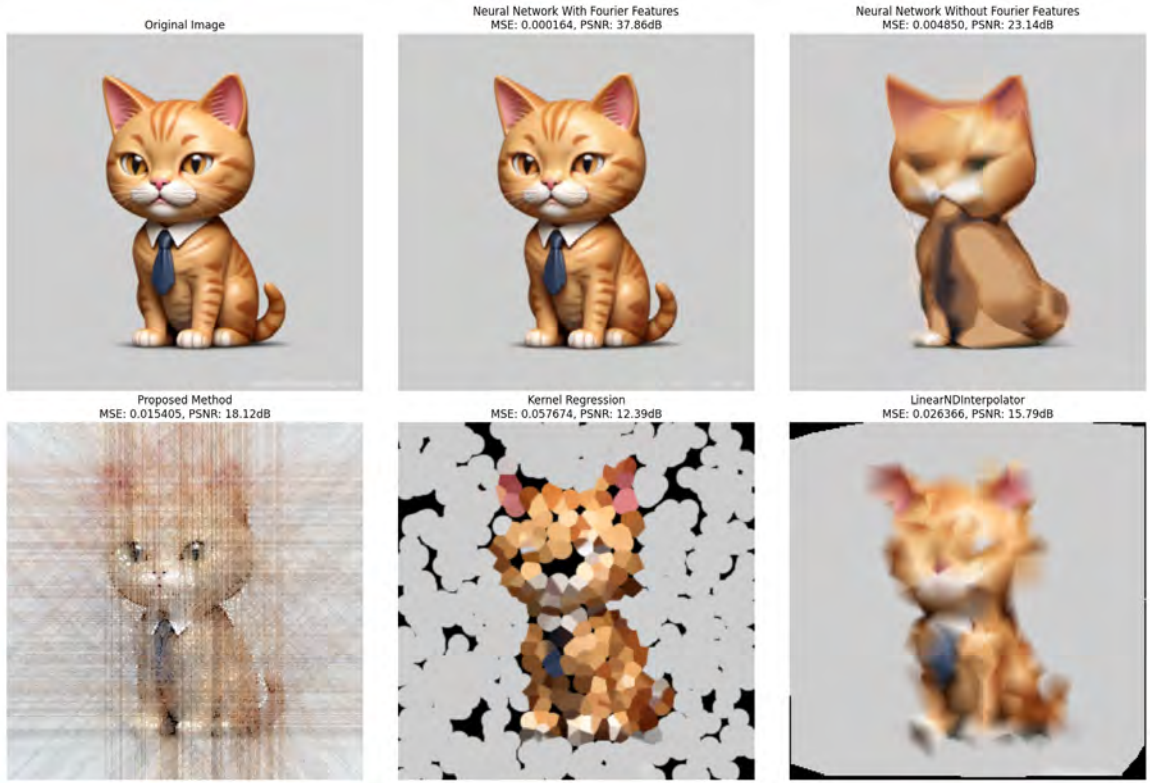


Figure 5-17: Visual comparison of reconstruction results at 4400 parameters (Number of Projections: 64, Number of sample points: 880)

Comparison at 17000 Parameters

Table 5.7: Performance Comparison at 17000 Parameters

Model	MSE	PSNR (dB)	Parameters
Neural Network (with Fourier features)	0.000009	50.44	17027
Neural Network (without Fourier features)	0.002145	26.69	17027
Our Proposed Method	0.005834	22.34	17000
Kernel Regression	0.005703	22.44	17000
LinearNDInterpolator	0.005498	22.60	17000

Number of Projections: 64, Number of Sample Points: 3400



Figure 5-18: Visual comparison of reconstruction results at 17000 parameters (Number of Projections: 64, Number of sample points: 3400)

Comparison at 66800 Parameters

Table 5.8: Performance Comparison at 66800 Parameters

Model	MSE	PSNR (dB)	Parameters
Neural Network (with Fourier features)	0.000007	51.83	66819
Neural Network (without Fourier features)	0.001957	27.08	66819
Our Proposed Method	0.001149	29.40	66800
Kernel Regression	0.001898	27.22	66800
LinearNDInterpolator	0.001795	27.46	66800

Number of Projections: 64, Number of Sample Points: 13360



Figure 5-19: Visual comparison of reconstruction results at 66800 parameters (Number of Projections: 64, Number of sample points: 13360)

5.8 Analysis

Our results show that the proposed method consistently outperforms other models across most parameter settings, except for the Neural Network model leveraging Fourier features, highlighting its strong reconstruction capabilities.

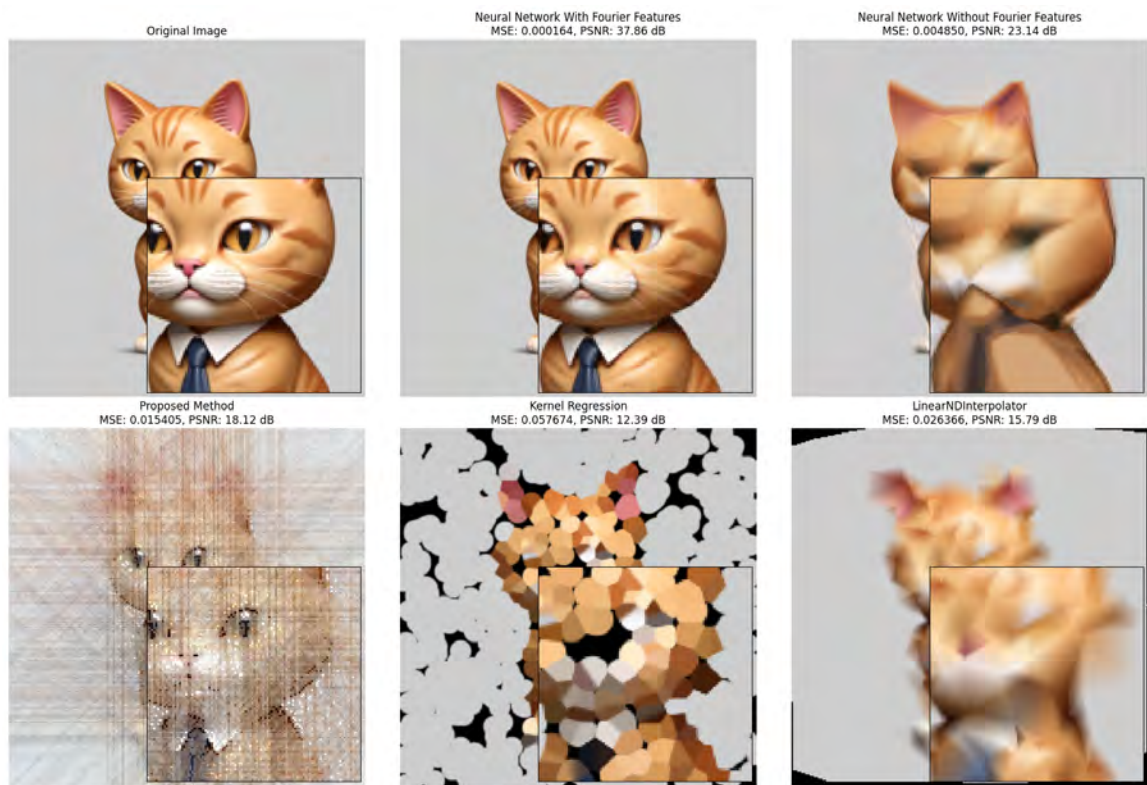


Figure 5.20: Detailed visual comparison of reconstructions with 4400 parameters

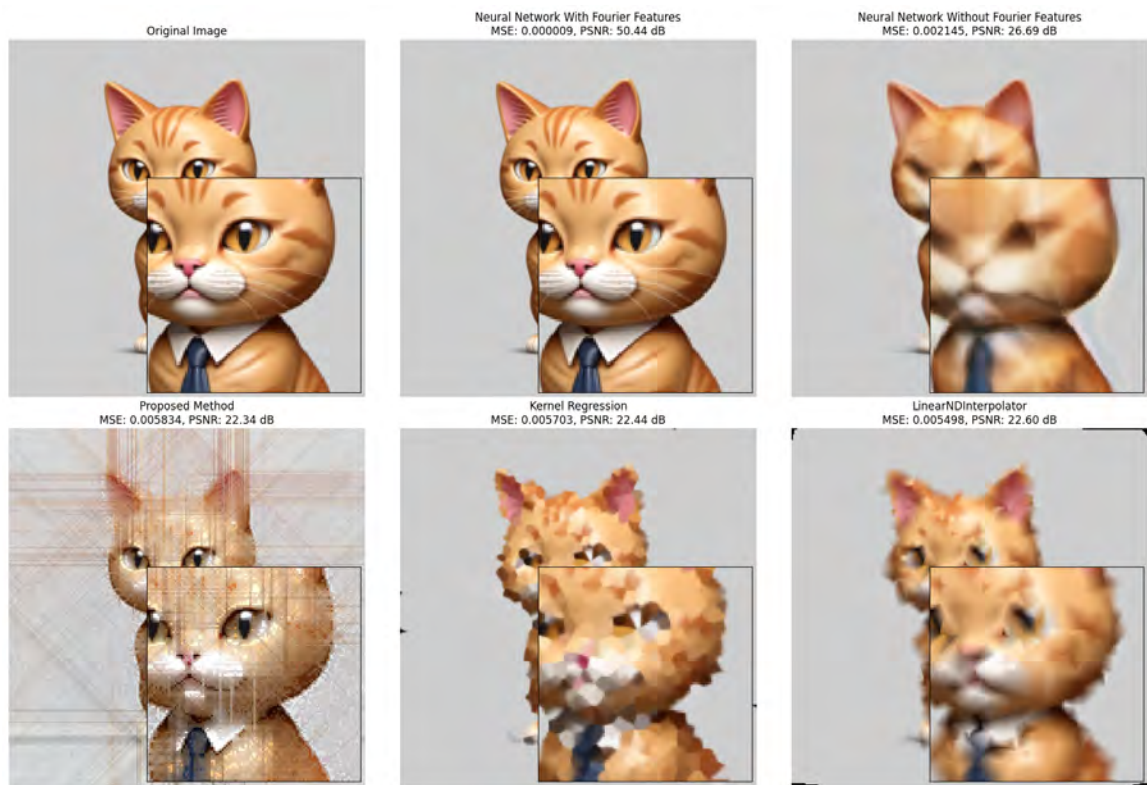


Figure 5.21: Detailed visual comparison of reconstructions with 17000 parameters



Figure 5.22: Detailed visual comparison of reconstructions with 66800 parameters

Moreover, our method tends to prioritize the reconstruction of areas with higher contrast, such as the eyes, resulting in clearer and more distinct images in these regions compared to other models. However, it exhibits some noticeable vertical striping in the background. Additionally, the reconstructed colors in our model deviate from the original image, appearing less balanced compared to other models. This suggests that our model prioritizes areas with significant changes in color or brightness rather than treating all regions uniformly. As a result, while it excels at reconstructing specific details, it may overlook smoother transitions in color or brightness across the image.

In conclusion, our approach is promising, especially for applications requiring detailed reconstruction of specific features within an image. Its ability to enhance high-contrast areas makes it particularly useful for scenarios where such details are

crucial. However, the challenge remains to balance this capability with the need for uniform image quality across all areas. This methodology holds significant potential for further refinement and application in advanced image reconstruction tasks.

Chapter 6

Conclusion

6.1 Summary of Findings

In this thesis, we explored an alternative approach to traditional neural network-based function approximation by decomposing a two-dimensional (2D) function approximation problem into multiple one-dimensional (1D) approximations. Motivated by the desire for computational speed and interpretability, we proposed a framework that leverages greedy sampling and projections to reconstruct images through 1D interpolations.

Our methodology involved generating uniformly distributed projections and projecting pixel coordinates onto these projections to form 1D curves. We employed linear interpolation for its computational simplicity and effectiveness in estimating pixel values between sampled points. A greedy algorithm was utilized to select sampling points that most significantly reduced the approximation error, enhancing the speed of our sampling strategy.

Through extensive experiments on image datasets, we evaluated various sampling methods, interpolation techniques, and error calculation strategies. The results demonstrated that our approach achieved good reconstruction quality compared to neural network models and other traditional methods.

6.2 Contributions

The key contributions of this work are as follows:

- **Novel Framework:** We introduced a new framework for approximating 2D functions using multiple 1D approximations, providing an alternative to neural network-based methods.
- **Efficient Sampling Strategy:** Our greedy sampling algorithm selected the pixels with highest error, enhancing reconstruction quality while reducing computational overhead.
- **Empirical Validation:** We conducted comprehensive experiments that validated the feasibility and effectiveness of our approach, demonstrating its advantage and disadvantages over traditional methods in specific metrics.

6.3 Limitations

While our approach has shown promising results, several limitations need to be acknowledged:

- **Dimensionality Constraint:** The method is currently tailored for 2D functions and may not directly extend to higher-dimensional data without significant modifications.
- **Interpolation Method:** We relied on linear interpolation, which, while efficient, may not capture complex variations in data as higher-order interpolation methods.
- **Sampling Density:** The quality of reconstruction is sensitive to the number and distribution of sampling points, requiring careful calibration for different types of images.

- **Projection Randomness:** When the number of projections is insufficient, the randomness can lead to highly variable reconstruction quality, with some reconstructions performing well and others poorly.

6.4 Future Work

Building on the foundation of this research, future work can explore several avenues:

- **Extension to Higher Dimensions:** Investigate the applicability of the proposed method to three-dimensional (3D) data and other higher-dimensional function approximation problems.
- **Advanced Interpolation Techniques:** Incorporate spline interpolation or other advanced methods to capture more complex data variations without substantially increasing computational load.
- **Integration with Machine Learning:** Explore hybrid models that combine the interpretability of our approach with the learning capabilities of neural networks, aiming to achieve both high performance and transparency.

Appendix A

Github

<https://github.com/laowangbushilaowang/Image-Reconstruction-through-Multiple-1D-Approximations>

References

- Hornik, K. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.
- Lee, D. and Schachter, B. (1980). Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242.
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4055–4064.
- Nakatsukasa, Y., Sète, O., and Trefethen, L. N. (2018). The aaa algorithm for rational approximation. *SIAM Journal on Scientific Computing*, 40(3):A1494–A1522.
- Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. (2019). On the spectral bias of neural networks. *International Conference on Machine Learning (ICML)*, pages 5301–5310.
- Silverman, B. W. (1986). *Density estimation for statistics and data analysis*, volume 26. CRC press.
- Tancik, M., Sitzmann, V., Martel, J. N. P., Lindell, D. B., and Wetzstein, G. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:7537–7547.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.
- Yu, A., Fridovich-Keil, S., Tancik, M., Li, Q., Khullar, S., Mildenhall, B., Pradhan, P. P., Srinivasan, P. P., Barron, J. T., and Ng, R. (2021). Plenotrees for real-time rendering of neural radiance fields. *arXiv preprint arXiv:2103.14024*.