G-RIPS Sendai 2026

Mitsubishi Group

Final Report

Authors:
Megan Joseph[1]
Marie Ange Malla Ndassi[2]
Han Nguyen[3]
Xiangwen Zhao[4]

Mentors:
Kean Fallon[+]
Ryuhei Takahashi[*]
Keito Arifuku[*]

[1] San Jose State University
[2] African Institute for Mathematical Sciences Cameroon
[3] Louisiana State University
[4] University of Osaka
[+] Academic Mentor, Iowa State University
[*] Industrial Mentor, Mitsubishi Electric Corporation

August 6, 2026

Contents

1 Introduction **2**

2 Prior Work **3**
2.1 Single-Pixel Imaging . . . 3
2.2 Single-Pixel Imaging Methods . . . 4
2.2.1 Single-Pixel Imaging Patterns . . . 4
2.2.2 Domain Sampling Methods . . . 6
2.3 Deep Learning Approaches for Single-Pixel Imaging . . . 7
2.3.1 Data-Driven Strategy . . . 8
2.3.2 Model-Driven Strategy . . . 8
2.4 Improvements of SPI Reconstruction . . . 10

3 Methods **10**
3.1 Overall Framework . . . 10
3.2 Sampling Strategies . . . 11
3.2.1 Round-Robin Adaptive Spectrum Sampling . . . 11
3.2.2 Iterative Threshold Relaxation . . . 11
3.3 Neural Networks Architecture . . . 12
3.3.1 U-Net . . . 12
3.3.2 ResNet . . . 14
3.3.3 SIREN . . . 14
3.3.4 Training Loss . . . 16
3.4 Data . . . 16
3.5 Neural Network Parameter Tuning . . . 18

4 Results **19**
4.1 Comparison with BFSI . . . 19
4.2 Comparison with FSI + DIP . . . 25

5 Conclusion **27**
5.1 Future Work . . . 28

1 Introduction

Imaging is one of the most important techniques for acquiring information about the physical world. Advances in imaging systems have enabled numerous applications, including biomedical imaging, remote sensing, industrial inspection, and autonomous navigation.

Conventional cameras use two-dimensional detector arrays, such as silicon-based charge-coupled devices (CCD) or complementary metal-oxide-semiconductor (CMOS) image sensors, consisting of hundreds of thousands to millions of pixels. Each pixel independently records the light intensity from a corresponding location in the scene. This approach enables high-resolution image acquisition in a single exposure and has become the standard technique in consumer cameras, smartphones, and many scientific imaging systems.

Despite its widespread success, conventional imaging is not suitable for every application. In particular, high-performance detector arrays are expensive or difficult to manufacture for certain wavelength ranges, including infrared, terahertz, and X-ray imaging. Furthermore, in low-light environments, each detector pixel receives only a limited number of photons, resulting in poor signal-to-noise ratio (SNR). Imaging through scattering media or under harsh environmental conditions also presents significant challenges for conventional cameras. These limitations motivate alternative imaging techniques that reduce the reliance on large detector arrays while maintaining imaging capability.

Single-pixel imaging (SPI) has emerged as a computational imaging technique that overcomes many of these limitations. Instead of using a detector array, SPI employs a spatial light modulator to generate a sequence of known modulation patterns, while a single-pixel detector measures the corresponding reflected or transmitted light intensity. Although each measurement contains no spatial information individually, the complete set of measurements together with the known illumination patterns allows the original image to be computationally reconstructed. Compared with conventional imaging systems, SPI offers several important advantages. Because only a single detector is required, the imaging hardware is significantly simpler and less expensive, particularly for wavelength bands where detector arrays remain costly or unavailable, such as hyperspectral imaging [1, 2]. In addition, because all the collected light is integrated by a single detector, SPI generally achieves a higher SNR, making it particularly suitable for low-light imaging [3, 4]. Furthermore, the same imaging principle can be applied across a wide range of wavelengths, from the visible spectrum to infrared [5, 6], terahertz [7, 8, 9, 10, 11], and ultraviolet regions [12, 13], where detector arrays are often expensive or difficult to manufacture. Moreover, advanced computational reconstruction algorithms enable high-quality image reconstruction from fewer measurements, making SPI an attractive solution for compressive imaging applications. Owing to these advantages, SPI has attracted increasing attention in biomedical imaging [14, 15, 16, 17, 18], remote sensing [19, 20], and other computational imaging applications [21, 22, 23, 24, 25]. Although SPI significantly reduces hardware complexity, it typically requires a large number of illumination patterns to achieve high-quality image reconstruction, resulting in long acquisition times and limiting its applicability in real-time imaging.

Previous studies have proposed a variety of approaches to reduce the number of measurements required by SPI [27, 46, 45]. Representative methods include compressed sensing [49], Fourier-domain sampling[45], and adaptive sampling strategies [46]. More recently, advances in deep learning have provided new opportunities to improve SPI reconstruction. Data-driven, model-driven, and hybrid methods have demonstrated improved reconstruction quality and faster reconstruction compared with conventional optimization-based methods. In particular, Deep Image Prior (DIP) is attractive because it exploits the implicit image prior induced by the architecture of an untrained neural network, enabling high-quality reconstruction without requiring any pre-training. These methods improve reconstruction efficiency from different perspectives, but achieving high image quality under extremely low sampling ratios remains a challenging problem.

A digital micromirror device (DMD) is a spatial light modulator consisting of an array of microscopic mirrors. Each mirror can rapidly switch between two angular positions, corresponding to the ON and OFF states, thereby modulating the reflected light into binary patterns. Motivated by this binary modulation mechanism, in this project, we investigate Binary Fourier Single-Pixel Imaging (BFSI), which employs binary Fourier illumination patterns that are directly compatible with DMDs. Unlike grayscale Fourier patterns, binary patterns can be displayed at the maximum switching speed of a DMD without requiring temporal dithering or multiple exposures, thereby significantly reducing the image acquisition time. Owing to this high-speed acquisition capability, BFSI is particularly suitable for real-time imaging applications. However, reducing the number of projected patterns inevitably decreases the amount of acquired information, leading

to degraded reconstruction quality at low sampling ratios. To address this challenge, we integrate BFSI with model-driven neural networks, to recover high-quality images from fewer measurements. By combining the fast acquisition capability of BFSI with the reconstruction ability of neural networks, the proposed framework aims to achieve both low acquisition time and high reconstruction quality under limited measurements.

2 Prior Work

2.1 Single-Pixel Imaging

SPI can be broadly classified into two approaches: structured illumination and structured detection. In structured illumination, the illumination patterns are projected onto the object before detection, whereas in structured detection, the reflected or transmitted light is spatially modulated before reaching the detector.

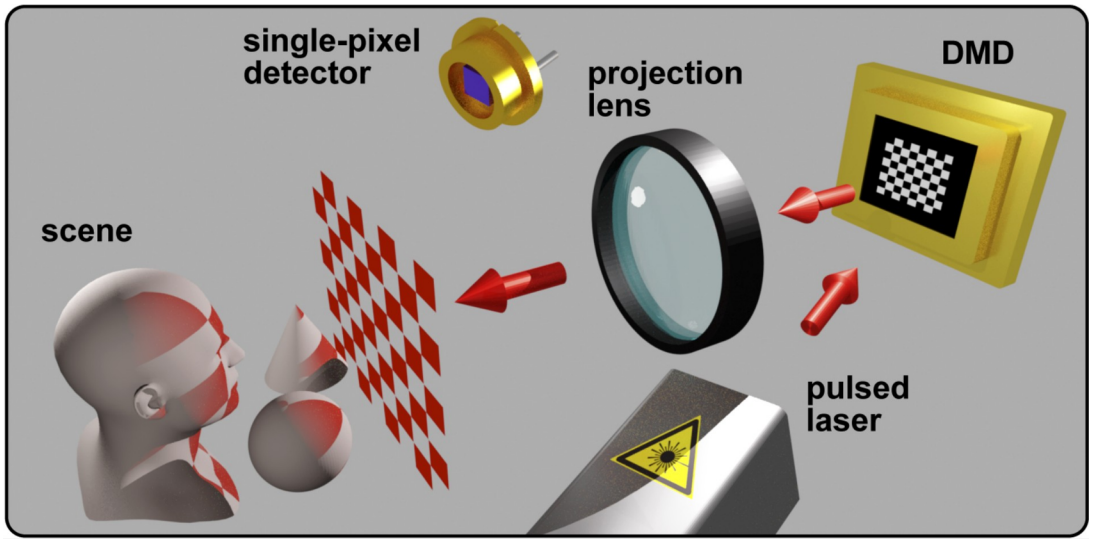


Figure 1: Structured illumination [26]. The DMD is placed before the scene so the patterns are projected onto the object before detection.

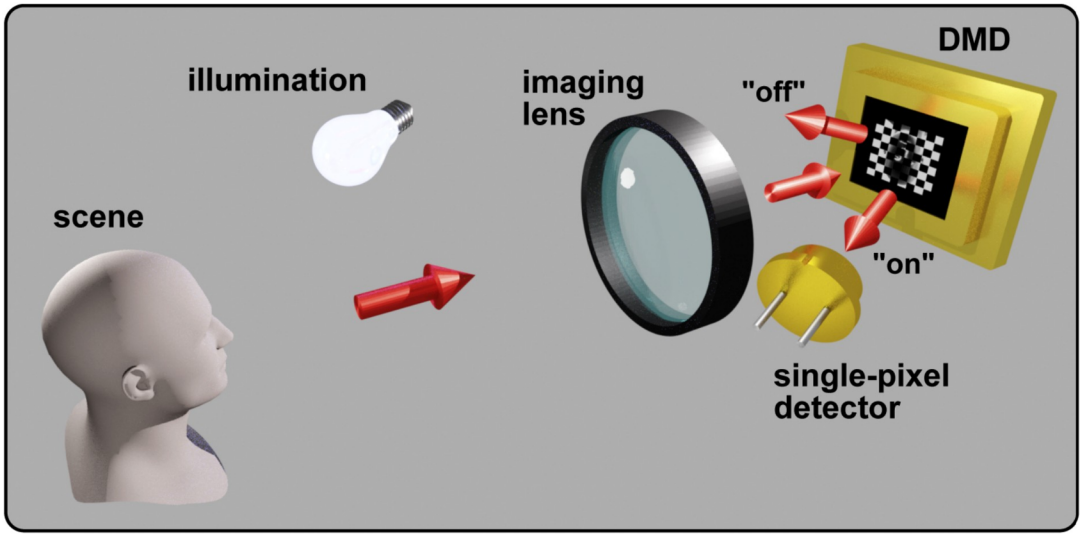


Figure 2: Structured detection [26]. The DMD is placed after the scene so the reflected light is spatially modulated before reaching the detector.

As in the DMD each micro mirror switches between two tilt states, corresponding to the ON and OFF reflection directions, the patterns used in SPI must be binary $(0, 1)$ patterns.

In the continuous domain, the light intensity y_i detected by the single pixel detector can be mathematically formulated as:

$$y_i = \int\int f(x, y)P_i(x, y)dxdy + n_i$$

where $f(x, y)$ denotes the luminance distribution of the target object, $P_i(x, y)$ is the i-th illumination pattern generated by a DMD, and n_i denotes additive measurement noise. For computational implementation, the continuous model is discretized by representing the object as a vector $\mathbf{x} \in \mathbb{R}^{n\times 1}$, the patterns as a matrix called the sensing matrix $A \in \mathbb{R}^{m\times n}$, the measurements $\mathbf{y} \in \mathbb{R}^m$, and the additive noise $\mathbf{n}$ is a random vector generated from Gaussian noise. The mathematical formulation is given by

$$\mathbf{y} = A\mathbf{x} + \mathbf{n}$$

To improve the quality or speed of SPI's reconstruction, extensive work has been done to examine and improve various aspects of single pixel imaging such as the pattern selection, reconstruction methods, or applying a neural network after the reconstruction of SPI.

2.2 Single-Pixel Imaging Methods

2.2.1 Single-Pixel Imaging Patterns

Often in image processing a natural image is transformed from the physical basis into another basis to extract more important information from the image. The Fourier transform is very useful in natural images since they encode the frequency of an image which can capture smooth edges in images especially at low frequencies. Similarly, the Hadamard transform can capture the shape of an image through careful basis selection [27]. In general, clever choice of illumination patterns allows for high quality reconstructions from few SPI measurements.

One such example of patterns is Hadamard Single-Pixel Imaging (HSI). For an image $x \in \mathbb{R}^{n\times n}$, HSI [28] generates measurements as follows :

$$\begin{aligned}\tilde{I}(u, v) :&= \sum_{x=1}^{n}\sum_{y=1}^{n} I(x, y)P(x, y; u, v)\\ &= \sum_{x=1}^{n}\sum_{y=1}^{n} I(x, y)H^{-1}(\delta(u, v))\\ &= \sum_{x=1}^{n}\sum_{y=1}^{n} I(x, y)\Big[P_+(x, y; u, v) - P_-(x, y; u, v)\Big]\end{aligned}$$

where (u, v) correspond to the measurement's location in the Hadamard domain, H^{-1} is the inverse Hadamard transform, and P_+ and P_- are the positive and negative Hadamard patterns

$$P_+(x, y; u, v) = [1 + H^{-1}(\delta(u, v))]/2 \qquad P_-(x, y; u, v) = [1 - H^{-1}(\delta(u, v))]/2.$$

Then to reconstruct $I(x, y)$, we calculate $H^{-1}(\tilde{I}(u, v)) \approx I(u, v)$

In a similar vein to HSI, Fourier single-pixel imaging transforms the image using the discrete Fourier

transform (DFT). The process is similar but works for images of any size $x \in \mathbb{R}^{n\times m}$:

$$\begin{aligned}\tilde{I}(u,v) :&= \sum_{x=1}^{n}\sum_{y=1}^{m} I(x,y)P(x,y;u,v)\\ &= \sum_{x=1}^{n}\sum_{y=1}^{m} I(x,y)\exp[-2\pi i(ux+vy)]\\ &= \frac{1}{1.5}\left[(2D_1 - D_2 - D_3) + \sqrt{3}i(D_2 - D_3)\right]\end{aligned}$$

where

$$D_i = \sum_{x=1}^{n}\sum_{y=1}^{n} I(x,y)\left[0.5 + 0.5\cos(2\pi(ux+vy)+\varphi_i)\right]$$

for $\varphi_i = 0, \frac{2\pi}{3}, \frac{4\pi}{3}$ and (u,v) are coordinates in the frequency domain. We then recover $I(x,y)$ by applying the inverse fast Fourier transform $(IFFT)$ to $\tilde{I}(u,v)$.

The cosine patterns are grayscale so they are not compatible with the DMD hardware. To mend this incompatibility and generate DMD compatible patterns, binary Fourier single-pixel imaging (BFSI) [29] takes the grayscale Fourier patterns and dithers the pattern to generate binary patterns. This dithering will produce quantization errors that are seen in the reconstruction. To minimize this quantization error, the BFSI method takes a Fourier pattern at scaled down resolution of the original image then upscales the pattern to the original resolution then applies Floyd-Steinberg dithering [30]. For example with a 256×256 pixel image, we obtain the grayscale FSI patterns corresponding to a 128×128 image, then upscale the pattern to a 256×256 pixel pattern through bicubic interpolation. Then we apply Floyd-Steinberg dithering, an iterative error diffusing dithering process that binarizes one pixel and diffuses the quantization error to neighboring pixels to obtain the binary pattern (see Figure 3).



Figure 3: FSI Pattern Dithering

After dithering, the projected illumination patterns become binary and are therefore compatible with the DMD hardware. Let $B_i(x,y;u,v)$ denote the binary dithered approximation of the the grayscale Fourier pattern with phase shift φ_i. The corresponding single-pixel measurements are

$$\hat{D}_i = \sum_{x=1}^{n}\sum_{y=1}^{m} I(x,y)B_i(x,y;u,v)$$

Since Floyd–Steinberg dithering preserves the local average intensity of the original grayscale pattern, $\hat{D}_i$ provides an approximation of the grayscale measurement D_i, Consequently, the complex Fourier coefficient is estimated using the same three-step phase-shifting formula as in FSI,

$$\tilde{I}(u,v) \approx \frac{1}{1.5}\left[(2\hat{D}_1 - \hat{D}_2 - \hat{D}_3) + \sqrt{3}i(\hat{D}_2 - \hat{D}_3)\right]$$

Repeating this process for all sampled frequencies reconstructs an approximation of the Fourier spectrum, after which the inverse fast Fourier transform (IFFT) is applied to obtain the final image. One caveat to this method is that the reconstructed image will be at the original FSI pattern's resolution i.e 128×128 so the magnitudes of the Fourier spectrum will be scaled by the downsampling scale. Although we lose spatial resolution, the original paper states that the reconstruction quality of the image is improved as the downsampling scale is increased.

Applying BFSI or FSI patterns using the DMD hardware, we can use the patterns corresponding to low level frequencies in the Fourier domain, which allows us to generate an approximation of the image using fewer measurements than using random binary patterns. In HSI we use sequency, which is the number of sign changes in the binary pattern, as a replacement to the frequency in FSI. Then we can use zigzag or cake-cutting methods to sample high sequency patterns to achieve similar reconstruction quality to the low frequency sampled FSI or BFSI.

2.2.2 Domain Sampling Methods

Since we are interested in sampling only a subset of the Fourier domain, an appropriate sampling strategy is required to determine which frequency coefficients should be acquired. Natural images exhibit strong spatial correlations, with most of their energy concentrated in the low-frequency region of the Fourier spectrum, while high-frequency components mainly represent fine details and edges. Consequently, Fourier-domain sampling strategies generally allocate more measurements to low-frequency coefficients, which contribute most significantly to image reconstruction, while using fewer measurements in high-frequency regions to improve sampling efficiency. Motivated by this property, a variety of Fourier-domain sampling strategies have been proposed.

A common way to sample the Fourier domain is a circular strategy, where we fix a radius based on our sampling rate defined by $SR = P/(n \times m)$, for an image $x \in \mathbb{R}^{n \times m}$ and P is the number of frequencies we sample (see Fig). 5. We then choose all frequencies which fall within the given radius.

Next, the variable density sampling method is a sampling method developed by Wenwen et al. [45]. In this method, each frequency in the Fourier domain has a different sampling probability given by

$$\rho = \begin{cases} 1, & r < R \\ (1-r)^{\epsilon}, & r > R \end{cases}.$$

Here r is the radius from the center of the Fourier spectrum, R is a predetermined radius threshold, and ϵ adjusts the probability of sampling high frequencies (see Fig. 5). This method balances keeping low frequency signals which are important to the overall shape of the image, while keeping some high frequency signals which are important for the fine details.

The final sampling strategy considered in this work is Adaptive Spectrum Sampling (ASS), which adaptively determines subsequent sampling locations based solely on Fourier coefficients that have already been measured and reconstructed [46]. Following the standard BFSI acquisition procedure, each binary Fourier illumination pattern is projected using phase-shifting measurements, from which the corresponding complex Fourier coefficient is recovered immediately after acquisition. The magnitude of the recovered coefficient is then used to update the accumulated energy of its associated radial sampling path and determine whether further sampling along that path should continue. Consequently, ASS requires no prior knowledge of unsampled Fourier coefficients and can be implemented entirely using information acquired during the measurement process. Unlike fixed sampling patterns, this enables the sampling distribution to adapt dynamically to the spectral characteristics of each individual image while remaining practical for real-world BFSI systems.

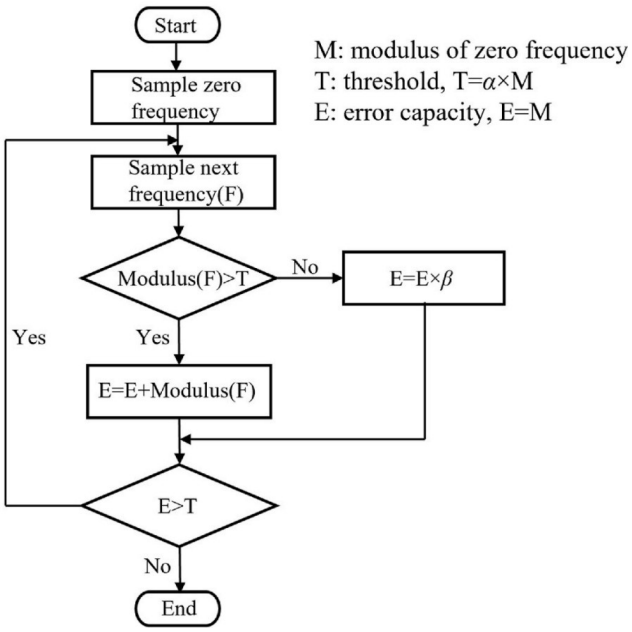


Figure 4: Sampling flow of ASS in the radial direction [46]

As shown in Fig. 4, ASS employs an adaptive thresholding criterion to identify high-magnitude Fourier coefficients and dynamically adjust the sampling distribution. Although this strategy improves sampling efficiency, the final sampling ratio is data-dependent and cannot be determined in advance. Instead, it is only known after the sampling process has been completed. Achieving a desired sampling ratio while preserving the adaptive nature of ASS therefore remains an open challenge. To address this issue, Section 3 presents an iterative threshold relaxation strategy that provides better control over the sampling ratio while maintaining the advantages of adaptive spectrum sampling.

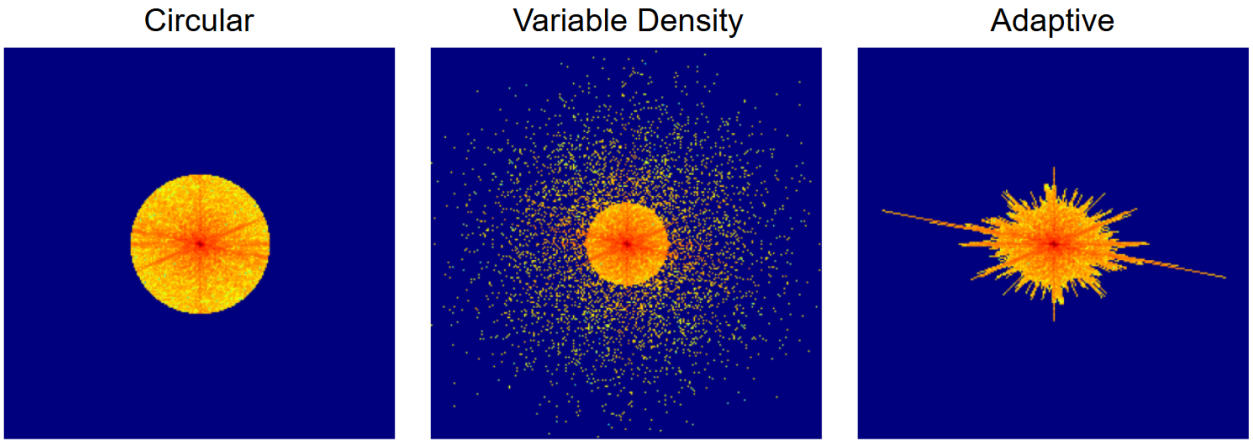


Figure 5: Sampled Fourier Spectrum at 10% Sampling Rate

2.3 Deep Learning Approaches for Single-Pixel Imaging

In single-pixel imaging, obtaining high quality images can be challenging when only a limited number of measurements are available. When the measurements are incomplete or undersampled, the resulting image may contain missing information and noise, making reconstruction more difficult. Therefore, effective methods are needed to recover the missing information and reduce the noise while preserving the important structures of the image.

Recently, deep learning based approaches have attracted significant attention due to their ability to achieve high-quality image reconstruction with faster processing time [31] compared with conventional iterative reconstruction methods. The inverse problem is commonly addressed by learning the mapping between the measurements and the target image using a deep neural network (DNN). Based on how the model and training data are incorporated into the learning process, existing approaches for deep learning-based SPI can generally be grouped into two categories: data-driven and model-driven strategy.

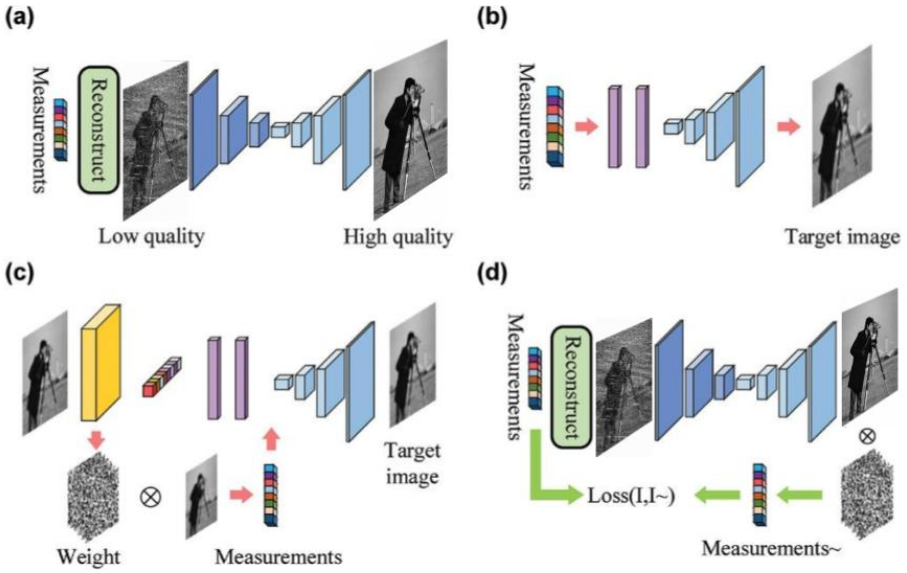


Figure 6: Deep learning strategies for single-pixel imaging. Data-driven approaches include (a) image-to-image mapping, (b) measurement-to-image mapping, and (c) pattern optimization, while (d) represents the model-driven approach. [31]

2.3.1 Data-Driven Strategy

Data-driven methods are one of the main strategies used to apply deep learning to SPI. These methods learn the relationship between the input data and the desired high-quality image from a training dataset.

As illustrated in Figure 6, data-driven approaches can follow different configurations. The network can learn a direct mapping from a low-quality image to a high-quality image (Figure 6(a)), map single-pixel measurements directly to a target image (Figure 6(b)), or jointly learn optimized measurement patterns and the corresponding image recovery process (Figure 6(c)).

In the simplest case, a DNN learns a mapping function from the input measurements or a low-quality image to the target image. During training the network parameters are optimized by minimizing the difference between the predicted image and the corresponding ground-truth image. Once training is completed, the network can recover an image from new measurements through a single forward pass. This avoids the repeated optimization steps required by many traditional iterative methods and can therefore provide much faster image recovery [32].

However, the performance of data-driven deep learning methods depends strongly on the similarity between the training data and the target scenes. Tian et al. [33] highlighted that supervised deep learning methods for single-pixel imaging generally rely on a strong similarity between training and testing data, and may exhibit poor performance when the target objects differ significantly from those seen during training. This dependence on the training distribution presents a major challenge for generalizing to new imaging conditions and previously unseen target scenes.

2.3.2 Model-Driven Strategy

The limitations of data-driven methods have motivated the development of model-driven strategies, which incorporate the physical model of the imaging system directly into the learning process. As illustrated in

Figure 6(d), the reconstruction network is connected to the SPI forward model, allowing the recovered image to be evaluated directly against the acquired measurements. Instead of learning the complete measurement-to-image mapping only from a large training dataset, the network is optimized while respecting the physical relationship between the image and the measurement [40, 42].

In this framework, the recovered image can be represented by a neural network as :

$$x = f_\theta(z)$$

where z is a fixed input and f_θ is a neural network parameterized by θ. The network parameters are optimized by passing the estimated image through the known SPI forward model and minimizing a task-specific loss E as follow :

$$\theta^* = \arg\min_\theta E(\Phi f_\theta(z); y),$$

if the L_2 norm distance is used to measure the discrepancy between the predicted and observed data, E will be define as :

$$E(\Phi f_\theta(z); y) = \|\Phi f_\theta(z) - y\|_2^2,$$

where Φ represents the SPI measurement process and y the acquired single-pixel measurement. This approach allows the physical imaging process to directly guide the optimization, reducing the dependence on external training datasets.

A particularly important approach in this context is the Deep Image Prior (DIP) framework introduced by Ulyanov et al. [43]. As illustrated in Figure 7, the method starts with a randomly initialized neural network and a fixed input z. The network parameters are then iteratively optimized to minimize a task-specific loss between the generated image and the available degraded observation. At each iteration, the generated image is evaluated, and the gradient of the loss is used to update the network parameters. In this way, the network architecture itself acts as an implicit prior for the image restoration task.

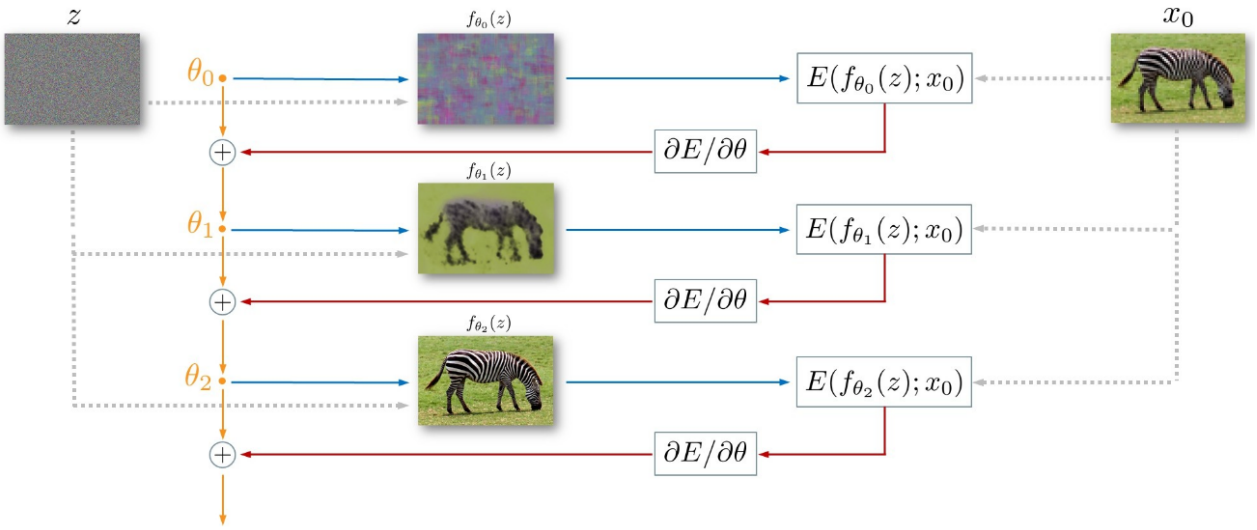


Figure 7: Image restoration using deep image prior [43].

In their work, Ulyanov et al. demonstrated that an untrained neural network can effectively address several inverse imaging problems, including image denoising, super-resolution, and inpainting, without using an external training dataset. Their results showed that the network architecture itself provides a useful prior, allowing meaningful image structures to be recovered from corrupted observations. In particular, the network tends to learn the main structures of an image before fitting the noise. Therefore, the optimization process is stopped at an appropriate point to obtain a denoised image before the network begins to reproduce the noise present in the measurements.

2.4 Improvements of SPI Reconstruction

To improve the reconstruction of an image after SPI, researchers have utilized the properties of natural images to post process the output of SPI. The first method applied is based off of compressed sensing (CS) [55, 54]. The original measurements of SPI without noise are

$$\mathbf{y} = A\mathbf{x}$$

where $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{x} \in \mathbb{R}^N$, and $A \in \mathbb{R}^{M \times N}$. In CS methods, we assume the image vector x is a sparse signal since we wish to reconstruct natural images. $\mathbf{x}$ being a sparse signal means $\mathbf{x} = \Psi\mathbf{s}$, where $\Psi \in \mathbb{R}^{N \times N}$ is a known sparse dictionary and $\mathbf{s} \in \mathbb{R}^N$ is a sparse vector i.e $\mathbf{s}$ contains many zeros. So we can reformulate our problem as

$$\mathbf{y} = A\mathbf{x} = A\Psi\mathbf{s} = \Phi\mathbf{s}$$

and to reconstruct our image we solve

$$\min_{\mathbf{s} \in \mathbb{R}^N} \|\mathbf{s}\|_{l_0} \text{ subject to } \Phi\mathbf{s} = \mathbf{y}$$

To further improve the post processing of the reconstruction, DNNs were utilized instead of the CS methods [56, 34]. Since these methods utilize DNNs, they sacrifice some comprehensibility to gain reconstruction improvements since DNNs are a black box method. In particular, we do not know if the DNN's outputs are physically consistent or if it is biased to any previous data. To address the latter issue [56, 34], utilizes untrained neural networks to stay unbiased to training data. in particular, [34] uses Ulyanov's DIP method, which is not physics informed, however showed increases in both computation speed and accuracy compared to CS reconstruction methods for subsampled FSI outputs.

3 Methods

3.1 Overall Framework

Our method aims to construct high-quality images from limited BFSI measurements using a model-driven neural network. We use BFSI due to it's compatibility with DMD hardware. Instead of tuning networks to images like other methods, we hope to configure different neural networks than can work on many different images. The general framework is summarized in Figure 8. It involves reconstructing an image using BFSI with limited patterns and optional noise to produce an image using a small number of patterns. Then, that image (which will be referred to as the neural network input) is sent into a neural network to denoise it and return a high quality image.



Figure 8: Our Pipeline

To achieve our goal, we first code our pipeline in Python [35]. We created one function for the BFSI reconstruction which returns the reconstructed image, reconstructed image with noise, patterns of the ground truth with noise (if any), measurements of the ground truth with noise if any, and information to help rebuild the spectrum. The second function is for denoising the output of BFSI with the neural network which outputs the final denoised image.

3.2 Sampling Strategies

In addition to circular and variable-density sampling, we propose an Iterative Threshold Relaxation Adaptive Sampling Strategy (ITR-ASS) based on the Adaptive Sampling Strategy(ASS) algorithm introduced in Section 2. A limitation of the original ASS algorithm is that, under certain parameter settings, the sampling process may terminate prematurely before sufficient spectral information has been collected. As a result, many low-magnitude but still informative Fourier coefficients remain unsampled, which may degrade the reconstruction quality, particularly at low sampling ratios. To address this issue, To address this limitation while preserving the adaptive nature of ASS, the proposed ITR-ASS introduces two main improvements:

- Round-robin adaptive spectrum sampling
- Iterative threshold relaxation

3.2.1 Round-Robin Adaptive Spectrum Sampling

In the original ASS algorithm, each radial sampling path is processed independently. Sampling continues along one path until no additional Fourier coefficients satisfy the adaptive threshold, after which the algorithm proceeds to the next path. Under low sampling ratios, this strategy may spend a large proportion of the sampling budget on only a few radial directions, while leaving many other directions unexplored. Consequently, the sampled Fourier spectrum becomes unevenly distributed, reducing spectral coverage.

To improve the spatial coverage of the Fourier spectrum, we adopt a round-robin sampling strategy. Instead of completely processing one radial path before moving to the next, only one Fourier coefficient is sampled from each path during one cycle. The algorithm then immediately switches to the next radial path and repeats this process until all paths have been visited. For each radial path, the accumulated spectral energy E is updated using only the Fourier coefficients sampled on that path. The updated path energy is then used to determine whether subsequent coefficients on the same path should continue to be sampled according to the adaptive threshold criterion.

3.2.2 Iterative Threshold Relaxation

In ASS, the adaptive sampling process terminates once no unsampled Fourier coefficient satisfies the current threshold. Consequently, the final sampling ratio depends on the image content and the selected parameters, and cannot be specified in advance.

To provide better control over the acquisition process, an iterative threshold relaxation strategy is introduced. After one round of adaptive sampling terminates, the current sampling ratio is compared with the user-defined target sampling ratio. If the desired sampling ratio has not yet been reached, the adaptive threshold is relaxed according to

$$T_{k+1} = \gamma T_k, \qquad 0 < \gamma < 1,$$

where T_k is the threshold used in the k-th iteration and γ is the threshold relaxation factor.

Adaptive sampling then resumes using the relaxed threshold. Unlike reinitializing the adaptive sampling process, all previously sampled Fourier coefficients and the accumulated path energies are retained. After the threshold has been updated, the radial sampling paths are reordered according to their accumulated energy. Paths with larger accumulated energy are assigned higher sampling priority, allowing informative regions of the Fourier spectrum to be explored earlier while avoiding redundant sampling of low-energy paths. This process is repeated until the target sampling ratio is achieved.

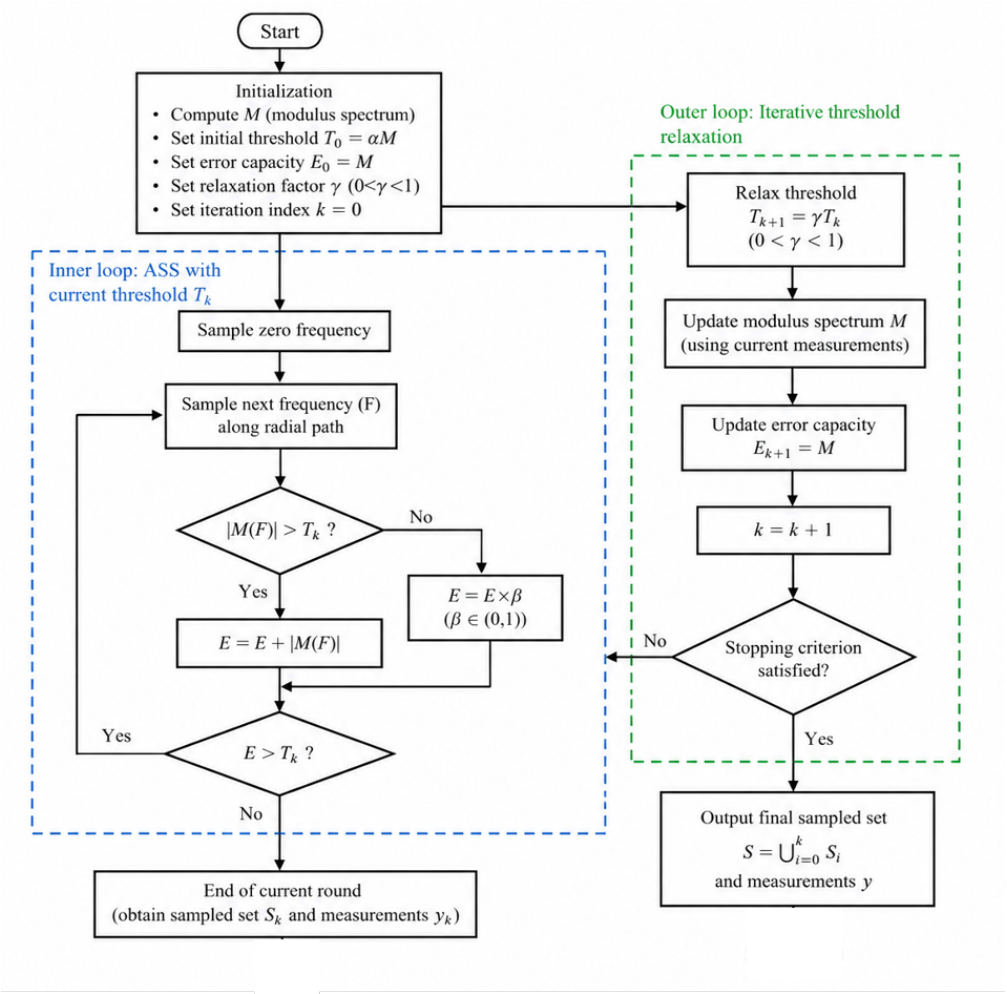


Figure 9: Sampling flow of iterative threshold relaxation ASS

3.3 Neural Networks Architecture

The neural network architecture considered in this work follow the deep image prior with different image parameterization approaches. Specifically, the experimental study considers two convolution-based neural networks, U-Net and ResNet, alongside a coordinate-based neural networks, the sinusoidal representation network (SIREN). These architectures were selected to investigate the applicability of different network designs within the DIP framework, including architectures that extend beyond those considered in the original DIP formulation. The selection was also guided by preliminary experimental observations and the computational and time constraints of the study.

3.3.1 U-Net

The U-Net architecture is a convolutional neural network based on an encoder-decoder structure, originally introduced for biomedical image segmentation [50]. In this work, the network receives an input tensor z and produces an output image $\hat{x}$ according to $\hat{x} = f_\theta(z)$, where f_θ denotes the U-Net parametrized by θ.

The encoder starts with an initial convolutional block, denoted by C_0, followed by four downsampling blocks. Each convolutional block contains two successive 3×3 convolutions, with instance normalization and ReLU activation applied after each convolution. The downsampling blocks use 2×2 max-pooling before applying the convolutional block. The feature maps can therefore be represented as

$$\mathbf{h}_0 = C_0(z),$$

and

$$C_i\left(\text{MaxPool}_{2\times 2}\left(\mathbf{h}_{i-1}\right)\right), \qquad i = 1, \ldots, 4.$$

This produces a hierarchical representation in which the spatial resolution decreases progressively while the number of feature channels increases.

The decoder consists of four corresponding upsampling blocks. At each level, the feature maps are first upsampled and then concatenated with the feature maps produced by the corresponding encoder level. If $\mathbf{u}_{i+1}$ denotes the feature map from the deeper decoder level and $\mathbf{h}_i$ the corresponding encoder feature map, the decoding operation can be expressed as

$$D_i\left(\text{Concat}\left[U_i\left(\mathbf{u}_{i+1}\right), \mathbf{h}_i\right]\right),$$

where U_i represents the upsampling operation and D_i denotes the subsequent convolutional block. These skip connections allow the decoder to directly access feature maps from the encoder at the same spatial scale.

In the implemented network, the upsampling operation can be selected between transposed convolution, bilinear upsampling, and nearest neighbor upsampling followed by convolution. The network also allows additional downsampling and upsampling levels to be introduced through the *more_layers* parameter. In the experiments, the selected configuration is defined by the corresponding tuning parameters discussed in Section 3.5.

The final decoder output is passed through a 1×1 convolution to map the final feature representation to the required number of output channels:

$$W_{\text{out}} * \mathbf{u}_0 + \mathbf{b}_{\text{out}},$$

where W_{out} represents the 1×1 convolutional kernel. A sigmoid activation is subsequently applied to obtain the final output:

$$\hat{x} = \sigma(\mathbf{y}),$$

constraining the output values to the interval $[0, 1]$. The resulting architecture is illustrated in Figure 10.

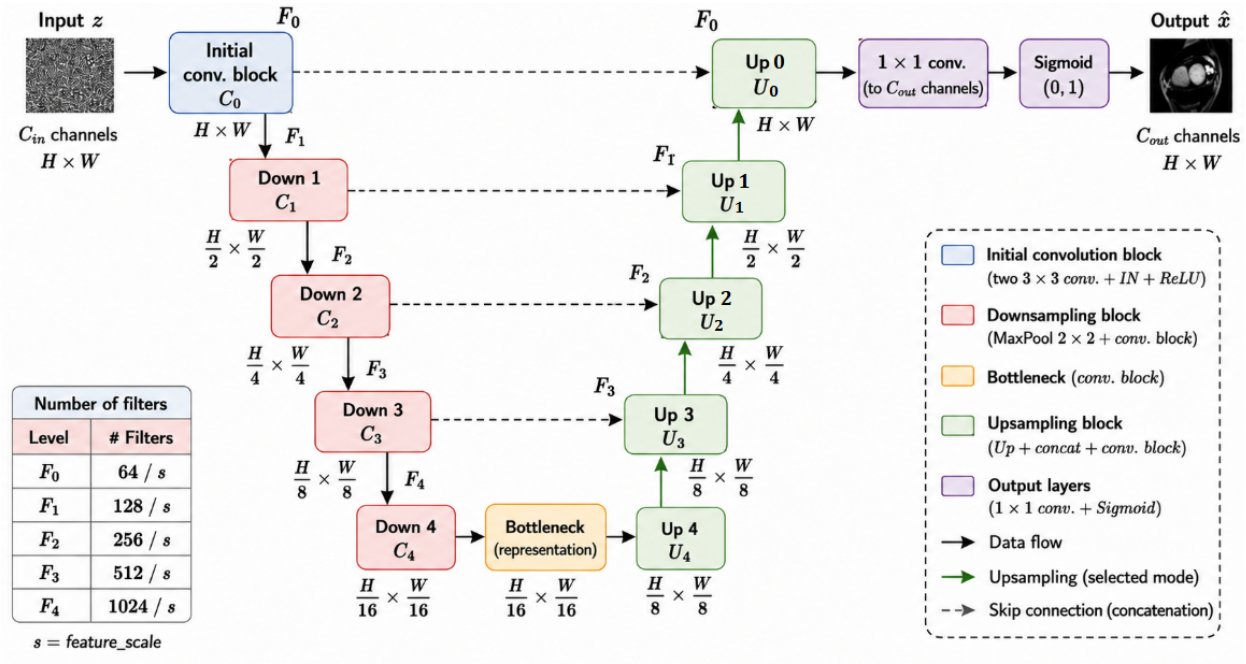


Figure 10: Schematic representation of the implemented U-Net architecture

The *feature_scale* parameter controls the width of the U-Net by uniformly scaling the number of feature channels in all encoder and decoder layers. A larger *feature_scale* results in fewer channels, thereby reducing the number of trainable parameters, memory usage, and computational cost.

3.3.2 ResNet

The ResNet architecture is a deep convolutional neural network based on learning residual functions originally introduced for image recognition [51]. In this work, they introduce skip layers to address the vanishing gradient problem and allows ResNet to have a much larger architecture than standard neural networks. The neural network takes as input a tensor z corresponding to an input images and outputs the classification of the image.

Formally, one block of ResNet is given by

$$\mathbf{y} = F(\mathbf{x}, \{W_i\}) + \mathbf{x}.$$

Then many blocks of ResNet are then stacked to form the entire architecture. In conventional neural networks, the network learns the the mapping $H(x)$, however, ResNet wishes to learn $F(\mathbf{x}) + \mathbf{x} := H(x)$ as the paper claims it is easier to learn the residuals rather than the unknown $H(x)$. They also introduce skip networks which allow the network to skip blocks as needed, see 11.

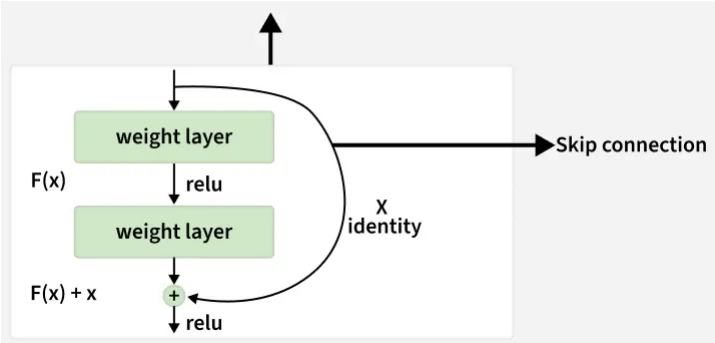


Figure 11: Skip Connections

Since the original ResNet was developed for image recognition, there have been developments in using ResNet for denoising [52, 53]. From these successful applications of ResNet in image classification and denoising, we believe it can be used in the DIP framework for a model-driven denoising network.

3.3.3 SIREN

SIREN is an implicit neural representation introduced by Sitzmann et al.[48], that models a continuous signal as a function of its spatial coordinates. For a two-dimensional image, the network receives a coordinate

$$\mathbf{p} = (p_x, p_y)$$

and predicts the corresponding image intensity,

$$\hat{x}(\mathbf{p}) = f_\theta(\mathbf{p})$$

where f_θ denotes the neural network parameterized by θ. In contrast to conventional neural networks that operate directly on image arrays, the image is therefore represented implicitly by the parameters of the coordinate network.

The main characteristic of SIREN is the use of periodic sine functions as activation functions. A SIREN layer can be expressed as

$$\sin\left(\omega_l \left(W_l \mathbf{h}_{l-1} + \mathbf{b}_l\right)\right),$$

where W_l and $\mathbf{b}_l$ denote the weights and biases of layer l, respectively, and ω_l is a frequency-scaling factor. The input to the first layer is the spatial coordinate $\mathbf{p}$, such that

$$\sin\left(\omega_0\left(W_0\mathbf{p}+\mathbf{b}_0\right)\right).$$

The periodic activation enables the network to represent spatial variations through sinusoidal components and provides a continuous and differentiable image representation.

In the implementation used in this work, the SIREN architecture consists of a coordinate input, a sequence of sinusoidal layers, and a linear output head. The input consists of two normalized spatial coordinates corresponding to the horizontal and vertical positions of each image point. The first layer maps these coordinates to a hidden representation using a sine activation with a dedicated frequency-scaling parameter. This is followed by a series of hidden sine layers, where each layer applies a linear transformation followed by a periodic activation. The resulting feature representation is then mapped to a single output channel through a linear layer. Finally, a sigmoid function is applied to constrain the reconstructed intensity values to the interval $[0, 1]$. The resulting reconstruction can therefore be written as

$$\sigma\left(W_{\text{out}}\mathbf{h}_L+\mathbf{b}_{\text{out}}\right),$$

where $\mathbf{h}_L$ denotes the output of the final sinusoidal layer, Wout and $\mathbf{b}_{\text{out}}$ are the parameters of the output layer, and $\sigma(\cdot)$ represents the sigmoid function.

The parameters defining the SIREN architecture include the hidden dimension, which determines the width of the sinusoidal layers; the number of hidden layers, which controls the depth of the network; the first-layer frequency parameter, which determines the frequency scaling applied to the first sine layer; the hidden-layer frequency parameter, which controls the frequency scaling of subsequent layers; and the output dimension, which corresponds to the number of reconstructed image channels. The network also requires a normalized coordinate domain defining the spatial locations supplied to the model.

The weights of the SIREN layers are initialized according to the initialization strategy proposed by Sitzmann et al. [48]. For a layer with n input features, the first layer is initialized according to

$$W_0 \sim \mathcal{U}\left(-\frac{1}{n}, \frac{1}{n}\right),$$

while the weights of subsequent sinusoidal layers are sampled from

$$W_l \sim \mathcal{U}\left(-\frac{\sqrt{6/n}}{\omega_l}, \frac{\sqrt{6/n}}{\omega_l}\right).$$

Biases are initialized to zero. This initialization is used to maintain suitable activation statistics throughout the network and to facilitate stable optimization of the periodic representation.

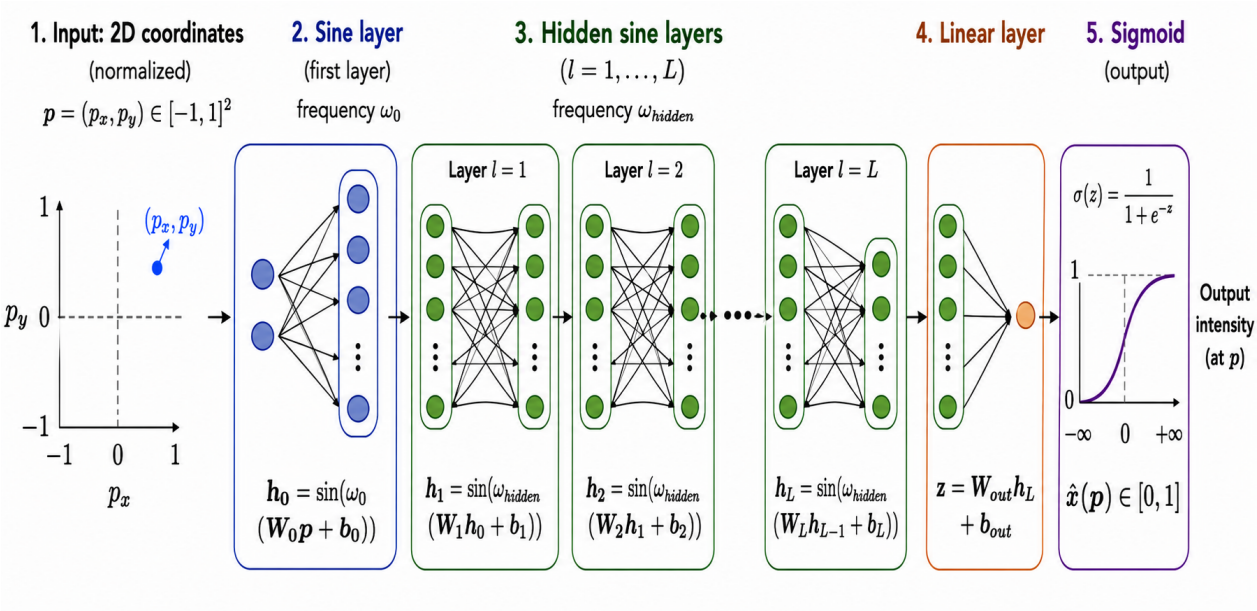


Figure 12: Schematic illustration of SIREN architecture

An overview of the SIREN architecture used in this work is presented in Figure 12. The network receives normalized two-dimensional coordinates and progressively transforms them through sinusoidal layers before producing the reconstructed image intensity at each coordinate. Evaluating the network over the complete coordinate grid yields the reconstructed image.

3.3.4 Training Loss

All neural networks are trained using a hybrid loss function that combines measurement-domain, Fourier-domain, and image-domain losses. The measurement consistency is enforced using the normalized squared error

$$\mathcal{L}_{\text{meas}} = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2}{\|\mathbf{y}\|_2^2},$$

while the Fourier-domain loss is defined as

$$\mathcal{L}_{\text{Fourier}} = \frac{\|\mathbf{S} - \hat{\mathbf{S}}\|_2^2}{\|\mathbf{S}\|_2^2 + \epsilon}.$$

An image-domain mean squared error is also included :

$$\mathcal{L}_{\text{Image}} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2,$$

where $\mathbf{x}$ is the BFSI reconstructed image and $\hat{\mathbf{x}}$ the network output. The overall objective is therefore

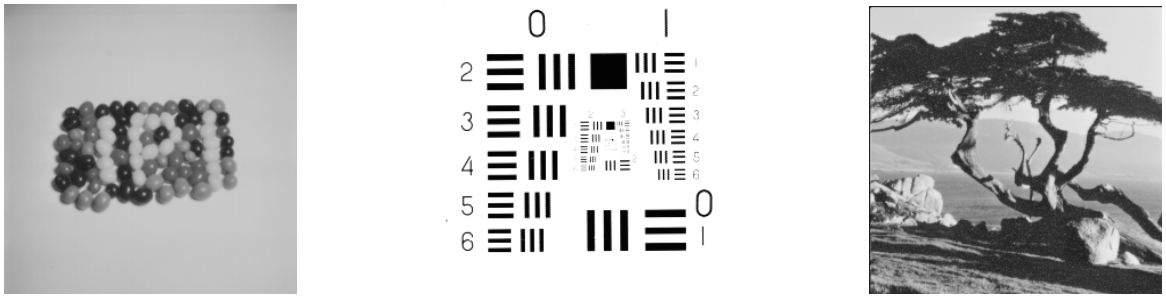
$$\mathcal{L}_{\text{objective}} = \mathcal{L}_{\text{meas}} + \mathcal{L}_{\text{Fourier}} + \lambda \mathcal{L}_{\text{image}},$$

The Fourier-domain loss is used directly as the primary data-fidelity term and therefore does not require an additional weighting coefficient. The regularization parameter λ was fixed at $\lambda = 0.001$ for all experiments.

3.4 Data

The images that we tested our framework on comes from the Kanagawa Institute of Technology Signal Processing Application Laboratory's Research Image Database[36] and the scikit-image Python package[37]. These were chosen because they are standard images for image processing research and have been used in

related papers. The images are shown in Figure 13. Images 13e and 13f were exclusively used for comparison with the FSI + DIP method as those images were also used in their experiments.



(a) Jelly beans (b) Resolution chart (c) Tree

(d) Cameraman (e) Fishing boat (f) Man

Figure 13: The six test images used to evaluate the proposed framework and tune the reconstruction parameters.

The primary metrics we calculated to measure quality are peak-signal-to-noise ratio (PSNR), structural similarity index (SSIM), and root mean squared error (RMSE). PSNR is a measure of the quality of an image. Specifically, it is the ratio of the maximum pixel value of an image (ground truth) and that of the corrupting noise (image of interest) with larger numbers being better [38]. It is defined as follows:

$$PSNR = 10\log_{10}\left(\frac{(L-1)^2}{MSE}\right) = 20\log_{10}\left(\frac{L-1}{RMSE}\right)$$

where L represents the maximum intensity levels in an image.

SSIM and RMSE both measure the similarity between two images, but SSIM takes into account the structure of an image like edges. SSIM specifically measures the luminance, contrast, and structural similarity over local neighborhoods then averages those values [39]. The possible values are between -1 and 1. Venkataramanan et al. [39] defines it as:

$$Q(i,j) = l(i,j) \cdot c(i,j) \cdot s(i,j)$$

where:

- $l(i,j) = \frac{2\mu_1(i,j)\mu_2(i,j)+C_1}{\mu_1^2(i,j)+\mu_2^2(i,j)+C_1}$ represents the luminance,
- $c(i,j) = \frac{2\sigma_1(i,j)\sigma_2(i,j)+C_2}{\sigma_1^2(i,j)+\sigma_2^2(i,j)+C_2}$ represents the contrast, and
- $s(i,j) = \frac{\sigma_{12}(i,j)+C_3}{\sigma_1(i,j)\sigma_2(i,j)+C_3}$ represents the structural similarity
- $\mu_1(i,j) = \sum_{m,n\in W_{ij}} w(m,n)I_1(m,n)$ represents the local mean of luminance for one image,
- $\mu_2(i,j) = \sum_{m,n\in W_{ij}} w(m,n)I_2(m,n)$ represents the local mean of luminance for the second image,

- $\sigma_1^2(i,j) = \sum_{m,n \in W_{ij}} w(m,n) I_1^2(m,n) - \mu_1^2(i,j)$ represents the local standard deviation of luminance for one image,
- $\sigma_2^2(i,j) = \sum_{m,n \in W_{ij}} w(m,n) I_2^2(m,n) - \mu_2^2(i,j)$ represents the local standard deviation of luminance for the second image, and
- $\sigma_{12}(i,j) = \sum_{m,n \in W_{ij}} w(m,n) I_1(m,n) I_2(m,n) - \mu_1(i,j)\mu_2(i,j)$ represents the local correlations of luminance between the two images
- $I_1(i,j)$ and $I_2(i,j)$ are the luminance images,
- W_{ij} represents a k x k region of the image, and
- $w(m,n)$ represents the weight attached at each index in the region.

3.5 Neural Network Parameter Tuning

In order to test our method, we needed to fix a set of neural networks, or ideally, a single neural network. To do so, we needed to tune the parameters of the neural networks that we're using so that they produce the best output. First, we set a seed so that our results are reproducible. Next, we used Optuna, an open source, automated hyperparameter optimization approach [41], to parameter tune U-Net, ResNet, and SIREN.

Rather than tuning to specific images, we tuned over all of the images. In BFSI, we also fix a set of parameters to simplify the tuning process. The values we tuned across all networks were learning rate, the total number of iterations, and exponential sliding window weight. The learning rate is how quickly the model learns, total number of iterations is how many steps it takes, and exponential sliding window weight is how much importance is applied to the previous output.

For the CNNs, we also tuned the standard deviation of the additive normal noise and the upsampling method. Standard deviation of the additive normal noise is how much regularization is applied to the input, and upsampling method how the network increases the size of a representation of the image at each layer where it is initially shrunken down. For SIREN, we tuned the respective frequencies in the first layer and in the hidden layers. The frequencies are how much we stretch the signal at the first layer and in the rest of the layers. We tested values based on what worked well in Ulyanov et al [43] and from early testing. Table 1 contains the combinations we were interested in testing for each neural network. Since Optuna works by sampling different combinations of hyperparameters and prunes the ones that are less likely to work well [41], not all of the combinations shown were tested.

Table 1: Hyperparameter combinations by neural network

Hyperparameter	**SIREN**	**ResNet**	**U-Net**
Learning rate (LR)	{1e-4, 5e-4, 5e-5}	{0.1, 0.01, 0.001, 0.02, 0.03, 0.05}	{0.1, 0.01, 5e-4, 0.001, 0.005}
Weight	{0.95, 0.99}	{0.99, 0.9, 0.95}	{0.99, 0.9, 0.95}
Number of iterations	{1000, 2000, 3000}	{1000, 2000}	{1000, 2000}
STD	0 (fixed)	$\{\frac{1}{20}, \frac{1}{30}, 0.03, 0.0, 0.01, 0.02\}$	$\{\frac{1}{20}, \frac{1}{30}, 0, 0.03, 0.07, 0.1\}$
Frequency 0 (ω_0)	{15, 20, 30, 35, 40}	None (fixed)	None (fixed)
Frequency hidden (ω_{hidden})	{1, 0.9, 2, 0.5}	None (fixed)	None (fixed)
upsampling	None (fixed)	{bilinear, nearest}	{bilinear, nearest}

After collecting this data, we analyzed it to select the best configuration for each neural network. To do so, we found the average change in PSNR and SSIM for each configuration of each neural network. Then, we ranked those values and summed them to get a combined rank. The configuration with the best rank for each neural network was picked. This is summarized in Table 2.

Table 2: Best hyperparameters and average percent change in performance by network type

NN	LR	STD	Weight	ω_0	ω_{hidden}	Upsampling	Mean PSNR Change	Mean SSIM Change
U-Net	0.0010	0.030000	0.99	NaN	NaN	bilinear	+48.050168%	+45.363316%
SIREN	0.0005	0.000000	0.95	40.0	1.0	NaN	+40.094340%	+42.726630%
ResNet	0.0300	0.033333	0.90	NaN	NaN	bilinear	+22.297600%	+22.143800%

4 Results

All of the following results were obtained using the neural networks as configured in Table 2. Note that these configurations are validated at the specific BFSI parameters used to tune and may not necessarily be the best at other parameter values. All experiments were conducted on the CREST GPU cluster. The primary computing node used in this work was *crestgpu6*, which is equipped with an AMD EPYC 7742 processor, 128 GB of DDR4-3200 memory, and two NVIDIA A100 40 GB PCIe GPUs.

4.1 Comparison with BFSI

We first compare our method with the BFSI method by Zhang et al [29]. Using our own 256×256-pixel images, we evaluate the PSNR and SSIM of both the output of the neural network and the BFSI reconstruction. Following the paper, we used bicubic interpolation with scales 1, 2, 4, and 8, which represent the downsampling factors. In this experiment, the sampling ratio is set to 100%, meaning that all binary Fourier patterns are used for reconstruction. Contrary to the paper, we also test with a noise level of 0.2 in addition to 0 noise. The results are summarized in Table 3. Values are highlighted in green when our method achieves higher PSNR/SSIM values and in red otherwise. To compute PSNR and SSIM at higher scale factors, the BFSI reconstruction was upscaled using bicubic interpolation to match the resolution of the original 256×256 images.

Table 3: PSNR and SSIM by different image-scale-network comparisons

		BFSI		SIREN		U-Net		ResNet	
image	**scale**	psnr	ssim	psnr	ssim	psnr	ssim	psnr	ssim
Cameraman	1	21.0155	0.6656	34.2840	0.9937	30.9560	0.8751	33.1285	0.9151
	2	21.0822	0.6864	29.9238	0.8723	29.3656	0.8771	28.1570	0.8407
	4	20.9527	0.6187	23.3974	0.7258	23.4984	0.7372	22.7664	0.6881
	8	15.2204	0.4890	16.6395	0.5953	19.8356	0.5784	16.7654	0.4687
Jelly beans	1	16.8098	0.6967	17.2472	0.9760	17.3585	0.9682	31.6014	0.9397
	2	22.4199	0.8850	22.7748	0.9831	23.7259	0.9787	28.5695	0.9527
	4	23.4136	0.9038	24.1875	0.9428	24.2810	0.9489	28.1031	0.9053
	8	21.3729	0.8241	22.1968	0.8481	22.4670	0.8685	22.5653	0.8404
Resolution chart	1	13.0860	0.5553	43.8339	0.9981	14.5247	0.9552	39.3407	0.9924
	2	14.4313	0.6883	30.4994	0.9683	19.0735	0.9074	29.0235	0.9580
	4	14.0969	0.6464	18.2582	0.8158	15.4161	0.7352	18.6312	0.8304
	8	12.3871	0.5241	12.5947	0.5528	12.7883	0.5574	11.7499	0.5417
Tree	1	19.1743	0.8070	19.8288	0.9715	26.0041	0.8891	31.8894	0.9153
	2	20.6244	0.8078	21.4514	0.8627	24.2152	0.7953	27.5195	0.8379
	4	19.3556	0.6626	20.1984	0.7165	20.1787	0.7173	22.6681	0.6658
	8	16.5003	0.4490	17.7598	0.5317	18.5455	0.4862	18.4462	0.4728

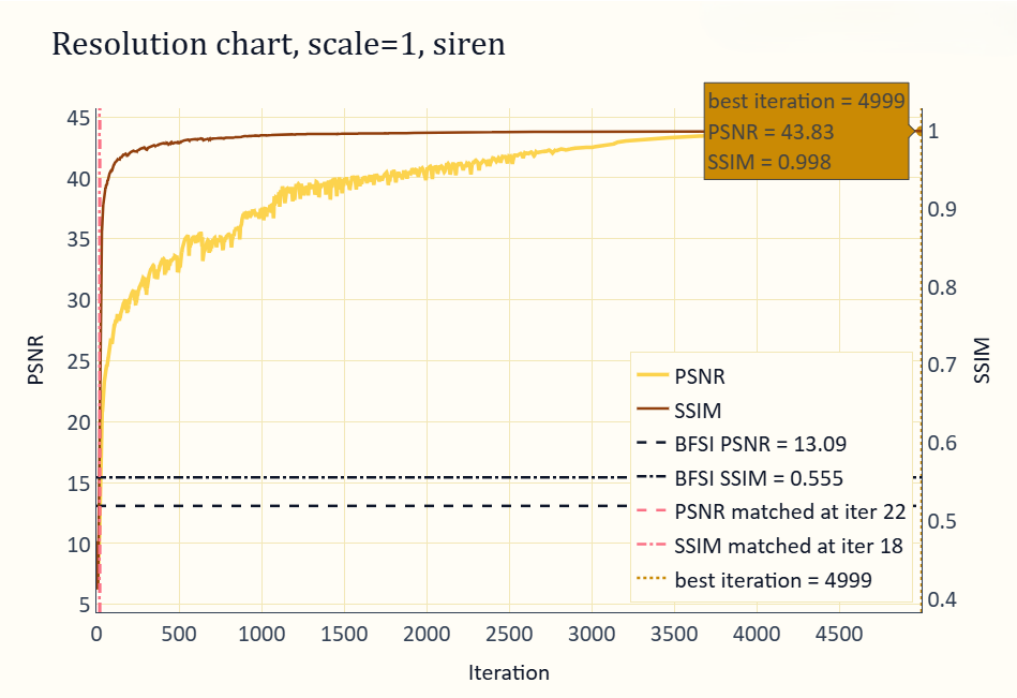
Table 4: PSNR and SSIM by different image-scale-network comparisons with noise

		BFSI		**SIREN**		**U-Net**		**ResNet**	
image	**scale**	psnr_bfsi	ssim_bfsi	psnr	ssim	psnr	ssim	psnr	ssim
Cameraman	1	14.3409	0.1946	20.6343	0.4289	17.2161	0.3075	16.8185	0.3653
	2	17.4838	0.2974	20.9653	0.4170	18.9759	0.3337	18.9255	0.3234
	4	17.9493	0.3732	15.5828	0.5236	19.6423	0.5049	20.6749	0.4132
	8	15.1632	0.4396	15.2368	0.4878	19.2644	0.4449	16.0014	0.4357
Jelly beans	1	12.1140	0.1794	12.2400	0.6480	6.8311	0.5370	18.4701	0.5247
	2	14.7314	0.3620	12.3000	0.6608	16.0663	0.3692	22.6510	0.7945
	4	17.5225	0.6011	11.8987	0.6773	10.5029	0.6693	21.8174	0.8119
	8	21.0755	0.7588	21.1220	0.6834	20.7340	0.6864	22.1877	0.7763
Resolution chart	1	10.7609	0.3248	20.8930	0.7758	13.1631	0.4064	20.5074	0.8513
	2	11.5877	0.4513	19.2682	0.7287	15.5066	0.5704	20.9174	0.7930
	4	13.3703	0.5282	14.5552	0.6005	13.8651	0.4793	17.1324	0.7471
	8	12.2990	0.4975	12.4025	0.4941	12.5883	0.4615	13.0125	0.5909
Tree	1	14.1136	0.3065	19.3059	0.6287	18.2933	0.5170	17.0924	0.6155
	2	17.3979	0.4738	18.9314	0.6189	20.4582	0.6076	19.8248	0.6289
	4	19.1467	0.5218	18.6166	0.5961	19.4396	0.5468	18.9298	0.5688
	8	17.0987	0.4277	17.4309	0.4995	18.3413	0.4644	18.7146	0.4732

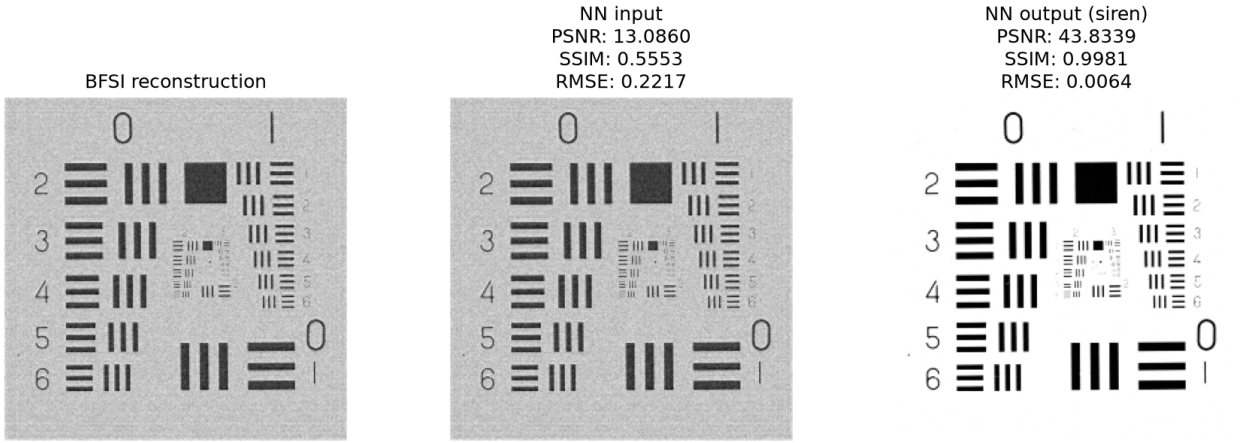
The data in Table 3 shows that higher scale factors generally lead to worse reconstruction quality, especially at scale 8. However, BFSI can often achieve better performance at scale 2. At scale 1, the input contains more high-frequency information from the original image, which explains why the neural network generally achieves higher PSNR and SSIM at scale 1. Across all 48 image-scale-network comparisons, our method achieves higher PSNR and SSIM than BFSI in most cases. Only three cases show lower SSIM or PSNR values: one from U-Net and two from ResNet, while SIREN consistently outperforms BFSI.

In Table 4, however, the models generally do not perform as well. U-Net and SIREN have five and six cases where they perform worse than the neural network input, whereas ResNet only has two cases where it fails. This may mean that ResNet is more robust to noise but will need to be investigated further.

Next, we tested our images using the bilinear and Lanczos upsampling method in addition to bicubic. As shown in Figure 14a, the reconstruction quality improves steadily throughout training when BFSI is combined with SIREN. Both curves rise rapidly and eventually converge to stable, high values, clearly surpassing the baseline indicated by the conventional BFSI reference lines. A visual comparison in Figure 14b further highlights this improvement. The conventional BFSI reconstruction retains noticeable noise and blurred fine structures, whereas the SIREN-enhanced output restores sharp edges and clean line patterns across the image. The background is also reconstructed as white which closely matches the ground truth.



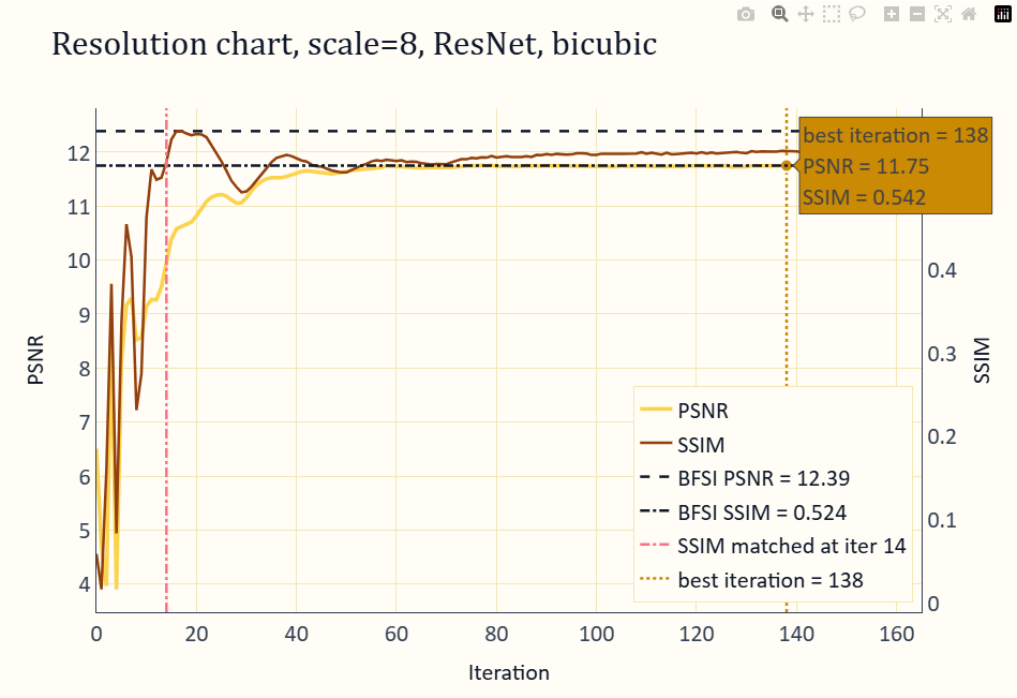
(a) Evolution of PSNR and SSIM under a 100% sampling rate: BFSI + SIREN compared with BFSI



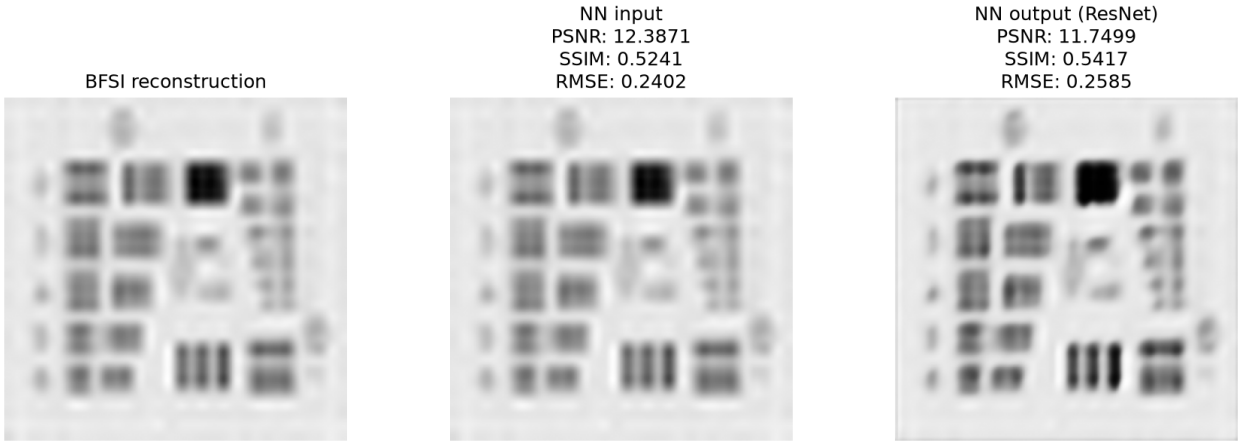
(b) Visual comparison

Figure 14: Performance gain of BFSI + SIREN over conventional BFSI under full sampling

Although most experiments showed clear improvements over the BFSI baseline, there were a few situations where the neural network failed to outperform BFSI. These cases occurred primarily at scale 8, where the amount of available information is heavily reduced and the reconstruction task becomes significantly more challenging.



(a) ResNet failure case at scale 8 : degraded PSNR and SSIM performance



(b) Visual comparison

Figure 15: Performance degradation of ResNet at scale 8

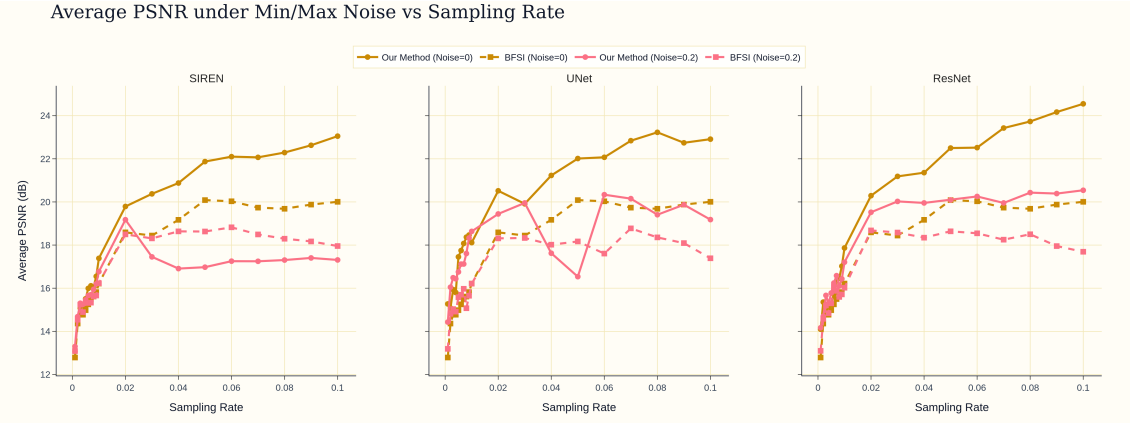
As shown in Figure 15a, the PSNR and SSIM curves highlight the limitations of ResNet when operating at scale 8 and the available information is heavily reduced. The PSNR curve stays consistently below the BFSI baseline throughout training, indicating that the network cannot recover enough detail from the sparse samples. In contrast, SSIM briefly reaches the BFSI reference early on and then stabilizes slightly above it, suggesting that the model preserves some structural similarity and contrast even though it fails to improve PSNR. The visual comparison in Figure 15b reflects this quantitative pattern. ResNet reconstruction is blurrier and less defined than the BFSI reconstruction, with noticeable loss of high-frequency detail. However, the slightly higher SSIM manifests as marginally stronger contrast in certain regions, giving the image a faint sense of coherence despite its overall degradation.

Our goal is to construct high-quality images under limited measurements. Next, we investigated how our method compares at 10% sampling rates to BFSI at 100% sampling. Table 5 shows the PSNR and SSIM for

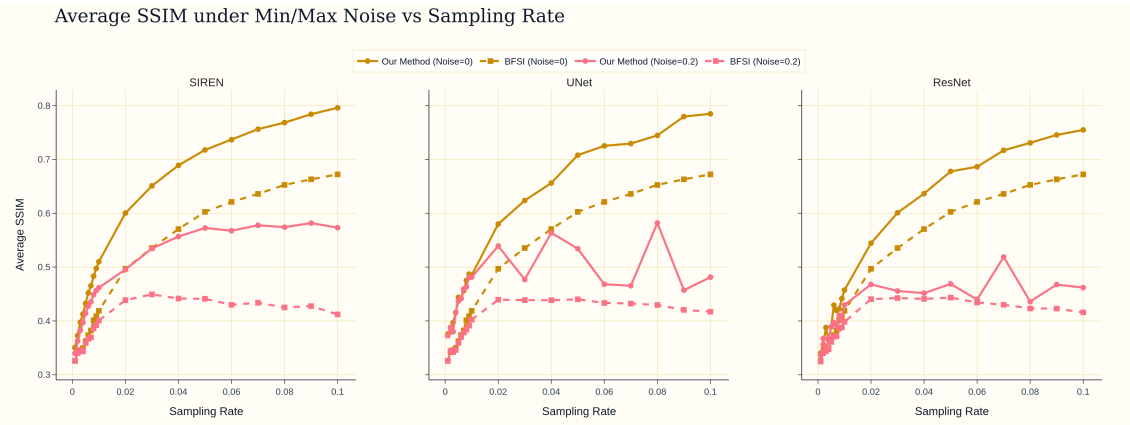
each neural network and image across two noise levels. These are fixed for scale 1 and bicubic upsampling. An average is also included at the bottom. At zero noise level, our method at 10% is comparable to BFSI at 100%, but our method is better at a noise level of 0.2.

Table 5: Our Method at 10% vs BFSI 100% (PSNR/SSIM)

		BFSI		**SIREN**		**U-Net**		**ResNet**	
noise_level_before	**image**	psnr	ssim	psnr	ssim	psnr	ssim	psnr	ssim
0.0	Resolution chart	21.47	0.47	23.56	0.92	12.41	0.82	22.62	0.90
	Tree	26.55	0.77	20.56	0.79	20.27	0.77	24.67	0.76
	Jelly beans	28.45	0.62	22.23	0.96	22.39	0.97	31.03	0.94
	Cameraman	27.10	0.65	25.54	0.80	25.55	0.80	24.42	0.75
	Average	**25.89**	**0.63**	**22.97**	**0.87**	**20.15**	**0.84**	**25.69**	**0.84**
0.2	Resolution chart	10.92	0.33	18.20	0.69	12.03	0.52	17.50	0.65
	Tree	14.20	0.31	18.90	0.61	18.24	0.55	20.59	0.57
	Jelly beans	11.54	0.18	11.80	0.67	16.42	0.76	21.97	0.85
	Cameraman	14.42	0.19	15.72	0.53	20.12	0.42	20.49	0.35
	Average	**12.77**	**0.25**	**16.16**	**0.63**	**16.70**	**0.56**	**20.14**	**0.61**



(a) Average PSNR for noise level 0 and 0.2 for each neural network across sampling rates compared to BFSI



(b) Average SSIM for noise level 0 and 0.2 for each neural network across sampling rates compared to BFSI

Figure 16: Comparing our method at limited measurements with BFSI at high sampling rates

We see that compared to BFSI alone our method has consistently higher SSIM for all three neural networks

at both 0 and 0.2 noise levels (see Fig. 16b). For PSNR, ResNet has higher PSNR than BFSI alone, however SIREN and U-Net perform worse when the noise level is 0.2 (see Fig. 16a).

Figure 17 shows the average PSNR of our method and BFSI alone across different sampling rates. We can see that our method using ResNet at 10% sampling with zero noise is comparable to BFSI at about 20% sampling. Additionally, the PSNR is consistently higher for our method.

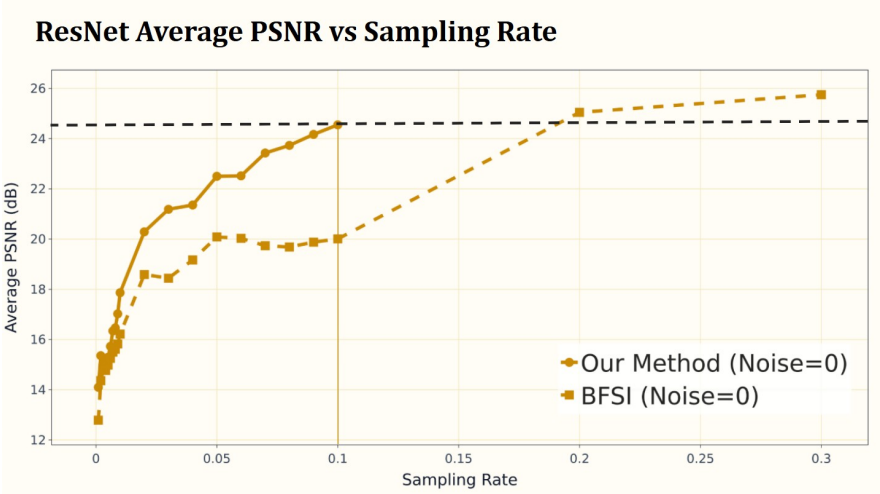


Figure 17: Average PSNR across sampling rates comparing our method for limited measurements and BFSI for higher measurements

Figure 18 shows a visual comparison between BFSI at a 100% sampling rate (Figure 18a) and our method at a 10% sampling rate (Figure 18b). The man in our method is completely denoised, and it is clear that he is holding a tripod. The BFSI output at 10% sampling rate is still noisy, however.



(a) BFSI at 100% samples



(b) Our method at 10% samples

Figure 18: Our method for the Cameraman image at scale 1 with noise=0.2 using U-Net

4.2 Comparison with FSI + DIP

In this experiment, we test our method in comparison to FSI and DIP [34]. We tested three images (Cameraman, Man, and Boat) at a 256×256 pixel resolution with added Gaussian noise of 0 mean and 0.1 standard deviation. We test each image at scale 1 (original spatial resolution) at 5 different sampling rates (5%, 10%, 15%, 20%, and 25%) of the Fourier spectrum with two different sampling methods (variable density sampling method [45] and our iterative threshold adaptive sampling method). After BFSI reconstruction, they were tested on all our neural networks. The results of our testing are in Table 6 and 7.

Table 6: PSNR results for the variable-density and adaptive sampling methods.

Image	Ratio	Reference DIP		U-Net		ResNet		SIREN	
		VD	Adaptive	VD	Adaptive	VD	Adaptive	VD	Adaptive
Boat	0.05	23.3503	**23.9625**	22.5220	**22.6497**	22.4148	**22.6186**	22.4259	**22.4979**
	0.10	23.6574	**25.4305**	21.4486	**23.4902**	22.2909	**23.5206**	22.6990	**23.7209**
	0.15	23.7826	**26.0569**	22.7753	**23.2019**	22.4089	**23.5210**	23.1817	**24.0631**
	0.20	24.2127	**26.1648**	22.5071	**23.1251**	22.9842	**23.7833**	23.4162	**24.0729**
	0.25	24.4248	**26.4952**	22.4969	**23.0901**	22.8389	**23.6543**	23.6988	**24.2491**
Cameraman	0.05	21.8752	**22.6251**	19.6335	**21.3237**	**21.4338**	21.2489	21.8036	**21.8639**
	0.10	22.1575	**24.5087**	20.6205	**22.9938**	20.8715	**23.1742**	22.6375	**23.6697**
	0.15	22.5808	**25.4237**	21.9777	**23.2931**	22.4654	**23.6399**	23.4136	**23.5531**
	0.20	22.8742	**25.8686**	22.1995	**23.6534**	22.8220	**23.5037**	24.0123	**24.3574**
	0.25	23.2686	**26.2456**	21.4908	**22.4620**	22.9910	**23.9661**	24.4133	**24.8286**
Man	0.05	24.0224	**24.8148**	20.7951	**21.1799**	**21.0198**	20.7491	20.3567	**20.6049**
	0.10	24.2687	**25.8613**	21.0837	**22.0289**	**21.3240**	21.3236	21.3366	**21.6983**
	0.15	24.5094	**26.2516**	21.2298	**22.0697**	21.2429	**21.2771**	21.5917	**22.5248**
	0.20	24.8775	**26.6977**	21.2189	**22.2005**	21.6358	**22.1135**	22.1480	**22.7767**
	0.25	25.1363	**26.9869**	21.6273	**22.2453**	21.4786	**22.2568**	22.5466	**23.0302**

Table 7: SSIM results for the variable-density and adaptive sampling methods.

Image	Ratio	Reference DIP		U-Net		ResNet		SIREN	
		VD	Adaptive	VD	Adaptive	VD	Adaptive	VD	Adaptive
Boat	0.05	0.5647	**0.5947**	0.4883	**0.5167**	0.5204	**0.5222**	**0.5282**	0.5281
	0.10	0.5818	**0.6723**	0.5280	**0.5558**	0.5000	**0.5786**	0.5410	**0.5864**
	0.15	0.5880	**0.6993**	0.5126	**0.5672**	0.4930	**0.5571**	0.5573	**0.6012**
	0.20	0.6019	**0.6899**	0.4847	**0.5470**	0.5309	**0.5786**	0.5674	**0.6011**
	0.25	0.6150	**0.7076**	0.5297	**0.5486**	0.5357	**0.5719**	0.5784	**0.6127**
Cameraman	0.05	**0.6180**	0.5974	**0.5693**	0.4811	0.4983	**0.5076**	**0.5474**	0.5472
	0.10	0.6124	**0.6838**	0.5247	**0.5403**	0.3834	**0.5124**	0.5402	**0.5790**
	0.15	0.6326	**0.7054**	0.3964	**0.5205**	0.4718	**0.5109**	0.5657	**0.5960**
	0.20	0.6287	**0.6955**	0.4243	**0.5035**	0.4660	**0.5345**	0.5844	**0.6307**
	0.25	0.6361	**0.6964**	0.5004	**0.5031**	0.5013	**0.5620**	0.5713	**0.6026**
Man	0.05	0.5495	**0.6056**	0.4857	**0.5085**	0.4996	**0.5052**	0.5247	**0.5278**
	0.10	0.5727	**0.6647**	0.4919	**0.5752**	0.5127	**0.5733**	0.5358	**0.5921**
	0.15	0.5836	**0.6770**	0.5075	**0.5724**	0.5307	**0.5699**	0.5613	**0.6149**
	0.20	0.6040	**0.7004**	0.5394	**0.5832**	0.5562	**0.5786**	0.5870	**0.6278**
	0.25	0.6111	**0.7142**	0.5399	**0.5889**	0.5318	**0.5984**	0.6040	**0.6447**

We report that our adaptive sampling method performs better than the variable density method in line with FSI + DIP. At different sampling rates, our method reports slightly worse average PSNR compared to FSI + DIP using the variable density sampling method (See Fig. 19 and 20).

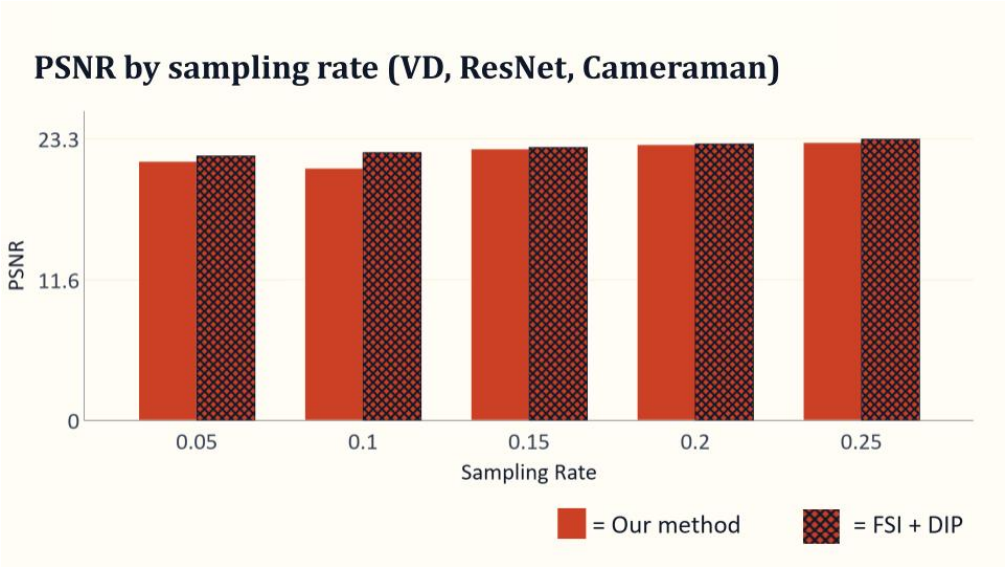


Figure 19: Difference between Sampling Rates of our method vs FSI+DIP with Variable Density sampling with ResNet on the Cameraman image

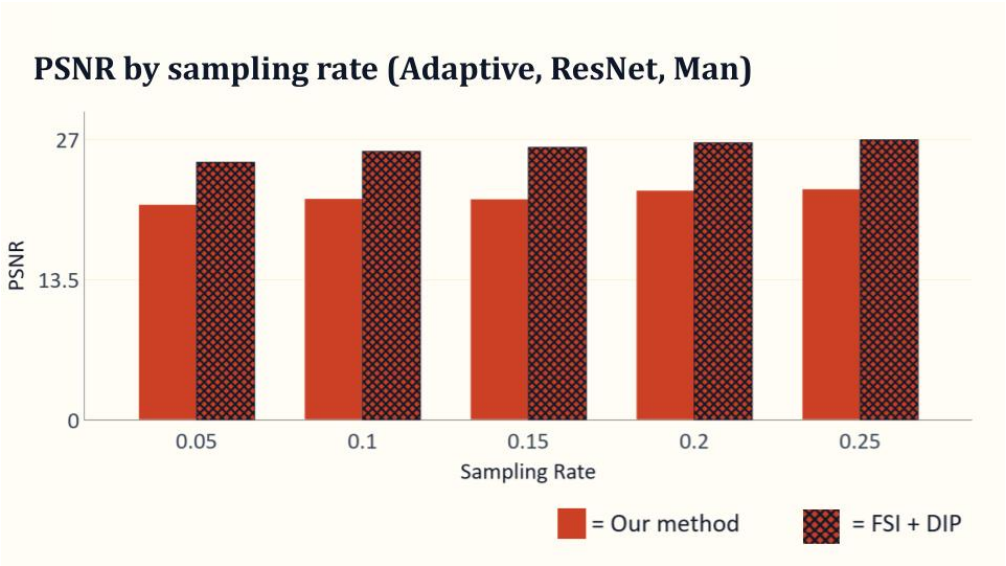


Figure 20: Difference between Sampling Rates of our method vs FSI+DIP with Adaptive sampling with ResNet on the Man image

Figures 19 and 20 display the difference in PSNR of our method and FSI+DIP for different images and sampling method on ResNet. We can see that in Fig. 19 the difference is minimal, however in Fig. 20 the difference is much more noticeable. We believe this is because our method did not parameter tune to the Man image and that their adaptive method is different from our adaptive method.

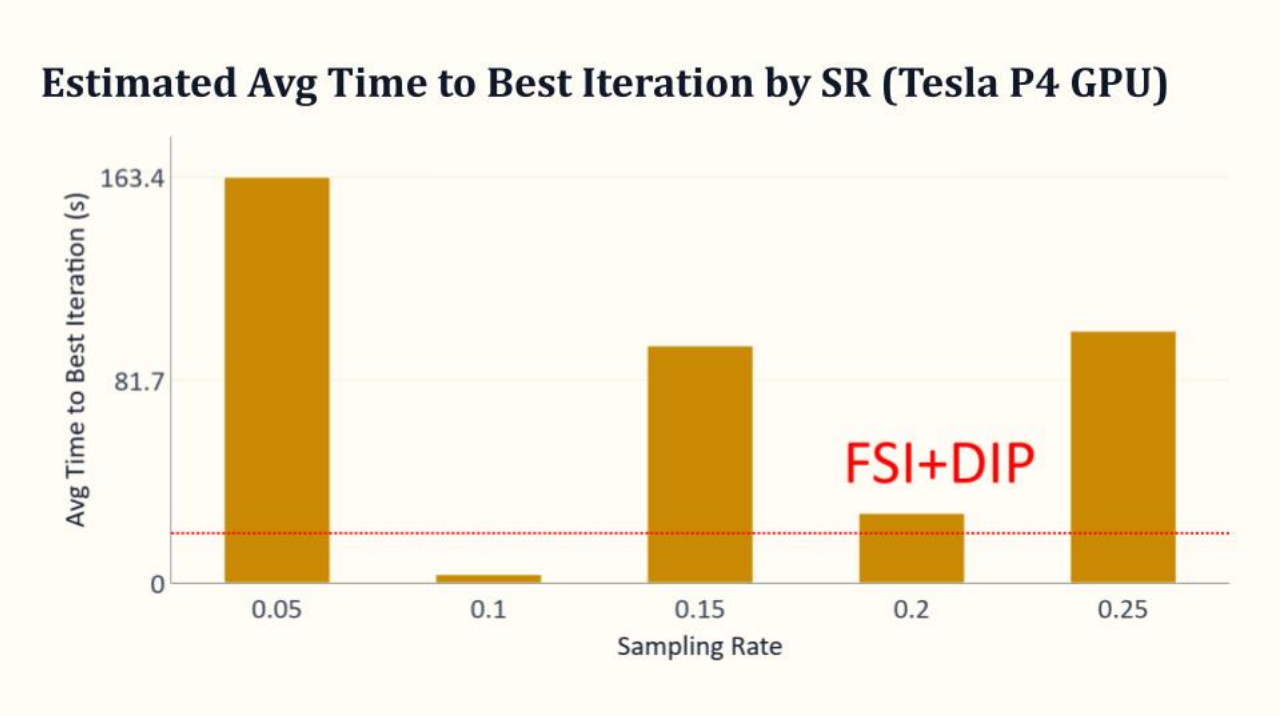


Figure 21: Time from Start to Best Iteration on average for each sampling rate

Figure 21 shows the average time from start to best iteration. FSI+DIP reports an average denoising time of 20 seconds We see that at 10% sampling rate our method converges faster than FSI+DIP. We believe this is because we did our parameter testing at a fixed 10% BFSI reconstruction. In general we expect our results to be slightly worse since FSI+DIP uses FSI's exact grayscale patterns and reconstruction.

5 Conclusion

This work presented a Binary Fourier Single-pixel Imaging reconstruction framework enhanced by neural network priors. Experimental results showed that all three investigated neural network architectures generally improved reconstruction quality over conventional BFSI under the evaluated sampling ratios, noise levels, image scales, and sampling strategies. These improvements demonstrate that neural network priors effectively complement the BFSI reconstruction process by recovering image structures that are not fully reconstructed using the physical model alone.

One of the most significant observations is the improvement in sampling efficiency. Compared with conventional BFSI, the proposed framework achieved higher reconstruction quality using fewer measurements, indicating that neural network priors can compensate for information loss caused by undersampling. This characteristic is particularly important for practical SPI systems because reducing the number of projected patterns directly shortens acquisition time.

The experiments also revealed distinct behaviors among the investigated neural network architectures. SIREN consistently produced the highest reconstruction quality across most experimental settings, whereas U-Net exhibited noticeable sensitivity to random weight initialization, leading to larger variations between optimization runs. These results suggest that reconstruction performance depends not only on the reconstruction framework itself but also on the characteristics of the selected neural network architecture.

Another important observation concerns the optimization process. For several experiments, the reconstruction quality reached a maximum well before the final optimization iteration, after which little improvement was observed and performance occasionally degraded. Since the reported results were obtained by selecting the iteration with the highest PSNR and SSIM, the reported performance represents an upper bound because these metrics require access to the ground-truth image. Consequently, although the results

provide a useful benchmark for evaluating reconstruction capability, they do not directly correspond to the stopping criterion available in practical SPI systems.

The comparison with the FSI+DIP framework further illustrates both the strengths and limitations of the proposed approach. Although the proposed method did not fully match the reconstruction quality reported for FSI+DIP, it substantially reduced the performance gap while maintaining the practical advantages of Binary Fourier illumination. The remaining difference is likely related to the binary approximation inherent to BFSI, which introduces quantization errors that are absent in grayscale Fourier measurements, as well as differences in the acquisition setup and reconstruction methodology.

While the proposed physics-guided loss function effectively combines measurement consistency and Fourier-domain information, the relative contributions of its different components remain empirically determined. Similarly, the hyperparameters used throughout this study were selected using 4 images and fix set of parameters for BFSI with no noise, providing a consistent experimental configuration but limiting the assessment of generalization.

5.1 Future Work

Although the proposed BFSI + NN framework demonstrates promising reconstruction performance, several challenges remain to be addressed before it can be applied reliably in practical single-pixel imaging systems.

A first direction concerns the development of adaptive sampling and parameter selection strategies. The current framework relies on manually tuned parameters that were selected empirically and may not be optimal for different images, image resolutions, or sampling conditions. Future work will therefore investigate adaptive sampling strategies and automatic parameter optimization methods capable of selecting near-optimal configurations while improving the robustness and generalization of the reconstruction process.

Another important research direction is the development of a practical early stopping criterion. In the current study, the best reconstruction was selected based on a weighted linear combination of PSNR and SSIM, which requires access to the ground-truth image and therefore represents an oracle stopping strategy rather than a practical implementation. Future work will focus on designing stopping criteria based only on information available during reconstruction. In particular, methods inspired by Wang et al. [57], which use the variance of the PSNR as an early stopping criterion for Deep Image Prior, will be investigated. Such approaches could significantly reduce the computational cost by avoiding unnecessary optimization up to the fixed 5000 iterations.

Unlike FSI+DIP, which employs the standard DIP denoising objective, the proposed method combines measurement consistency with Fourier-domain constraints to preserve the physical consistency of the reconstructed image. However, determining the optimal balance between the measurement loss, Fourier spectrum loss, image-domain loss, and regularization remains an open problem. Another issue of our loss function is that we store must all the patterns into RAM for our GPU to use. For higher resolution images we require much more memory than the simple image loss ex. for a 512×512 pixel image at 10% sampling rate will use approximately 164 GBs of RAM. Future work will investigate adaptive weighting strategies to improve reconstruction quality and robustness across different imaging conditions.

Closely related to the loss function is the normalization strategy adopted during optimization. In the current implementation, the measurement outputs are normalized before computing the loss. While this stabilizes optimization, it also removes the scaling relationship between the measured signals and the predicted measurements. Future work will investigate alternative normalization schemes that preserve this physical scaling while identifying neural network parameters that remain stable under the revised formulation.

The experimental results also highlighted the importance of network architecture and initialization. In particular, U-Net exhibited a noticeable sensitivity to random weight initialization, whereas SIREN produced more stable reconstructions. Future work will therefore investigate more robust initialization techniques and additional neural network architectures to improve both reconstruction accuracy and optimization stability.

Finally, additional work is required to improve the computational efficiency and practical applicability of the proposed framework. This includes developing faster optimization procedures, investigating automatic hyperparameter selection methods such as cross-validation, and validating the proposed reconstruction framework on a real Binary Fourier Single-Pixel Imaging system. Experimental evaluation under realistic acquisition conditions, including measurement noise and hardware imperfections, will provide a more com-

prehensive assessment of the method's practical performance.

References

[1] Xing Lin, Gordon Wetzstein, Yebin Liu, and Qionghai Dai. Dual-coded compressive hyperspectral imaging. *Optics letters*, 39(7):2044–2047, 2014.

[2] Qi Pian, Ruoyang Yao, Lingling Zhao, and Xavier Intes. Hyperspectral time-resolved wide-field fluorescence molecular tomography based on structured light and single-pixel detection. *Optics letters*, 40(3):431–434, 2015.

[3] Wen-Kai Yu, Xu-Ri Yao, Xue-Feng Liu, Ruo-Ming Lan, Ling-An Wu, Guang-Jie Zhai, and Qing Zhao. Compressive microscopic imaging with "positive–negative" light modulation. *Optics Communications*, 371:105–111, 2016.

[4] Wen-Kai Yu. Super sub-nyquist single-pixel imaging by means of cake-cutting hadamard basis sort. *Sensors*, 19(19):4122, 2019.

[5] Yinqi Wang, Kun Huang, Jianan Fang, Ming Yan, E Wu, and Heping Zeng. Mid-infrared single-pixel imaging at the single-photon level. *Nature Communications*, 14(1):1073, 2023.

[6] Matthew P Edgar, Graham M Gibson, Richard W Bowman, Baoqing Sun, Neal Radwell, Kevin J Mitchell, Stephen S Welsh, and Miles J Padgett. Simultaneous real-time visible and infrared video with single-pixel detectors. *Scientific reports*, 5(1):10669, 2015.

[7] Weili Li, Xuemei Hu, Jingbo Wu, Kebin Fan, Benwen Chen, Caihong Zhang, Wei Hu, Xun Cao, Biaobing Jin, Yanqing Lu, et al. Dual-color terahertz spatial light modulator for single-pixel imaging. *Light: Science & Applications*, 11(1):191, 2022.

[8] Luca Zanotto, Riccardo Piccoli, Junliang Dong, Roberto Morandotti, Luca Razzari, et al. Single-pixel terahertz imaging: a review. *Opto-Electron. Adv.*, 3(9):20001201, 2020.

[9] Wai Lam Chan, Kriti Charan, Dharmpal Takhar, Kevin F Kelly, Richard G Baraniuk, and Daniel M Mittleman. A single-pixel terahertz imaging system based on compressed sensing. *Applied Physics Letters*, 93(12), 2008.

[10] Rayko Ivanov Stantchev, Xiao Yu, Thierry Blu, and Emma Pickwell-MacPherson. Real-time terahertz imaging with a single-pixel detector. *Nature communications*, 11(1):2535, 2020.

[11] Luca Zanotto, Giacomo Balistreri, Andrea Rovere, O-Pil Kwon, Roberto Morandotti, Riccardo Piccoli, and Luca Razzari. Terahertz scanless hypertemporal imaging. *Laser & Photonics Reviews*, 17(8):2200936, 2023.

[12] Jun-Tian Ye, Chao Yu, Wenwen Li, Zheng-Ping Li, Hai Lu, Rong Zhang, Jun Zhang, Feihu Xu, and Jian-Wei Pan. Ultraviolet photon-counting single-pixel imaging. *Applied Physics Letters*, 123(2), 2023.

[13] Jiaqi Song, Baolei Liu, Yao Wang, Chaohao Chen, Xuchen Shan, Xiaolan Zhong, Ling-An Wu, and Fan Wang. Computational and dark-field ghost imaging with ultraviolet light. *Photonics Research*, 12(2):226–234, 2024.

[14] Daixuan Wu, Jiawei Luo, Guoqiang Huang, Yuanhua Feng, Xiaohua Feng, Runsen Zhang, Yuecheng Shen, and Zhaohui Li. Imaging biological tissue with high-throughput single-pixel compressive holography. *Nature Communications*, 12(1):4712, 2021.

[15] Zhiheng Du, Wenjing Zhao, Aiping Zhai, Zhaoxia Zhang, and Dong Wang. Dmd-based single-pixel off-axis interferometry for wavefront reconstruction of a biological sample. *Applied Physics Letters*, 123(3), 2023.

[16] Ya-Nan Zhao, Hong-Yun Hou, Jia-Cheng Han, Shan Gao, Sheng-Wei Cui, De-Zhong Cao, Bao-Lai Liang, Hong-Chao Liu, and Su-Heng Zhang. Single-pixel phase microscopy without 4f system. *Optics and Lasers in Engineering*, 163:107474, 2023.

[17] Wenjing Zhao, Zhiheng Du, Aiping Zhai, and Dong Wang. Wavefront imaging of a biological sample using dmd-based single-pixel phase-shifting interferometric techniques: An experimental comparison. *Optics & Laser Technology*, 172:110483, 2024.

[18] Ling-Yu Dou, De-Zhong Cao, Lu Gao, and Xin-Bing Song. Dark-field ghost imaging. *Optics Express*, 28(25):37167–37176, 2020.

[19] Jianwei Ma. Single-pixel remote sensing. *IEEE Geoscience and Remote Sensing Letters*, 6(2):199–203, 2009.

[20] Baris I Erkmen. Computational ghost imaging for remote sensing. *Journal of the Optical Society of America A*, 29(5):782–789, 2012.

[21] Graham M Gibson, Baoqing Sun, Matthew P Edgar, David B Phillips, Nils Hempler, Gareth T Maker, Graeme PA Malcolm, and Miles J Padgett. Real-time imaging of methane gas leaks using a single-pixel camera. *Optics express*, 25(4):2998–3005, 2017.

[22] Margie P Olbinado, David M Paganin, Yin Cheng, and Alexander Rack. X-ray phase-contrast ghost imaging using a single-pixel camera. *Optica*, 8(12):1538–1544, 2021.

[23] Y Klein, A Schori, IP Dolbnya, K Sawhney, and S Shwartz. X-ray computational ghost imaging with single-pixel detector. *Optics express*, 27(3):3284–3293, 2019.

[24] O Sefi, Y Klein, E Strizhevsky, IP Dolbnya, and S Shwartz. X-ray imaging of fast dynamics with single-pixel detector. *Optics Express*, 28(17):24568–24576, 2020.

[25] Huiquan Li, Zhiyuan Ye, Tianyu Yuan, Lu Hua, Yuchuan Zhong, Jinyue Hu, Jun Xiong, Tianxi Sun, and Xuepeng Sun. Magnified x ray ghost imaging with tapered polycapillary optics free of the penumbra effect. *Optics Letters*, 49(2):274–277, 2024.

[26] Graham M Gibson, Steven D Johnson, and Miles J Padgett. Single-pixel imaging 12 years on: a review. *Optics express*, 28(19):28190–28208, 2020.

[27] Yan Cai, Shijian Li, Wei Zhang, Hao Wu, Xuri Yao, and Qing Zhao. A detail-enhanced sampling strategy in hadamard single-pixel imaging. *Chinese Optics Letters*, 21(7):071101, 2023.

[28] Rui Sun, Jiale Long, Yi Ding, Jiaye Kuang, and Jiangtao Xi. Hadamard single-pixel imaging based on positive patterns. *Photonics*, 10(4), 2023.

[29] Zibang Zhang, Xueying Wang, and Jingang Zhong. Fast fourier single-pixel imaging using binary illumination, 2016.

[30] Robert W. Floyd and Louis Steinberg. An Adaptive Algorithm for Spatial Greyscale. *Proceedings of the Society for Information Display*, 17(2):75–77, 1976.

[31] Kai Song, Yaoxing Bian, Dong Wang, Runrui Li, Ku Wu, Hongrui Liu, Chengbing Qin, Jianyong Hu, and Liantuan Xiao. Advances and challenges of single-pixel imaging based on deep learning. *Laser & Photonics Reviews*, 19(7):2401397, 2025.

[32] Shan-Guo Feng, Da Chen, Hua-Hua Wang, Jia-Ning Cao, Zhi-Wei Zhang, Yun-Song Gu, Hong-Yue Xiao, Chang-Qi Zhang, Ze Zhang, and Lu Gao. Ghost image of rotating object with group frame neural network combining frame correction algorithm. *Optics & Laser Technology*, 170:110241, 2024.

[33] Ye Tian, Ying Fu, and Jun Zhang. Joint supervised and unsupervised deep learning method for single-pixel imaging. *Optics & Laser Technology*, 162:109278, 2023.

[34] Wei Lun Tey, Sing Yee Chua, Mau-Luen Tham, and Yeong-Nan Phua. Fourier single pixel imaging reconstruction using deep image prior. In *Proceedings of the 2024 7th Artificial Intelligence and Cloud Computing Conference*, AICCC '24, page 215–221, New York, NY, USA, 2025. Association for Computing Machinery.

[35] Python Software Foundation. *Python Language Reference*. Python Software Foundation. Version 3.11.13.

[36] Signal Processing Application Laboratory, Kanagawa Institute of Technology. Standard image database. http://www.ess.ic.kanagawa-it.ac.jp/app_images_j.html, 2022. Accessed: 2026-07-1.

[37] Stefan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Neinert, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. 2:e453, 2014.

[38] GeeksforGeeks. Python — peak signal-to-noise ratio (psnr). https://www.geeksforgeeks.org/python/python-peak-signal-to-noise-ratio-psnr/, 2025. Accessed: 2026-07-24.

[39] Abhinau K. Venkataramanan, Chengyang Wu, Alan C. Bovik, Ioannis Katsavounidis, and Zafar Shahid. A hitchhiker's guide to structural similarity. *IEEE Access*, 9:28872–28896, 2021.

[40] Fei Wang, Chenglong Wang, Mingliang Chen, Wenlin Gong, Yu Zhang, Shensheng Han, and Guohai Situ. Far-field super-resolution ghost imaging with a deep neural network constraint. *Light: Science & Applications*, 11:1, 01 2022.

[41] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[42] Suqin Nan, Lin Luo, Xuanpengfan Zou, Yang Guo, Xianwei Huang, Wei Tan, Xiaohui Zhu, Teng Jiang, Chuang Li, Yanfeng Bai, and Xiquan Fu. Wide-field scanning ghost imaging based on a local binary pattern and untrained neural network. *Opt. Express*, 32(23):41644–41656, Nov 2024.

[43] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *International Journal of Computer Vision*, 128(7):1867–1888, 2020.

[44] Hengkang Wang, Taihui Li, Zhong Zhuang, Tiancong Chen, Hengyue Liang, and Ju Sun. Early stopping for deep image prior. *Transactions on Machine Learning Research*, 2023.

[45] Meng Wenwen, Dongfeng Shi, Yuan Kee, Yingjian Wang, and Chengyu Fan. Sparse fourier single-pixel imaging. *Optics Express*, 27:31490–31503, 10 2019.

[46] Hongjie Liu, Yuki Matsuda, Shujun Zheng, Yusuke Saita, Xiao Lin, Takanori Nomura, and Xiaodi Tan. Adaptive sampling strategy for fourier single-pixel imaging. *Optics Communications*, 573:131027, 2024.

[47] Aman Dalmia. dalmia/siren, June 2020.

[48] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions, 2020.

[49] Zhenyu Liang, Dabin Yu, Zhengdong Cheng, Xiang Zhai, and Yangdi Hu. Compressed sensing fourier single pixel imaging algorithm based on joint discrete gradient and non-local self-similarity priors. *Optical and Quantum Electronics*, 52, 08 2020.

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[52] Haoyu Ren, Mostafa El-Khamy, and Jungwon Lee. Dn-resnet: Efficient deep residual network for image denoising, 2018.

[53] Hyekyoung Kang, Chanrok Park, and Hyungjin Yang. Evaluation of denoising performance of resnet deep learning model for ultrasound images corresponding to two frequency parameters. *Bioengineering*, 11(7), 2024.

[54] Wenjing Zhao, Lei Gao, Aiping Zhai, and Dong Wang. Comparison of common algorithms for single-pixel imaging via compressed sensing. *Sensors (Basel)*, 23(10):4678, May 2023.

[55] Zhenyu Liang, Dabin Yu, Zhengdong Cheng, Xiang Zhai, and Yangdi Hu. Compressed sensing fourier single pixel imaging algorithm based on joint discrete gradient and non-local self-similarity priors. *Opt. Quantum Electron.*, 52(8), August 2020.

[56] Jiaosheng Li, Bo Wu, Tianyun Liu, and Qinnan Zhang. Urnet: High-quality single-pixel imaging with untrained reconstruction network. *Optics and Lasers in Engineering*, 166:107580, 2023.

[57] Hengkang Wang, Taihui Li, Zhong Zhuang, Tiancong Chen, Hengyue Liang, and Ju Sun. Early stopping for deep image prior. *Transactions on Machine Learning Research*, 2023.