FINAL PRESENTATION FOR MITSUBISHI ELECTRIC CORPORATION

Model-Assisted Binary Fourier Single-Pixel Imaging

Megan Joseph - San Jose State University

Marie Ange Malla Ndassi - African Institute for Mathematical Sciences Cameroon

Han Nguyen - Louisiana State University

Xiangwen Zhao - University of Osaka

Acknowledgments

Advisors & Mentors

Academic Mentor

Kean Fallon, *Iowa State University*

Industry Mentors

Keito Arifuku, *Mitsubishi Electric Corporation*

Ryuhei Takahashi, *Mitsubishi Electric Corporation*

Funding & Support

Graduate Research in Industrial Projects (**G-RIPS**)

Advanced Institute for Materials Research (**AIMR**)

Institute for Pure & Applied Mathematics (**IPAM**)

Mathematical Science Center for Co-creative Society (**MathCCS**)

Tohoku Forum for Creativity (**TFC**)

Mitsubishi Electric Corporation

GOAL

Reconstruct high-quality images from limited BFSI measurements using a model-driven NN.

BFSI : Binary Fourier Single-pixel Imaging
NN : Neural Network

Agenda

01 Background & Motivation

02 Methods

03 Results

04 Conclusion & Discussion

SECTION 01

Background & Motivation

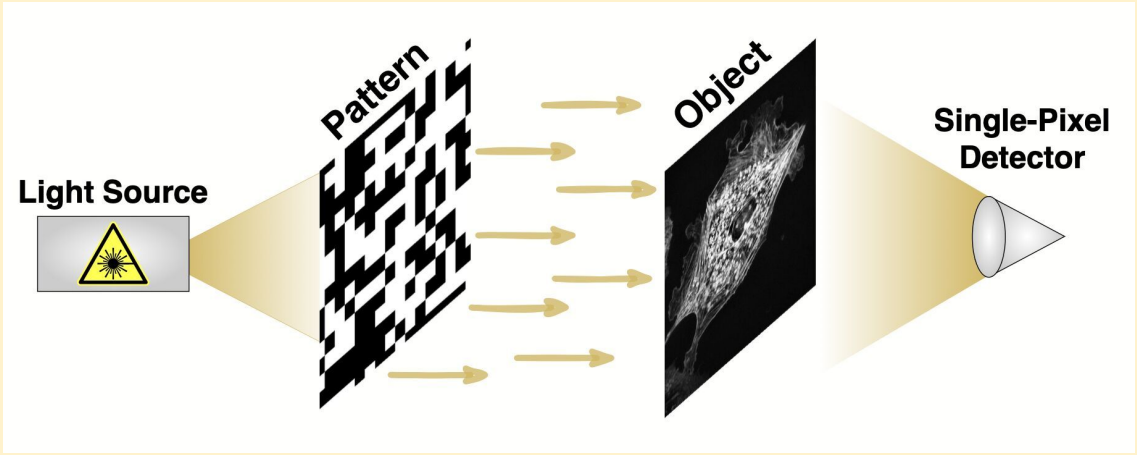


Background

Conventional Imaging

- Requires **hundreds of thousands to millions** of detector elements
- **High hardware cost** in some wavelengths (IR, THz, X-ray)

Single-Pixel Imaging (SPI)



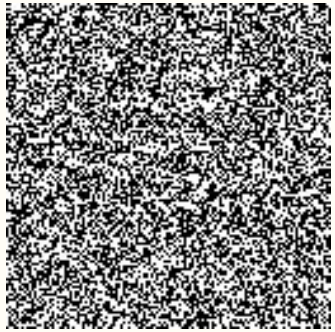
✓ **Low-cost hardware**

✓ **High sensitivity**

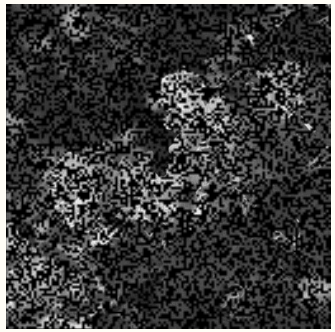
✓ **Flexible wavelength selection**

Single-Pixel Imaging

Goal: Optimize



Repeat



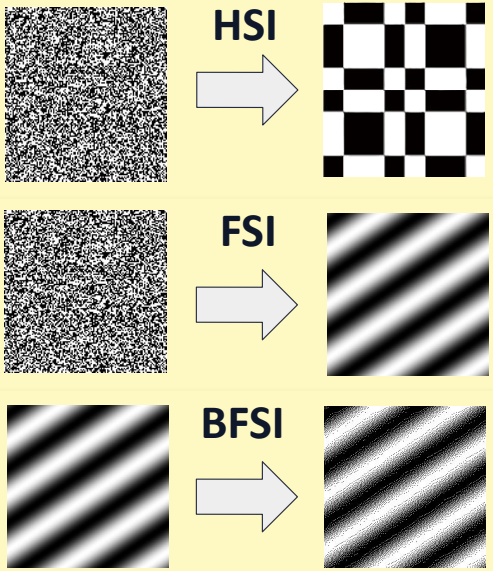
Measurement

Measurements

Reconstruct



Previous Work



Fewer measurements

Pros: Binary Patterns, Better than Random

Cons: Ineffective Subsampled Reconstruction

Pros: Effective Subsampled Reconstruction

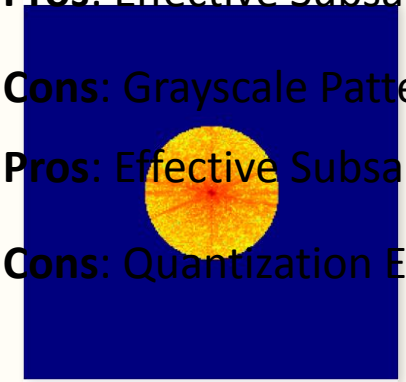
Cons: Grayscale Patterns

Pros: Effective Subsampled Reconstruction, Binary Patterns

Cons: Quantization Errors

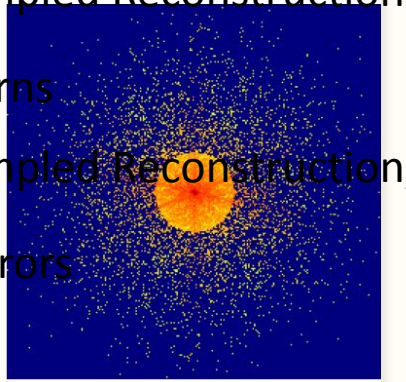
Sampling Strategies

Circular



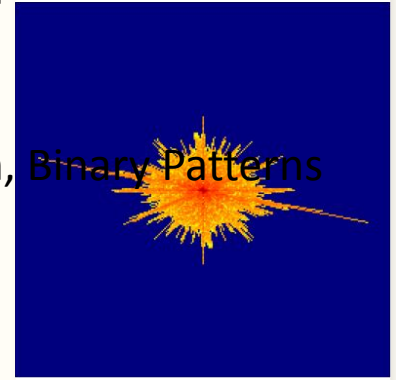
Simplest

Variable Density



High frequencies

Adaptive

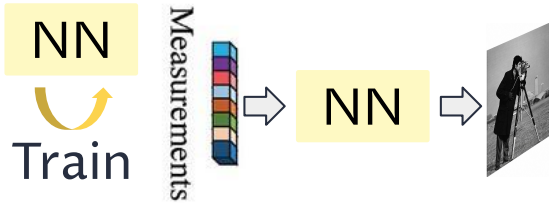


Important frequencies

Previous Work

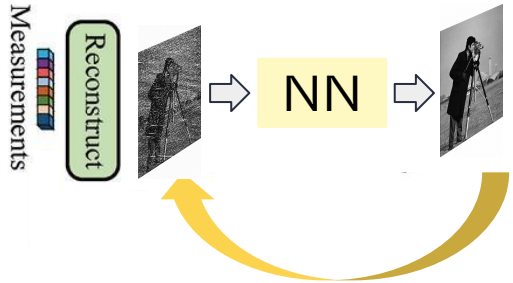
Neural Networks (NN)

Data-Driven NN



- Bias to training Data
- Slow training
- Fast

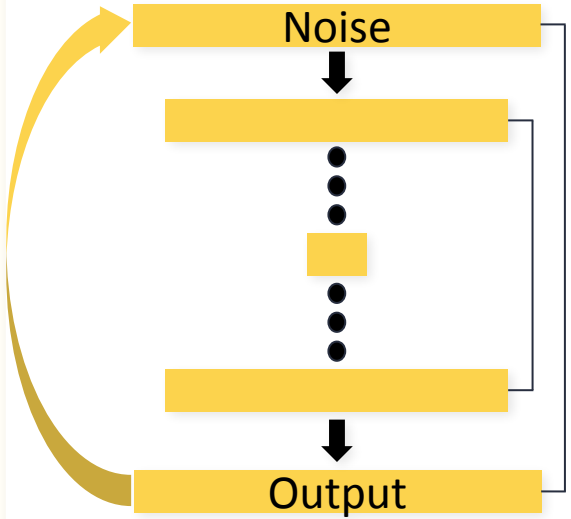
Model-Driven NN



- Unbiased to data
- Integrate SPI physics
- Slower

Deep Image Priors

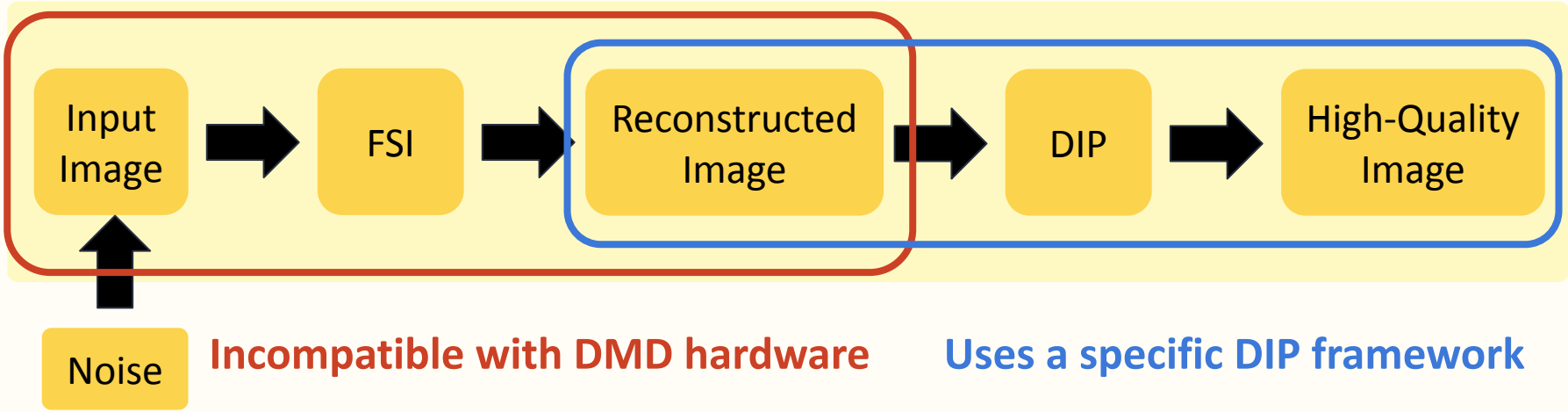
DIP



- Only uses Input Image

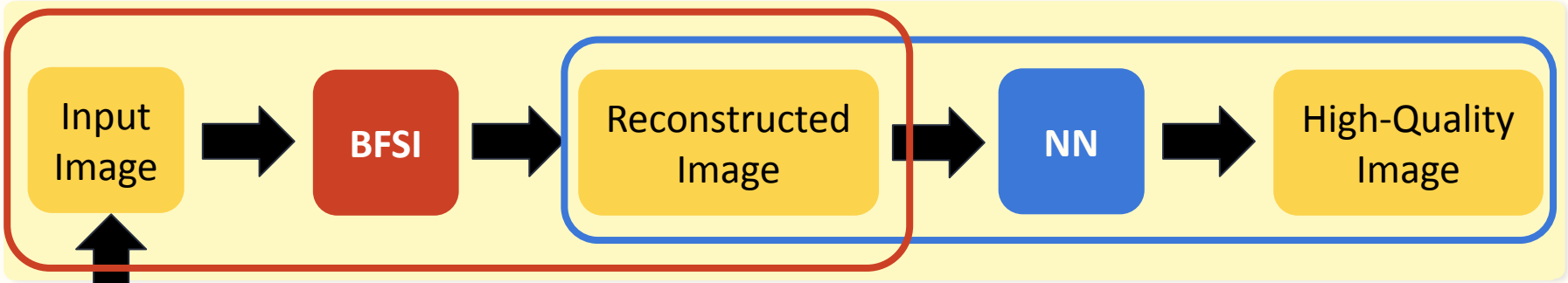
Previous Work

FSI + DIP
[5]



Our Work

BFSI + NN = High Quality Images



- Compatible Patterns
- Fast Reconstruction

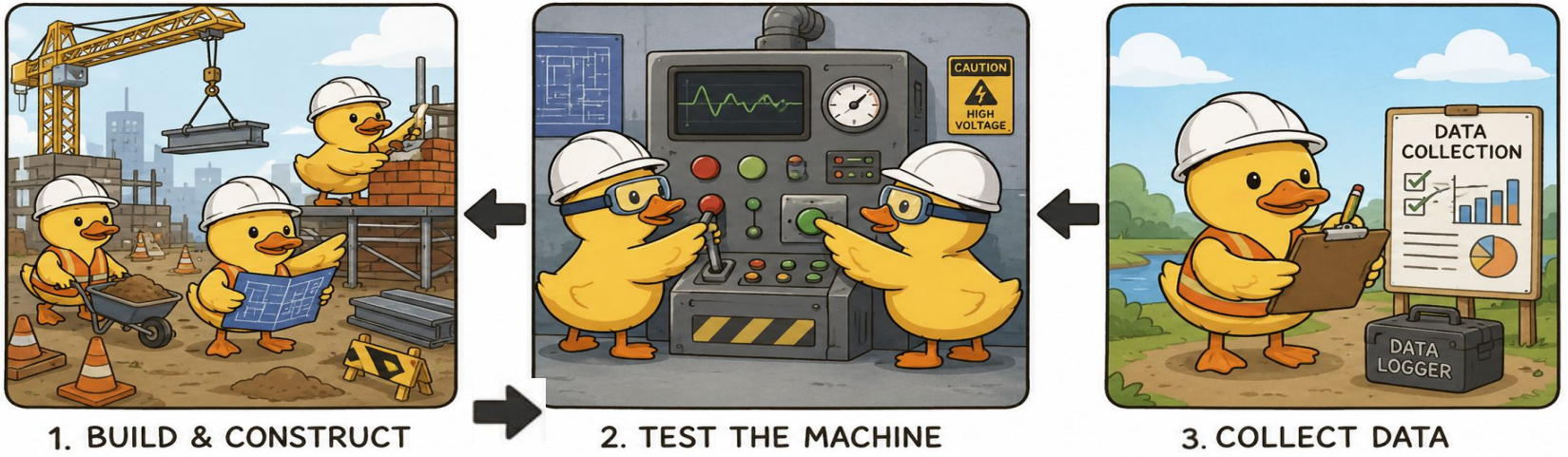
- Improved Reconstruction
- Adaptable
- Physics Informed

SECTION 02

Methods



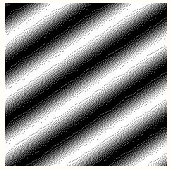
Setup


CAUTION
HIGH VOLTAGE
DATA COLLECTION
DATA LOGGER
1. BUILD & CONSTRUCT
2. TEST THE MACHINE
3. COLLECT DATA

BFSI



Original Image



BFSI





BFSI Output

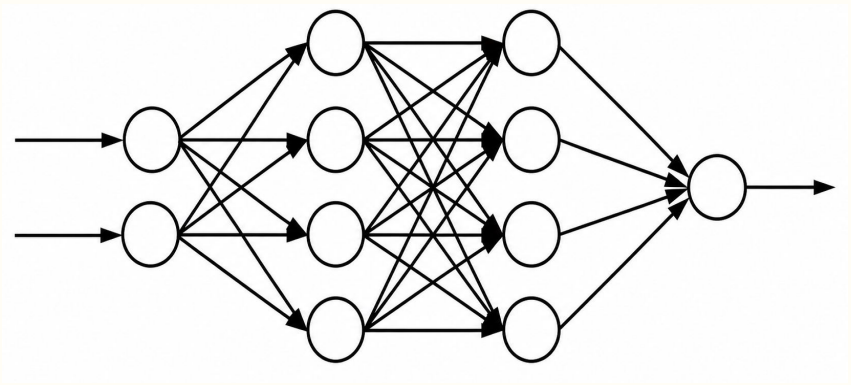
Parameters

- Sampling method
- Sampling rate
- Scale
- Upscaling method
- Noise level

Neural Networks



Neural Network Input
(BFSI Output)





Neural Network Output

Coordinate-based Networks
- SIREN

Convolution-based Networks
- ResNet
- U-Net

Loss Function
- Difference between measurements
- Difference between images
- Difference between frequencies

Parameter Tuning
- Learning rate
- Number of iterations
- Early stopping, etc

Data

Key metrics:

- Peak signal-to-noise ratio (PSNR)
- Structural Similarity (SSIM)
- Root Mean Squared Error (RMSE)
- # Iterations

Success

- **Baseline** NN output **>** NN input
- **Preferred** NN output **>>>** NN Input



[3, 4]

SECTION 03

Results

SECTION 03 › Results

Comparison with BFSI

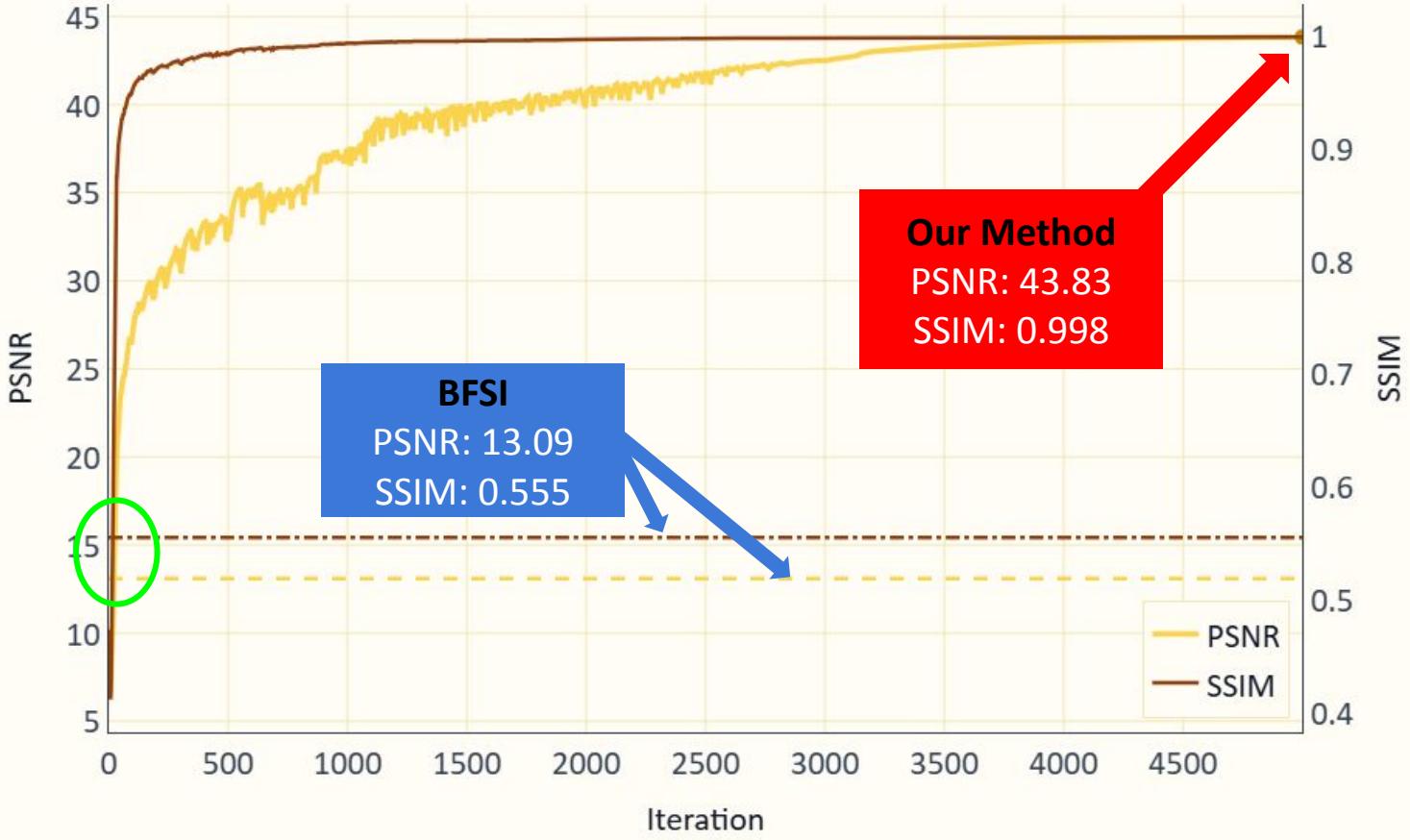
Summary (Scale 2)

Image	psnr_bfsi	avg_psnr	ssim_bfsi	avg_ssim	Noise
Cameraman	21.02	32.79	0.666	0.928	0
Jelly Beans	16.81	22.07	0.697	0.961	
Resolution Chart	13.09	32.57	0.555	0.982	
Tree	19.17	25.91	0.807	0.925	
Cameraman	14.42	18.22	0.194	0.367	0.2
Jelly Beans	11.54	12.51	0.176	0.570	
Resolution Chart	10.92	18.19	0.327	0.678	
Tree	14.20	18.23	0.307	0.587	

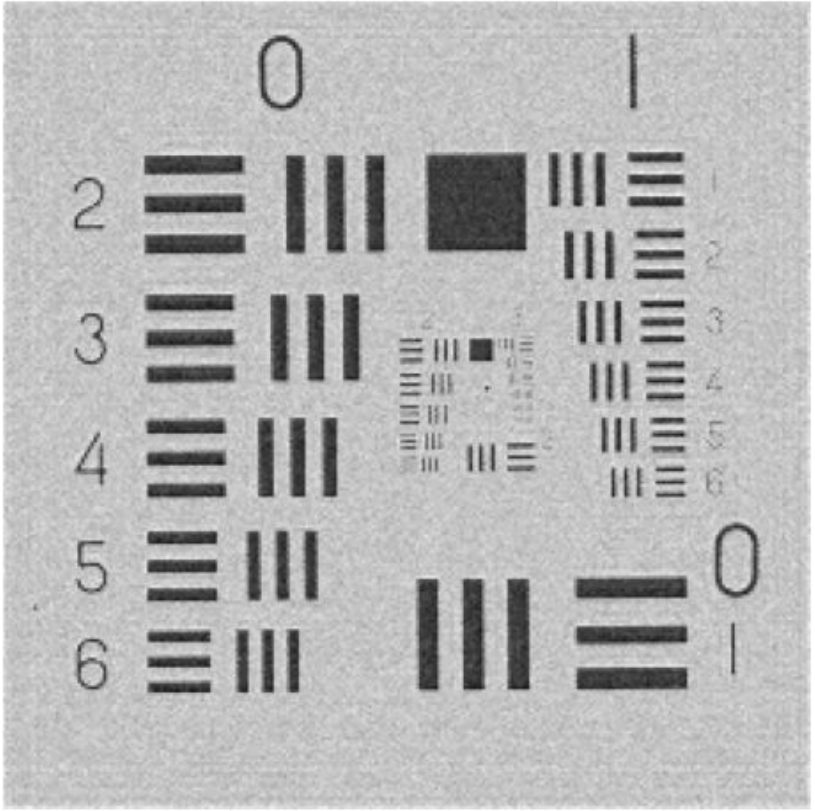
Our method consistently outperforms conventional BFSI.



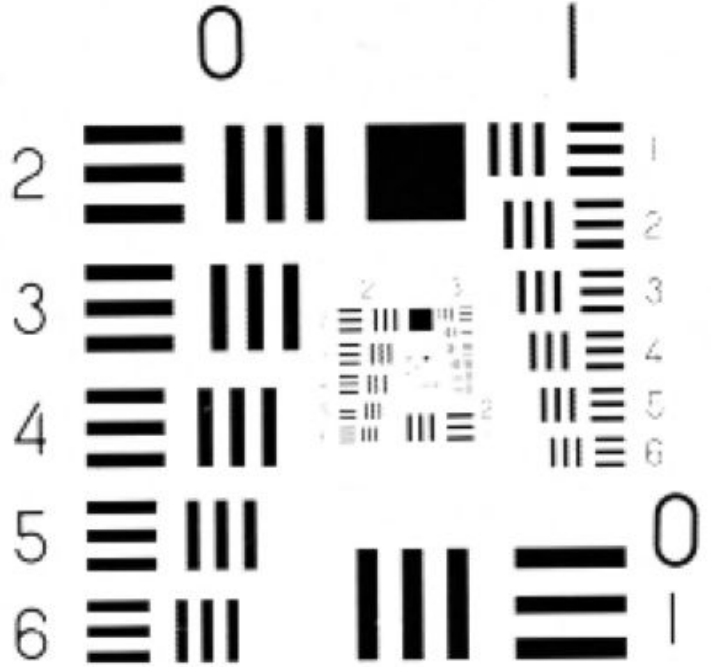
Resolution chart, scale 1, SIREN



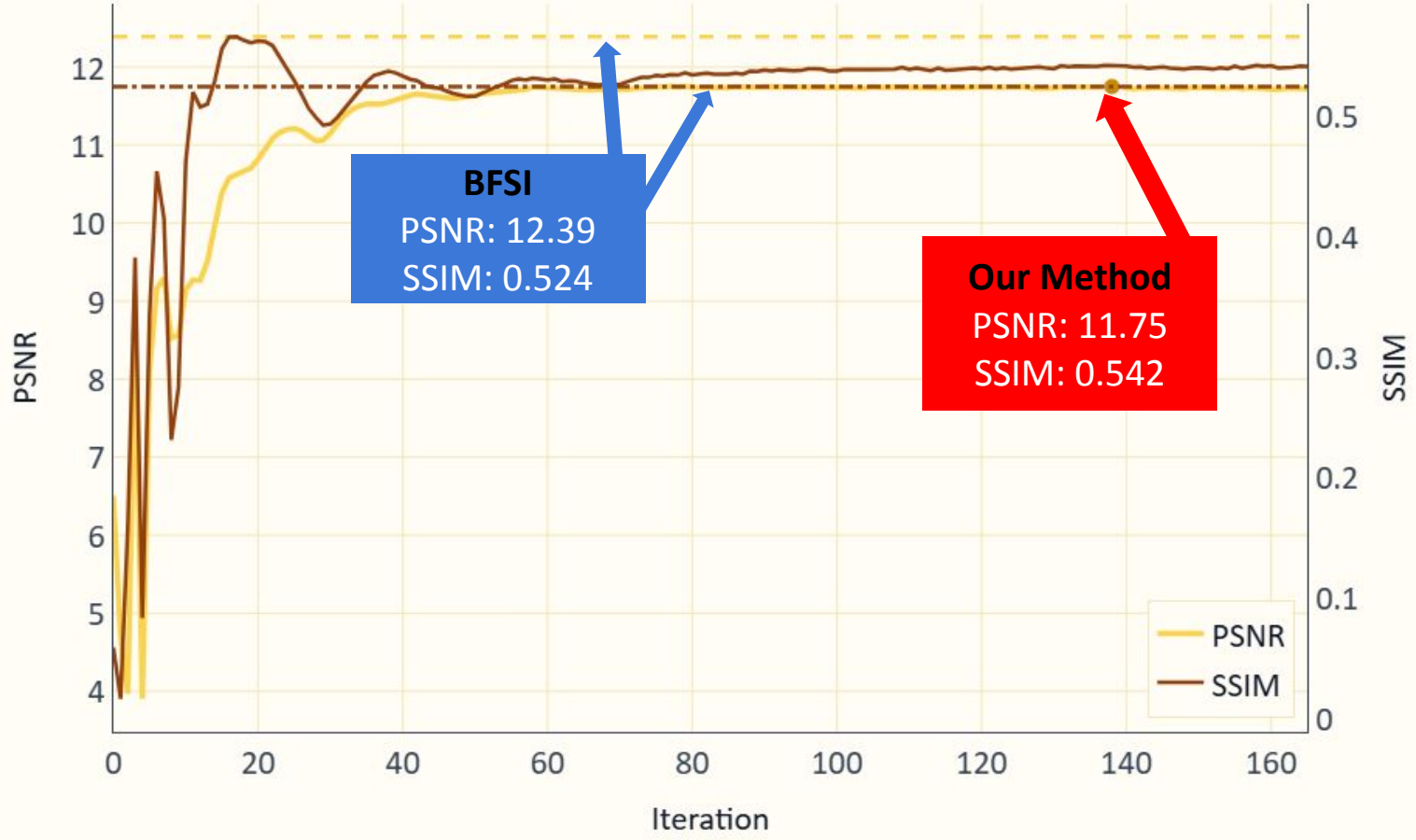
NN input
PSNR: 13.0860
SSIM: 0.5553
RMSE: 0.2217



NN output (siren)
PSNR: 43.8339
SSIM: 0.9981
RMSE: 0.0064



Resolution chart, scale 8, ResNet, bicubic

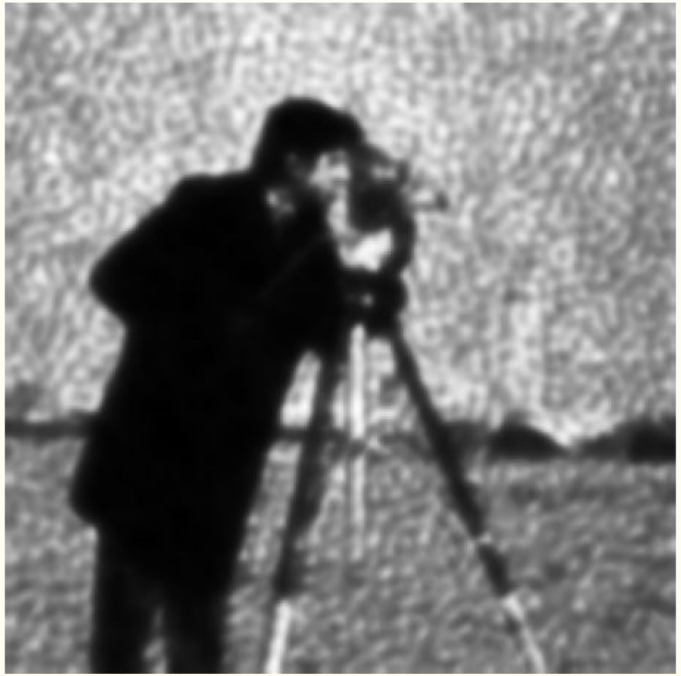


Our Method 10% vs BFSI 100% (PSNR/SSIM)

	Resolution Chart	Tree	Jelly Beans	Cameraman	Average PSNR/SSIM	Noise Level
BFSI	21.5 / 0.47	26.5 / 0.77	28.5 / 0.62	27.1 / 0.65	25.9 / 0.63	0
SIREN	23.6 / 0.92	20.6 / 0.87	22.2 / 0.97	25.5 / 0.80	22.97 / 0.87	
U-Net	12.4 / 0.82	20.3 / 0.77	22.4 / 0.97	25.6 / 0.80	20.15 / 0.84	
ResNet	22.6 / 0.90	24.7 / 0.76	31.0 / 0.94	24.4 / 0.75	25.69 / 0.84	
BFSI	10.9 / 0.33	14.2 / 0.31	11.5 / 0.18	14.4 / 0.19	12.8 / 0.25	0.2
SIREN	18.2 / 0.69	18.9 / 0.61	11.8 / 0.67	15.7 / 0.53	16.1 / 0.62	
U-Net	12.0 / 0.52	18.2 / 0.55	16.4 / 0.76	20.12 / 0.42	16.7 / 0.56	
ResNet	17.5 / 0.65	20.6 / 0.57	22.0 / 0.85	20.49 / 0.35	20.1 / 0.61	

Cameraman, scale 1, U-Net, Noise 0.2

Our Method 10% Sampling Rate



BFSI 100% Sampling Rate



SECTION 03 › Results

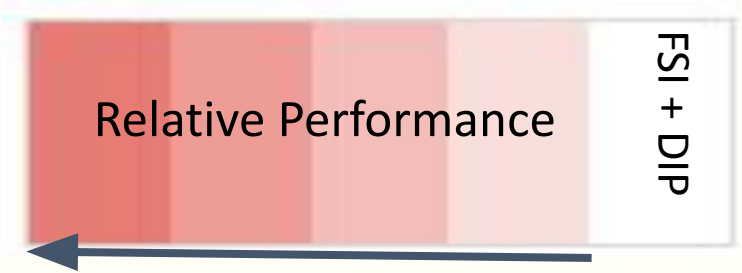
Comparison with FSI + DIP

Summary (Variable density, 10% sampling rate)

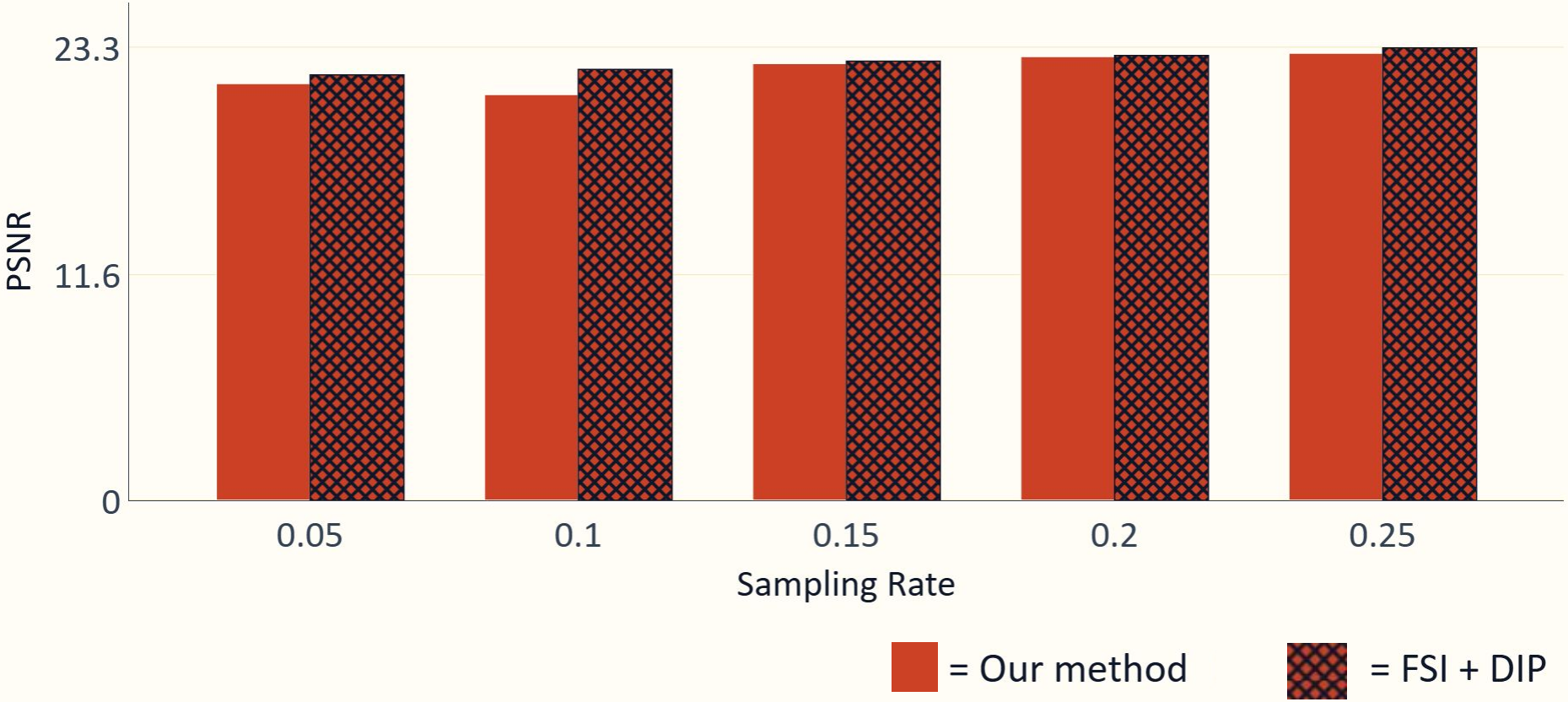
image	fsi_dip_psnr	avg_psnr	fsi_dip_ssim	avg_ssim	fsi_dip_rmse	avg_rmse
Cameraman	22.16	21.38	0.612	0.483	0.078	0.086
Fishing Boat	23.66	22.15	0.582	0.523	0.066	0.078
Man	24.27	21.25	0.573	0.513	0.061	0.087

FSI+DIP
Tuned a NN to each image

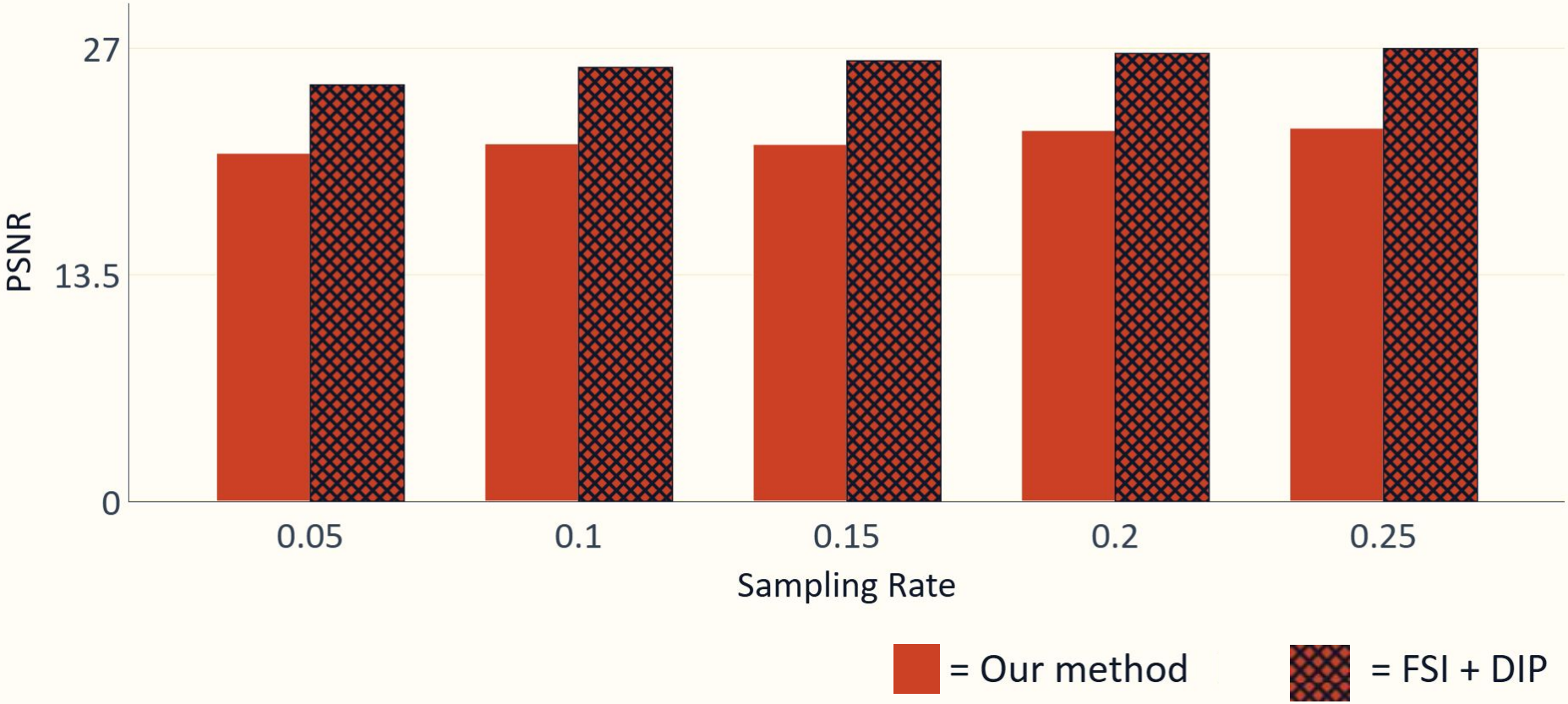
Our Method
Fixed Parameters for NN



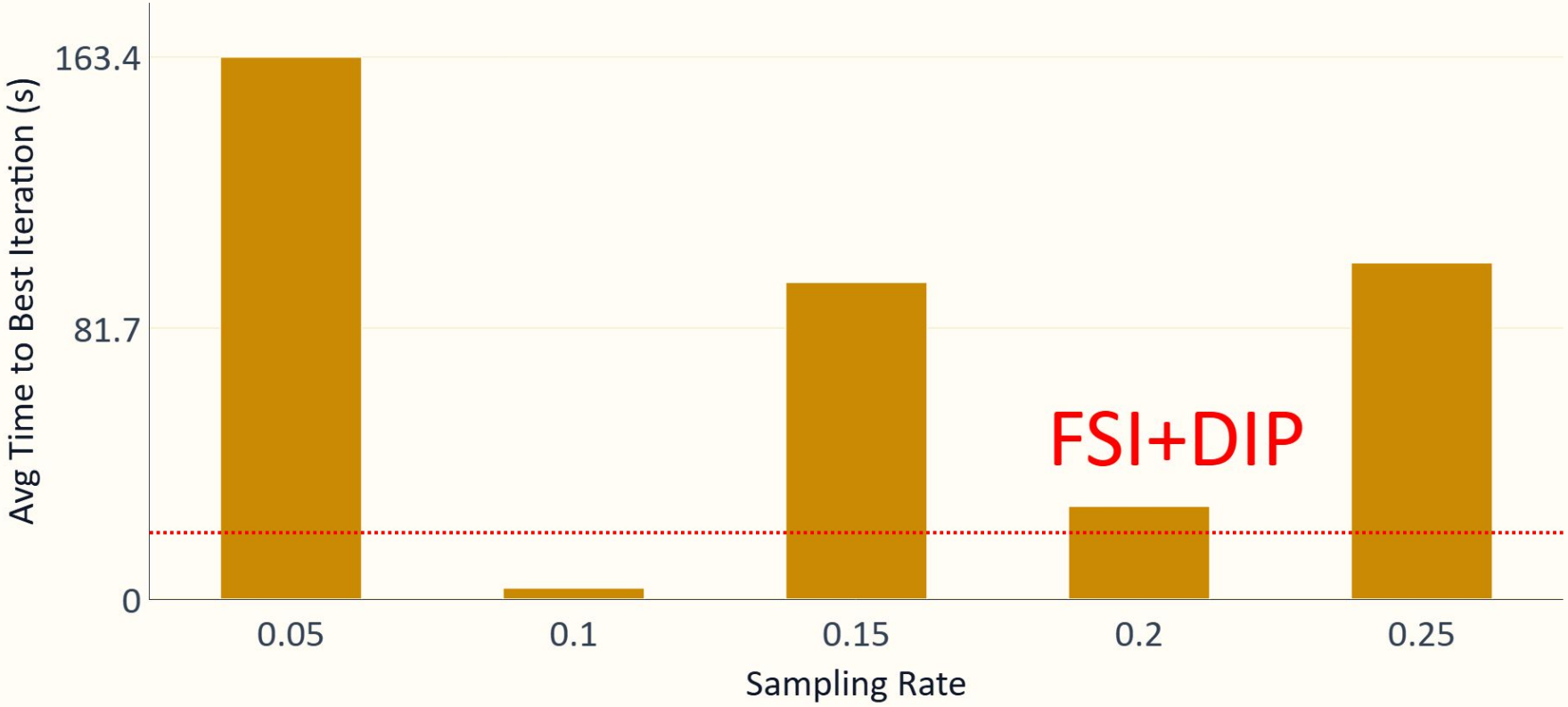
PSNR by sampling rate (VD, ResNet, Cameraman)



PSNR by sampling rate (Adaptive, ResNet, Man)



Estimated Avg Time to Best Iteration by SR (Tesla P4 GPU)



SECTION 04

Conclusion & Discussion

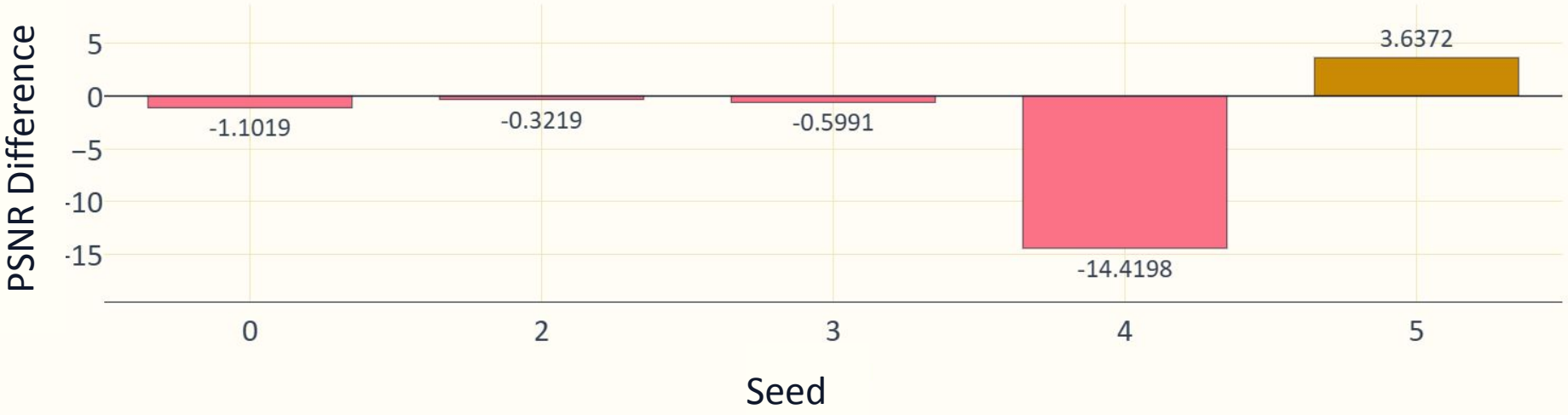
Conclusion

BFSI + NN = High Quality Images

- **Improved quality**
  Higher PSNR and SSIM than BFSI
- **Sampling efficiency**
  Higher quality with fewer measurements
- **Practical advantage**
  BFSI patterns remain compatible with high-speed hardware
- **Competitive with FSI + DIP**

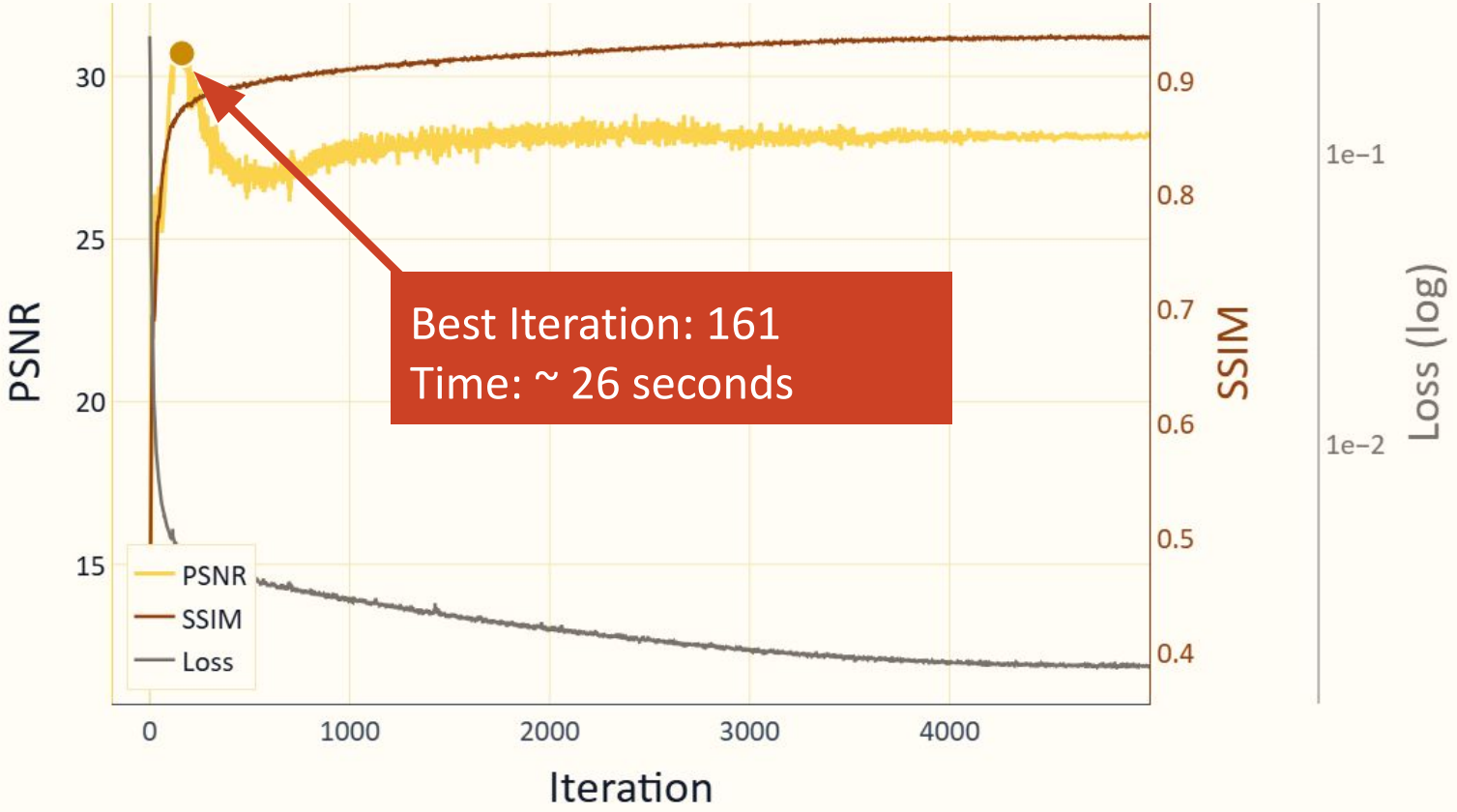
Discussion - Sensitivity to network initialization

PSNR Difference vs. BFSI Baseline Across Seeds (UNet, Resolution Chart)

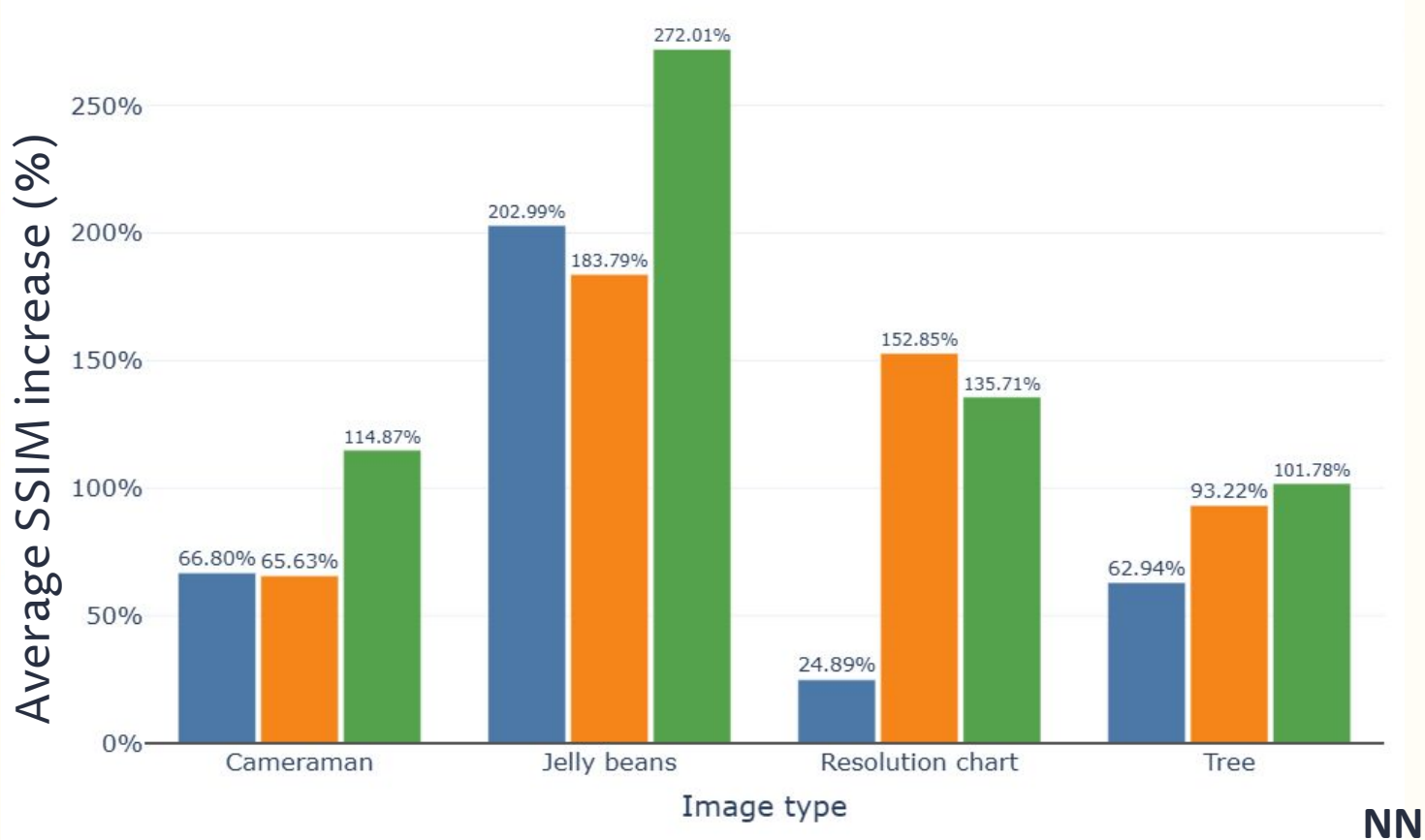


Performance varies across initializations.

Discussion - Early peak phenomenon



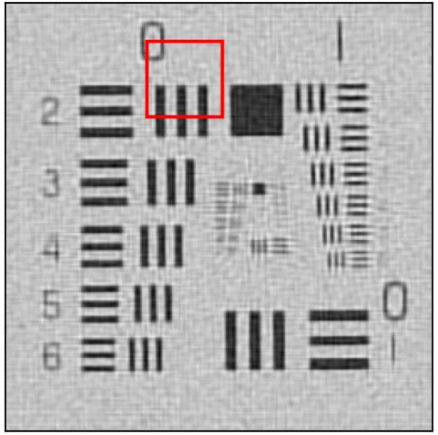
Discussion - Neural Network Comparison

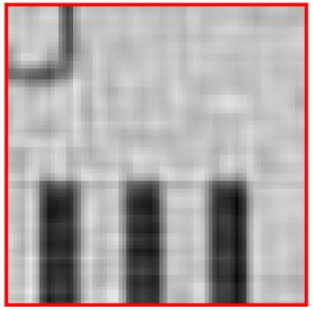


NN choice depends

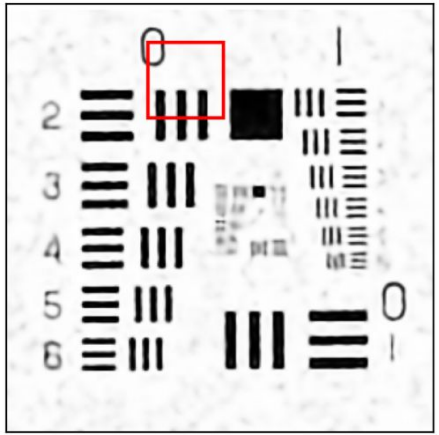
Discussion - Neural Network Comparison

BFSI



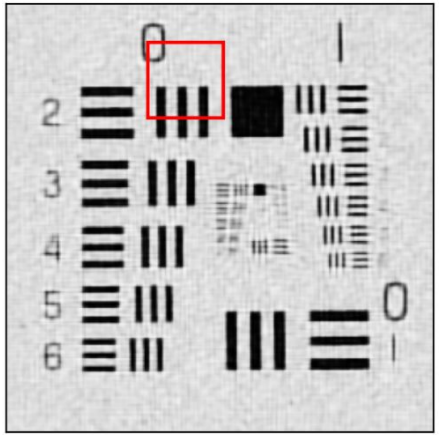


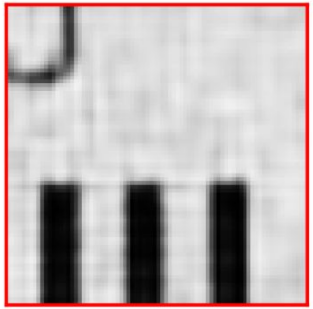
SIREN



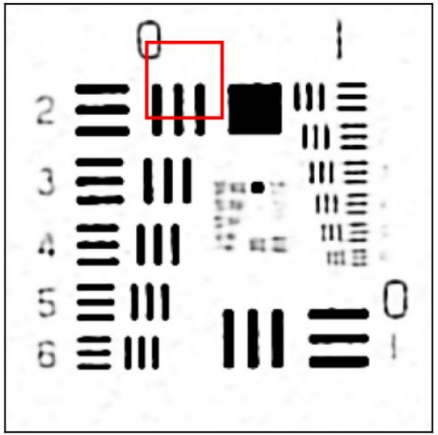


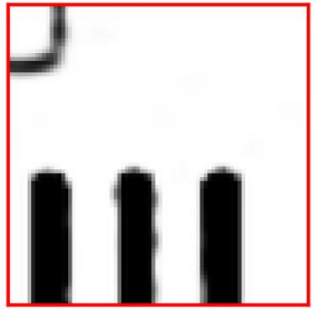
U-Net





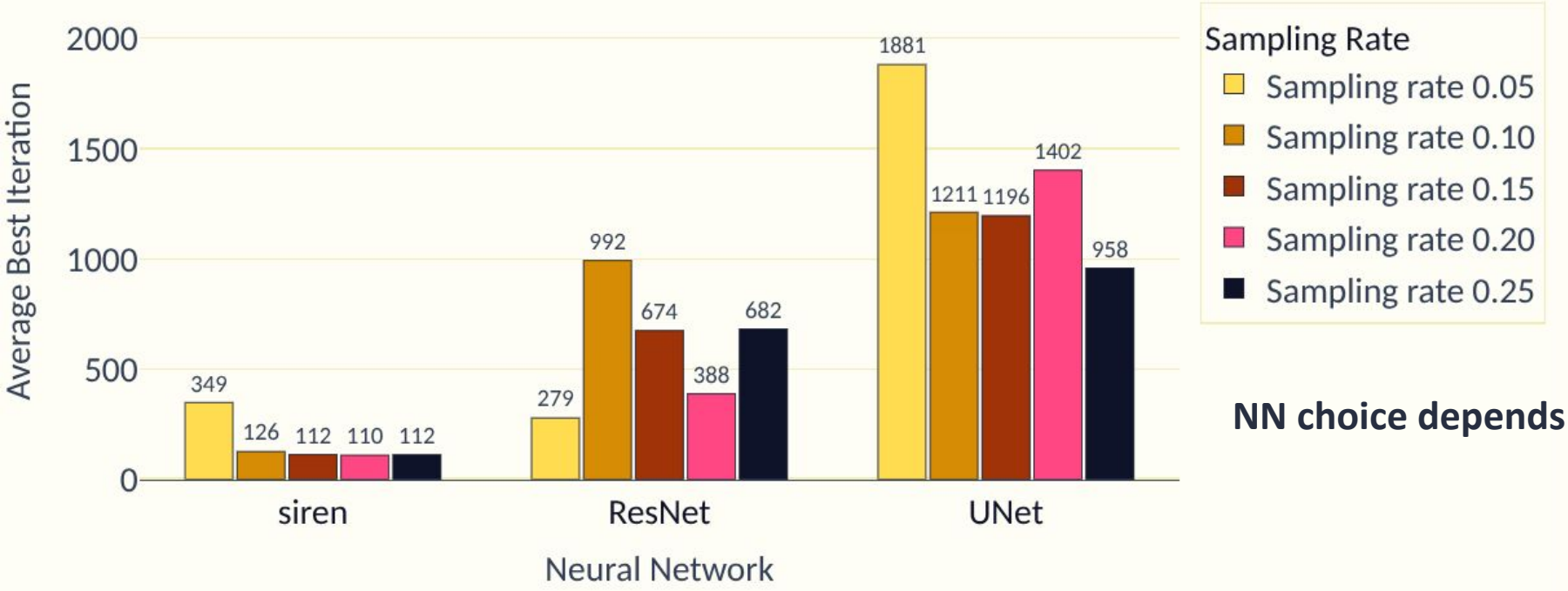
ResNet





Discussion - Neural Network Comparison

Average Best Iteration by Network and Sampling Rate



NN choice depends

Future work

- Address initialization

Thank You

Questions & Discussion

References

[1] S. C. Tudosie, A. Denker, Z. Kereta, and S. Arridge, "Learning binary sampling patterns for single-pixel imaging using bilevel optimisation," *arXiv*, arXiv:2508.19068, Aug. 2025.

[2] K. Song et al., "Advances and challenges of single-pixel imaging based on deep learning," *Laser & Photonics Reviews*, vol. 19, no. 7, Art. no. 2401397, 2025.

[3] Standard Image Database [Kanagawa Institute of Technology, Signal Processing Application Laboratory]

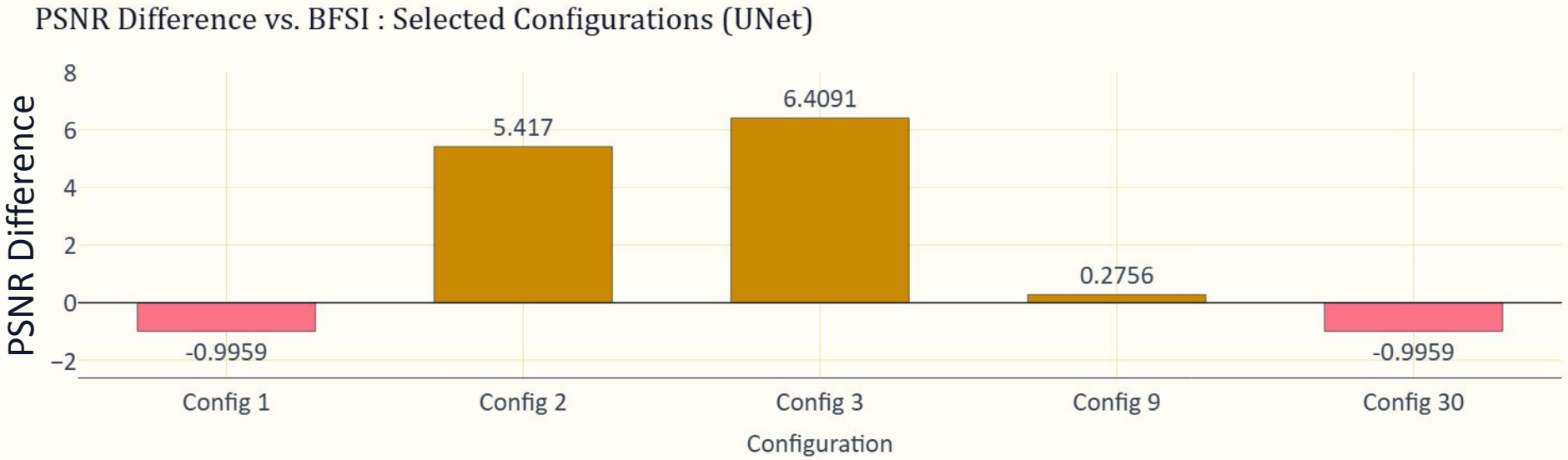
[4] General-purpose images — skimage 0.26.1rc0.dev0 documentation

[5] Wei Lun Tey, Sing Yee Chua, Mau-Luen Tham, and Yeong-Nan Phua. 2025. Fourier Single Pixel Imaging Reconstruction using Deep Image Prior. In Proceedings of the 2024 7th Artificial Intelligence and Cloud Computing Conference (AICCC '24). Association for Computing Machinery, New York, NY, USA, 215–221. https://doi.org/10.1145/3719384.3719415

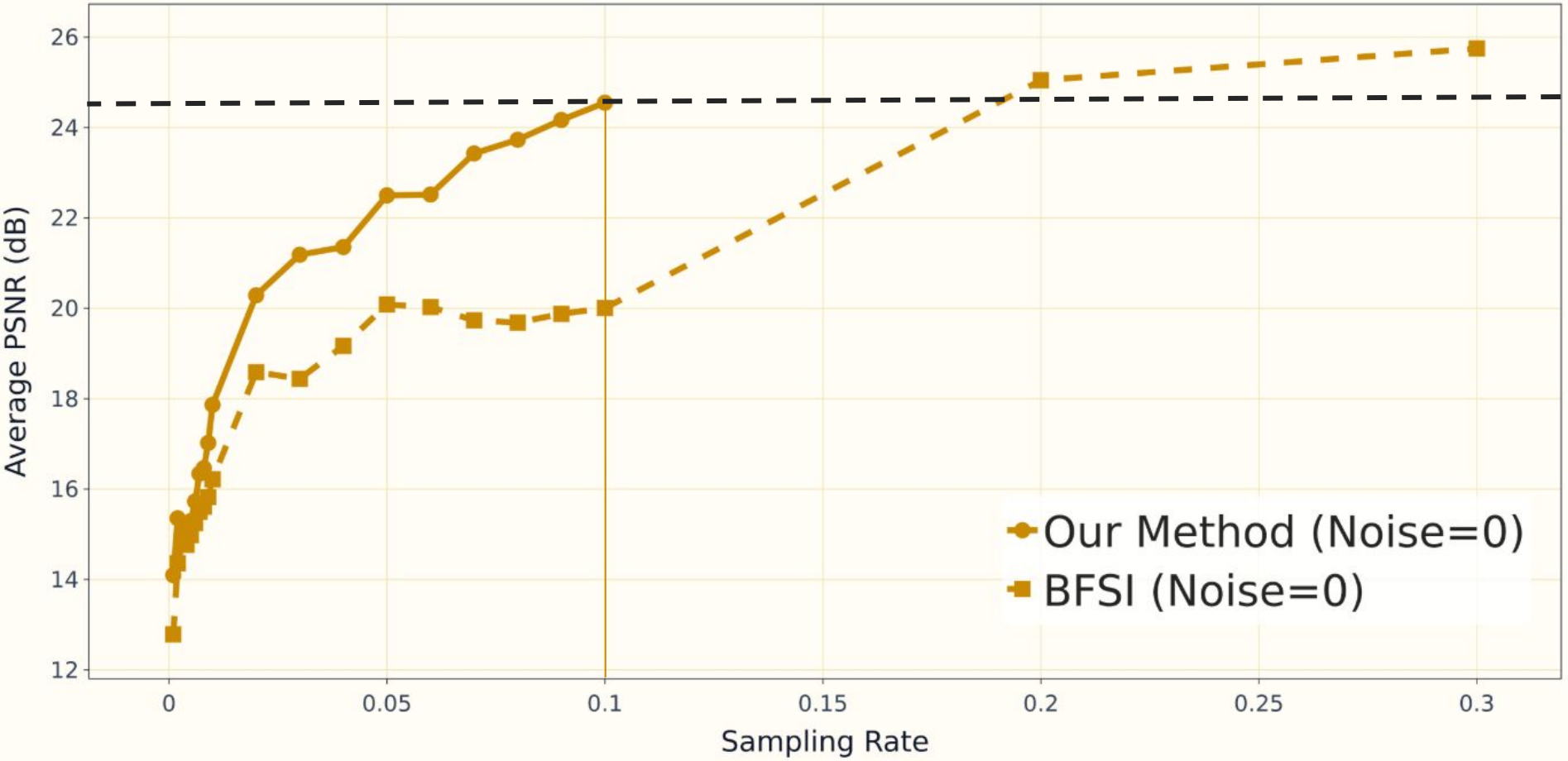
[6] Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky. "Deep image prior." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

Discussion - Sensitivity to network initialization

PSNR Difference vs. BFSI Baseline Across Seeds (UNet)



ResNet Average PSNR vs Sampling Rate



Testing Across Percent of Patterns (U-Net)



NN input
PSNR: 12.3871
SSIM: 0.5241
RMSE: 0.2402



NN output (ResNet)
PSNR: 11.7499
SSIM: 0.5417
RMSE: 0.2585



SECTION 03 › Results

With Noise

Testing With Noise (U-Net)

(PSNR/SSIM) ➡

Ground Truth



(17.67 / 0.448)



(20.88 / 0.685)



(20.27 / 0.768)



Noise Level

0

(17.97 / 0.443)



1 %

(19.35 / 0.532)



5 %

(18.24 / 0.545)

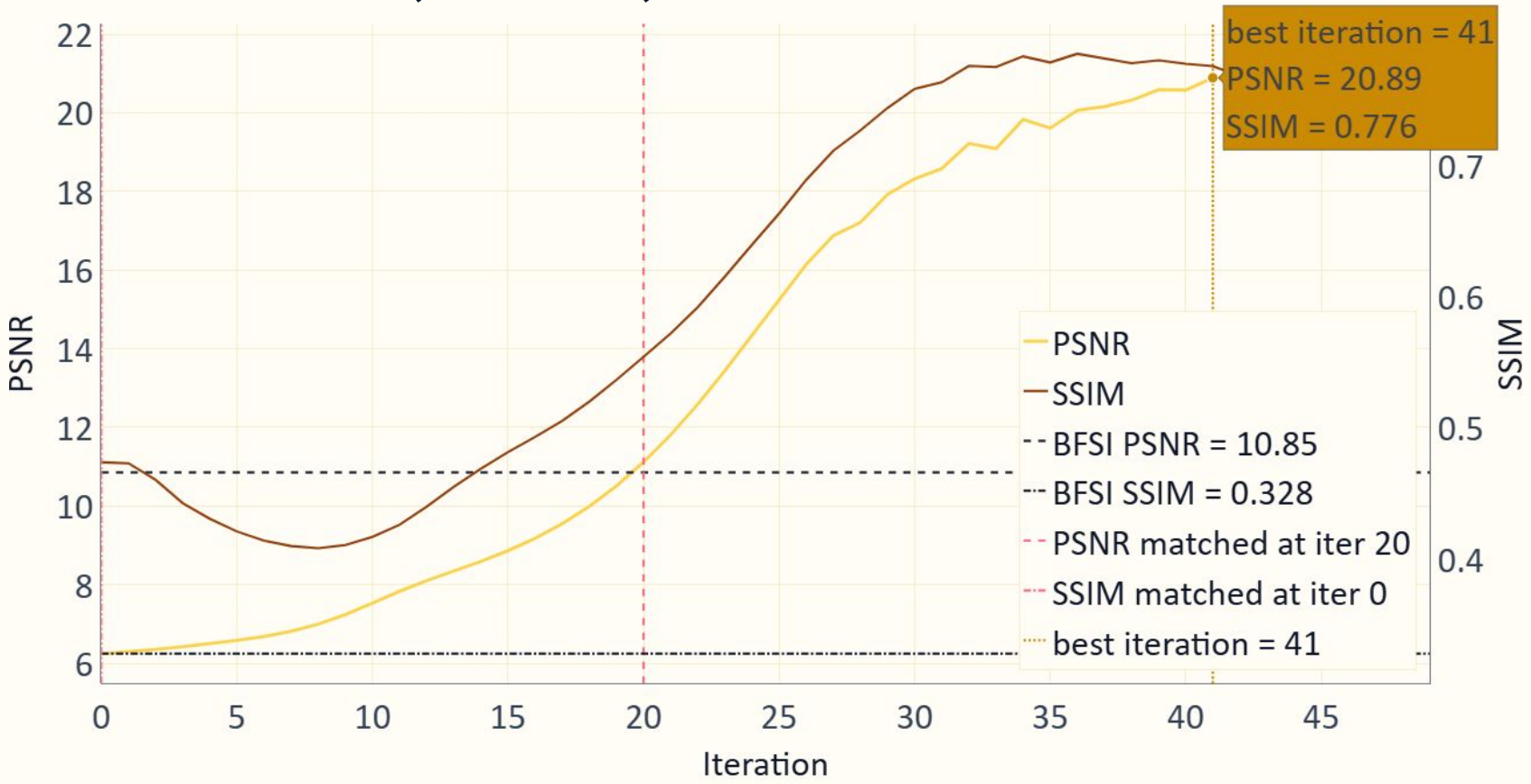


10%

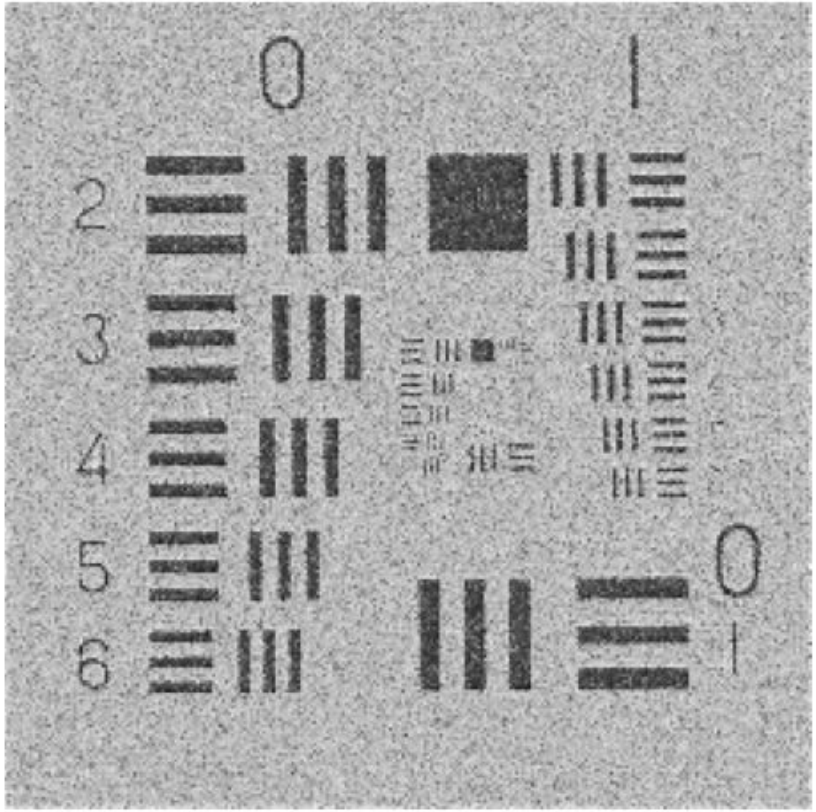
0.2

SLIDESMANIA.COM

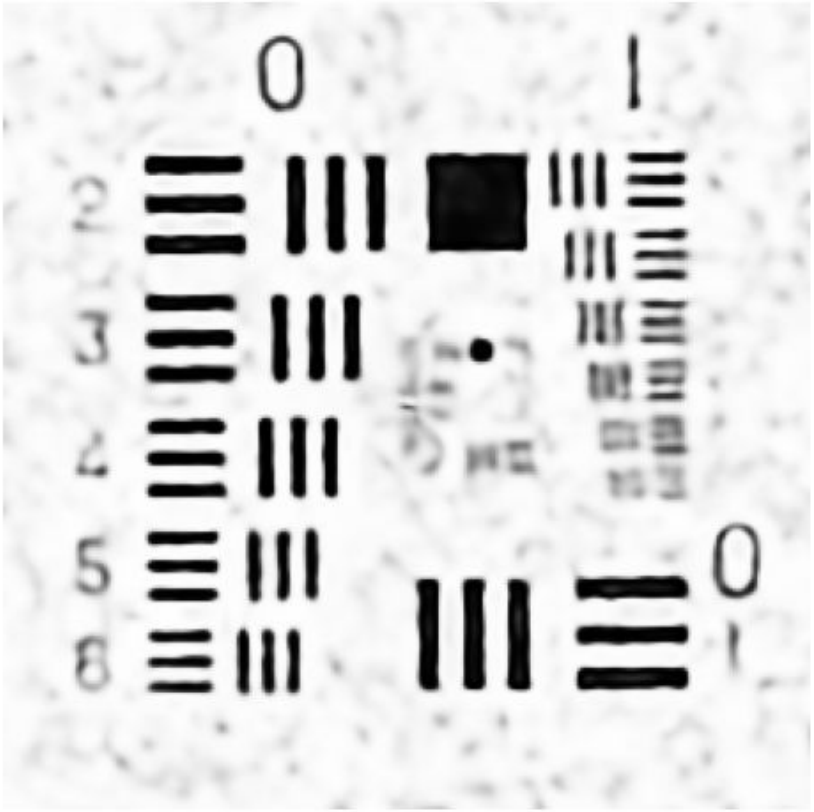
Resolution chart, scale=1, SIREN



NN input
PSNR: 10.8544
SSIM: 0.3278
RMSE: 0.2866

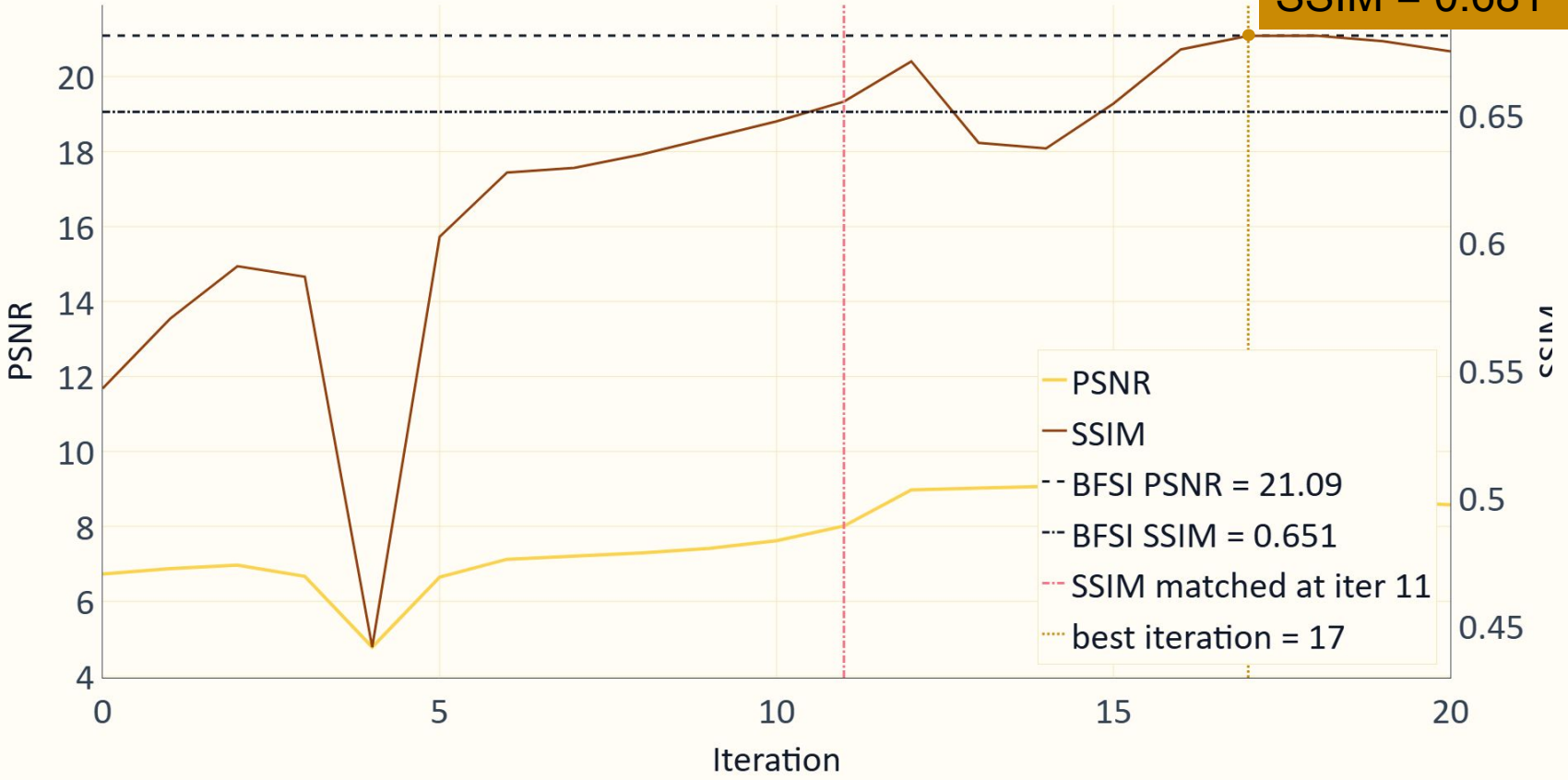


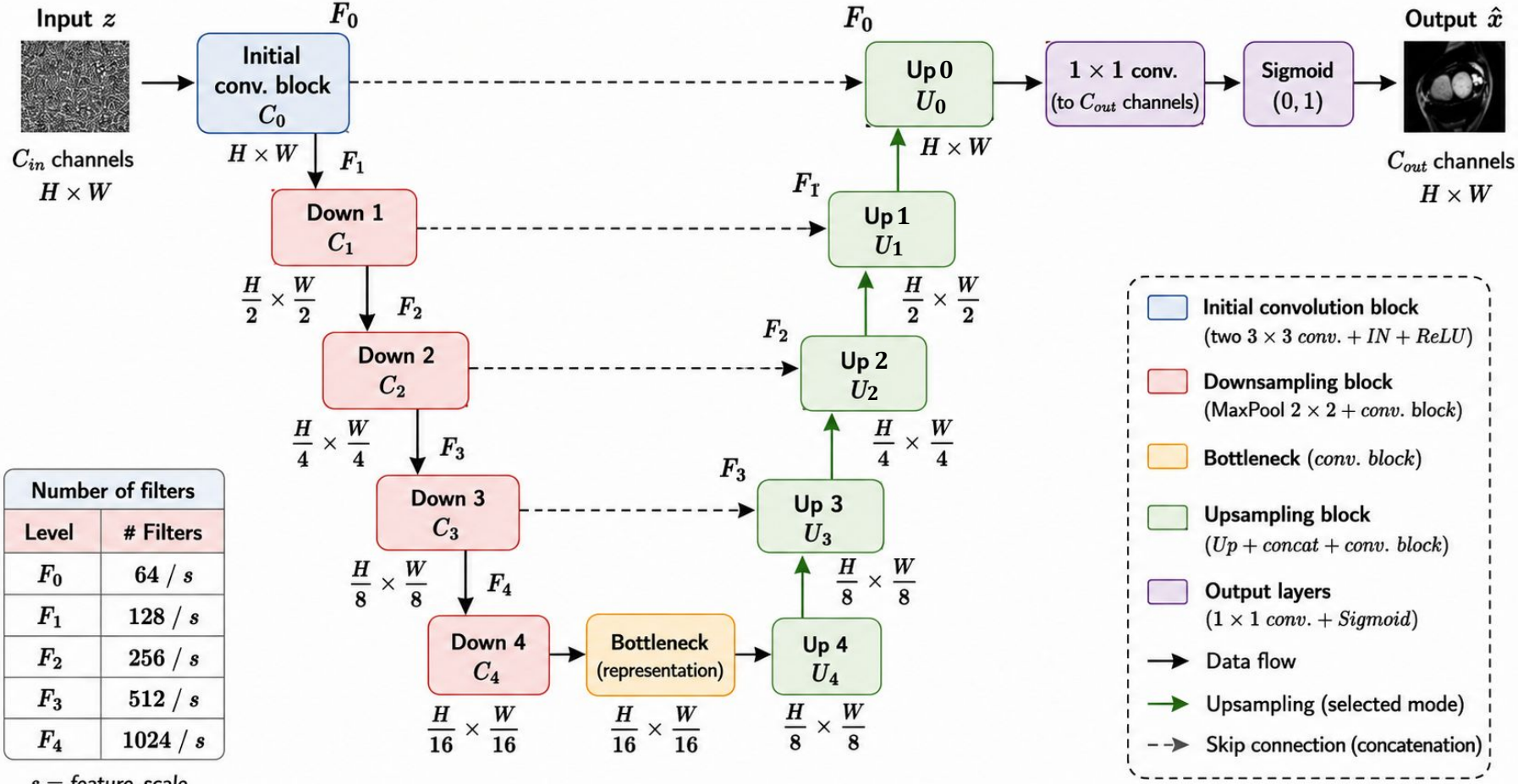
NN output (siren)
PSNR: 20.8930
SSIM: 0.7758
RMSE: 0.0902



Jelly Beans, scale=4, UNet, bilinear

Best iteration = 17
PSNR = 9.0163
SSIM = 0.681

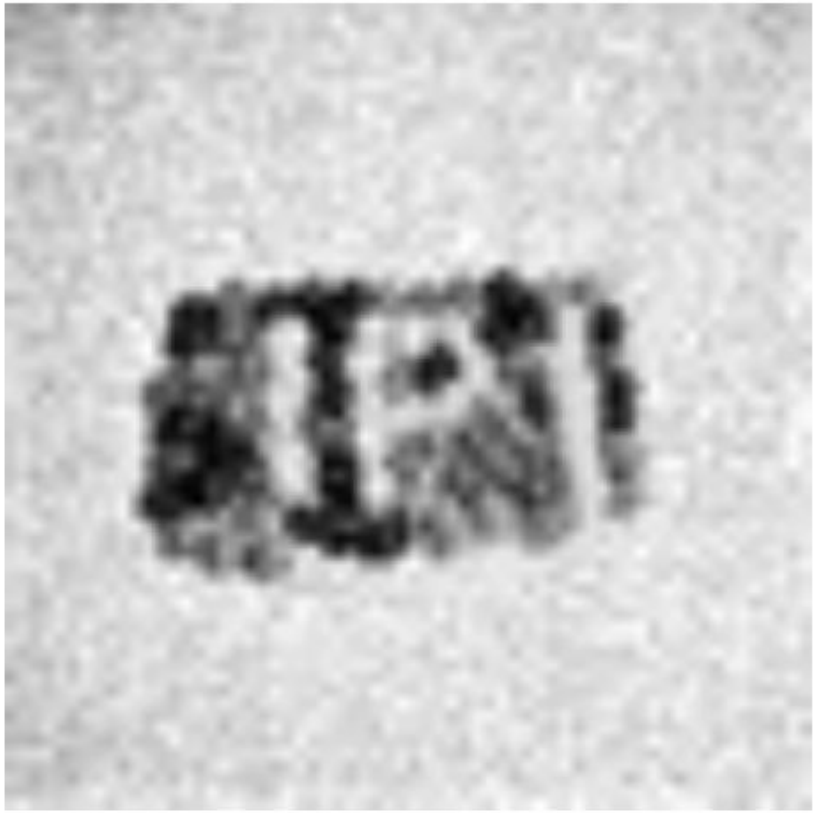




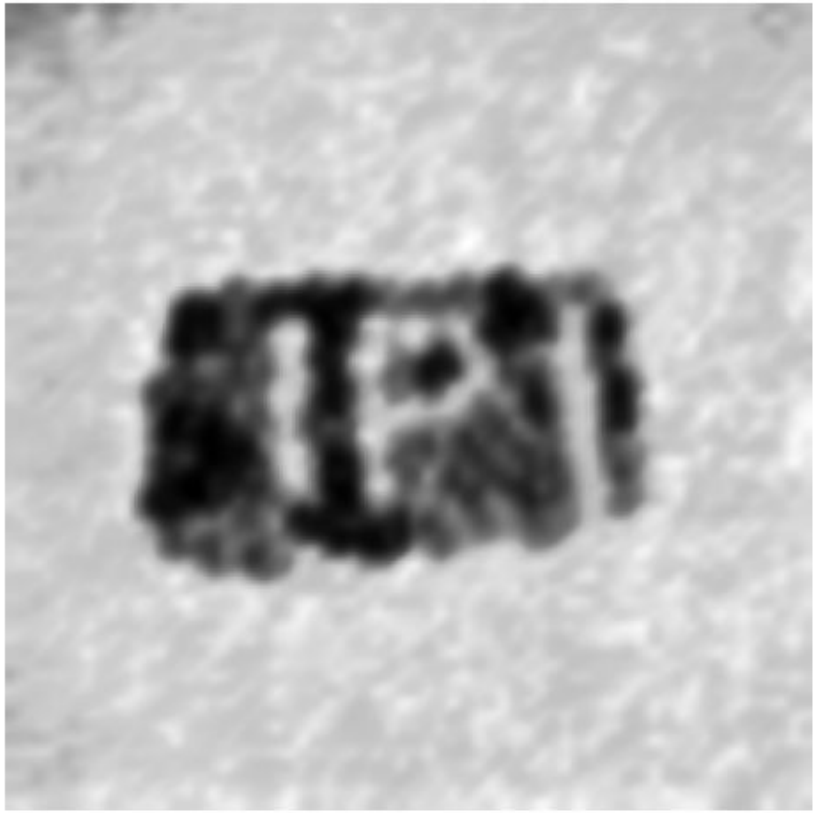
Number of filters	
Level	# Filters
F_0	$64 / s$
F_1	$128 / s$
F_2	$256 / s$
F_3	$512 / s$
F_4	$1024 / s$

s = feature_scale

NN input
PSNR: 21.0945
SSIM: 0.6512
RMSE: 0.0882



NN output (UNet)
PSNR: 9.0163
SSIM: 0.6810
RMSE: 0.3541



SECTION 03 › Results

Neural Network Comparison

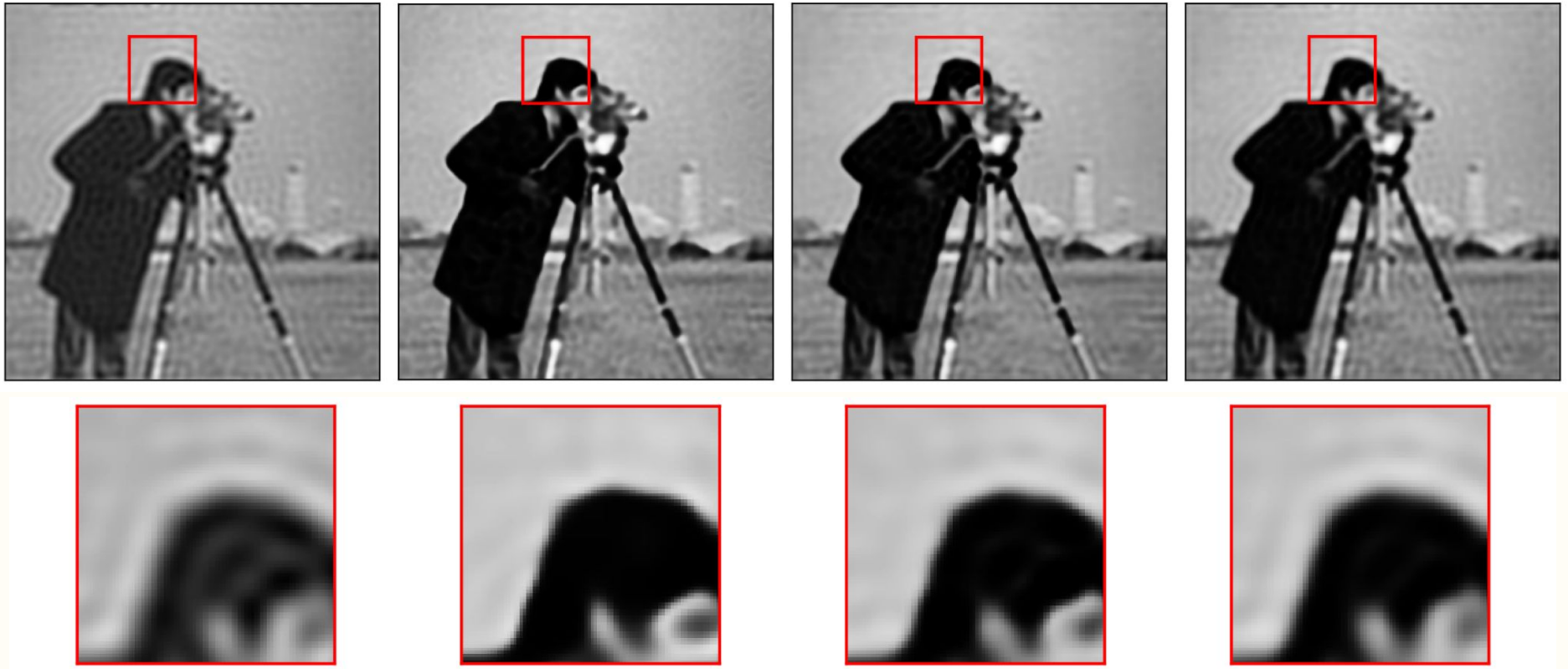
Neural network performance comparison

BFSI reconstruction at
SR = 5%, no noise, scale 1

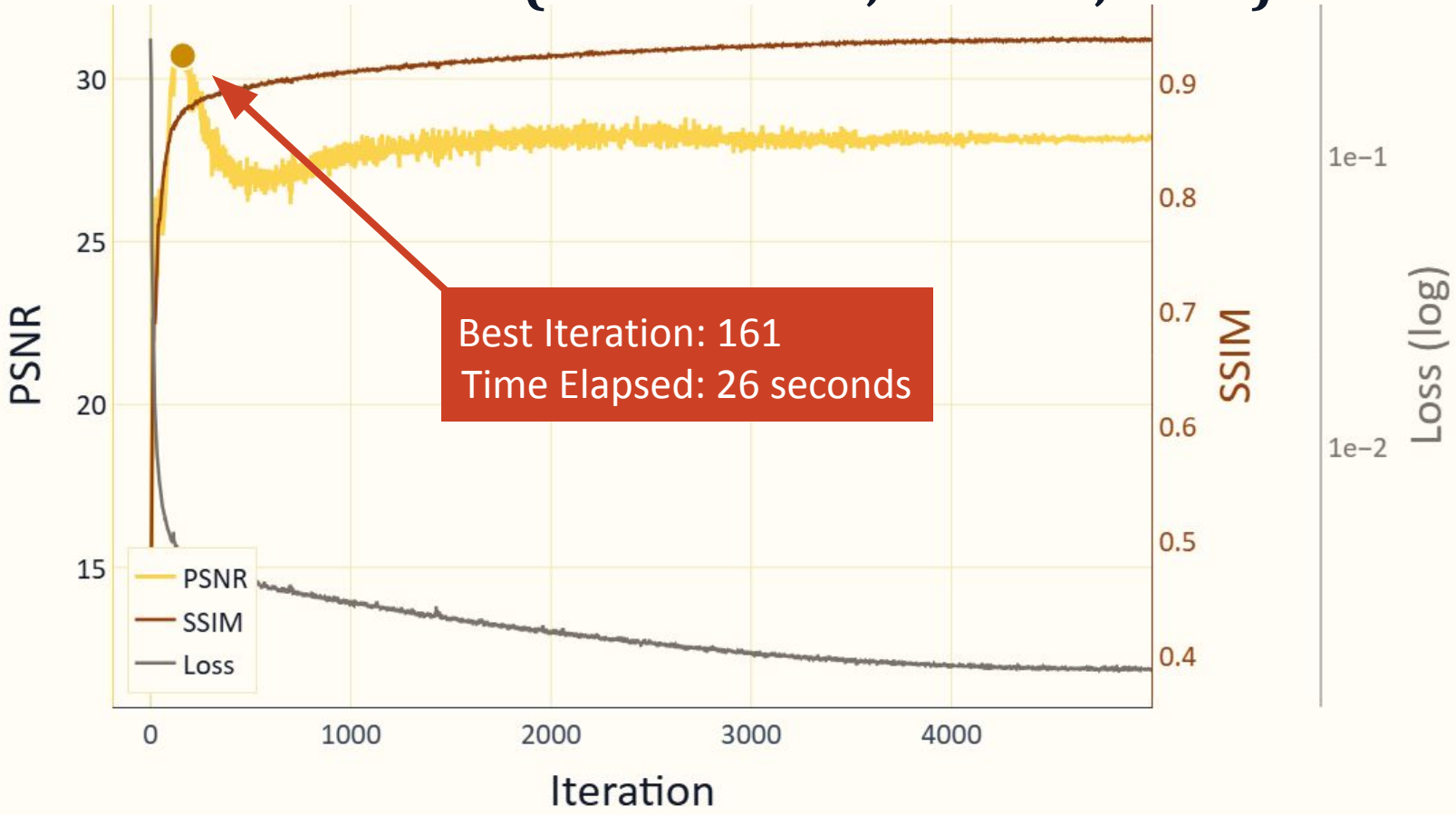
SIREN

U-Net

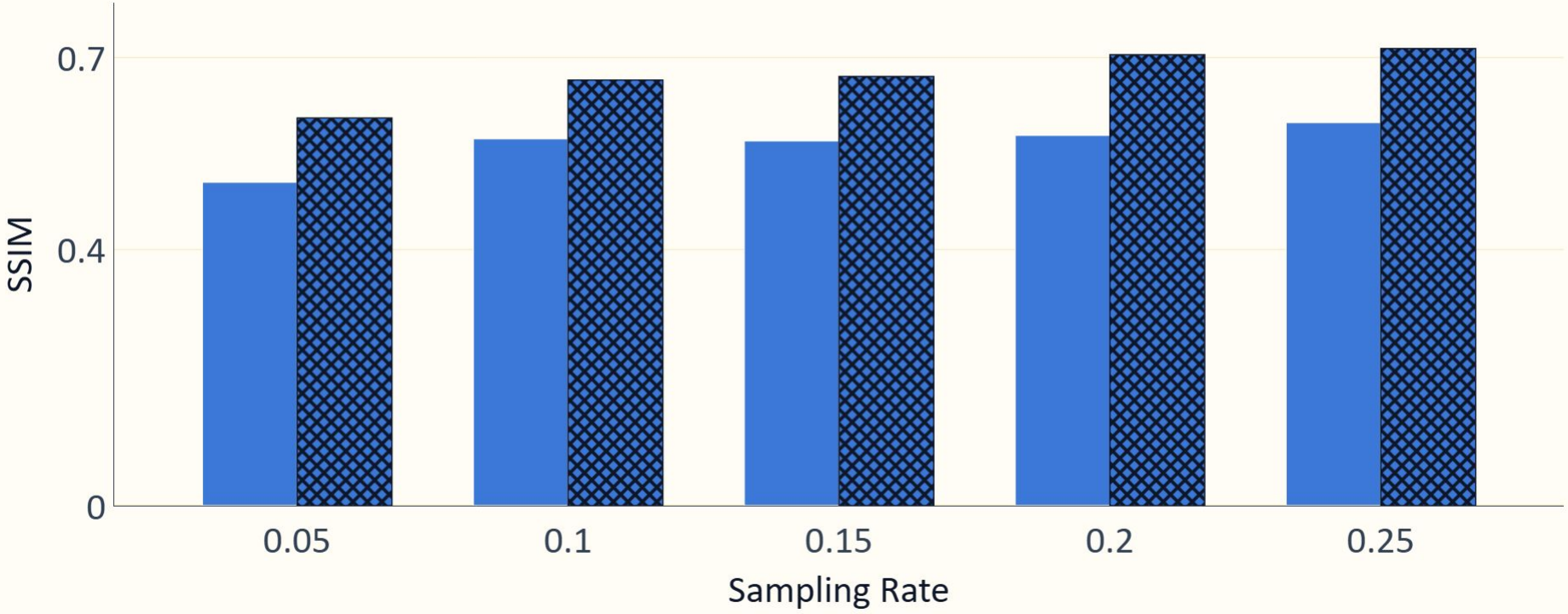
ResNet



Time to Best Iteration (Cameraman, scale=1, UNet)

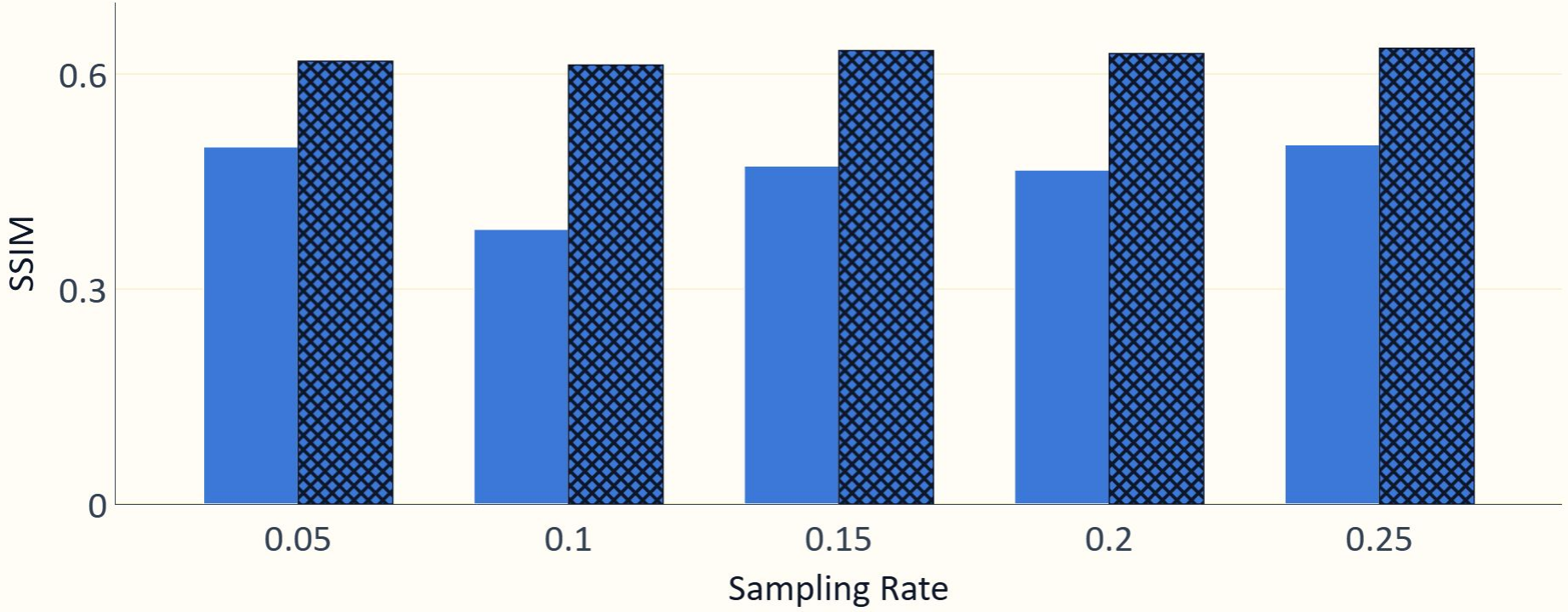


Average SSIM by SR (Adaptive, ResNet, Man)



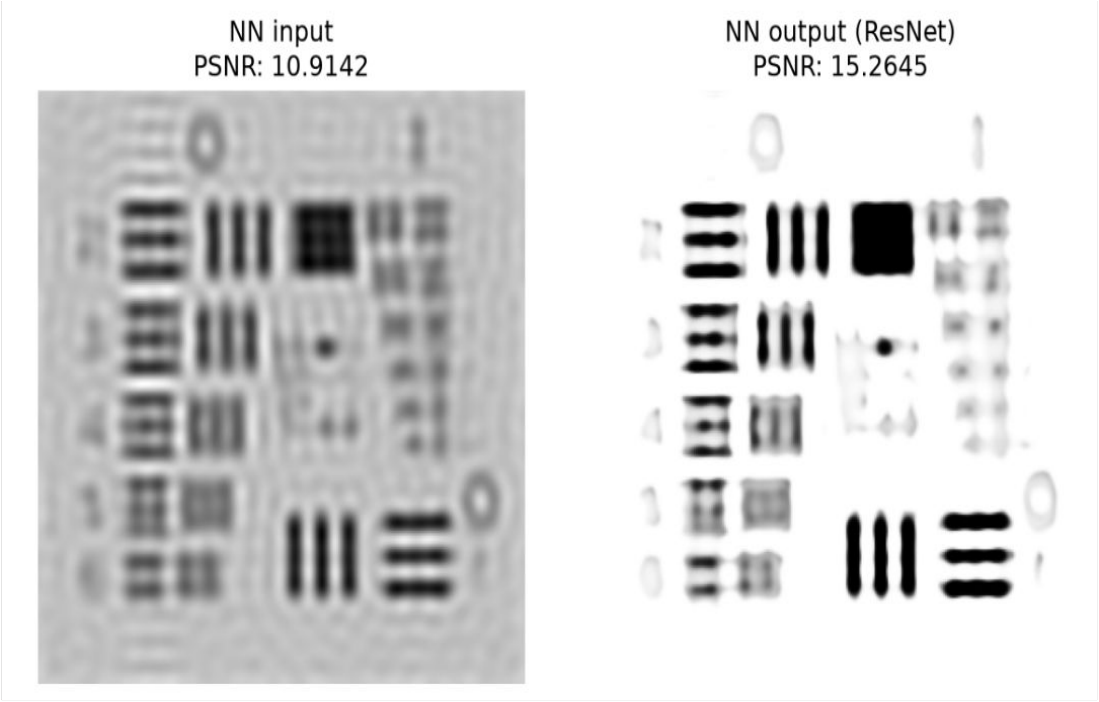
Solid = Our method Hatched = FSI + DIP

Average SSIM by SR (VD, ResNet, Cameraman)



Solid = Our method Hatched = FSI + DIP

Results



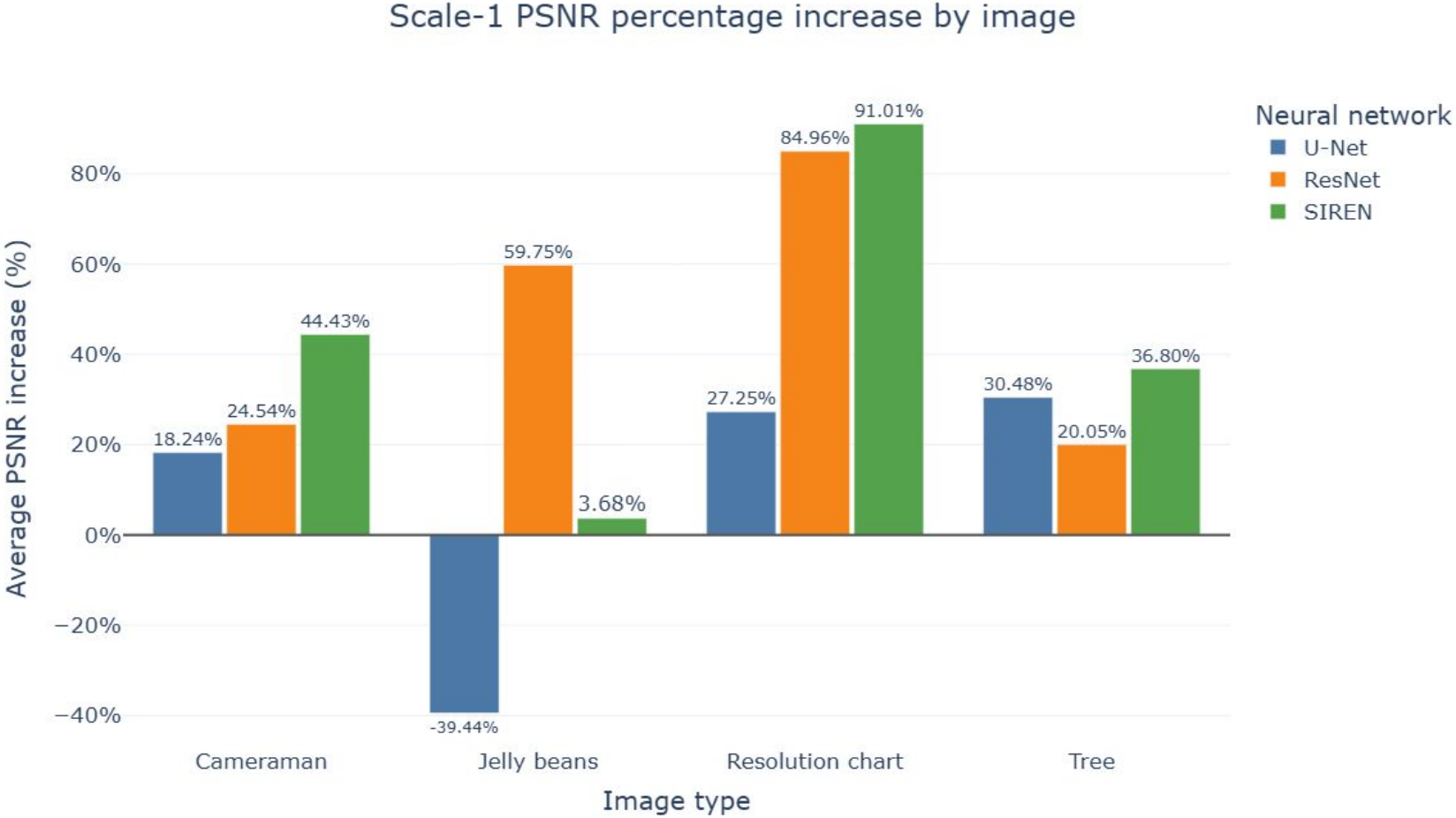
69.6 seconds

Time for full pipeline

Detailed Results

Summary Table

Image	Skip PSNR	UNet PSNR	ResNet PSNR	Texture-Net PSNR	Siren PSNR	BACON PSNR
Clock			21.1356			
Jelly Beans			22.8628			
Tree			11.1960			
Resolution Chart			18.0873			
Camera Man			**23.4298**			


Scale-1 PSNR percentage increase by image
Neural network
U-Net
ResNet
SIREN
Average PSNR increase (%)
80%
60%
40%
20%
0%
−20%
−40%
18.24%
24.54%
44.43%
-39.44%
59.75%
3.68%
27.25%
84.96%
91.01%
30.48%
20.05%
36.80%
Cameraman
Jelly beans
Resolution chart
Tree
Image type

Conclusion

Conventional BFSI → **Requires many measurements** → **+ DIP (SIREN, U-Net, ResNet)** → **Better reconstruction at low sampling**

Better reconstruction at low sampling ↓

Competitive with FSI + DIP; More practical in real life

- **Improved quality**
  Higher PSNR and SSIM than BFSI
- **Sampling efficiency**
  Higher quality with fewer measurements
- **Practical advantage**
  BFSI patterns remain compatible with high-speed hardware

Discussion - Neural Network Comparison

