AI in Humanities – Image Restoration

L^1 Minimization and Fourier Transformation

Bryce Ou, Claire Cho, Elma Hsieh, KangCheng Zhao,
Kate Kim

Agenda

Theorical Background – sparsity assumption

Fourier Transform and Singal Recovery

L^1 Minimization

Research Goup Process

L^1 Minimization vs. SRCNN (Fixed Mask)

Real World Applications

Sparsity Assumption and Signal Recovery

"Often, signals are sparse in the frequency domain."

If a signal has a structured (but not overly regular) pattern, the **uncertainty principle** allows us to fully recover it using only a few non-zero frequency components."

Spatial Domain

(How the signal looks in time/space)

Fourier Domain

(Sine and Cosine building blocks)

Fourier Transform

Fourier transform can convert signals from spatial domain to frequency domain.

Real world images can be viewed as discrete sampling from some continuous functions.

Therefore, according to **uncertainty principle**, we can exploit the sparsity in the frequency domain by applying Fourier transform to real world images for image recovery.

Then, for an M by N image, we can define its Fourier transform function as:

$$\hat{f}(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\, e^{-i2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}$$

Signal Recovery Process-settings

We have

$$\hat{f}(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\, e^{-i2\pi\left(\frac{km}{M} + \frac{ln}{N}\right)}$$

And suppose a mask function.

$$mask(m,n) = \begin{cases} 1, & (m,n) \notin S \\ 0, & (m,n) \in S \end{cases}$$

Then, for our real-world images, observed(m, n) is defined as:

$$observed(m,n) = f(m,n) \cdot mask(m,n)$$

Signal Recovery Process-L^1 Minimization

It will be NP-hard to count all kinds of possible combinations.

Under the assumption of sparsity, the minimum value of the sum of absolute values in the frequency domain is most likely to be an original signal.

So we use the L^1-norm to sum up our values. We then have

$$arg\min_{g} \|Fg\|_1 \quad \text{subject to } g(m,n) = observed(m,n) \text{ for } (m,n) \notin S$$

and under the condition of

$$|E||S| < \frac{N^2}{2}$$

we can recover the original image perfectly

Our Reseach implementation

1. Transform images from RGB to Grayscale, resize when necessary.
2. Randomly damage some of the images' pixels
3. Convert grayscale images to frequency domain by using Fourier transform
4. Use L_1 minimization to recover missing frequency cause by damaging in pixels
5. Reconstruct images by doing inverse Fourier transform

How does this image recovery algorithm differ from pre-existing models?

Our Model (L^1 minimization with FT)

- **Goal:** Recover damaged images by L^1 minimization.

- No need for training dataset
- High accuracy
- Easy to understand(pure Math)

Pre-Existing model: SRCNN

SRCNN – SUPER RESOLUTION CONVOLUTIONAL NEURAL NETWORK

TRAINED WITH **CIFAR-10 DATASET** (SOURCE:HTTPS://WWW.CS.TORONTO.EDU/~KRIZ/CIFAR.HTML)

FAST & EFFICIENT – SUITABLE FOR REAL-TIME OR LARGE-SCALE APPLICATIONS

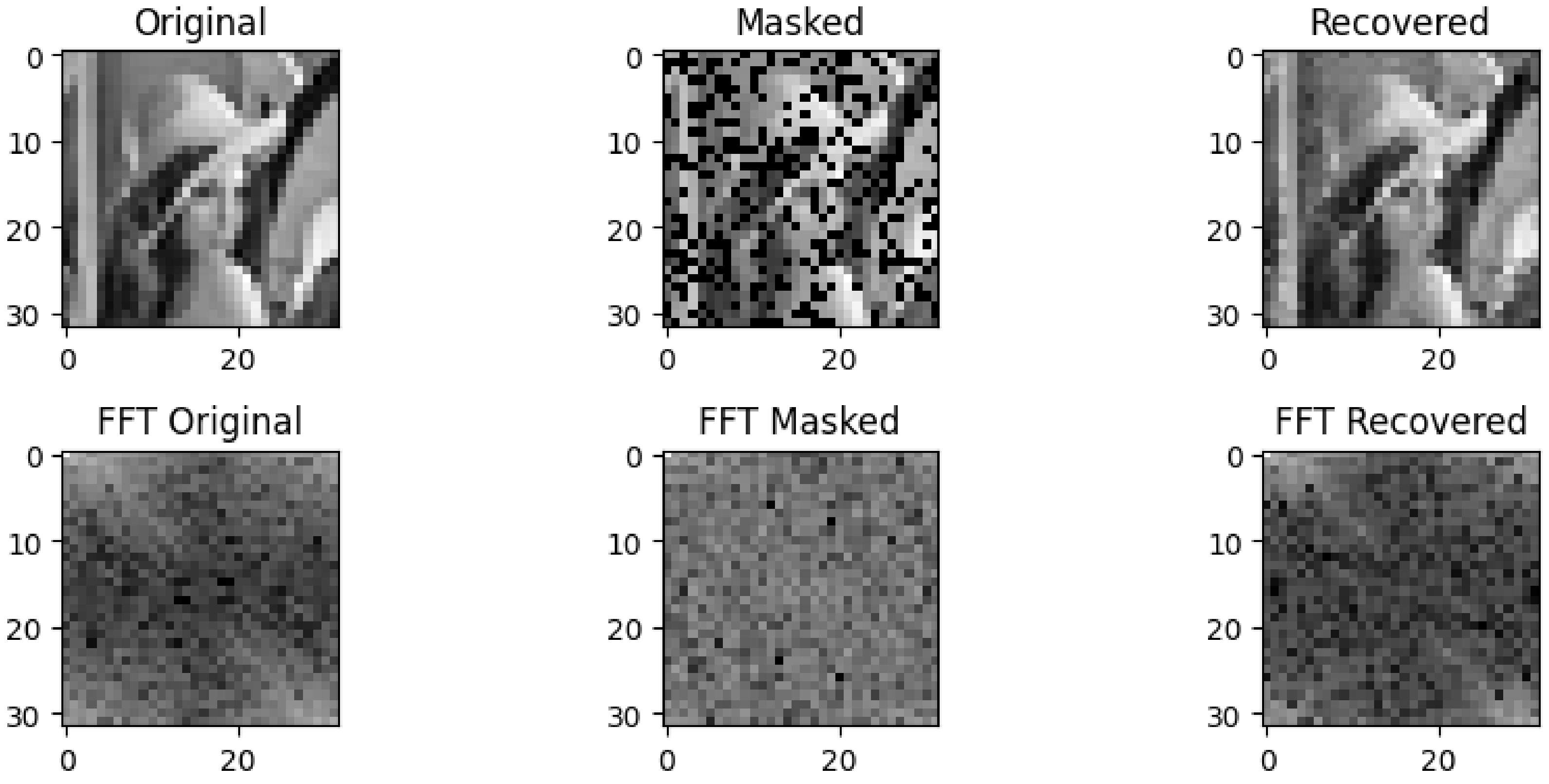
SCALES WELL

L^1 Minimization
Resize & Restore

With Fixed Mask

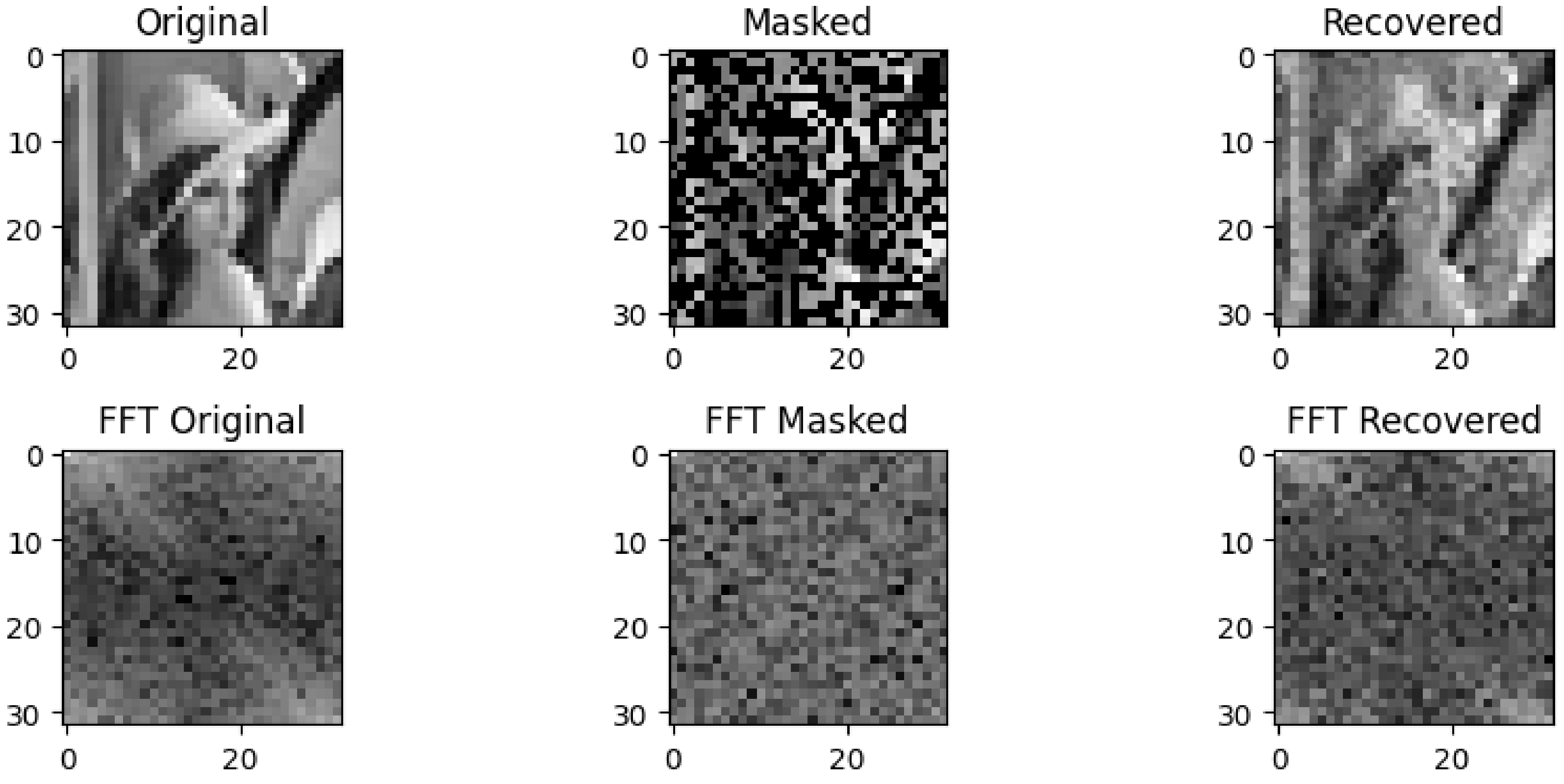
(512x512)

L^1Minimization
(32x32) + 30% Mask



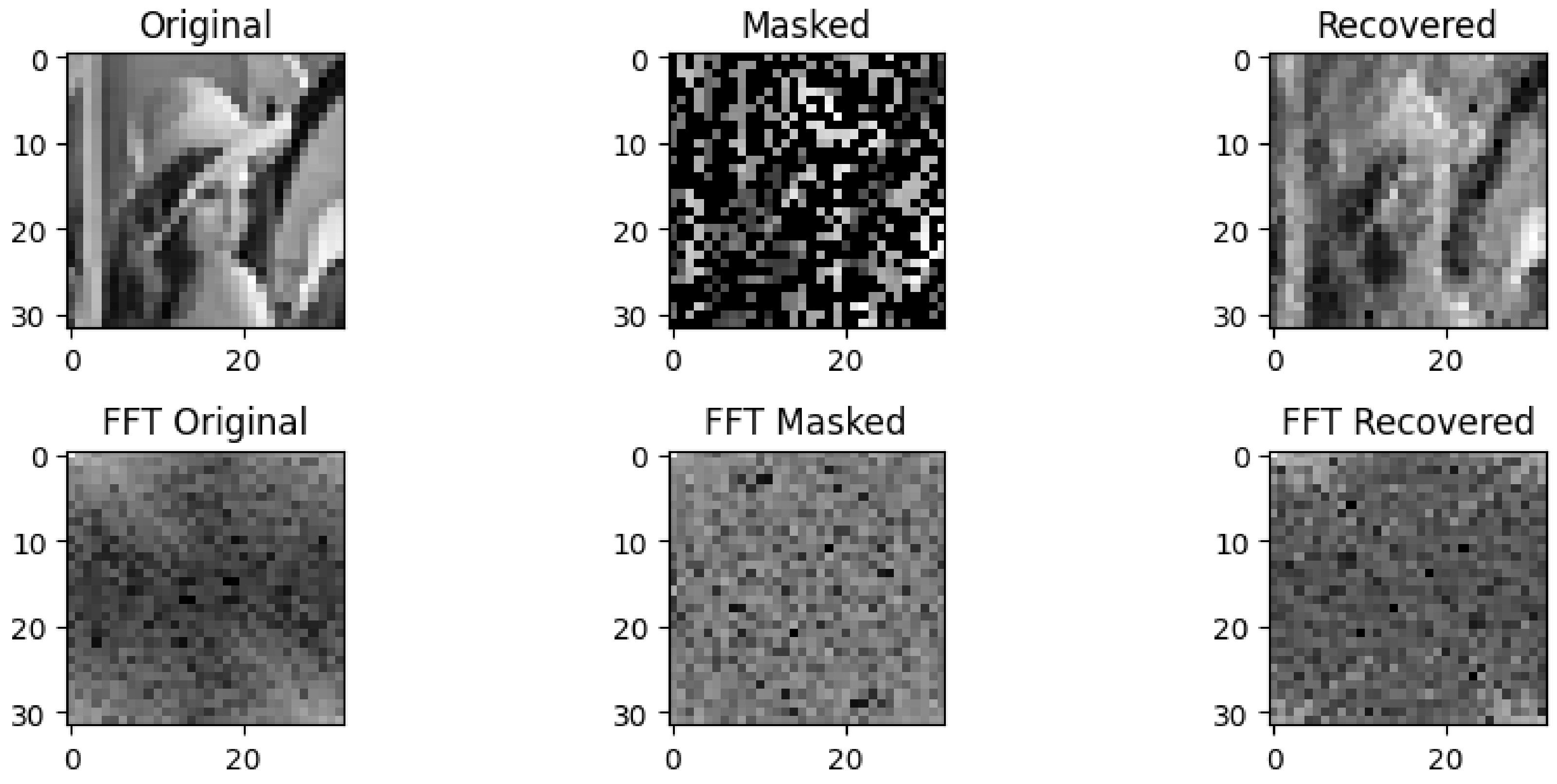
PSNR: 26.05
SSIM: 0.9411
Time: 11.19 sec

L^1 Minimization (32x32) + 50% Mask



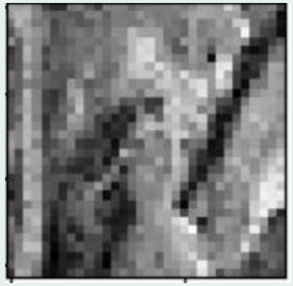
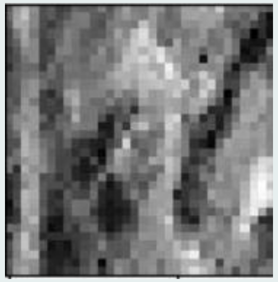
PSNR: 22.13
SSIM: 0.8191
Time: 19.21 sec

L^1 Minimization (32x32) + 60% Mask

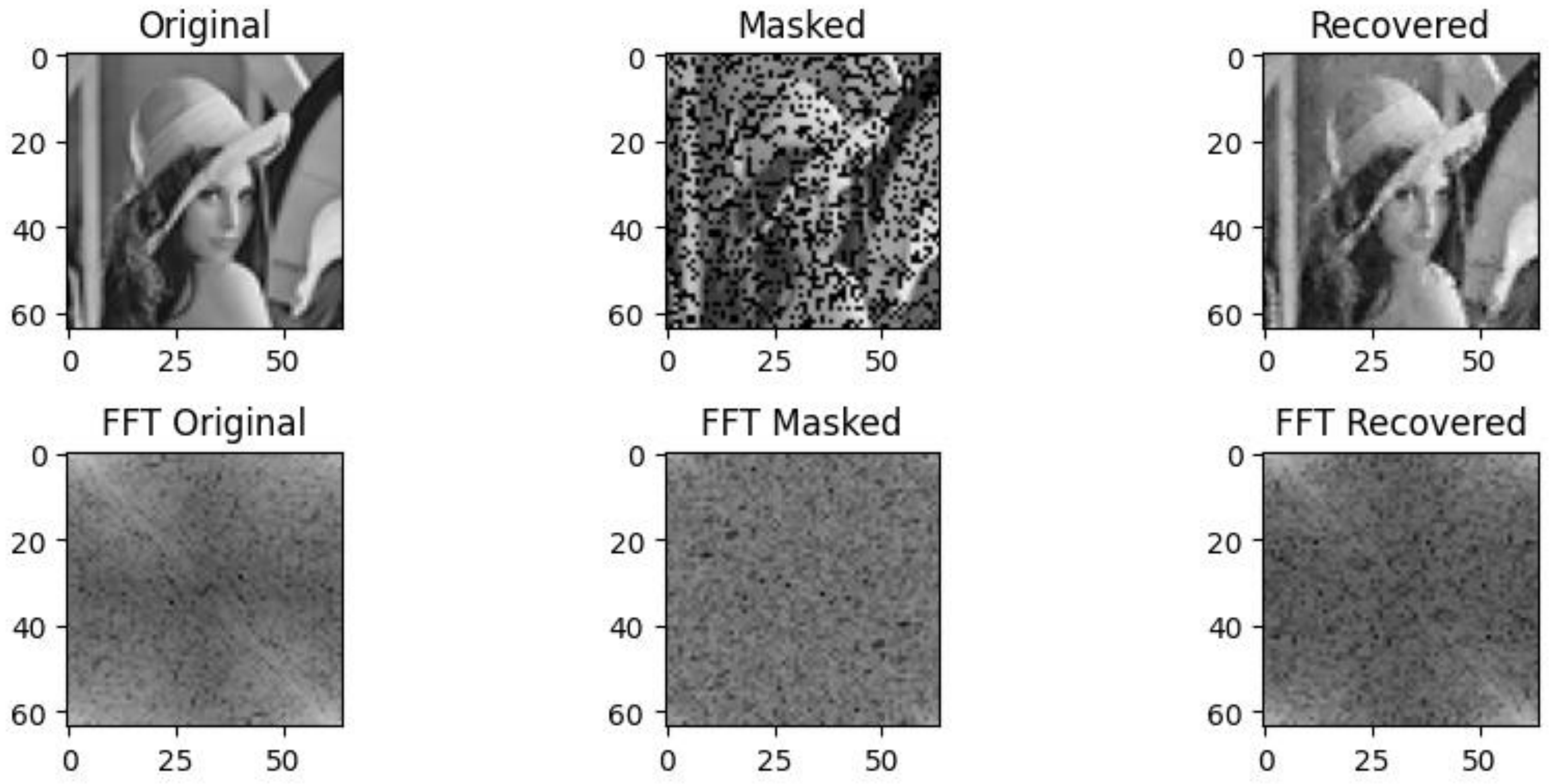


PSNR: 20.71
SSIM: 0.7713
Time: 38.35 sec

32x32 Comparison

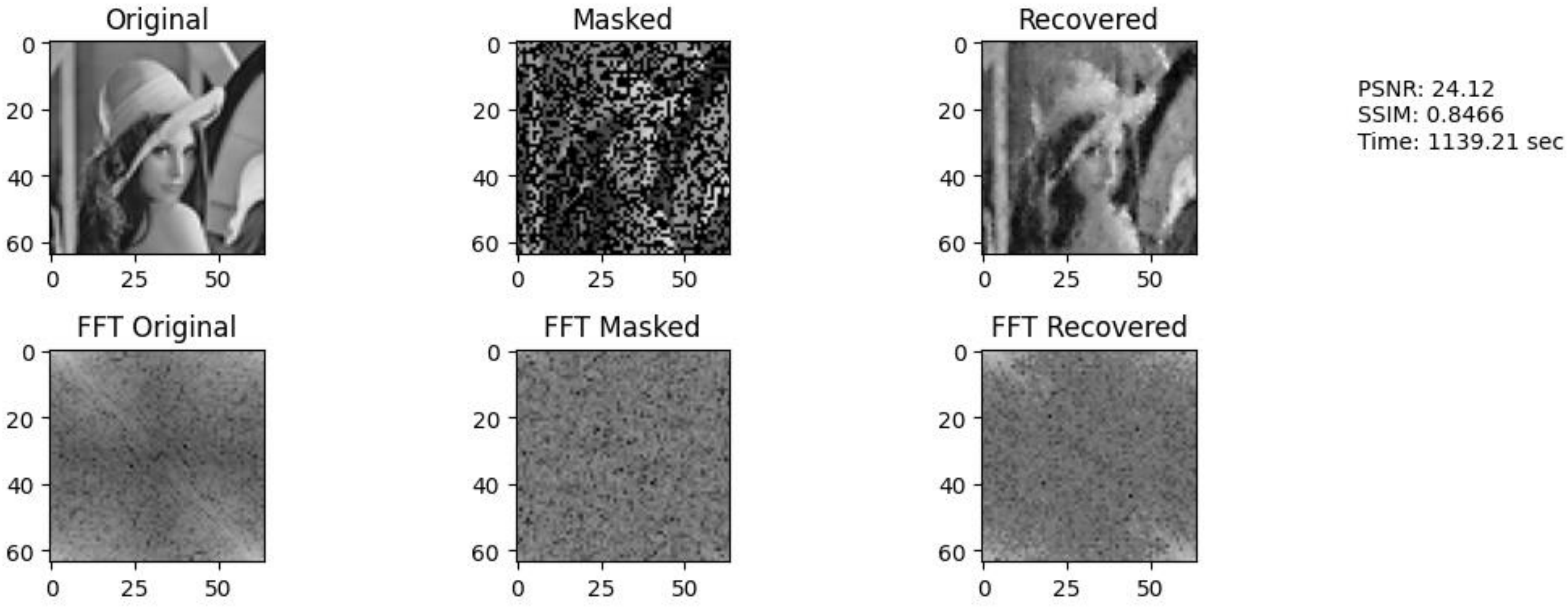
Mask	30%	50%	60%
PSNR	26.05	22.13	20.71
SSIM	0.9411	0.8191	0.7713
Time	11.19	19.21	38.35
Image			

L^1 Minimization (64x64) + 30% Mask



PSNR: 27.71
SSIM: 0.9311
Time: 751.94 sec

L^1 Minimization (64x64) + 50% Mask



64x64 Comparison

Mask	30%	50%
PSNR	27.71	24.12
SSIM	0.9311	0.8466
Time	751.94	1139.21
Image		

L^1 Minimization (128x128) + 30% Mask

```
(venv) glsn-mini-01:L1_minimization ehsieh2$ python3 main.py
run resize recovery function...
[INFO] Saved mask with 4915 coordinates to fixed_mask_128.npy
Starting clock
/Users/ehsieh2/STEMforAll-25/venv/lib/python3.9/site-packages/scs/__init__.py:83: UserWarning: Converting A to a CSC (compressed sparse column) matrix; may take a while.
  warn(
Killed: 9
```

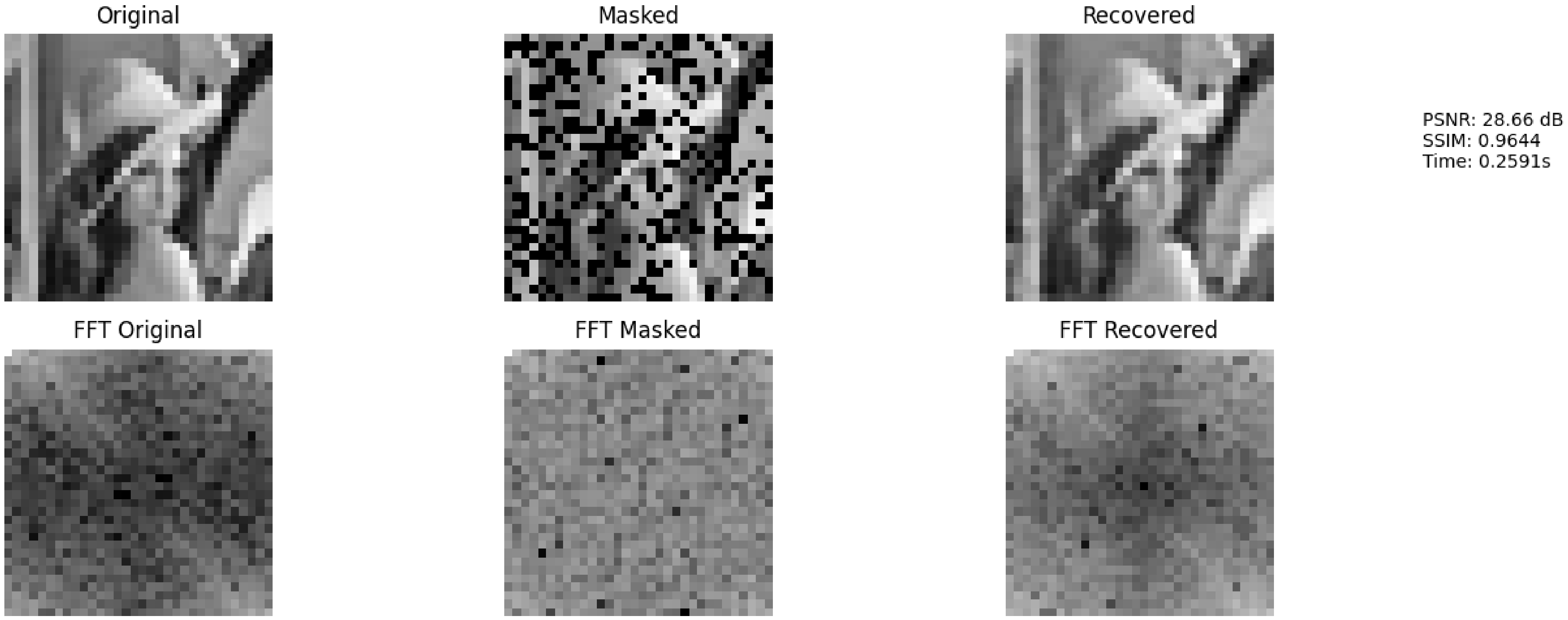
Operating system killed the operation because it was using too much memory (RAM). This is common in convex optimization with large problem sizes, especially when working with pixel-wise constraints on even moderately-sized images like 128x128.

Operation killed after more than 1 hour of running.

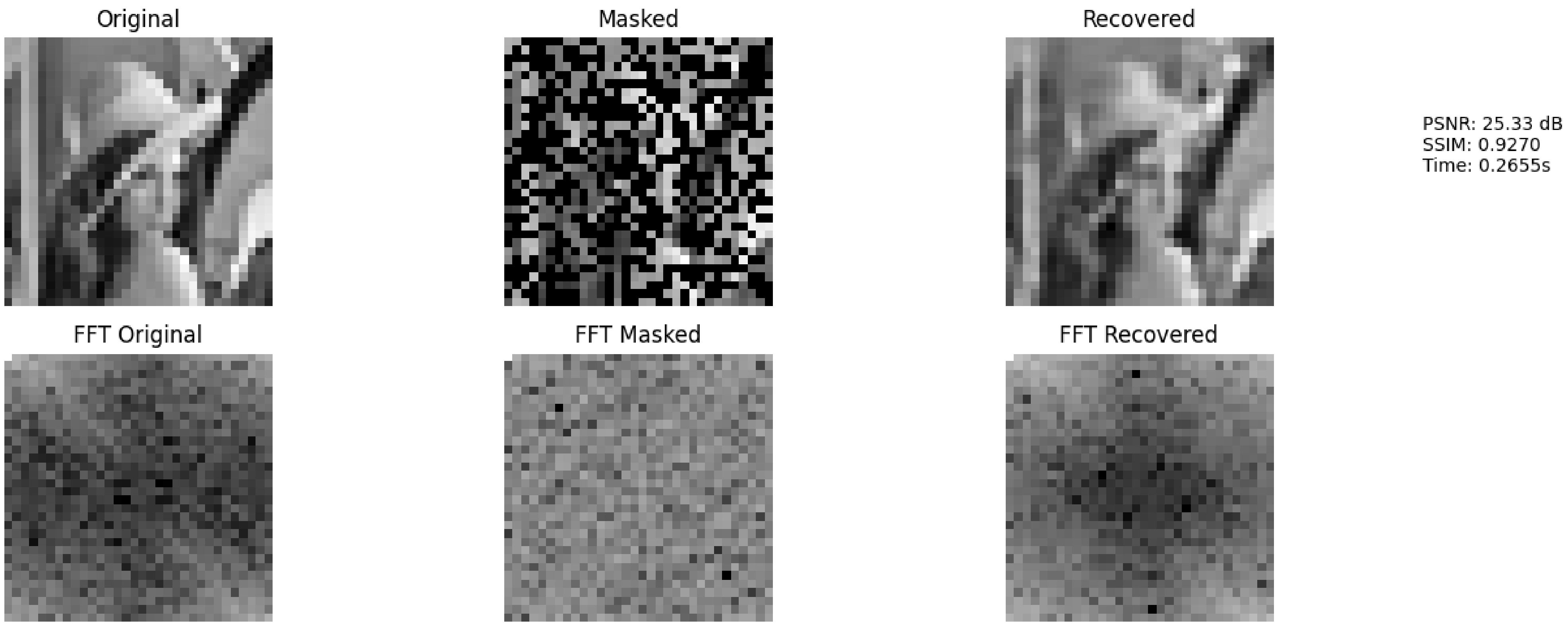
SRCNN
Resize & Restore

With Fixed Mask

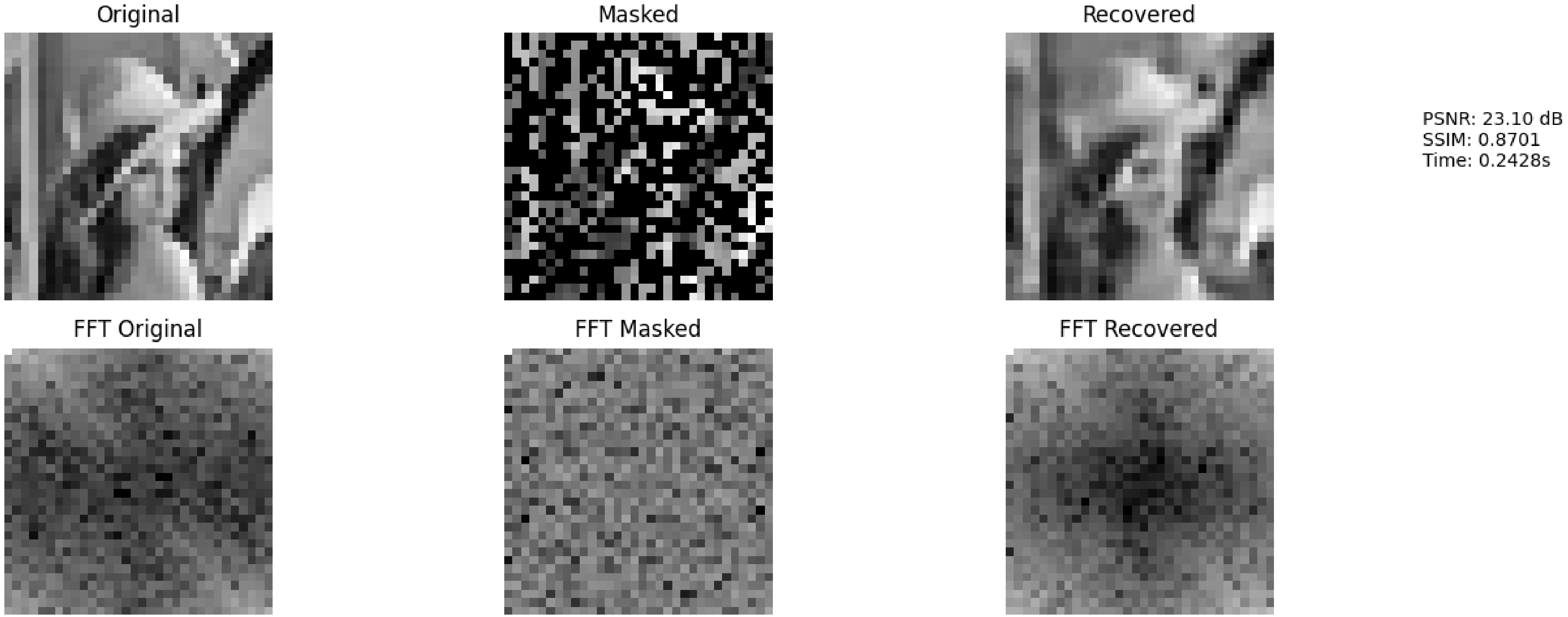
SRCNN
(32x32) + 30%



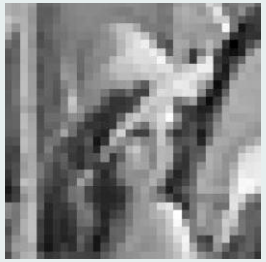
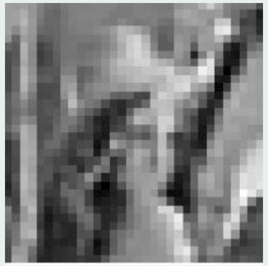
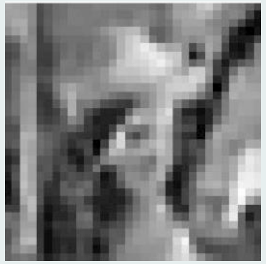
SRCNN
(32x32) + 50%



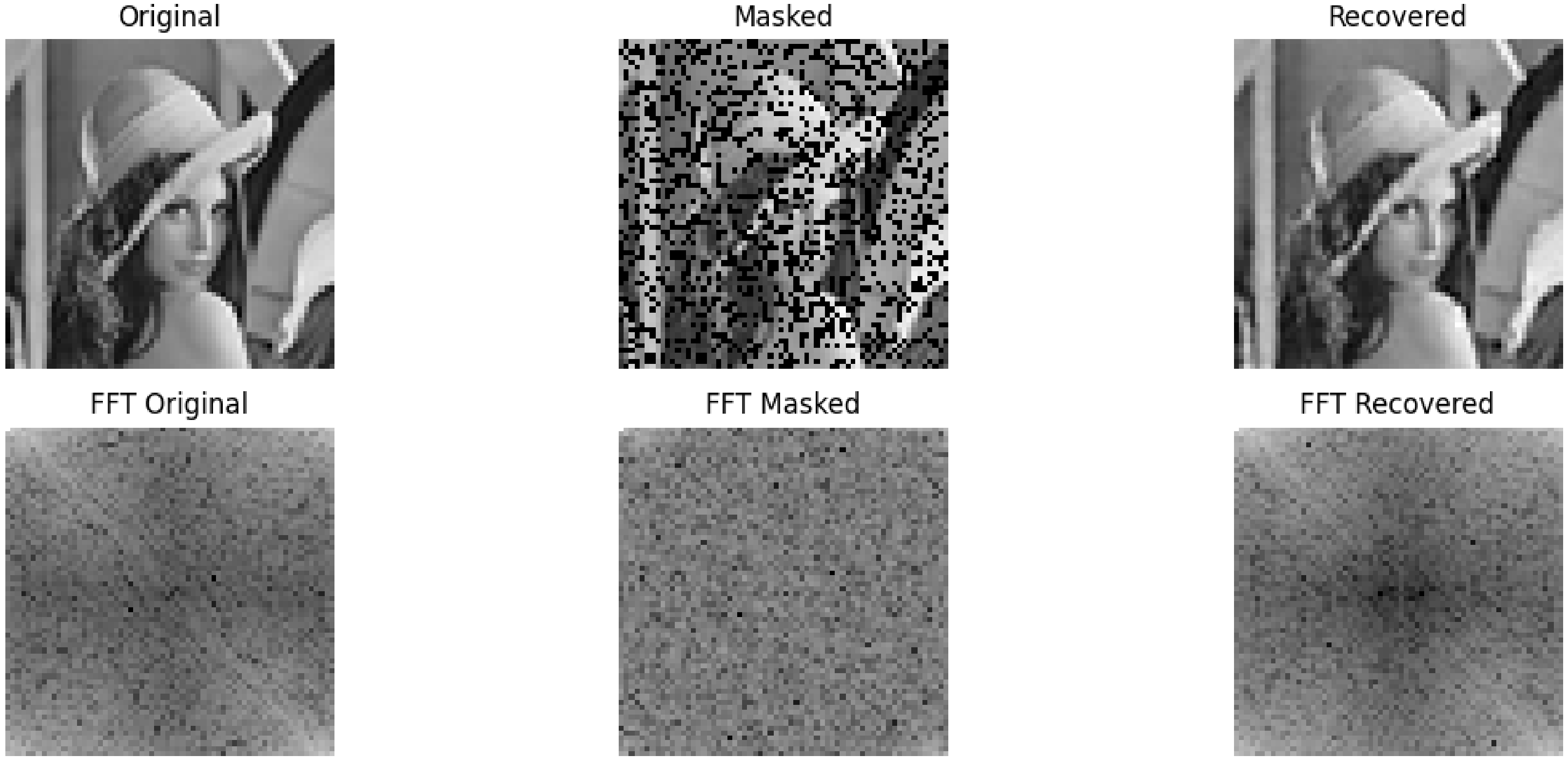
SRCNN
(32x32) + 60%



32x32 Comparison

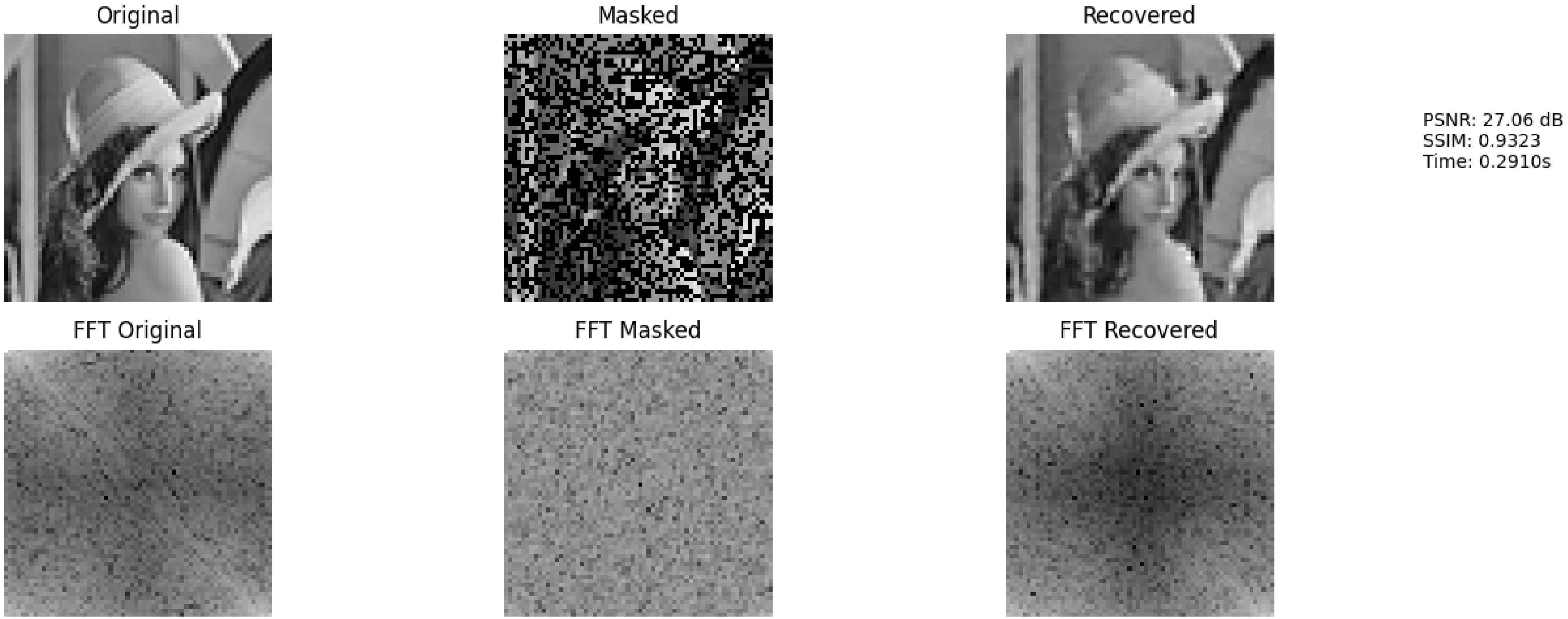
Mask	30%	50%	60%
PSNR	28.66	25.33	23.10
SSIM	0.9644	0.9270	0.8701
Time	0.2591	0.2655	0.2428
Image			

SRCNN
(64x64) + 30%



PSNR: 30.94 dB
SSIM: 0.9693
Time: 0.2525s

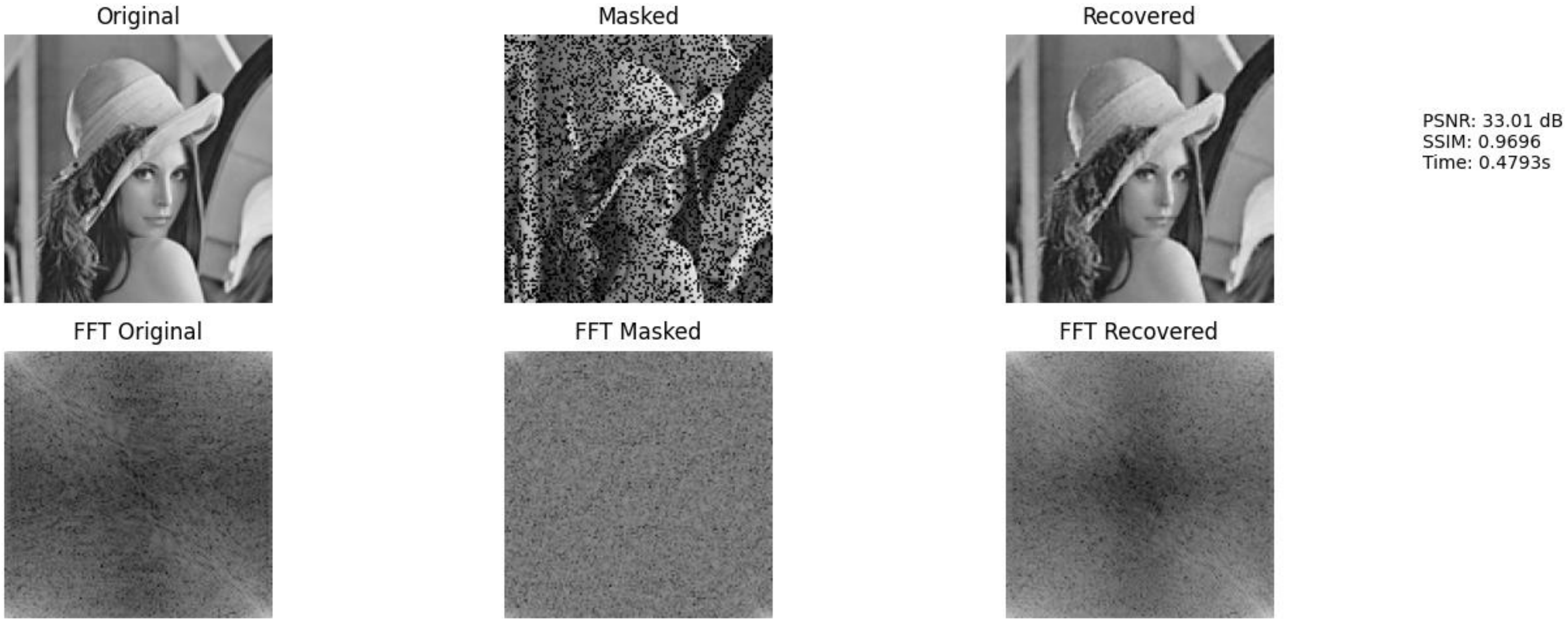
SRCNN
(64x64) + 50%



64x64 Comparison

Mask	30%	50%
PSNR	30.94	27.06
SSIM	0.9693	0.9323
Time	0.2525	0.2910
Image		

SRCNN
(128x128) + 30%



L^1Minimization & SRCNN Comparison

	Model\ Size + Mask	32x32 30%	32x32 50%	32x32 60%	64x64 30%	64x64 50%	128x128 + 30%
PSNR	L^1Minimization	26.05	22.13	20.71	27.71	24.12	X
	SRCNN	28.66	25.33	23.10	30.94	27.06	33.01
SSIM	L^1Minimization	0.9411	0.8191	0.7713	0.9311	0.8466	X
	SRCNN	0.9644	0.9270	0.8701	0.9693	0.9323	0.9696
SECONDS	L^1Minimization	11.19	19.21	38.35	751.94	1139.21	X
	SRCNN	0.2591	0.2655	0.2428	0.2525	0.2910	0.4793

Conclusion

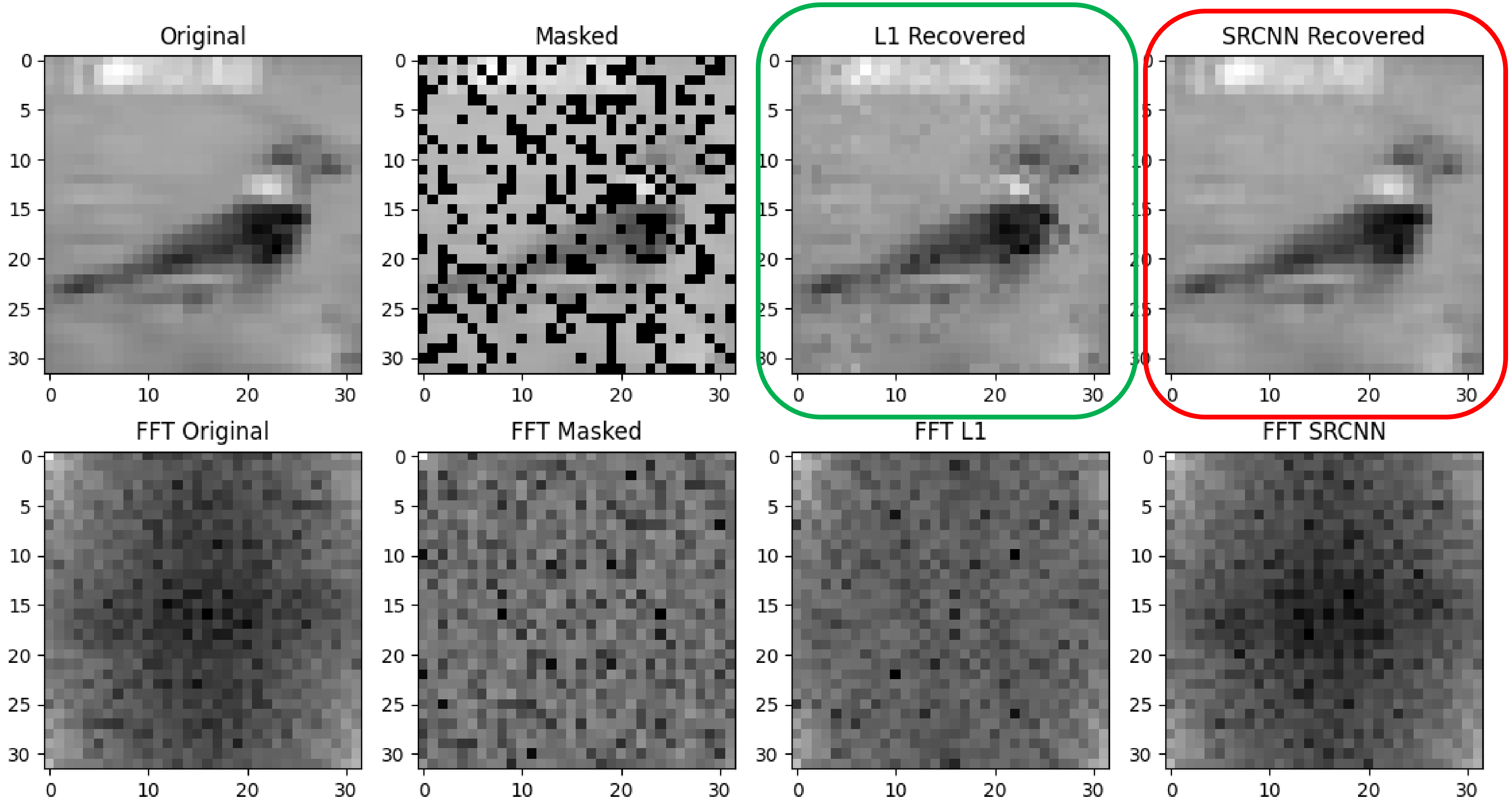
PSNR (Peak Signal-to-Noise Ratio) - higher is better (image quality)

SSIM (Structural Similarity Index) - higher is better (perceptual similarity)

Time (in seconds) - lower is better

Criteria	L^1	SRCNN
Image Quality	Lower PSNR/SSIM	Higher PSNR/SSIM
Speed	Very slow, poor scalability	Very fast, consistent runtime
Scalability	Fails or impractical on large images	Works well on all sizes

L^1 Minimization vs SRCNN

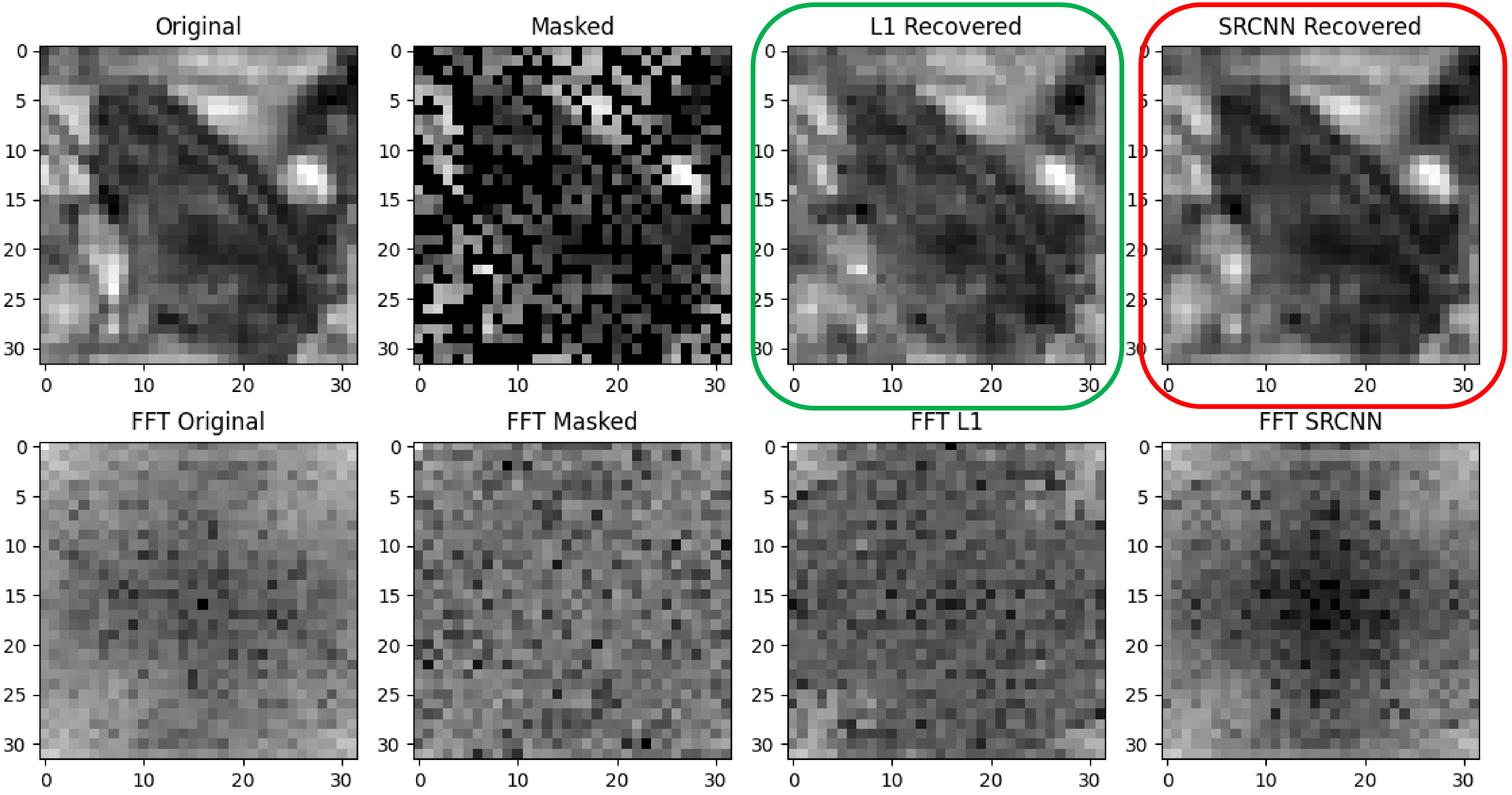


L1 PSNR: 32.37
L1 SSIM: 0.9255
L1 Time: 25.2640s

SRCNN PSNR: 38.57
SRCNN SSIM: 0.9852
SRCNN Time: 0.0050s

30% Mask

L^1 Minimization vs SRCNN

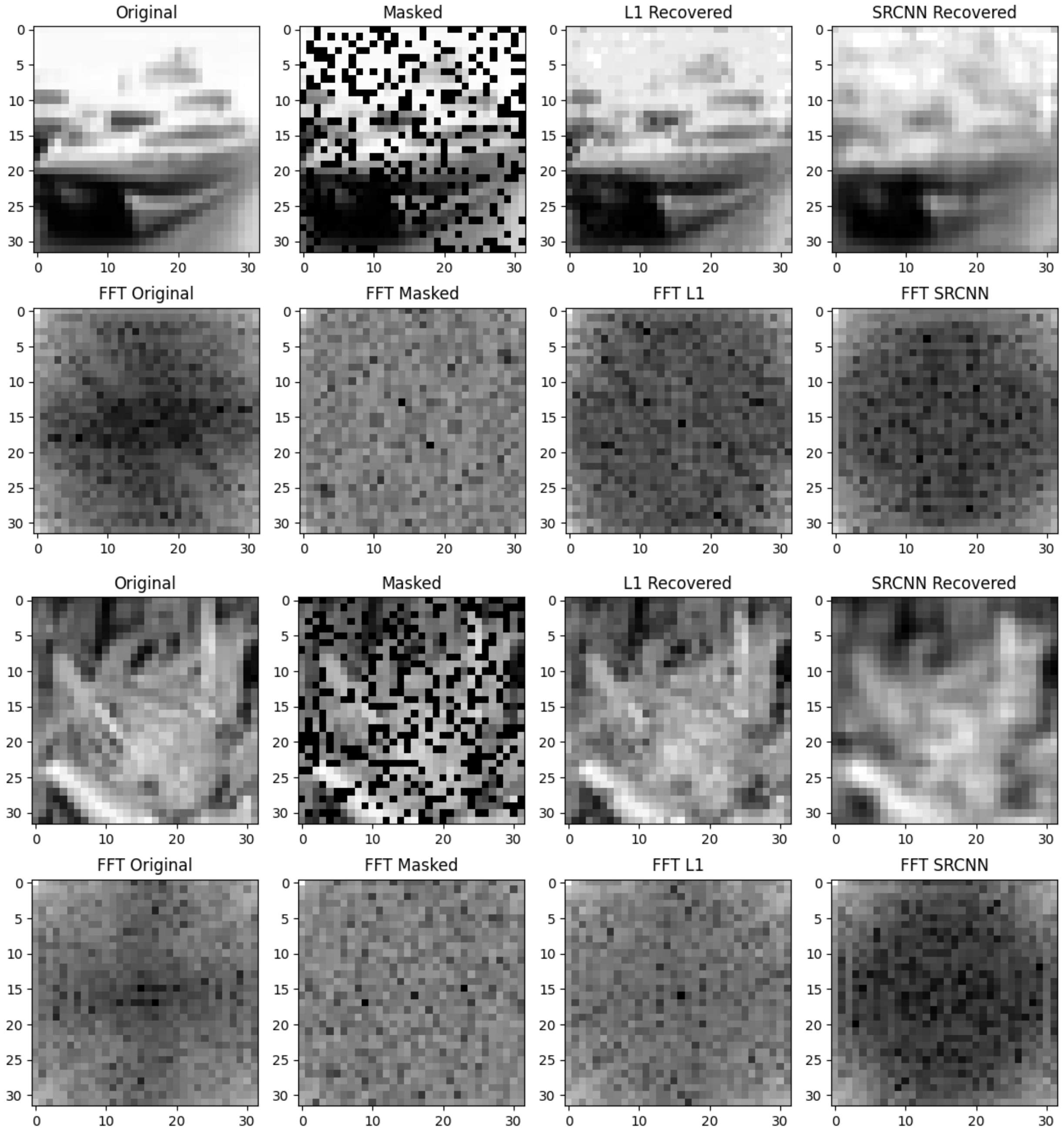


L1 PSNR: 25.37
L1 SSIM: 0.8347
L1 Time: 44.1199s

SRCNN PSNR: 29.54
SRCNN SSIM: 0.9223
SRCNN Time: 0.0039s

50% Mask

Report Evaluation for 30% Mask with 500 training set.



L1 PSNR: 26.60
L1 SSIM: 0.9308

SRCNN PSNR: 21.95
SRCNN SSIM: 0.7546

L1 PSNR: 27.04
L1 SSIM: 0.8950

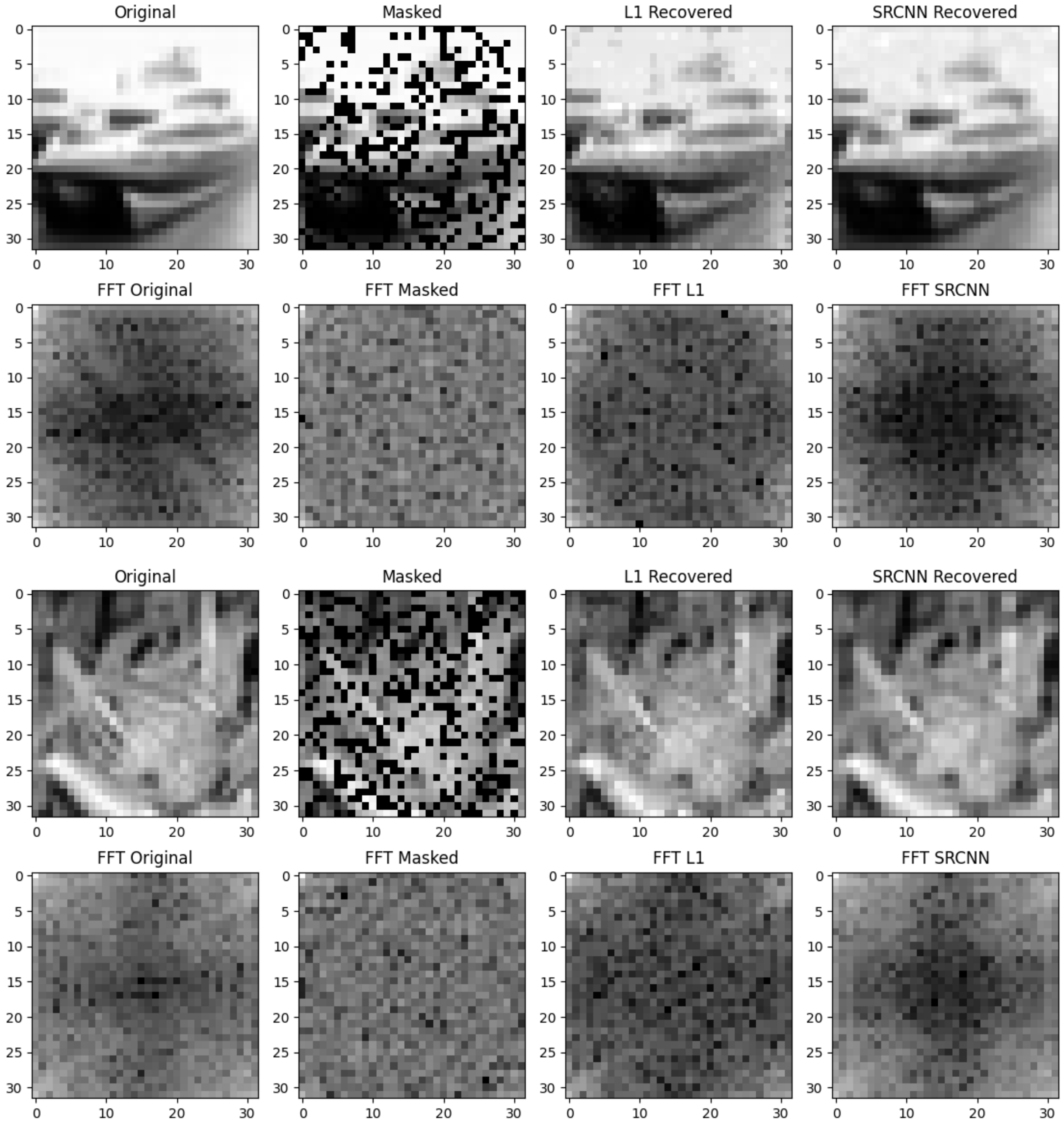
SRCNN PSNR: 22.64
SRCNN SSIM: 0.7131

[L_1]:

- Avg PSNR: 27.64 dB
- Avg SSIM: 0.9176
- Avg Error Rate: 9.69%
- Avg Time Usage: 26.66 seconds

[SRCNN]:

- Avg PSNR: 23.12 dB
- Avg SSIM: 0.7584
- Avg Error Rate: 10.47%
- Avg Time Usage: 0.002 seconds



L1 PSNR: 26.89
L1 SSIM: 0.9262

SRCNN PSNR: 30.27
SRCNN SSIM: 0.9561

L1 PSNR: 26.66
L1 SSIM: 0.9155

SRCNN PSNR: 28.00
SRCNN SSIM: 0.9303

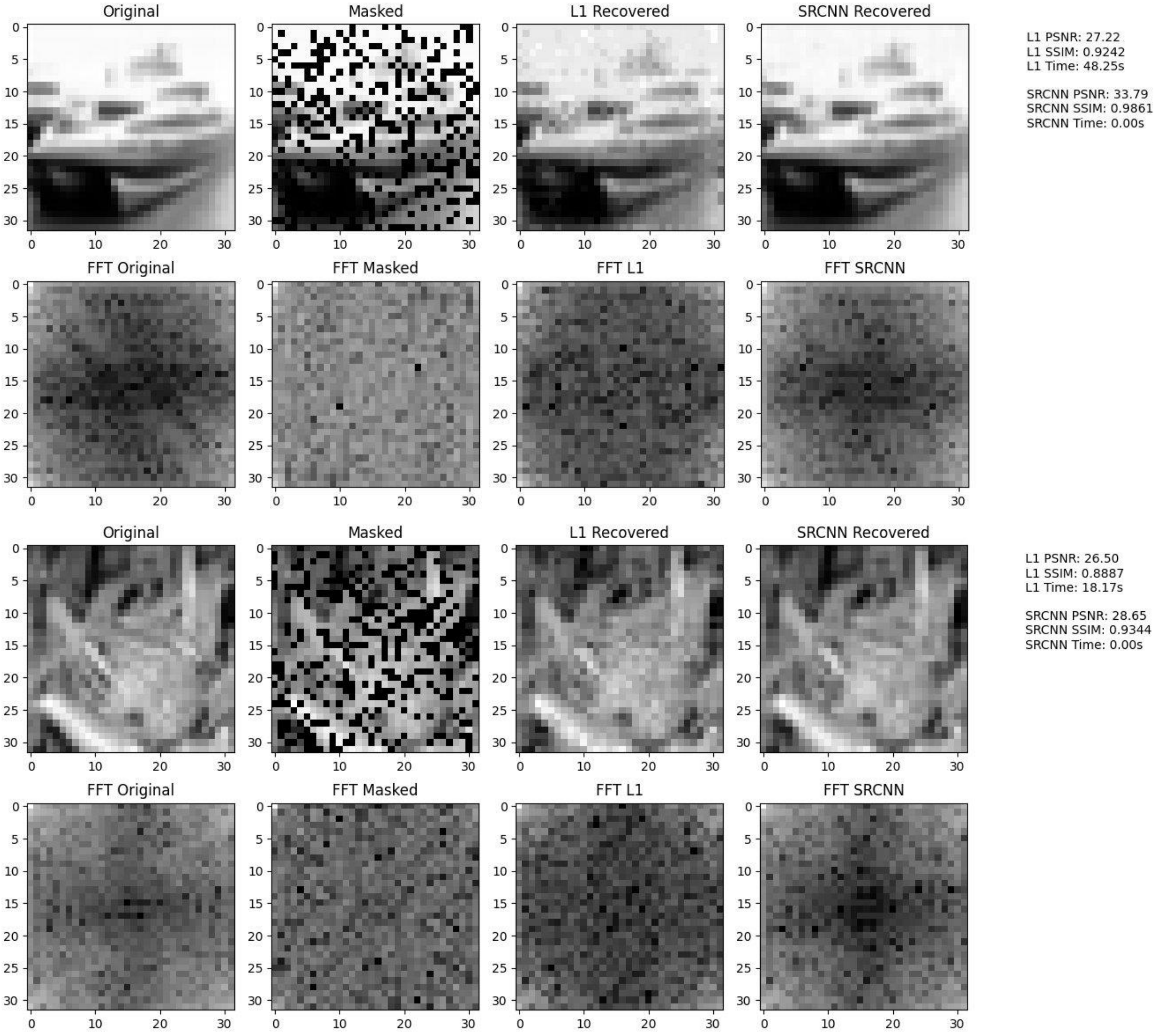
Report Evaluation for 30% Mask with 5,000 training set.

[L_1]:

- Avg PSNR: 27.96 dB
- Avg SSIM: 0.9280
- Avg Error Rate: 9.42%
- Avg Time Usage: 26.51 seconds

[SRCNN]:

- Avg PSNR: 30.13 dB
- Avg SSIM: 0.9525
- Avg Error Rate: 6.95%
- Avg Time Usage: 0.001 seconds



Report Evaluation for 30% Mask with 50,000 training set.

[L_1]:

- Avg PSNR: 27.95 dB
- Avg SSIM: 0.9211
- Avg Error Rate: 8.44%
- Avg Time Usage: 25.14 seconds

[SRCNN]:

- Avg PSNR: 32.23 dB
- Avg SSIM: 0.9698
- Avg Error Rate: 5.33%
- Avg Time Usage: 0.0095 seconds

Exploring real-world implementations

How the technology is currently being used

Real World Implications

Without needing training data, it can accurately and efficiently recover the image and signal.

Belgium, Antwerpen, Cathedral of Our Lady

Wall mural with small patches restored utilizing AI algorithms and DALL- E Outpainting (OpenAI's image generating feature).

With ***our*** algorithm, you would not need the large data set that it took to train the AI to restore this mural

Medical Imaging

It has been integrated with MRI machines for fewer measurement and more accurate result.

Compressed Storage and Transmission

Canmore efficiently stream large videos and compress cloud storage

Wireless Communications

Used in 5g, real-time system, and next-gen IoT to improve call/ data quality from corrupted noisy signals.

Sources

- Iosevich, A., & Mayeli, A. (2023, November 7). *Uncertainty principles on finite Abelian groups, restriction theory, and applications to sparse signal recovery*. arXiv.org. https://arxiv.org/abs/2311.04331
- Turan, Ayca. "Ai Assisted Restoration: Time Travel through Creative Technology." *Medium*, Medium, 10 Apr. 2023, medium.com/@aycaturan/ai-assisted-restoration-time-travel-through-creative-technology-c638643d2a54.
- Sachdev, A. (2021, December 12). Spatial and frequency Domain — Image Processing - VITHelper - Medium. *Medium*. https://medium.com/vithelper/spatial-and-frequency-domain-image-processing-83ffa3fc7cbc
- Sharda, A. (2022, January 6). Understanding Gaussian Blur Filters | Medium. *Medium*. https://aryamansharda.medium.com/image-filters-gaussian-blur-eb36db6781b1
- Tiantian, W., Hu, Z., & Guan, Y. (2024). An efficient lightweight network for image denoising using progressive residual and convolutional attention feature fusion. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-60139-x
- Krizhevsky, A. (2009). *CIFAR-10 and CIFAR-100 datasets*. Toronto.edu. https://www.cs.toronto.edu/~kriz/cifar.html
- Singh, H. (2021, March 1). *Neural Network | Introduction to Neural Network | Neural Network for DL*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/03/basics-of-neural-network/
- Hardik. (2021, October 26). *Fourier Transformation in Image Processing*. CrossML Blog. https://medium.com/crossml/fourier-transformation-in-image-processing-84142263d734
- Ongie, Greg., Jacob, Matthews (2015, Febeuary 3). Arxiv.org. Recovery of Piecewise Smooth Images from Few Fourier Sampleshttps://arxiv.org/pdf/1502.00705

Q & A