High Perceptual Quality Image Denoising via Neural Compression

Nam Nguyen

School of Electrical Engineering and Computer Science
Oregon State University

January 26, 2026

Compression-Based Denoising

- **Signal:** $\mathbf{x} = (x_1, \ldots, x_n)$
- **Observation:** $\mathbf{y} = (y_1, \ldots, y_n)$

$\Rightarrow$ **Goal:** *recover* $\mathbf{x}$ *from its noisy observation* $\mathbf{y}$

Compression-Based Denoising

- **Signal:** $\mathbf{x} = (x_1, \ldots, x_n)$
- **Observation:** $\mathbf{y} = (y_1, \ldots, y_n)$

$\Rightarrow$ **Goal:** *recover* $\mathbf{x}$ *from its noisy observation* $\mathbf{y}$

Key idea: *Structured signals* are more **compressible** than noisy ones.

- Optimal lossy compression $\Rightarrow$ asymptotically optimal denoising [Weissman et al. 2005].
- Neural compression as a denoising mechanism [Zafari et al. 2025].

Compression-Based Denoising

- **Signal:** $\mathbf{x} = (x_1, \ldots, x_n)$
- **Observation:** $\mathbf{y} = (y_1, \ldots, y_n)$

$\Rightarrow$ **Goal:** *recover* $\mathbf{x}$ *from its noisy observation* $\mathbf{y}$

Key idea: *Structured signals* are more **compressible** than noisy ones.

- Optimal lossy compression $\Rightarrow$ asymptotically optimal denoising [Weissman et al. 2005].
- Neural compression as a denoising mechanism [Zafari et al. 2025].

Limitation: Minimizing distortion **only** often produces **over-smoothed/blurry** outputs.

Compression-Based Denoising

- **Signal:** $\mathbf{x} = (x_1, \ldots, x_n)$
- **Observation:** $\mathbf{y} = (y_1, \ldots, y_n)$

$\Rightarrow$ **Goal:** *recover* $\mathbf{x}$ *from its noisy observation* $\mathbf{y}$

Key idea: *Structured signals* are more **compressible** than noisy ones.

- Optimal lossy compression $\Rightarrow$ asymptotically optimal denoising [Weissman et al. 2005].
- Neural compression as a denoising mechanism [Zafari et al. 2025].

Limitation: Minimizing distortion **only** often produces **over-smoothed/blurry** outputs.

$\Rightarrow$ **Our approach:** Use a **WGAN-based discriminator** to guide reconstructions toward the clean-image manifold.

Perception-enhanced Neural Compression Denoiser

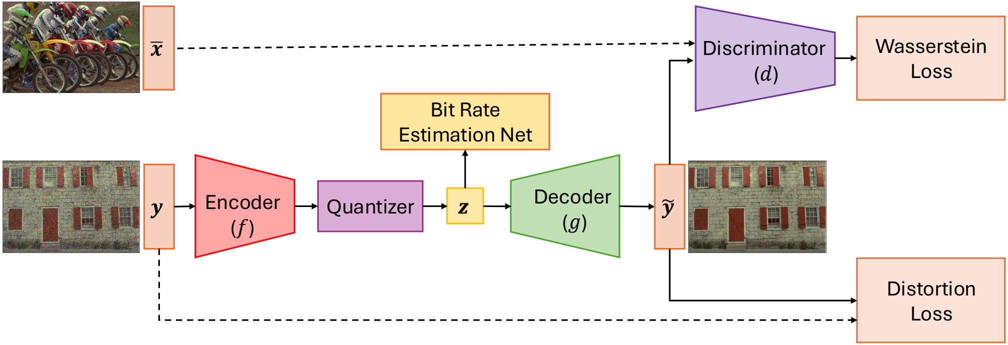

Figure: Overview of the architecture. $\bar{\mathrm{x}}$ is drawn from the clean image dataset (unpaired with x).

Experiment Setup

- **Loss Function:**

$$\boxed{\mathcal{L} = \underbrace{\mathbb{E}[\|Y - \tilde{Y}\|^2]}_{\text{Distortion loss}} + \lambda_r \underbrace{\log \mathbb{P}(Q(f(Y)))}_{\text{Compression rate}} + \lambda_p \underbrace{W_1(p_{\bar{X}}, p_{\tilde{Y}})}_{\text{Wasserstein loss}}}$$

Experiment Setup

- **Loss Function:**

$$\boxed{\mathcal{L} = \underbrace{\mathbb{E}[\|Y - \tilde{Y}\|^2]}_{\text{Distortion loss}} + \lambda_r \underbrace{\log \mathbb{P}(Q(f(Y)))}_{\text{Compression rate}} + \lambda_p \underbrace{W_1(p_{\bar{X}}, p_{\tilde{Y}})}_{\text{Wasserstein loss}}}$$

- **Dataset:**
  - **Training:** BSDS500 dataset (481×321 natural images, high texture and structure) [Arbelaez et al. 2011].
  - **Testing:** Kodak-24 dataset (768×512 high-quality color images) [Company 1991].

Experiment Results

Category	Method	PSNR (dB) ↑	SSIM ↑	PI ↓
Non-learning	JPEG-2K [Taubman et al. 2002]	26.4408	0.7357	7.4794
	BM3D [Dabov et al. 2007]	31.8757	0.8687	2.6503
Supervised	N2C [Zhang et al. 2017]	<u>32.2114</u>	<u>0.8865</u>	2.5446
	N2N [Lehtinen et al. 2018]	**32.2723**	**0.8877**	2.5439
Unsupervised	DeCompress [Zafari et al. 2025]	27.8315	0.7519	2.7979
	OTDenoising [Wang et al. 2023]	31.2893	0.8677	**2.0095**
	Ours	28.0435	0.8035	<u>2.1670</u>

Table: Comparison of denoising performance on the KODAK dataset corrupted by Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma = 25$. Best values are **bold** and second-best values are <u>underlined</u>.

Real-World Denoising: Microscopy & Smartphone

- Fluorescence microscopy (Mouse Nuclei) [Buchholz et al. 2020] and real smartphone photos (SIDD) [Abdelhamed et al. 2018].
- Our method achieves good PSNR/SSIM and low perceptual distortion across datasets.

Mouse Nuclei (Gaussian noise)

σ	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**DISTS↓**
10	33.03	0.805	0.044	0.140
20	30.59	0.803	0.073	0.168

SIDD (smartphone noise)

PSNR↑	**SSIM↑**	**LPIPS↓**	**DISTS↓**
33.61	0.904	0.323	0.237

Experiment Results

Category	Method	PSNR ↑	SSIM ↑	LPIPS ↓	PI ↓	FID ↓
Non-learning	JPEG-2K	26.4381	0.7479	0.4001	7.4368	109.1468
	BM3D	31.8757	0.8687	0.2214	3.8550	68.2196
Supervised	**DiffDeComp**, $\rho = 0$	30.1119	0.8475	0.1456	2.8558	50.2368
	DiffDeComp, $\rho = 0.9$	28.0348	0.8086	0.1163	2.4571	<u>24.2271</u>
	CGanDeComp	28.8619	0.8106	**0.0959**	2.5179	**21.9491**
	N2C	32.2117	0.8864	0.1269	2.5578	47.8364
	N2N	<u>32.2749</u>	<u>0.8877</u>	0.1263	2.5316	43.7995
	Restormer	**32.4120**	**0.8967**	<u>0.1032</u>	2.6429	<u>35.8829</u>
Unsupervised	**GanDeCompress**	27.8523	0.8033	0.1983	<u>2.1615</u>	77.9838
	DeCompress	27.8057	0.7518	0.2627	2.7967	83.2373
	OTDenoising	30.7174	0.8603	0.1385	**2.0005**	58.5344
	DIP	28.5314	0.7882	0.2112	2.7356	61.9785
	DD	26.5443	0.7551	0.4244	3.6312	110.8884

Table: Denoising performance comparison on the KODAK dataset with Gaussian noise $\mathcal{N}(0, \sigma^2)$, Thank you!