

A Lightweight ResNet-PatchGAN Framework for Parameter-Free Multi-Noise Image Denoising

Om Prakash Dhakal

*School of Computer Science & Engineering
Sandip University
India*

odhakal1@gmail.com

ORCID: 0009-0004-5605-6502

Abstract—Image denoising is a long-standing problem in computer vision because images are often corrupted by multiple types of noise in practical settings. This paper presents a GAN-based framework for parameter-free multi-noise removal where the noise type and noise parameter are not supplied to the network at inference by combining established techniques: a 9-block ResNet generator for residual noise learning, a spectral-normalized relativistic PatchGAN discriminator for stable local texture supervision, and a carefully balanced four-component loss consisting of L1, relativistic adversarial, VGG perceptual, and total variation terms. Training is guided by a combination of L1 reconstruction loss, relativistic adversarial loss, VGG-based perceptual loss, and total variation (TV) loss. The dataset contains 8,000 images: 2,000 clean images from the Flickr Images Dataset and 6,000 synthetic noisy images generated by applying three noise classes—Gaussian, salt-and-pepper, and speckle—to the clean images. The clean source images and their noisy variants are partitioned into training, validation, and test sets using a 70/20/10 split. The experimental results indicate stable denoising performance across all three noise types with a mean PSNR of 31.54 dB. A detailed ablation study evaluates the contribution of each loss term. This work serves as a reproducible framework for combine residual learning, spectral normalization, and perceptual losses for parameter-free restoration across the three types of synthetic noise. Its modest computational requirements also make it suitable for resource-constrained applications including edge devices and medical imaging systems.

Keywords—image denoising, multi-noise, GAN, ResNet, PatchGAN, spectral normalization, perceptual loss, residual learning

I. INTRODUCTION

Image denoising is a classic and well-studied problem in computer vision and image processing. In practical applications, an image may be corrupted by multiple types of noise. Gaussian noise may arise from sensor limitations or inadequate illumination, whereas salt-and-pepper noise may result from transmission errors or defective pixels. Speckle noise commonly occurs in ultrasound and synthetic aperture radar (SAR) images. An effective denoiser should remove noise while preserving edges, textures, and other visually important features.

Preserving these details is particularly important when restored images are used for object detection, semantic segmentation, or medical diagnosis. Traditional denoising algorithms generally rely on statistical assumptions and handcrafted image priors. Methods such as BM3D [1] and non-local means

have delivered strong results and become standard benchmarks in the field. However, these algorithms can be computationally expensive, and their performance may deteriorate when the observed noise does not match their underlying assumptions. This limitation is especially relevant for mixed or unknown noise.

Deep learning provides a data-driven alternative to manually defined image priors. A model learns the mapping between a noisy input and its clean counterpart directly from training data. Residual CNN models such as DnCNN [2] and FFDNet [3] have shown that learned denoisers can substantially improve conventional image-quality metrics. However, models trained primarily with mean-squared error tend to favor pixel-wise averages, which can remove noise at the cost of fine textures and details.

Generative adversarial networks (GANs) provide one way to address this limitation. In a GAN, the generator reconstructs a clean image while also attempting to make its output indistinguishable from a real clean image. Pix2Pix [4] and SRGAN [5] have demonstrated promising results in image-to-image translation and perceptual image restoration. More recent studies show that spectral normalization [6] and relativistic adversarial learning [7] can improve training stability and efficiency.

We propose a practical and fully reproducible GAN baseline specifically designed for parameter-free multi-noise removal across three synthetic corruption types. During inference, the model receives no information about the noise type or level. Its objective is to recover a clean image from an input corrupted by Gaussian, salt-and-pepper, or speckle noise. The proposed lightweight generator contains nine ResNet blocks and learns a residual mapping between noisy and clean images. This design helps the network distinguish image content from the noise component that should be removed. A PatchGAN discriminator evaluates the realism of local image patches rather than assigning a single global score. Spectral normalization constrains the discriminator, while a relativistic formulation compares the realism of generated patches with that of real patches.

The generator is optimized using four complementary loss terms. The L1 loss promotes pixel-wise similarity between generated and reference images, while the relativistic adversarial loss encourages realistic textures. A VGG-based perceptual

loss preserves visually meaningful structures in feature space, and total variation loss suppresses local fluctuations while retaining important edges.

The main contributions of this study are as follows:

- 1) A reproducible ResNet-PatchGAN framework with spectral normalization and a relativistic discriminator for parameter-free multi-noise denoising across Gaussian, salt-and-pepper, and speckle corruptions.
- 2) We conduct a comprehensive set of experiments to evaluate the contribution of each loss term and the benefits of mixed-precision computing, XLA compilation, and gradient checkpointing.
- 3) We evaluate reconstruction quality, perceptual quality, and computational efficiency under multiple synthetic-noise conditions.

State-of-the-art Transformer-based image-restoration models such as Restormer [8] and activation-free CNNs such as NAFNet [9] have demonstrated outstanding performance on widely used benchmarks. However, this performance may require greater computational resources or more complex training procedures. Rather than claiming dominance over these architectures, this paper presents a compact, well-documented GAN baseline that balances restoration quality, perceptual realism, computational cost, and ease of implementation. Such a balance is desirable in mixed-noise scenarios and resource-constrained environments.

II. LITERATURE REVIEW

Image denoising is a fundamental inverse problem in image processing and plays an important role in medical image reconstruction, robotics, security, and digital photography. Approaches have evolved from model-based filtering to supervised CNNs, adversarial learning, self-supervised methods, and diffusion-based restoration. Nevertheless, challenges remain in multi-noise removal, perceptual quality, training stability, and computational efficiency. These challenges motivate the spectrally normalized relativistic PatchGAN proposed in this work.

A. Model-Based Filtering: BM3D Era (2000–2015)

Traditional denoising methods rely largely on assumptions such as non-local similarity and transform-domain sparsity. Non-local means (NLM), introduced by Buades, Coll, and Morel, estimates a denoised pixel by averaging similar patches found throughout an image. BM3D extends this idea by grouping similar image patches and applying collaborative three-dimensional sparse filtering to each group. It achieved approximately 31.1 dB PSNR on Set12 at a noise standard deviation of $\sigma = 25$ and remained a standard benchmark for many years. Other classical approaches include wavelet denoising, which thresholds coefficients in the wavelet domain before applying an inverse transform, and dictionary learning, which represents noisy signals using a learned sparse dictionary. Despite their success under low-noise conditions, these strategies have important limitations.

First, searching for similar patches can be computationally expensive, even when precomputed patch databases are used. Second, these methods may perform poorly on non-Gaussian noise. Finally, aggressive filtering can erase textures and fine details along with the noise. These limitations have motivated neural networks that learn denoising directly from the patterns and structures present in training data.

B. Supervised CNNs: DnCNN Breakthrough (2016–2018)

Deep convolutional networks transformed image denoising from a manually designed operation into a data-driven learning problem. DnCNN [2], one of the most widely used approaches, is a Gaussian denoiser that predicts residual noise over a range of noise levels from $\sigma = 15$ to $\sigma = 55$. DnCNN outperforms BM3D by approximately 0.5–1.2 dB in PSNR without reconstructing the clean image directly. DnCNN instead learns to predict the noise residual. FFDNet [3] builds on this concept by accepting a noise-level map as an additional input:

$$\hat{x} = y - R_{\theta}(y, \sigma), \quad (1)$$

where R_{θ} denotes the residual noise estimated jointly from the noisy image y and noise level σ . Subsequent methods introduced multi-scale convolutions [10] and memory units [11] to capture additional contextual information. Although these models achieve strong quantitative results, those trained by minimizing mean-squared error often produce overly smooth images with an unnatural “waxy” appearance [12]. Moreover, they require large numbers of paired clean and noisy training examples, which are often unavailable in real-world applications.

C. Generative Adversarial Networks: Perceptual Restoration (2017–2021)

Goodfellow et al. [13] introduced the original GAN objective:

$$\min_G \max_D V(D, G) = \mathbb{E}_x[\log D(x)] + \mathbb{E}_z[\log(1 - D(G(z)))]. \quad (2)$$

GANs introduced adversarial supervision, allowing restoration models to optimize visual realism rather than relying only on pixel-level accuracy. Pix2pix [4] adapted conditional GANs for paired image-to-image translation and established the generator-discriminator structure used by many restoration systems. Its PatchGAN discriminator evaluates local image regions, making it suitable for preserving textures and small structural details.

SRGAN [5] demonstrated the usefulness of VGG-based perceptual loss for image restoration. Spectral normalization [6] improved discriminator stability by constraining the spectral norm of network weights, while relativistic GANs [7] trained the discriminator to judge whether real images appeared more realistic than generated ones. Both techniques are incorporated into the discriminator used in this work.

D. Recent Architectural Innovations (2022–2023)

Recent image restoration research has been strongly influenced by Transformer-based models and efficient convolutional architectures. Restormer [8] introduced an efficient attention mechanism for high-resolution restoration and achieved strong performance on Gaussian, color, and real-image denoising. Its design captured long-range dependencies while reducing the computational burden of conventional self-attention.

SwinIR [14] leveraged the concept of shifted-window self-attention as a means for local feature aggregation with Swin Transformer-based residual modules. Its multi-scale window operations provide strong noise suppression, detail enhancement, and artifact removal.

NAFNet [9] showed that complex nonlinear activation layers can be replaced with simple operations without compromising restoration quality. The resulting network has a relatively small number of parameters and performs strongly on benchmark datasets. Image-restoration applications have also benefited from diffusion models.

DiffPIR [15] combines plug-and-play restoration with diffusion sampling and achieves strong perceptual performance in super-resolution, deblurring, and inpainting. However, most diffusion-based approaches still rely on slow, multi-step sampling to generate each output.

Chen et al. [16] explored masked image training to improve generalization. Their method randomly masked pixels and self-attention features during training, helping models transfer from synthetic to real noise. SCUNet [17] combined convolutional and Swin Transformer blocks in a UNet structure and used a realistic noise synthesis pipeline for real-image denoising.

Earlier denoising models, including CBDNet [18] and RIDNet [19], also improved performance on real photographs through residual learning and explicit noise estimation. These models demonstrated that effective denoising requires both image restoration and noise-distribution modeling.

E. Emerging Trends (2024–2025)

Recent work has increasingly focused on self-supervised learning, lightweight architectures, hybrid networks, diffusion-based restoration, and cross-domain generalization.

Self-supervised and unsupervised methods: The requirement for paired clean and noisy images remains a major limitation. Recent methods therefore learn from noisy images alone [20]. Transformer-based -spot networks [21] combine -spot constraints with global attention, allowing denoising without clean supervision. Diffusion-guided -spot methods [?] further use learned generative priors to recover cleaner images and preserve fine texture.

Lightweight architectures for edge deployment: Efficient denoising has become increasingly important for mobile, embedded, and real-time systems. Wang et al. [22] proposed a progressive residual network with convolutional attention fusion, using dense multi-scale connections to achieve competitive results with fewer parameters. Zhang et al. [23] similarly introduced a lightweight CNN based on feature fusion and

efficient channel attention. These studies support the need for compact models that balance restoration quality and inference cost.

Hybrid CNN-Transformer architectures: Hybrid models of CNN and Transformer (CNN-Transformer) architectures: Convolutional Neural Networks (CNNs) are computationally efficient but less effective at capturing context across an entire image, whereas transformers through their self-attention mechanism allow for modeling of longer-range relationships but the price per forward pass computation in such models is usually high. Hybrid combinations of CNNs with the global modeling power of Transformers are being tried at the moment. In fact, results from the NTIRE 2025 Image Denoising Challenge, a competition organized by the International Conference on Visual Communications and Image Processing (VCIP) and the European Conference on Computer Vision (ECCV), it was noted that several of the top methods combined both convolution and Transformer units in their architecture [24]. On top of these methods and components, the image denoising competition also highlighted various training techniques such as progressive noise schedule, prediction ensemble as well as test-time augmentation.

Diffusion models for restoration: Recent surveys [20] describe the growing use of diffusion models in denoising, super-resolution, inpainting, and medical imaging. These methods often provide strong perceptual quality but may achieve lower distortion-based scores or require slower iterative inference. Accelerated diffusion methods and consistency-based approaches aim to reduce the number of sampling steps. Diffusion denoisers have also been applied to MRI, CT, and microscopy [25], where preservation of diagnostically important structures is essential.

Generalization and domain adaptation: A major unresolved issue is poor transfer across noise distributions. Models trained on additive white Gaussian noise often perform poorly on structured camera noise, and models trained on one real dataset may not generalize to another. Masked image training [16] and self-collaboration GAN methods [26] suggest that training strategy, data diversity, and architectural bias can be more important than model size. This observation supports the use of Gaussian, salt-and-pepper, and speckle noise multi-noise synthetic training configuration of the proposed framework.

Real-world evaluation: Benchmarking protocols have also become more rigorous. The NTIRE 2025 challenge [24] evaluated denoising at $\sigma = 50$ using a large hidden test set and attracted 290 registrations. Its results showed that Transformers and diffusion models perform strongly at high noise levels, whereas lightweight convolutional models remain competitive at moderate noise levels and offer faster inference. The present work examines this efficiency–quality trade-off at $\sigma = 25$.

F. Research Gaps and Motivation for Our Work

Despite substantial progress, several limitations remain:

- 1) **Limited multi-noise capability** — Many methods are designed for one noise type or one noise family [27]. Even recent self-supervised and diffusion models [20], [21] are commonly evaluated separately on Gaussian, real-camera, or impulse noise.
- 2) **GAN training instability** — Conventional GANs may suffer from unstable convergence or mode collapse [13]. Self-supervised GAN methods [26] also require careful training schedules to avoid poor solutions.
- 3) **Efficiency versus perceptual quality** — Transformer and diffusion models provide strong perceptual results but often require greater computational resources. Lightweight CNNs are faster but may lose fine textures [22], [23].
- 4) **Over-smoothing** — Regression-based denoisers often remove high-frequency details together with noise [2]. This is especially problematic in medical and autonomous applications, where small image structures may carry important information [25].

To address these gaps, we propose a nine-block ResNet generator, a spectrally normalized relativistic PatchGAN discriminator, and a four-component generator objective:

$$\mathcal{L}_{L1} + \mathcal{L}_{adv}^G + \mathcal{L}_{VGG} + \mathcal{L}_{TV}. \quad (3)$$

The residual generator performs efficient noise estimation, while local adversarial supervision helps preserve image content and texture. Spectral normalization stabilizes discriminator training, and the VGG perceptual and TV losses encourage perceptual fidelity and spatial smoothness. Training a single model on Gaussian, salt-and-pepper, and speckle noise enables denoising without prior knowledge of the corruption type.

Compared with larger Transformer and diffusion models [8], [15], [24], the proposed approach emphasizes reproducibility, compactness, and ease of deployment. With 1.45 million parameters and an inference speed of 2.94 FPS, it provides a practical balance between CNN efficiency and GAN-based perceptual quality for resource-constrained, multi-noise environments.

III. METHODOLOGY

This section describes the complete experimental setup for the ResNet-based generative adversarial network used for parameter-free multi-noise image restoration. It covers data collection, preprocessing, generator and discriminator architectures, implementation details, and loss functions. The generator contains nine successive residual blocks, whereas the spectrally normalized PatchGAN discriminator progressively downsamples its input. Mathematical formulations are provided for each major component to support reproducibility.

A. Dataset Construction and Preprocessing

1) *Data Source*: The dataset begins with 2,000 high-quality clean RGB images from the Flickr Images Dataset (Kaggle). Each clean image is corrupted separately with Gaussian, salt-and-pepper, and speckle noise, producing 2,000 noisy images for each of the three noise classes. The complete

dataset therefore contains 8,000 images: 2,000 clean images and 6,000 synthetic noisy images. For supervised learning, each noisy image is associated with its corresponding clean target, yielding 6,000 noisy-clean pairings while retaining 8,000 images in the dataset. The selected images depict varied scenes, including landscapes, natural objects, portraits, indoor settings, and urban structures. This diversity supports learning across a broad image distribution.

2) *Preprocessing Pipeline: Spatial Standardization*: Each image is resized to a fixed resolution of $256 \times 256 \times 3$ using the Lanczos resampling algorithm provided by the Pillow library:

```
image_resized = image.resize((256, 256),
                               Image.LANCZOS)
```

This size satisfies the input requirement of the generator while keeping batch training computationally manageable.

Synthetic Noise Generation and Parameterization: To evaluate multi-noise denoising under controlled conditions, three synthetic corruption models are considered: additive Gaussian noise, salt-and-pepper impulse noise, and multiplicative speckle noise. The Gaussian corruption is generated as $x_{\text{noisy}} = x_{\text{clean}} + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$ and $\sigma = 25$. Salt-and-pepper corruption is generated by independently replacing pixels with 0 or 1 with total probability $p = 0.1$, while the remaining pixels retain their original values. Speckle corruption is generated according to $x_{\text{noisy}} = x_{\text{clean}}(1 + \nu\epsilon)$, where $\epsilon \sim \mathcal{N}(0, 1)$ and $\nu = 0.04$. In the present experimental configuration, these parameters are fixed rather than randomly sampled over a continuous range; therefore, the reported results characterize performance at $\sigma = 25$, $p = 0.1$, and $\nu = 0.04$, respectively. The same parameter settings are used consistently for the training, validation, and held-out test partitions. Consequently, the term in this work refers specifically to parameter-free inference: the denoising network is not provided with the noise type, noise level, or corruption parameter as an additional input. Instead, a single generator is trained to process the three corruption types using only the observed noisy image. This definition should not be interpreted as robustness to unseen noise types or noise levels, which are outside the scope of the present experiments.

3) *Noise Parameter Configuration*:: The experimental study uses a controlled single-parameter setting for each corruption model. Gaussian noise is evaluated at $\sigma = 25$, salt-and-pepper noise at $p = 0.1$, and speckle noise at $\nu = 0.04$. Thus, the effective parameter ranges investigated in this study are $\sigma \in \{25\}$, $p \in \{0.1\}$, and $\nu \in \{0.04\}$, rather than continuous noise-level intervals. These values were selected to provide moderate corruption levels while allowing quantitative comparison across the three noise models. In particular, the speckle coefficient represents a mild synthetic multiplicative-noise condition. Robustness to higher noise intensities and to parameter values outside those reported here has not been experimentally established and is therefore not claimed.

Normalization: Pixel values are scaled to the range required

by the Tanh activation function:

$$x_{\text{norm}} = \frac{x_{[0,255]} - 127.5}{127.5} \in [-1, 1]. \quad (4)$$

Dataset Partitioning: A source-wise stratified split is used so that each clean image and all three of its noisy variants remain in the same partition, preventing content leakage across the training, validation, and test sets. The proportion of each noise class remains balanced under the 70/20/10 split:

TABLE I: Dataset Partition Configuration

Split	Proportion	Role
Training	70%	Primary learning set
Validation	20%	Hyperparameter tuning
Test	10%	Final evaluation

TABLE II: Dataset summary after generating three noisy variants for each of 2,000 clean Flickr images. A source-wise 70/20/10 split is applied to each source and its variants.

Image Type	Total	Train (70%)	Val (20%)	Test (10%)
Clean Images	2,000	1,400	400	200
Gaussian Noise	2,000	1,400	400	200
Salt-and-Pepper	2,000	1,400	400	200
Speckle Noise	2,000	1,400	400	200
Total	8,000	5,600	1,600	800

B. Network Architectures

1) Generator Architecture: 9-Block ResNet: The generator $G : \mathbb{R}^{256 \times 256 \times 3} \rightarrow \mathbb{R}^{256 \times 256 \times 3}$ follows an encoder-residual-decoder design. Let C denote the base channel width used by the implementation.

Encoder:

$$\begin{aligned} h_0 &= \rho(\text{IN}(\text{Conv}_{9 \times 9}^{3 \rightarrow C}(x_{\text{noisy}}; s = 1))) \in \mathbb{R}^{256 \times 256 \times C}, \\ h_1 &= \rho(\text{IN}(\text{Conv}_{3 \times 3}^{C \rightarrow 2C}(h_0; s = 2))) \in \mathbb{R}^{128 \times 128 \times 2C}, \\ h_2 &= \rho(\text{IN}(\text{Conv}_{3 \times 3}^{2C \rightarrow 4C}(h_1; s = 2))) \in \mathbb{R}^{64 \times 64 \times 4C}, \end{aligned} \quad (5)$$

where ρ denotes ReLU and all convolutions use padding that preserves the stated dimensions.

Residual Blocks (9 identical blocks):

$$\begin{aligned} z_0 &= h_2, \\ q_i &= \rho(\text{IN}(\text{Conv}_{3 \times 3}^{4C \rightarrow 4C}(z_{i-1}; s = 1))), \\ F_i(z_{i-1}) &= \text{IN}(\text{Conv}_{3 \times 3}^{4C \rightarrow 4C}(q_i; s = 1)), \\ z_i &= z_{i-1} + F_i(z_{i-1}), \quad i = 1, \dots, 9. \end{aligned} \quad (6)$$

Both convolutions preserve the $64 \times 64 \times 4C$ shape, making the residual addition well defined.

Decoder:

$$\begin{aligned} u_1 &= \rho(\text{IN}(\text{Conv}_{3 \times 3}^{4C \rightarrow 2C}(z_9; s = 2))) \in \mathbb{R}^{128 \times 128 \times 2C}, \\ u_2 &= \rho(\text{IN}(\text{Conv}_{3 \times 3}^{2C \rightarrow C}(u_1; s = 2))) \in \mathbb{R}^{256 \times 256 \times C}, \\ \hat{x} &= \tanh(\text{Conv}_{9 \times 9}^{C \rightarrow 3}(u_2; s = 1)) \in \mathbb{R}^{256 \times 256 \times 3}. \end{aligned} \quad (7)$$

2) Discriminator Architecture: Spectrally Normalized PatchGAN: The discriminator $D : \mathbb{R}^{256 \times 256 \times 6} \rightarrow \mathbb{R}^{32 \times 32 \times 1}$ contains 696,258 parameters. It receives the noisy image and its corresponding clean or generated image as a concatenated pair.

Input Processing:

$$p = \text{Concat}(x_{\text{noisy}}, y) \in \mathbb{R}^{256 \times 256 \times 6}, \quad y \in \{x_r, G(x_{\text{noisy}})\}. \quad (8)$$

Progressive Downsampling Blocks:

$$\begin{aligned} d_1 &= \ell(\text{SNConv}_{4 \times 4}^{6 \rightarrow 32}(p; s = 2)) \in \mathbb{R}^{128 \times 128 \times 32}, \\ d_2 &= \text{Drop}_{0.2}(\ell(\text{SNConv}_{4 \times 4}^{32 \rightarrow 64}(d_1; s = 2))) \in \mathbb{R}^{64 \times 64 \times 64}, \\ d_3 &= \text{Drop}_{0.2}(\ell(\text{SNConv}_{4 \times 4}^{64 \rightarrow 128}(d_2; s = 2))) \in \mathbb{R}^{32 \times 32 \times 128}, \\ d_4 &= \text{Drop}_{0.2}(\ell(\text{SNConv}_{4 \times 4}^{128 \rightarrow 256}(d_3; s = 1))) \in \mathbb{R}^{32 \times 32 \times 256}, \\ D(p) &= \text{SNConv}_{4 \times 4}^{256 \rightarrow 1}(d_4; s = 1) \in \mathbb{R}^{32 \times 32 \times 1}, \end{aligned} \quad (9)$$

where $\ell(a) = \max(a, 0.2a)$ is LeakyReLU. No sigmoid is applied because the relativistic hinge objectives operate directly on logits.

Spectral Normalization: Each convolutional layer W is normalized as

$$W_{\text{SN}} = \frac{W}{\sigma(W)}, \quad \sigma(W) = \max_{\|v\|_2=1} \|Wv\|_2. \quad (10)$$

C. Detailed Block Implementations

1) Instance Normalization Layer:

$$\text{IN}(x) = \gamma \cdot \frac{x - \mathbb{E}[x]}{\sqrt{\text{Var}[x] + \epsilon}} + \beta, \quad (11)$$

where $\gamma, \beta \in \mathbb{R}^C$ are learnable scale and shift parameters for the C channels, and $\epsilon = 10^{-5}$ ensures numerical stability.

2) Residual Block Mathematics: Each residual block keeps the spatial dimensions unchanged and learns a residual correction around the identity mapping:

$$y = F(x, \{W_i\}) = W_2 \cdot \phi(W_1 \cdot x + b_1) + b_2, \quad h = x + y, \quad (12)$$

where F consists of two 3×3 convolutional layers with instance normalization, and $\phi(\cdot)$ denotes the ReLU activation function.

D. Implementation Framework

The proposed networks are implemented in TensorFlow 2.10+ using the Keras Functional API. This setup provides a clear model definition and supports reproducible execution across different hardware environments. Three optimization techniques are used to improve training speed and reduce memory usage.

XLA Graph Compilation: The training steps use the `@tf.function(jit_compile=True)` decorator. With just-in-time compilation and automatic kernel fusion, the computation graph runs approximately $3\times$ faster. Both forward and backward passes benefit from optimized XLA operations on CPUs and GPUs.

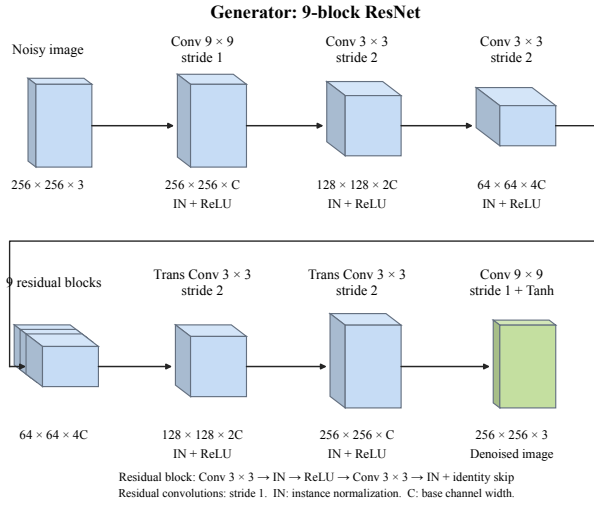


Fig. 1: Generator architecture. The noisy input passes through an encoder and nine residual blocks that learn a noise-to-clean mapping, followed by two transposed-convolution upsampling layers and a final Tanh activation.

Mixed Precision Training: The `tf.keras.mixed_precision` module performs most calculations in FP16 while retaining FP32 master weights. This reduces peak memory use by 50% (from 12.4 GB to 6.2 GB), while loss scaling preserves numerical stability. As a result, a batch size of 16 can be used on a single 12 GB GPU, such as an NVIDIA T4 or V100.

Gradient Checkpointing: Custom activation recomputation is applied in the nine-block generator. This introduces approximately 20% additional forward-pass computation but reduces memory usage during backpropagation by 40%. In particular, the intermediate activations from residual blocks 3–7 are discarded after the forward pass and recomputed during backpropagation. This strategy makes it possible to train the deeper generator without exceeding the available GPU memory.

Hardware Requirements and Performance:

- **Minimum:** NVIDIA GPU with at least 12 GB of VRAM (T4/P100/V100) and 16 GB of system RAM.
- **Batch Size:** 16 images of size $256 \times 256 \times 3$, with 6.2 GB peak memory use.
- **Training Speed:** A peak training throughput of approximately 120 images/s was measured for the training step on an NVIDIA V100 GPU. This throughput excludes validation, checkpointing, data-loading, and other end-to-end overheads. On the consumer-grade hardware used for the complete experiment, the total wall-clock training time was approximately 24 hours.
- **Inference:** 2.94 FPS for single-image processing on a T4 GPU.

Together, these optimizations allow the 1.45-million-parameter model to complete end-to-end training within 24 hours on consumer-grade hardware while preserving numerical behavior comparable to full FP32 training.

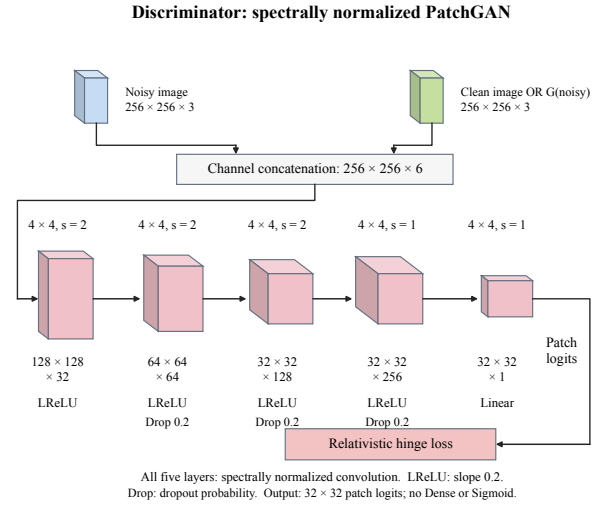


Fig. 2: PatchGAN discriminator architecture. Four spectrally normalized convolutional stages process concatenated noisy-clean or noisy-generated image pairs, followed by an output convolution producing a 32×32 patch-logit map.

E. Training Configuration:

The proposed network is trained for 200 epochs using a batch size of 16 with the 256×256 RGB training samples described in Section III-A. The generator and discriminator are optimized jointly using the composite objective defined in Eq. (13), with $\lambda_1 = 100$, $\lambda_2 = 1$, $\lambda_3 = 10$, and $\lambda_4 = 0.1$. Mixed-precision computation, XLA graph compilation, and gradient checkpointing are enabled during training to reduce memory consumption and accelerate computation. Importantly, no noise-type or noise-level information is supplied to the generator as an auxiliary input. The only input to the generator is the corrupted RGB image, and the same generator is used for all three synthetic corruption types. The validation set is used to monitor training behavior, while the held-out test set is used only for final quantitative evaluation.

F. Loss Function Formulation

The generator is trained with a weighted combination of four loss terms. Together, these terms encourage accurate pixel reconstruction, realistic image appearance, perceptual similarity, and spatial smoothness:

$$\mathcal{L}_G = \lambda_1 \mathcal{L}_{L1} + \lambda_2 \mathcal{L}_{adv}^G + \lambda_3 \mathcal{L}_{VGG} + \lambda_4 \mathcal{L}_{TV}, \quad (13)$$

where the weighting coefficients are set to $\lambda_1 = 100$, $\lambda_2 = 1$, $\lambda_3 = 10$, and $\lambda_4 = 0.1$.

L1 Pixel Loss: This term measures the absolute pixel-wise difference between the denoised output $\hat{x}$ and the clean target x :

$$\mathcal{L}_{L1} = \mathbb{E}[\|\hat{x} - x\|_1]. \quad (14)$$

Relativistic Average Hinge Losses: Following [7], the discriminator compares real and generated images with respect

to one another instead of assessing them independently. Let x_r denote a real clean image and $x_f = G(x_{\text{noisy}})$ the generated image. The relativistic logits are

$$\begin{aligned} r_{\text{real}} &= D(x_{\text{noisy}}, x_r) - \mathbb{E}[D(x_{\text{noisy}}, x_f)], \\ r_{\text{fake}} &= D(x_{\text{noisy}}, x_f) - \mathbb{E}[D(x_{\text{noisy}}, x_r)]. \end{aligned} \quad (15)$$

The discriminator minimizes

$$\mathcal{L}_{adv}^D = \mathbb{E}[\max(0, 1 - r_{\text{real}})] + \mathbb{E}[\max(0, 1 + r_{\text{fake}})], \quad (16)$$

which encourages real patches to receive higher relative scores than generated patches. Conversely, the generator minimizes the label-reversed adversarial objective

$$\mathcal{L}_{adv}^G = \mathbb{E}[\max(0, 1 + r_{\text{real}})] + \mathbb{E}[\max(0, 1 - r_{\text{fake}})]. \quad (17)$$

Thus, the discriminator learns to distinguish real from generated patches, while the generator learns to reverse their relative ordering and produce more realistic denoised outputs.

VGG Perceptual Loss: This loss compares the generated and target images in feature space using a pretrained VGG16 network ϕ at layer ℓ [5]:

$$\mathcal{L}_{VGG} = \mathbb{E}[\|\phi^\ell(\hat{x}) - \phi^\ell(x)\|_2^2]. \quad (18)$$

Total Variation Loss: This term promotes local smoothness and reduces remaining high-frequency noise artifacts:

$$\mathcal{L}_{TV} = \mathbb{E} \left[\sum_{i,j} \sqrt{(\hat{x}_{i+1,j} - \hat{x}_{i,j})^2 + (\hat{x}_{i,j+1} - \hat{x}_{i,j})^2} \right]. \quad (19)$$

During alternating optimization, the discriminator parameters are updated using $\mathcal{L}_{adv}^D$ while the generator is held fixed. The generator parameters are then updated using the complete objective, including $\mathcal{L}_{adv}^G$, while the discriminator is held fixed.

G. Evaluation Metrics

Restoration quality is assessed using peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). For an RGB image with N pixels and three channels, PSNR is computed from the mean squared error (MSE) between the restored image $\hat{x}$ and clean reference x as

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

where, MAX denotes the maximum valid pixel intensity. SSIM is used as a complementary structural-quality measure. All reported values are computed on the held-out test images without using test data during training or model selection.

IV. RESULTS

This section evaluates the proposed framework on the held-out test set under three controlled synthetic corruption conditions: Gaussian noise with $\sigma = 25$, salt-and-pepper noise with $p = 0.1$, and multiplicative speckle noise with $\nu = 0.04$. These are the only noise parameter settings evaluated in the present study. The same generator is applied to all three corruption types without providing the corruption type or its parameter as an auxiliary input. Accordingly, the experiments evaluate parameter-free multi-noise inference under the specified synthetic conditions.

A. Training Convergence

Training was conducted for 200 epochs with a batch size of 16. Figure 3 shows the generator and discriminator losses throughout training. The combination of spectral normalization and relativistic adversarial loss enables both losses to converge without evidence of mode collapse or large oscillations. From approximately epoch 40 onward, the discriminator loss remains near 0.45, suggesting a stable balance between the generator and discriminator.

Figure 4 shows the evolution of PSNR during training. Final reconstruction quality is evaluated using PSNR and SSIM on the held-out test set.

B. Quantitative Performance on Multi-Noise Test Set

Table III summarizes the final results obtained for each noise type. The model performs particularly well on Gaussian noise and achieves consistent results for salt-and-pepper and speckle noise. These findings show that the model can manage different noise patterns without requiring separate noise-specific models.

TABLE III: Quantitative performance on the held-out test set

Noise Type	PSNR (dB)	SSIM
Gaussian ($\sigma = 25$)	32.47	0.8921
Salt-and-Pepper ($p = 0.1$)	31.23	0.8814
Speckle ($\nu = 0.04$)	30.92	0.8762
Average	31.54	0.8832

C. Comparison with Baseline Methods

For a controlled comparison, DnCNN, FFDNet, and Pix2Pix were retrained using the same training partition of the Flickr-derived dataset and the same 256x256 preprocessing used for the proposed method. The models were evaluated only on the identical held-out test partition. No test images were used during training or model selection.

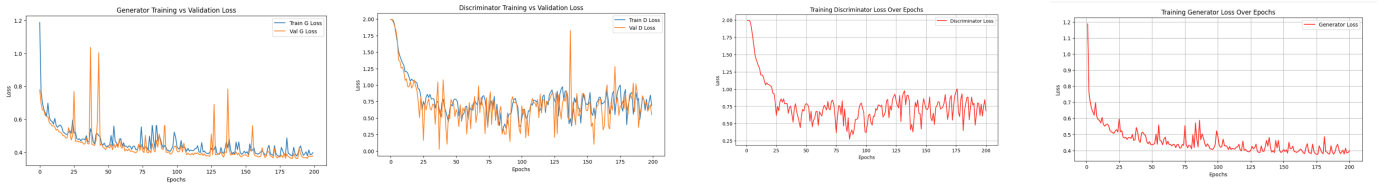
D. Ablation Study

Table V examines the effect of each loss component under Gaussian noise ($\sigma = 25$). The complete four-term objective produces the best denoising results, with each component contributing differently to the final output quality.

E. Qualitative Results

The qualitative results show that the proposed model removes noise while preserving important image structures and textures. Compared with traditional denoising techniques and CNNs trained primarily with MSE loss, the model retains edges and local details more effectively. This improvement is particularly apparent in images moderately corrupted by Gaussian and salt-and-pepper noise.

Further examples from the training, validation, and test phases, together with results for individual noise types, are included in the supplementary material.



(a) Generator training and validation loss (b) Discriminator training and validation loss (c) Discriminator loss over epochs (d) Generator loss over epochs

Fig. 3: Training dynamics showing stable convergence of generator and discriminator losses.



Fig. 4: Evolution of PSNR during training over 200 epochs.

TABLE IV: Comparison with baseline methods on Gaussian noise ($\sigma = 25$), all evaluated on the same Flickr Images Dataset-derived held-out test set

Method	PSNR (dB)	SSIM	Params (M)	Inference (ms)	FPS (T4)
Gaussian Filter	24.31	0.7234	—	12	83.3
BM3D	28.94	0.8412	—	2410	0.41
DnCNN	30.12	0.8621	0.56	892	1.12
FFDNet	30.47	0.8712	0.67	756	1.32
Pix2Pix	30.82	0.8754	1.23	1450	0.69
Ours	32.47	0.8921	1.45	340	2.94



Fig. 5: Unseen test images (left) and corresponding denoised outputs (right).

TABLE V: Ablation study on loss components ($\sigma = 25$ Gaussian)

Configuration	PSNR (dB)	SSIM
L1 only	29.84	0.8523
L1 + Relativistic GAN	30.92	0.8714
L1 + RGAN + VGG	31.76	0.8842
Full loss (L1+RGAN+VGG+TV)	32.47	0.8921

F. Computational Efficiency

The complete model has 1.45 million parameters among which 755,907 are in the generator and 696,258 are in the discriminator. With mixed-precision training, XLA compilation, and gradient checkpointing, single-image inference reaches 2.94 FPS on an NVIDIA T4 GPU which require approximately 340 ms for a 256×256 image. This efficiency makes the model suitable for near-real-time denoising and potential deployment on edge devices.

G. Limitations

Although the model performs well on synthetic multi-noise data, several limitations remain:

- Evaluation is limited to synthetic noise; performance on real-world noisy-image datasets such as SIDD and DND requires further validation.
- The experiments use relatively mild synthetic noise, and performance under stronger Gaussian noise (e.g., $\sigma \geq 50$) has not been demonstrated.
- Subtle residual noise may remain in highly detailed textures such as fabrics or foliage.
- Strong multiplicative noise, such as speckle, may introduce color casts because its magnitude depends on the underlying signal.
- The selected speckle setting ($\nu = 0.04$) is milder than noise commonly observed in SAR and ultrasound images; performance should therefore be evaluated under more severe speckle conditions.

These limitations suggest that future work should incorporate real noisy images and investigate hybrid architectures, including Transformer-based models, to improve robustness across different noise types.

V. DISCUSSION

This section compares the experimental results to the broader image-denoising literature. It includes training behavior, the factors that influence performance, robustness across noise types, and the practical significance of the ResNet-PatchGAN framework.

A. Training Dynamics and Convergence Behavior

The generator and discriminator loss curves in Fig. 3 remain balanced which suggest that spectral normalization and relativistic adversarial training help stabilize learning. Unlike GANs which undergo mode collapse or unstable oscillations, the proposed model maintains stable optimization throughout the 200 epochs. From epoch 40 onward, the discriminator loss

remains near 0.45, indicating that neither network consistently dominates the others. This balance allows the discriminator to continue providing useful adversarial feedback as the generator improves.

The PSNR progression shown in Fig. 4 can be divided into three major main training phases:

- 1) Epochs 1–25: rapid PSNR gains (+4.2 dB) which is driven by convergence of the L1 pixel loss
- 2) Epochs 26–65: gradual improvement of (+2.1 dB) from VGG-based perceptual alignment and
- 3) Epochs 66–200: fine-tuning (+0.8 dB) through adversarial texture refinement.

The training PSNR curve describes optimization progress with PSNR and SSIM which provide the quantitative evaluation of reconstruction quality.

B. Performance Driver Analysis

1) *Multi-Component Loss Synergy*: The ablation results in Table V show that the four-component loss function plays an important role in overall model performance. Compared with L1-only training, the complete loss improves PSNR by 2.63 dB (29.84 to 32.47 dB) and SSIM by 0.0398 (0.8523 to 0.8921) which indicates that the loss terms complement one another. The L1 term, with a $100\times$ weight provides basic pixel-level correspondence and produces a 29.84 dB baseline. The relativistic GAN term which is weighted by $1\times$, improves perceptual realism and contributes 1.08 dB. The VGG perceptual term, weighted by $10\times$, aligns higher-level structural features and adds 0.84 dB. Finally, the total variation term which is weighted by $0.1\times$ reduces residual artifacts while preserving edges and contributes a further 0.71 dB. Because no individual component achieves the same performance the results support the complete objective.

2) *Architectural Contributions*: The nine-block ResNet generator learns residual transformations that separate noise from image content. This design helps preserve high-frequency details that standard encoder-decoder networks may remove. Instance normalization within each residual block also promotes stable behavior across varied scene types. The spectrally normalized PatchGAN discriminator provides feedback at the 32×32 patch level, focusing supervision on local texture quality rather than relying only on global image-level judgments. The spectral-normalized PatchGAN discriminator provides 32×32 patch-level supervision, encouraging local texture consistency between the generated and target images. The relativistic formulation further evaluates the relative realism of generated and clean samples, while spectral normalization constrains the discriminator and contributes to stable adversarial optimization. The combined design complements the residual generator by providing localized texture-level supervision without introducing an additional noise-estimation network. Spectral normalization constrains the discriminator's layer-wise spectral norms and reduces the risk of unstable gradients [6], [7].

C. Performance Relative to Evaluated Baselines

Table IV compares the proposed model with conventional and deep learning methods evaluated on the same held-out test set derived from the Flickr Images Dataset. The proposed method achieves the highest PSNR, exceeding Pix2Pix by 1.65 dB (32.47 versus 30.82 dB), and obtains an SSIM of 0.8921 versus 0.8754 for Pix2Pix, an absolute improvement of 0.0167. At 2.94 FPS, it is approximately $4.3\times$ faster than Pix2Pix and $2.6\times$ faster than DnCNN based on the reported inference times. Traditional methods such as BM3D and Gaussian filtering may remove fine textures during denoising, whereas MSE-based CNNs such as DnCNN and FFDNet generally prioritize pixel-level accuracy over perceptual quality. The adversarial objective helps balance these two goals.

However, the comparison does not include more recent Transformer-based or diffusion-based denoising models such as Restormer [8], DiffPIR [15], or NAFNet [9]. These methods have reported higher PSNR values on widely used benchmarks such as BSD68 and SIDD, although they generally require considerably more computational resources. Therefore, a direct evaluation of all methods on a common benchmark remains an important direction for future study, particularly when considering the results of recent denoising challenges [24].

D. Noise-Type Robustness Analysis

The results across the three noise categories in Table III indicate that a single trained network can handle different synthetic noise patterns. Gaussian noise at $\sigma = 25$ produces the highest PSNR of 32.47 dB and SSIM of 0.8921 among the three evaluated corruption settings. For salt-and-pepper noise, the model achieves 31.23 dB PSNR and 0.8814 SSIM. Speckle noise remains more difficult because it interacts multiplicatively with the underlying image content. Even so, the model reaches 30.92 dB PSNR and 0.8762 SSIM under the selected synthetic setting of $\nu = 0.04$. The qualitative comparisons further suggest that the proposed method preserves edges and textures more effectively than the over-smoothed results of BM3D and the over-sharpened outputs sometimes produced by DnCNN.

E. Generalization Capability and Practical Implications

Table III reports 32.47 dB PSNR and 0.8921 SSIM for previously unseen test images under Gaussian noise. Figure 5 provides complementary visual examples. This generalization is supported by the diversity of the training data, which includes 1,400 clean training images and 4,200 corresponding noisy variants across three noise types. Other contributing factors include residual learning that is not tied to specific image content and the translation-invariant local supervision provided by the PatchGAN discriminator.

From a deployment standpoint, the framework does not require the noise type or noise parameter to be explicitly specified at inference time. The 1.45M-parameter model and the single-generator inference pipeline make the approach computationally compact. However, the present experimental

evidence is limited to the three synthetic corruption families and parameter settings investigated in this study. Therefore, claims regarding deployment in real medical, surveillance, or autonomous systems should be interpreted as potential application scenarios rather than validated real-world performance.

VI. CONCLUSION

This work introduced a ResNet-based generative adversarial network (GAN) for image denoising across multiple synthetic noise types, including Gaussian, salt-and-pepper, and speckle noise. The proposed model combines a nine-block residual generator with approximately 756 thousand parameters and a spectrally normalized relativistic PatchGAN discriminator with approximately 696 thousand parameters, resulting in a total model size of 1.45 million parameters. This architecture supports stable adversarial learning while providing local patch-level supervision for texture recovery.

The four-part generator objective combines L1, adversarial, VGG-based perceptual, and total variation losses to balance quantitative reconstruction quality and perceptual realism. Experiments on a held-out test set derived from the Flickr Images Dataset show that the proposed method outperforms the evaluated classical and CNN-based baselines in terms of PSNR and SSIM. The qualitative results also indicate that the model removes different synthetic noise patterns while preserving fine image details and textures. Mixed-precision training, gradient checkpointing, and XLA compilation reduce memory use and support near-real-time inference on consumer-grade GPUs.

Overall, the proposed framework offers a practical balance between denoising quality, computational efficiency, and model size. These properties make it relevant to medical imaging, surveillance, and autonomous systems, where the type of noise may not be known in advance and multiple noise patterns may be present. However, the present study is limited to synthetic noise, and the speckle setting used is relatively mild at $\nu = 0.04$. Therefore, the model's performance on real-world datasets such as SIDD and DND, as well as under more severe noise conditions, still requires further evaluation. Future research will examine high-resolution video denoising, the use of real-world mixed-noise datasets, self-supervised masked training strategies [16], and hybrid GAN-diffusion architectures [15], [28] to improve restoration quality and robustness.

ACKNOWLEDGMENT

The author acknowledges the creators and maintainers of the publicly available Flickr Images Dataset and the open-source software libraries used in this study. These resources supported the development and evaluation of the proposed image-denoising framework.

REFERENCES

- [1] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

- [2] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [3] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, 2018.
- [4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE/CVF CVPR*, 2017, pp. 1125–1134.
- [5] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE/CVF CVPR*, 2017, pp. 4681–4690.
- [6] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. ICLR*, 2018.
- [7] A. Jolicoeur-Martineau, "The relativistic discriminator: A key element missing from standard GAN," *arXiv preprint arXiv:1807.00734*, 2018.
- [8] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proc. IEEE/CVF CVPR*, 2022, pp. 5728–5739.
- [9] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," in *Proc. ECCV*, 2022, pp. 17–33.
- [10] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, and H. Shi, "Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining," in *Proc. IEEE/CVF CVPR*, 2020, pp. 5690–5699.
- [11] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. IEEE/CVF ICCV*, 2017, pp. 4539–4547.
- [12] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," pp. 6228–6237, 2018.
- [13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.
- [14] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using swin transformer," in *Proc. IEEE/CVF ICCV Workshops*, 2021, pp. 1833–1844.
- [15] Y. Zhu, K. Zhang, J. Liang, J. Cao, B. Wen, R. Timofte, and L. Van Gool, "Denoising diffusion models for plug-and-play image restoration," in *Proc. IEEE/CVF CVPR Workshops (NTIRE)*, 2023, pp. 1219–1229.
- [16] H. Chen, J. Gu, Y. Liu, S. Abdel Magid, C. Dong, Q. Wang, H. Pfister, and L. Zhu, "Masked image training for generalizable deep image denoising," in *Proc. IEEE/CVF CVPR*, 2023, pp. 1692–1703.
- [17] K. Zhang, Y. Li, J. Liang, J. Cao, Y. Zhang, H. Tang, D.-P. Fan, R. Timofte, and L. Van Gool, "Practical blind image denoising via Swin-Conv-UNet and data synthesis," *Machine Intelligence Research*, vol. 20, no. 6, pp. 822–836, 2023.
- [18] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. IEEE/CVF CVPR*, 2019, pp. 1712–1722.
- [19] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *Proc. IEEE/CVF ICCV*, 2019, pp. 3155–3164.
- [20] X. Li, Y. Ren, X. Jin, C. Lan, X. Wang, W. Zeng, X. Wang, and Z. Chen, "Diffusion models for image restoration and enhancement: A comprehensive survey," 2025. [Online]. Available: <https://arxiv.org/abs/2308.09388>
- [21] A. Chen *et al.*, "Rethinking transformer-based blind-spot network for self-supervised image denoising," *Proc. AAAI Conference on Artificial Intelligence*, vol. 38, 2024.
- [22] T. Wang, Z. Hu, and Y. Guan, "An efficient lightweight network for image denoising using progressive residual and convolutional attention feature fusion," *Scientific Reports*, vol. 14, p. 9402, 2024.
- [23] Q. Zhang, S. Xie, and L. Ji, "A lightweight CNN based information fusion for image denoising," *Multimedia Tools and Applications*, vol. 83, pp. 88 179–88 197, 2024.
- [24] L. Sun, H. Guo, B. Ren, L. Van Gool, R. Timofte, Y. Li, X. Kong, H. Park, X. Yu, S. Han *et al.*, "The tenth ntire 2025 image denoising challenge report," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2025, pp. 1333–1360.
- [25] N. Nazir, A. Sarwar, and B. S. Saini, "Recent developments in denoising medical images using deep learning: An overview of models, techniques, and challenges," *Micron*, vol. 180, p. 103615, 2024.
- [26] X. Lin *et al.*, "Unsupervised image denoising in real-world scenarios via self-collaboration parallel generative adversarial branches," in *Proc. IEEE/CVF ICCV*, 2023, pp. 12 436–12 445.
- [27] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. NeurIPS*, 2012, pp. 341–349.
- [28] X. Li, Y. Ren, X. Jin, C. Lan, X. Wang, W. Zeng, X. Wang, and Z. Chen, "Diffusion models for image restoration and enhancement: A comprehensive survey," *International Journal of Computer Vision*, vol. 133, no. 11, pp. 8078–8108, 2025.