BİNGÖL UNIVERSITY

Turkish Journal of Nature and Science

e-ISSN: 2149-6366
www.dergipark.gov.tr/tdfd

Research Article

Volume 15, Issue 2, Page 83-91, 2026

Received: 26.01.2026
Accepted: 03.03.2026
Published: 30.06.2026

DOI: https://doi.org/10.46810/tdfd.1872328

IGHF-DL: Parameter-Efficient Hybrid Image Denoising Using Classical Mathematical Filtering and Deep Learning

Fatih MARAŞLI[1*], Serkan ÖZTÜRK[2]

[1]Tokat Gaziosmanpaşa University, Pazar Vocational School, Department of Computer Technology, Tokat, Türkiye
[2]Erciyes University, Faculty Of Engineering, Department of Computer Engineering, Kayseri, Türkiye

**Corresponding author: fatih.marasli@gop.edu.tr*

Abstract

There is a critical balance between denoising and detail preservation that requires careful consideration. There is also a need for mathematically grounded, computationally efficient denoising models that are practical for real-world applications. In this study, we propose the IGHF-DL model, which improves the mathematical edge preservation power of the classic Inverse Gaussian Harmonic Filter (IGHF) by incorporating a CNN structure that includes six residual blocks, a channel attention mechanism, and an edge-sensitive enhancement block. In our model, training was performed with a low computational cost, reducing the number of DnCNN parameters by 54%. In comprehensive experiments performed on the BSD68, CBSD68, McMaster, Kodak24, Set12, and Urban100 datasets, our model achieved a PSNR value of 28.72 dB for BSD68, outperforming the BM3D (+0.15 dB) model and reaching 98.3% of the DnCNN performance, demonstrating competitive performance. Our model is successful not only in terms of metrics but also in terms of edge preservation and perceptual quality. On datasets with textured images such as Urban100, IGHF-DL outperforms the IGHF model in edge preservation and perceptual quality with FOM: 0.8252 and LPIPS: 0.1148, demonstrating its robustness on a mathematical basis and showing potential for further development in integration with next-generation methods. At the same time, compared to the IGHF model, IGHF-DL showed the best improvement in metrics at high noise levels with a +7.44 dB improvement. The proposed hybrid approach offers a practical and efficient solution with parameter efficiency that reduces computational costs for resource-constrained environments.

Keywords

IGHF Image Denoising, Deep Learning, Hybrid CNN, Perceptual Quality and Edge Preservation, Parameter Efficiency

BİNGÖL ÜNİVERSİTESİ

Türk Doğa ve Fen Dergisi

e-ISSN: 2149-6366
www.dergipark.gov.tr/tdfd

Araştırma Makalesi

Cilt 15, Sayı 2, Sayfa 83-91, 2026

Alınış: 26.01.2026
Kabul: 03.03.2026
Basım: 30.06.2026

DOI: https://doi.org/10.46810/tdfd.1872328

IGHF-DL: Klasik Matematiksel Filtreleme ve Derin Öğrenme ile Parametre Verimli Hibrit Görüntü Gürültü Giderme

Fatih MARAŞLI[1*], Serkan ÖZTÜRK[2]

[1]Tokat Gaziosmanpaşa Üniversitesi, Pazar Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Tokat, Türkiye
[2]Erciyes Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Kayseri, Türkiye

**Sorumlu yazar: fatih.marasli@gop.edu.tr*

Özet

Görüntü gürültüsünün giderilmesi ile ayrıntının korunması arasında dikkat gerektiren bir denge bulunmaktadır. Aynı zamanda matematiksel temelli ve pratikte kullanılabilen düşük hesaplama maliyetli gürültü giderme modellerine ihtiyaç duyulmaktadır. Bu çalışmada klasik Ters Gauss Harmonik Filtreleme (Inverse Gaussian Harmonic Filter- IGHF) filtresinin matematiksel kenar koruma gücünü altı artık blok, kanal dikkat mekanizması ve kenar duyarlı iyileştirme bloğu içeren CNN yapısı ile iyileştiren IGHF-DL modelini öneriyoruz. Modelimizde, DnCNN parametre sayısının %54'ü kadarıyla düşük hesaplama maliyeti sağlayan bir eğitim gerçekleştirilmiştir. BSD68, CBSD68, McMaster, Kodak24, Set12 ve Urban100 veri setleri üzerinde yapılan kapsamlı deneylerde modelimiz BSD68 için 28.72 dB PSNR değeri elde ederek BM3D (+0.15 dB) modelini geçmiş ve DnCNN performansının %98.3'üne ulaşarak rekabetçi performans göstermektedir. Modelimiz sadece metrik olarak değil aynı zamanda kenar koruma ve algısal kalite açısından da başarılıdır. Urban100 gibi dokulu görüntülerin olduğu veri setinde IGHF modeline göre IGHF-DL, kenar koruma ve algısal kalitede FOM: 0.8252 değeri, LPIPS: 0.1148 ile matematiksel temeldeki sağlamlığını göstererek yeni nesil yöntemlerle entegrasyonunda daha da gelişebileceğini göstermektedir. Aynı zamanda IGHF modeline göre IGHF-DL metrik olarak en iyi gelişimi yüksek gürültü seviyelerinde +7.44 dB gelişim göstermiştir. Önerilen hibrit yaklaşım, kaynak kısıtlı ortamlar için hesaplama maliyetini düşüren parametre verimliliği ile pratik ve verimli bir çözüm sunmaktadır.

Anahtar kelimeler

IGHF Gürültü Giderme, Derin Öğrenme, Hibrit CNN, Algısal Kalite ve Kenar Koruma, Parametre Verimliliği

1. INTRODUCTION

Image processing technologies are of great importance in the fields of medicine, industry, astronomy, and geology. Image denoising is a critical preprocessing step for tasks such as object recognition and feature extraction. Due to environmental conditions and the hardware structure of the imaging device, images can be exposed to many different types of noise. In real-world imaging, additive white Gaussian noise (AWGN) originating from sensor electronics and environmental conditions remains the most significant degradation factor [1], [2], [3]. Despite years of research, image denoising remains a challenging task, as it is difficult to achieve an optimal balance between denoising and detail preservation. Following classical filters, developments in image denoising have continued with CNN-based methods, GAN-based methods, transformation-based methods, and diffusion-based methods.

BM3D [4], Non-Local Means [5], and Inverse Gaussian Harmonic Filtering (IGHF) [6], like our previous paper, provide mathematical precision, theoretical interpretability, and proven edge preservation capabilities. However, these approaches often struggle with complex noise models and cannot achieve the performance levels required by modern applications. In contrast, deep learning approaches such as DnCNN [7] and FFDNet [8] achieve impressive performance. However, these methods are not efficient in environments with limited hardware due to high computational costs and high parameter counts (DnCNN: 556K). In particular, embedded systems such as mobile devices and IoT, as well as real-time applications, require structures with fewer parameters.

Classical image denoising has contributed significantly to the development of modern approaches. Non-local denoising [5] introduced the concept of utilizing self-similarity in images, calculating the weighted averages of similar patches within an image. This approach demonstrated that adaptive weighting based on similarity measurements can effectively preserve details while denoising, laying the foundation for attention-like mechanisms.

BM3D achieved outstanding performance by developing this self-similarity concept through collaborative filtering in transformation domains, grouping similar patches and applying collaborative Wiener filtering. The success of this method stems from its ability to utilize both local and non-local iterations. This principle is also compatible with modern attention mechanisms in transformers.
In our previous work on Inverse Gauss Harmonic Filtering (IGHF), we introduced a new mathematical framework based on the harmonic meaning of local and global statistics. This method provides superior edge preservation due to its mathematical formulation (Equation 1).

This harmonic mean approach demonstrates that it is a powerful preprocessing stage that will form the basis for deep learning networks.

The transition from classical denoising to deep learning-based denoising began with CNN-based approaches such as DnCNN and FFDNet. DnCNN achieved impressive results using a 17-layer CNN with backpropagation [7]. FFDNet demonstrated a different approach as a model that operates sensitively to noise levels [8]. These approaches proved that learned features could perform better than manually constructed features. However, the high number of parameters in these methods continued to pose a disadvantage. To overcome this issue, attention mechanisms such as the Squeeze-and-Excitation (SE) block have been integrated into hybrid CNN architectures to enhance feature discrimination while minimizing the number of parameters. Fırat and Üzen have demonstrated that integrating SE blocks into hybrid CNN architectures improves classification accuracy while minimizing the number of parameters [9].

Recently, studies on model compression and efficiency have emerged. Although methods such as MemNet [10] and CBDNet [11] propose architectures that yield more efficient and competitive results, they still present disadvantages due to the increased number of parameters. Challenge reports using AWGN-noisy images at the $\sigma = 50$ noise level, without computational complexity or model size constraints, also indicate continued progress in the field. In the relevant challenge report, the SRC-B model achieved a PSNR value of 31.20 with its dynamic fusion block innovation, demonstrating successful performance. Again in the same challenge report, the SNUCV model achieved a PSNR value of 29.95 by creating a hybrid model from the MambaIRv2 [12], Xformer [13], and Restormer [14] models [15].

Real-world noisy image studies have become increasingly important. Roman Flepp et al. presented the MIDD dataset, consisting of over 400,000 images obtained with different parameters to overcome the lack of mobile datasets [16]. Similarly, diffusion models have demonstrated their strengths in new-generation studies in this field. Yang et al. proposed a linear intermediate value diffusion model obtained from intermediate noisy images, original images, and corresponding real-world noisy images [17]. Zou et al. proposed their self-supervised DCD-Net model by iteratively training a noise estimation model and a denoiser. The model achieved superior results compared to similar methods on both synthetic and real noise [18]. Zhu and Chen proposed a new diffusion model called DDN, which includes a feature extraction module to extract image features and a diffusion noise estimation module to predict low-level noise. They succeeded in outperforming state-of-the-art methods in both blind noise and real-world noise [19]. Li et al. proposed an adaptive DMID method that reduces noise in the cleaned image by placing the noisy image into a pre-trained unconditional diffusion model. This method successfully overcomes the disadvantages of diffusion models [20]. Li et al. developed a Mamba-based method (MAIR) that delivers strong performance in multiple restoration tasks, consisting of a nested S-shaped scanning strategy that captures globality in addition to locality and a sequence shuffling attention block, thereby reducing computational cost [21]. Wang et al. used a technique

called Zero-Shot, employing double sampling without a clean image during training. They added more noise to the original noisy image, convolved it with an empty filter, and trained the system using the resulting blurred double-sampled image as input. The biggest advantage of the method is that it does not require prior knowledge of information such as the additional noise model, parameters, and estimated noise level. The method has demonstrated superior performance with these innovative approaches [22].

Our approach aims to deliver both efficient and competitive performance with support for CNNs with a few parameters. To overcome this limitation, we propose a new approach called IGHF-DL, which preserves texture by supporting classical mathematical IGHF filtering with a CNN architecture that requires fewer parameters. Our fundamental view is that the power of local and global features in IGHF mathematics, obtained through harmonic average weighting, can be enhanced by supporting it with a minimal parameter CNN. Building upon our previous work on IGHF, we have developed it by supporting its proven mathematical foundation with a deep learning architecture. Our main contributions are:

1. We present a hybrid model that combines our IGHF core, which is based on the harmonic weighting of local and global features, with a CNN enhancement module that uses six residual blocks, two SE channel attention mechanisms, an edge-sensitive enhancement block, and fixed scaling factors.
2. When creating a parameter-efficient model with 301,058 parameters, we report the contribution of IGHF preprocessing and DL refinement through ablation experiments.

Our proposed hybrid method outperforms the classical BM3D with only 301,058 parameters ($\sigma = 25$: $+0.15\,dB$). It achieves competitive performance for resource-limited environments by using 46% fewer parameters than DnCNN and reaching 98.3% of DnCNN's performance. Furthermore, a comprehensive evaluation was performed on six different datasets using extended metrics (PSNR, SSIM, FOM, LPIPS).

2. MATERIAL AND METHOD

Our approach is based on the Inverse Gauss Harmonic Filtering (IGHF) mathematical framework, which provides superior edge preservation through the harmonic weighting of local and global statistics. IGHF is combined with classical filtering and deep learning to implement a hybrid approach, as shown in Figure 1.

Step 1: Classic IGHF preprocessing (parameter: 0). IGHF preprocessing provides three key advantages: (1) Through harmonic mean formulation, adaptive weight estimation naturally balances local and global statistics and creates an attention mechanism based on statistical properties; (2) By adjusting filter strength based on local variance, edge-preserving initialization provides CNN with a cleaner starting point; (3) noise-adaptive behavior through statistical parameters that become increasingly valuable at higher noise levels.

Step 2: Six residual blocks for feature extraction, two attention mechanisms (SE_Block) for adaptive channel weighting, CNN enhancement with fixed scaling factors (base_scale=0.10, edge_scale=0.08), and CNN enhancement with sensitive enhancement support in edge regions. Our model has 301,058 parameters, which is 46% fewer than its closest competitor, DnCNN.

The CNN module complements IGHF by: (1) recovering textures that IGHF may over-smooth; (2) Adaptive residual learning that provides stable enhancement with fixed scaling factors (3) Edge-aware enhancement via the Sobel-based enhancement branch.

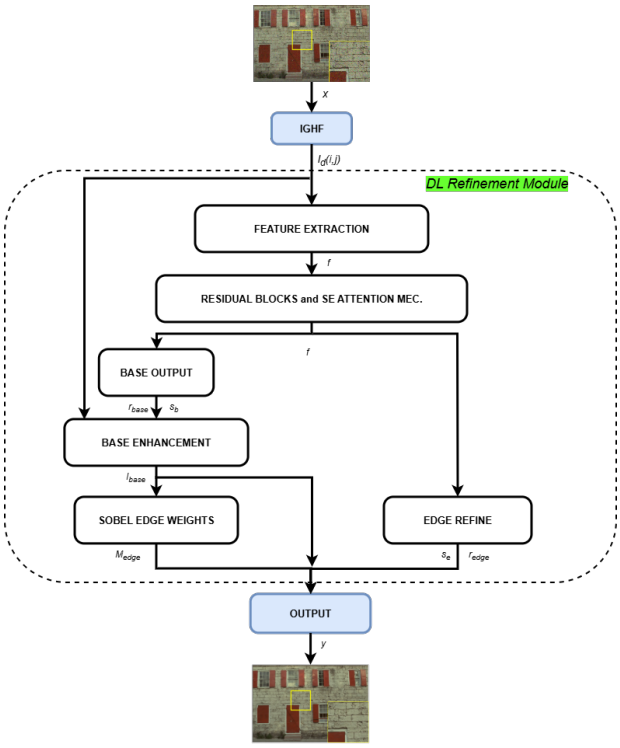

Figure 1. Proposed IGHF-DL Architecture

2.1. Classic IGHF Pre-processing

The basic IGHF formula calculates adaptable weights using the following (Equation 1):

$$w_{harmonic}(i,j) = \frac{2}{\frac{w_g}{\sigma_g+\epsilon} + \frac{w_l(i,j)}{\sigma_l(i,j)+\epsilon}} \quad (1)$$

Where $w_g = min(1, \sigma_{global} \cdot \alpha)$ global weight, $w_l(i,j) = min(1, \sigma_{local}(i,j) \cdot \alpha)$ local weight and α represents the weight factor. The harmonic meaning naturally balances local and global information and provides a mathematical basis for adaptive spatial weighting.

The IGHF core formula (Equation 2) converts these weights into filter coefficients:

$$I_d(i,j) = \sqrt{2w^2_{harmonic}(i,j).max(0, \; -ln(x_{i,j} \cdot w_{harmonic}(i,j) \cdot \sqrt{2\pi})} + \mu_{local} \quad (2)$$

Where $x_{i,j}$ noisy pixel, $I_d(i,j)$ denoised pixel, the max function is a clamping measure taken to prevent NaN values from occurring and μ_{local} is the mean of the local mask. According to this mathematical framework, adaptive weights are determined based on statistical features. This establishes a solid theoretical foundation by determining the importance of each pixel through attention-like mechanisms. IGHF achieves 27.17 dB performance at $\sigma = 25$ on BSD68. This provides robust starting point for this deep learning architecture. The preprocessing parameters are fixed across all noise levels during training: contrast stretching with $\gamma = 1.3$, [lower_limit, upper_limit]=[0.01, 0.95], window size $w = 3$, and harmonic weighting factor $\alpha = 30$. The value $\alpha = 30$ ensures that harmonic weights w_g and w_l reach the 1.0 saturation bound rapidly, providing balanced filtering between noise suppression and edge preservation.

2.2. DL Refinement

The fundamental idea guiding our approach and ensuring conceptual consistency is that the IGHF basis can be enhanced with CNN. A CNN architecture with 301,058 parameters is used. As detailed in Algorithm 1, learning is performed using six residual blocks, two SE attention mechanisms, and a Sobel-enhanced edge enhancement block with fixed scaling factors.

The CNN-based DL refinement module takes the IGHF pre-processed output x (Algorithm 1 input) and extracts spatial features through two sequential 3×3 convolutions with ReLU activations, producing intermediate map f_0 and feature map f, where b = 48 denotes the base feature channel count and C the number of input channels. The feature map f is then passed through K = 6 residual blocks and two SE channel-attention blocks [23], where residual blocks strengthen deep feature representations and SE blocks recalibrate channel weights. The feature map f is then split into two parallel sub-stages. In the base refinement sub-stage, a base residual output r_t is produced from f and added to x with fixed scale factor s_b = 0.10, yielding the base-refined output $I_t = x + 0.10 \cdot r_t$. Simultaneously, a normalised edge-weight mask M_{edge} is computed by applying the Sobel [24] operator to I_t. In the edge refinement sub-stage, an edge residual output $r_{edge} \in [-1, 1]$ with Tanh activation is produced from f. The final denoised output y is obtained by fusing both sub-stages with fixed scale factors s_b = 0.10 and s_e = 0.08, and clamped to [0, 1] (Equation 3). All convolutional weights were initialised using the Kaiming normal distribution [25]; biases were set to zero.

$$y = I_t + s_e \cdot M_{edge} \cdot r_{edge},\ y \leftarrow clamp(y, 0, 1) \quad (3)$$

where s_e = 0.08 is the edge refinement scale factor.

2.3. Training

The BSD500 and DIV2K datasets were used during the training phase. Although the IGHF process may extend the training time, preprocessing with IGHF was also performed. Training was performed on this pre-computed file. Training was conducted using $\sigma = \in \{15, 25, 35, 50\}$ noise levels in the form of 105,000 128x128 patches (103,000 training, 2,000 validation).
As the loss function (Equation 3);

$$L = L\_char + 0.3 * L\_SSIM + 0.5 * L\_edge + 0.4 * L\_grad \quad (4)$$

Where; L_char Charbonnier loss (smooth L1), L_SSIM loss of structural similarity and L_edge Sobel edge-preservation loss and L_grad gradient magnitude loss. The FOM metric has been optimized using high weights such as 0.5 and 0.4 for edge preservation.

All convolutional layer weights were initialized using Kaiming normal distribution; biases were initialized to zero. In training, the specified loss values were optimized using AdamW, with a learning rate of lr = 2e-4, a batch size of 64, 150 epochs, and CosineAnnealing as the schedule.

3. RESULTS

The IGHF baseline phase was performed on Truba systems prior to training, generating 105,000 patches. During the training phase, PyTorch 2.0 and NVIDIA A100 GPU Google Colab were used for hardware and software. Training duration was approximately 6.5 hours. Comparisons were made with BM3D, DnCNN, and FFDNet models, which are similar to our working methodology. We evaluate our model on six standard benchmark datasets. The datasets used are BSD68 (grayscale), CBSD68 (color), Kodak24 (color), McMaster (color), Set12 (grayscale), and Urban100 (grayscale). For the color datasets, AWGN noise was added to the RGB image, then converted to YCbCr, and all evaluations were performed on the Y channel.

Table 1. Noise Comparison (BSD68, Gray Scale, PSNR)

Model	Year	σ=15	σ=25	σ=50	Number of Param.	Method
BM3D	2007	31.07	28.57	25.62	-	Classical Block Matching
DnCNN	2017	31.73	29.23	26.23	556K	Deep Learning
FFDNet	2018	31.63	29.19	26.29	486K	Deep Learning
IGHF	2025	27.70	27.17	18.40	0	Classical Spatial
IGHF-DL(Ours)	2026	30.44	28.72	25.83	301K	Hybrid

Table 1 shows the PSNR values obtained at different noise levels ($\sigma = \in \{15, 25, 50\}$) on the BSD68 dataset. The results show that our model can compete with classical and deep learning-based models by achieving similar values. It outperformed the BM3D model by 0.15 dB at a noise level of σ=25. The most important factor is that it offers a hybrid structure that balances parameter

efficiency and perceptual quality with 46% fewer parameters than DnCNN.

Table 2. Ablation Study (BSD68)

σ	Model	PSNR (dB)	SSIM	FOM	LPIPS
15	CNN-Only	**31.41**	**0.8897**	**0.7764**	**0.1244**
	IGHF	27.70	0.7672	0.3967	0.2494
	IGHF - DL	30.44	0.8679	0.7525	0.1401
25	CNN-Only	**28.93**	**0.8231**	0.6605	0.2027
	IGHF	27.17	0.7466	0.5161	0.2376
	IGHF - DL	28.72	0.8217	**0.6809**	**0.1818**
50	CNN-Only	**25.91**	0.7051	0.4657	0.3315
	IGHF	18.39	0.3403	0.6241	0.6727
	IGHF - DL	25.83	**0.7141**	**0.5699**	**0.2816**

Note: CNN-Only: Same architecture (301K parameters) trained directly on noisy -- > clean mapping without IGHF preprocessing.

Table 2 shows the contribution of deep learning. The ablation study was performed on the entire BSD68 dataset. The contributions of DL are clearly visible at all noise levels. The greatest improvement occurred at high noise levels with a decrease in the LPIPS value. The reason for the decrease in the FOM metric at high noise levels is that the Canny edge detection used in the calculation was not accurate due to its sensitivity to high noise. Our correct contribution, which shows improvement, is demonstrated by the visual results and LPIPS. CNN-Only represents the same CNN architecture trained without IGHF preprocessing. While CNN-Only achieves higher PSNR at low noise levels, IGHF-DL demonstrates superior perceptual quality (FOM, LPIPS) at $\sigma \geq 25$ and confirms the contribution of IGHF preprocessing. The CNN-Only model achieves a higher PSNR (+0.97 dB) and slightly better LPIPS values at $\sigma = 25$ noise level. This suggests that, at low noise levels, the IGHF preprocessing step may over-smooth the input and thus provide a less sharp starting point compared to a directly noisy input. This low-noise limitation will be addressed in future work through adaptive noise estimation.

Table 3. IGHF-DL Model Multi-Dataset Comparison (σ=25)

Dataset	PSNR	SSIM	FOM	LPIPS
BSD68	28.72	0.8217	0.6809	0.1818
CBSD68	30.01	0.8571	0.7470	0.1405
Kodak24	31.25	0.8602	0.7319	0.1570
McMaster	31.96	0.8819	0.7829	0.1283
Set12	29.94	0.8534	0.7192	0.1594
Urban100	29.15	0.8707	0.8252	0.1148

Table 3 shows the performance of our model on different datasets. The Urban 100 dataset yielded the lowest LPIPS (0.1148) and highest FOM (0.8252) values. This value is successful in itself for our perceptual quality goal in images from texture-dense datasets.

Table 4. Computational Complexity (256x256 Gray Image)

Method	Number Of Parameters	Layers	Channels	CPU Time (ms)	MACs (G)	Model Size (MB)	GPU Time (ms)
DnCNN	556K	17	64	220.2	36.57	2.13	3.27
FFDNet	486K	15	64	38.55	7.99	1.86	1.78
IGHF-DL	301K	8	48	126.94	19.61	1.15	2.50

Table 4 compares the computational complexity of our DnCNN, FFDNet, and IGHF-DL models on 256x256 grayscale images. DnCNN has a higher MAC value due to its deeper architecture, i.e., greater number of layers. FFDNet achieves the lowest computational complexity due to downsampling. In contrast, our IGHF-DL model establishes a balance between efficiency and model capacity. Our model achieves a competitive MAC value and inference speed using fewer parameters than the compared models. These results demonstrate that IGHF-DL offers an efficient and practical design without excessive computational overhead. The reported times for IGHF-DL represent only the CNN module inference. To accelerate training, the basic IGHF stage has been saved as a pre-data file using Python Pickle. The full pipeline includes IGHF preprocessing (approximately 3075 ms on CPU for a 256×256 image) and CNN refinement (2.50 ms on GPU). Total inference time: ~3078 ms. Note: IGHF is implemented in pure Python/NumPy; preprocessing can be pre-computed offline for batch processing scenarios.

Table 5. Loss Function Sensivity Analysis

Config.	λ_char	λ_ssim	λ_edge	λ_grad	PSNR (dB)	FOM
Ours (Balanced)	1.0	0.3	0.5	0.4	27.25	0.711
High SSIM	1.0	0.5	0.2	0.2	27.24	0.735
High Edge	1.0	0.2	0.7	0.5	27.26	0.703
PSNR-Focused	1.0	0.1	0.3	0.2	27.27	0.700
Equal Weights	1.0	0.4	0.4	0.4	27.25	0.709

Different weight configurations were evaluated in the validation set. Our balanced configuration (λ_char=1.0, λ _ssim=0.3, λ _edge=0.5, λ _grad=0.4) provides an optimal balance between PSNR and FOM for edge-preserving image denoising. Charbonnier loss (λ =1.0) functions as the primary regularization term and provides robustness against outliers compared to MSE. SSIM loss (λ =0.3) preserves structural similarity without over-smoothing. Edge loss (λ =0.5) and gradient magnitude loss (λ =0.4) directly optimize edge preservation by targeting the FOM metric. Table 5 presents a sensitivity analysis that validates our weight selection. The PSNR-focused configuration achieves marginally higher PSNR (+0.02 dB) but compromises edge preservation (FOM: 0.700 vs. 0.711). The high SSIM configuration yields a higher FOM (0.735) despite lower edge weights, as SSIM

loss implicitly constrains local structural patterns that indirectly benefit edge preservation.

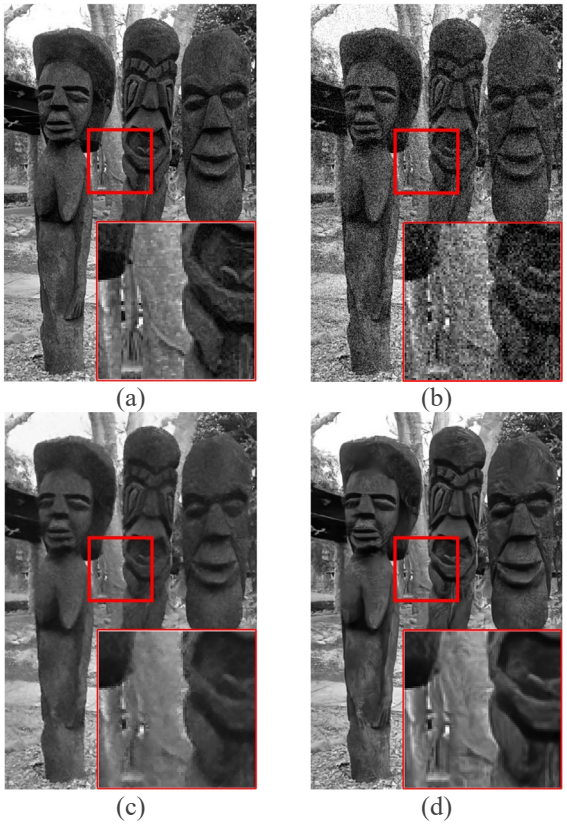

Figure 2. An Example of $\sigma = 25$ Noise Level from the BSD68 Gray Data Set: (a) Zoomed Original Image, (b) Noisy Image PSNR: 20.52 dB, (c) Image with IGHF Applied PSNR: 24.69 dB, (d) IGHF - DL PSNR: 25.63 dB

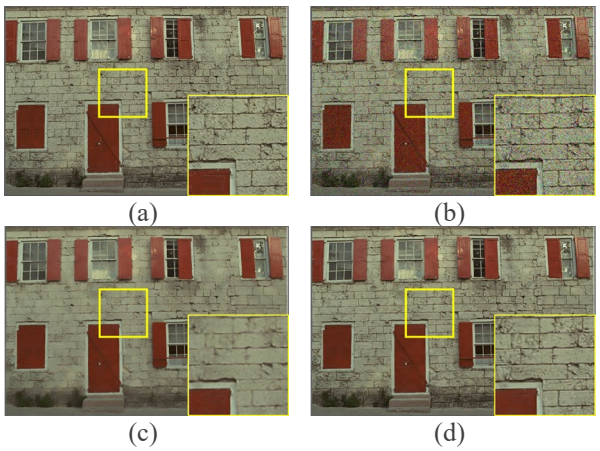

Figure 3. An Example of $\sigma = 25$ Noise Level from the Kodak24 Color Data Set: (a) Zoomed Original Image, (b) Noisy Image PSNR: 23.73 dB, (c) Image with IGHF Applied PSNR: 25.33 dB, (d) IGHF - DL PSNR: 28.23 dB

Figures 2 and 3 show how successfully both metric and visual quality have been improved in the experimental results. Preserving wall lines, especially in a texture-heavy dataset such as Kodak24, is a significant achievement.

4. DISCUSSION AND CONCLUSION

The results obtained demonstrate that our hybrid approach is competitive with similar models. While the classical IGHF provides mathematical robustness and theoretical interpretability, the CNN structure overcomes the disadvantages of IGHF. The model shows its greatest improvement, particularly at high noise levels, with a gain of +7.44 dB. Furthermore, the LPIPS and FOM metrics indicate that perceptual quality is preserved. While IGHF preprocessing limits real-time use (~3075 ms CPU), it can be pre-computed offline for batch scenarios, leaving only the 2.50 ms CNN refinement at inference.

This study presents a novel method for denoising that combines the classical IGHF mathematical framework with a deep learning architecture in a hybrid manner. Our method shows competitive results at different noise levels with fewer parameters compared to other methods. An interpretable, modular, and extensible structure is presented.

In future work, optimizing the IGHF processing speed, working on real-world noisy datasets, and trying new improvements with current methods will be explored.

Acknowledgement

The numerical calculations reported in this paper were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

REFERENCES

[1] B. Charles, “Image Noise Models,” in *Handbook of Image and Video Processing*, Elsevier, 2005, pp. 397–409. doi: 10.1016/B978-012119792-6/50087-5.

[2] U. Erkan, D. N. H. Thanh, L. M. Hieu, and S. Enginoglu, “An Iterative Mean Filter for Image Denoising,” *IEEE Access*, vol. 7, pp. 167847–167859, 2019, doi: 10.1109/ACCESS.2019.2953924.

[3] D. N. H. Thanh and S. D. Dvoenko, “A method of total variation to remove the mixed Poisson-Gaussian noise,” *Pattern Recognit. Image Anal.*, vol. 26, no. 2, pp. 285–293, Apr. 2016, doi: 10.1134/S1054661816020231.

[4] 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.

[5] A. Buades, B. Coll, and J.-M. Morel, “A Non-Local Algorithm for Image Denoising,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, San Diego, CA, USA: IEEE, 2005, pp. 60–65. doi: 10.1109/CVPR.2005.38.

[6] F. Marasli and S. Ozturk, “Inverse Gaussian Harmonic Filter (IGHF): Spatial Filter with Contrast Stretching Priority,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 39, no. 12, p. 2534001, Sep. 2025, doi: 10.1142/S0218001425340018.

[7] 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.

[8] 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, Sep. 2018, doi: 10.1109/TIP.2018.2839891.

[9] H. Fırat and H. Üzen, "Detection of Pneumonia Using A Hybrid Approach Consisting of MobileNetV2 and Squeeze-and-Excitation Network," *Türk Doğa Ve Fen Derg.*, vol. 13, no. 1, pp. 54–61, Mar. 2024, doi: 10.46810/tdfd.1363218.

[10] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A Persistent Memory Network for Image Restoration," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017, pp. 4549–4557. doi: 10.1109/ICCV.2017.486.

[11] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward Convolutional Blind Denoising of Real Photographs," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 1712–1722. doi: 10.1109/CVPR.2019.00181.

[12] H. Guo *et al.*, "MambaIRv2: Attentive State Space Restoration," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2025, pp. 28124–28133. doi: 10.1109/CVPR52734.2025.02619.

[13] J. Zhang, Y. Zhang, J. Gu, J. Dong, L. Kong, and X. Yang, "Xformer: Hybrid X-Shaped Transformer for Image Denoising," presented at the The Twelfth International Conference on Learning Representations, ICLR, Feb. 2024. Accessed: Jan. 09, 2026. [Online]. Available: https://openreview.net/forum?id=vXrIQLzIKY

[14] 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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 5718–5729. doi: 10.1109/CVPR52688.2022.00564.

[15] L. Sun *et al.*, "The Tenth NTIRE 2025 Image Denoising Challenge Report," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Nashville, TN, USA: IEEE, Jun. 2025, pp. 1333–1360. doi: 10.1109/CVPRW67362.2025.00125.

[16] R. Flepp, A. Ignatov, R. Timofte, and L. Van Gool, "Real-World Mobile Image Denoising Dataset with Efficient Baselines," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2024, pp. 22368–22377. doi: 10.1109/CVPR52733.2024.02111.

[17] C. Yang, C. Wang, L. Liang, and Z. Su, "Real-world image denoising via efficient diffusion model with controllable noise generation," *J. Electron. Imaging*, vol. 33, no. 04, Jul. 2024, doi: 10.1117/1.JEI.33.4.043003.

[18] Y. Zou, C. Yan, and Y. Fu, "Iterative Denoiser and Noise Estimator for Self-Supervised Image Denoising," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France: IEEE, Oct. 2023, pp. 13219–13228. doi: 10.1109/ICCV51070.2023.01220.

[19] Y. Zhu and Y. Chen, "A new diffusion method for blind image denoising," *Pattern Anal. Appl.*, vol. 27, no. 4, p. 124, Dec. 2024, doi: 10.1007/s10044-024-01346-0.

[20] T. Li, H. Feng, L. Wang, L. Zhu, Z. Xiong, and H. Huang, "Stimulating Diffusion Model for Image Denoising via Adaptive Embedding and Ensembling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 8240–8257, Dec. 2024, doi: 10.1109/TPAMI.2024.3432812.

[21] B. Li, H. Zhao, W. Wang, P. Hu, Y. Gou, and X. Peng, "MaIR: A Locality-and Continuity-Preserving Mamba for Image Restoration," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2025, pp. 7491–7501. doi: 10.1109/CVPR52734.2025.00702.

[22] S. Wang, C. Zhao, Q. Wang, M. Liu, C. Mou, and F. Xu, "Zero-shot image denoising with hollow pair sampling and noise-aware attention," *Pattern Recognit.*, vol. 167, p. 111779, Nov. 2025, doi: 10.1016/j.patcog.2025.111779.

[23] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 7132–7141. doi: 10.1109/CVPR.2018.00745.

[24] I. Sobel and G. Feldman, "A 3x3 Isotropic gradient operator for image processing," *Pattern Classif. Scene Anal.*, pp. 271–272, 1973.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile: IEEE, Dec. 2015, pp. 1026–1034. doi: 10.1109/ICCV.2015.123.

Appendices

Appendix A. IGHF-DL Algorithm

Algorithm 1: IGHF-DL – Deep Learning Refinement Module

Input:

$I_d(i,j) \in \mathbb{R}^{B\times C\times H\times W}$

IGHF output image (typically C = 1)

Hyperparameters:

b = 48 # base feature channels

K = 6 # number of residual blocks

α_base = 0.10 # base residual scale (fixed, s_b)

α_edge = 0.08 # edge refinement scale (fixed, s_e)

Modules:

FeatureExtract($I_d(i,j)$):

$f_0 \leftarrow$ ReLU(Conv3×3($I_d(i,j)$; C → b))

f ← ReLU(Conv3×3(f_0; b → b))
return f
SEBlock(f):
s ← GlobalAvgPool(f) # [B, b, 1, 1]
s ← Sigmoid(FC(ReLU(FC(s; b → b/8)); b/8 → b)) # [B, b]
return f ⊙ s # channel-wise multiplication
ResBlock(f):
u ← ReLU(Conv3×3(f; b → b))
v ← Conv3×3(u; b → b)
return f + v # residual connection
BaseOutput(f):
h ← ReLU(Conv3×3(f; b → b/2))
r ← Conv3×3(h; b/2 → C)
return r
EdgeWeight($I_d(i,j)$):
gx ← Conv($I_d(i,j)$, Sobel_x)
gy ← Conv($I_d(i,j)$, Sobel_y)
grad ← sqrt($gx^2 + gy^2 + \varepsilon$)
return grad / max(grad)
normalized edge map ∈ [0, 1]
EdgeRefine(f):
h_1 ← ReLU(Conv3×3(f; b → 32))
h_2 ← ReLU(Conv3×3(h_1; 32 → 16))
r ← Tanh(Conv3×3(h_2; 16 → C))
return r # r ∈ [−1, 1]

Forward Pass:

1: f ← FeatureExtract($I_d(i,j)$)
2: for k = 1 to K do
3: f ← ResBlock(f)
4: if k = 2 then f ← SEBlock(f) # first attention
5: if k = 4 then f ← SEBlock(f) # second attention
6: end for
7: r_base ← BaseOutput(f)
8: y_base ← $I_d(i,j)$ + α_base · r_base
base refinement
9: w_edge ← EdgeWeight(y_base)
10: r_edge ← EdgeRefine(f)
11: y ← y_base + α_edge · w_edge · r_edge
edge-aware refinement
12: y ← clamp(y, 0, 1)
13: return y

Output:

$y \in \mathbb{R}^{B\times C\times H\times W}$ # Denoised image