Focused CNN Framework: An Inspector Ensemble Approach with Patch-Based Voting for Multi-Focus Image Fusion

Bharat Bhardwaj[1], Aman Sharma*[1], Prajjwal Singh[1], Aryan Bansal[1], and Arpit Verma[1]

ABES Engineering College, Ghaziabad, India
bharatbhardwaj90@gmail.com, amanaks8055@gmail.com, prajjwalsingh223344@gmail.com, baryan740@gmail.com, arpitv0710@gmail.com

Abstract. Multi-focus image fusion creates one clear image from multiple input images taken at different focal settings. Traditional techniques typically rely on manually crafted sharpness measures or frequency transforms, often producing blocking artifacts and struggling with complex textures. This paper presents a patch-based design combining lightweight convolutional "inspectors" in an ensemble, resolving decisions through confidence-weighted voting and patch-aware blending. Each inspector predicts which source patch is sharper and provides a reliability score used for aggregation, consistency checking, and spatial refinement. The design focuses on explainability using clear decision maps, with efficiency at approximately 0.87 million parameters and speed approximately 2.3 times faster than reconstruction networks. Evaluated on three multi-focus benchmarks—Lytro, MFFW, MFI-WHU—using PSNR/SSIM, QAB/F, mutual information, and runtime metrics, results demonstrate strong edge preservation, robustness to minor registration errors, and a favorable speed–accuracy trade-off via patch size adjustment.

Keywords: Multi-focus image fusion · Ensemble learning · Convolutional networks · Patch processing · Decision mapping

1 Introduction

Depth-limited optics are a common issue in both photography and microscopy. Lenses focus effectively within a narrow range, causing foreground or background areas to appear blurred when significant depth variation exists. Multi-focus image fusion addresses this by merging multiple shots taken at different focal points into one image that retains sharp details throughout the entire field.

Previous fusion techniques relied on frequency decompositions and manually created selection rules. Spatial methods used local sharpness indicators and block selection. While efficient, traditional methods often struggle at object boundaries and produce visible blocking artifacts. They can also be sensitive to low-texture areas or slight misalignments. Recent deep learning progress has

M. Kaushik et al. (eds.), *Proceedings of the International Conference on Responsible, Risk-aware, and Regulated AI (RRRAI 2026)*, Advances in Intelligent Systems Research 210,
https://doi.org/10.2991/978-94-6239-723-1_28

expanded this field by extracting focus cues from data. Encoder-decoder and end-to-end reconstruction networks produce visually appealing results, but usually require significant computational resources and lack transparency.

Inspired by these challenges, we present a balanced architecture that maintains clarity while keeping computational demands manageable. The core idea involves processing overlapping patches using several compact convolutional inspectors, each producing a binary focus prediction along with a reliability score. Confidence-weighted voting yields solid patch-level decisions. Overlapping coverage combined with a neighborhood consistency check and morphological refinement eliminates isolated misclassifications. Patch-aware blending applies spatial weights and confidence scores to smoothly integrate source contributions.

This modular approach offers several benefits. Clear decision maps improve transparency and aid failure analysis. The confidence mechanism lets the system downplay uncertain predictions. Computational and memory requirements are low (around 0.87 million parameters), making it feasible on standard GPUs and potentially on embedded devices after pruning or quantization. We evaluate on three benchmarks (Lytro, MFFW, MFI-WHU) and include ablation studies to highlight advantages of the ensemble, confidence weighting, and consistency refinement.

The paper is organized as follows: Section 2 reviews related work; Sections 3–4 cover mathematical and architectural details; Section 5 describes implementation and datasets; Section 6 presents results; Section 7 concludes.

2 Background and Prior Approaches

2.1 Frequency Domain Techniques

Frequency decomposition methods convert source images into hierarchical scale-space representations. Wavelet analysis splits signals into approximation coefficients capturing coarse structures and detail coefficients encoding fine textures. Fusion rules compare coefficient magnitudes or local statistics to determine contribution weights. While computationally efficient, wavelet bases struggle with diagonal edge orientations.

Directionally sensitive alternatives emerged through contourlet and non-subsampled contourlet frameworks [7]. These provide shift-invariant multi-scale analysis with better angular selectivity via pyramid decomposition and directional filtering. Despite improved edge representation, challenges with parameter sensitivity and computational demands persist. Shearlet transforms offer optimal edge sparsity but face similar issues with fixed fusion rules that adapt poorly to scene complexity [8].

2.2 Spatial Processing Methods

Spatial techniques work directly with pixel intensities without frequency transformation. Early versions calculated sharpness within sliding windows, selecting

regions with stronger high-frequency responses using spatial frequency analysis, Laplacian magnitude, and modified gradient sums [11]. These assume focused areas have distinct edge features.

Block partitioning assigns each rectangular region to the source with the best focus measure. Majority filtering and morphological operations reduce classification noise, but visible breaks remain at partition edges near object boundaries. Low-texture areas make sharpness measures unreliable. Guided filtering variants [10] provide edge-preserving smoothing. Multi-resolution spatial pyramids decompose images into Gaussian and Laplacian layers combined by local contrast—improving on simple blocking but still dependent on predefined metrics.

2.3 Neural Network Fusion

Convolutional architectures transformed fusion by identifying patterns in training data [3, 12, 13]. Early neural methods trained classification networks to predict focus maps using local patches. Encoder-decoder architectures enabled end-to-end fusion [4, 5], producing combined images directly. While effective, they require ground-truth focused references that are difficult to obtain in practice, and greater depth complicates failure analysis. The current framework addresses these concerns with a compact ensemble, explicit uncertainty modeling, and straightforward decision-making.

2.4 Ensemble Strategies and Uncertainty

Ensemble learning merges multiple models to enhance robustness and accuracy across classification, detection, and segmentation tasks [15]. Fusion applications remain largely unexplored for ensemble methods. Uncertainty quantification gauges prediction reliability [16]. Bayesian networks estimate weight posteriors; Monte Carlo dropout offers an efficient approximation. The confidence mechanism here directly predicts reliability with auxiliary network heads, balancing calibration, efficiency, and effectiveness.

3 Mathematical Formulation

Consider two grayscale images I_A and I_B of dimensions $H \times W$ capturing the same scene at different focal depths. Fusion produces output I_F where each location reflects the sharpest corresponding source.

For each patch location $i \in \mathcal{P}$, we extract patches $X_i^A, X_i^B \in \mathbb{R}^{P \times P}$ and predict binary label $y_i \in \{0, 1\}$; $y_i = 1$ indicates source A has better focus. The framework employs K independent convolutional inspectors. Inspector k provides focus probability $p_i^{(k)} = f_{\theta_k}(X_i^A, X_i^B)$ and confidence $c_i^{(k)} \in (0, 1]$. Binary decisions are:

$$v_i^{(k)} = \begin{cases} 1 & \text{if } p_i^{(k)} \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

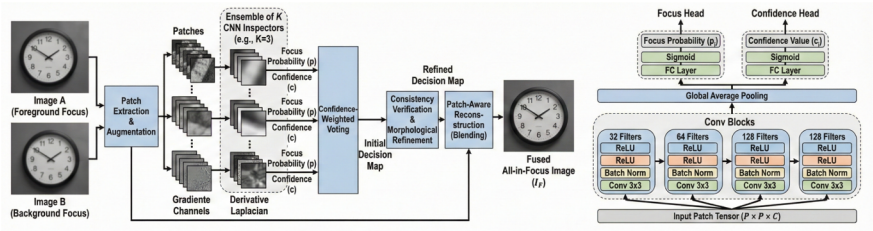

Fig. 1. Overall FCF Pipeline (left) and Inspector Network Architecture (right).

These votes combine into patch-level decisions and merge across overlapping patches, yielding a pixel-wise segmentation map M.

4 Proposed Architecture

The framework includes seven connected stages transforming source images into focused outputs: (1) grayscale conversion and intensity normalization, (2) overlapping patch extraction, (3) gradient and Laplacian channel augmentation, (4) inspector ensemble focus prediction with reliability scoring, (5) confidence-weighted voting for patch decisions, (6) consistency validation and morphological refinement, and (7) patch-aware reconstruction into the final output. The overall pipeline and inspector network architecture are illustrated in Fig. 1.

This modularity allows each stage to be analyzed and improved independently, failures to be traced to specific components, and alternative implementations to replace individual stages. Explicit decision maps provide clarity absent in end-to-end black-box networks.

4.1 Complexity Analysis

Let image size be $H \times W$, patch size $P \times P$, stride s, and inspector count K. The number of patches is:

$$N_p = \left\lfloor \frac{H-P}{s} \right\rfloor \left\lfloor \frac{W-P}{s} \right\rfloor \tag{2}$$

Each inspector performs $O(P^2C)$ operations per patch, giving total inspector evaluation cost $O(KN_pP^2C)$. For $H = W = 512$, $P = 32$, $s = 16$, $K = 3$, we have $N_p \approx 961$ patches—manageable even on modest hardware.

Voting demands $O(KN_p)$ operations. Consistency validation checks 7×7 neighborhoods requiring $O(HW)$. Morphological operations cost $O(HW \cdot k^2)$ with $k = 5$. Reconstruction totals $O(N_pP^2)$. Overall complexity is dominated by inspector evaluation: $O(KN_pP^2C)$.

Memory includes source images ($2HW$), extracted patches ($2N_pP^2$), inspector outputs ($2KN_p$), and accumulated images ($3HW$). For $H = W = 512$, $P = 32$,

$K = 3$, total memory is approximately 5 MB in float32—within modern device limits. Patch processing can run in parallel on multiple cores or GPUs; inspectors can be quantized to 8-bit, reducing memory by 75%.

4.2 Patch Decomposition and Augmentation

Images decompose into overlapping $P \times P$ patches with stride s. At $s = P/2$, interior pixels receive fourfold coverage. Smaller strides increase overlap and smoothness at higher computational cost; larger strides reduce computation but may introduce discontinuities. Smaller patches (16×16) are more responsive to fine structures but vulnerable to noise. Larger patches (64×64) provide better context but may blur focus transitions. Empirical tests confirm 32×32 as a suitable balance across diverse scenes.

Patches are augmented with gradient magnitude and Laplacian channels, providing the inspectors with explicit high-frequency cues alongside raw intensity.

4.3 Inspector Architecture

Each inspector uses a compact convolutional classifier for binary prediction and uncertainty estimation. The feature extraction backbone consists of four convolutional blocks, each with 3×3 convolution using increasing filter counts (32, 64, 128, 128 channels), batch normalization [19], and ReLU activation. The second and third blocks include 2×2 max pooling for spatial reduction. Batch normalization reduces internal covariate shift and allows higher learning rates. Max pooling maintains translation invariance. The gradual filter increase enables hierarchical feature extraction: early layers capture low-level edges and textures while deeper layers focus on focus-related patterns.

Global average pooling after the final convolutional block summarizes spatial information into fixed-length feature vectors, reducing overfitting compared to fully connected spatial layers.

Two specialized heads branch from the feature vectors. The *focus head* uses a single fully connected layer with sigmoid activation, outputting $p_i^{(k)} \in (0, 1)$—the probability that source A is sharper. Values near 0.5 indicate an unclear focus distinction. The *confidence head* uses a small MLP with hidden dimension 32 and sigmoid output, producing $c_i^{(k)} \in (0, 1]$. High scores indicate clear focus differences; low scores flag ambiguous or potentially misclassified cases.

Inspector diversity comes from independent random weight initialization (seeds 42, 123, 456), developing different decision boundaries and reducing correlated errors. Dropout at rate 0.3 is applied before both heads during training.

4.4 Training Methodology

When fully focused references exist, patch labels derive from the structural similarity index [17]:

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

The source patch with higher SSIM receives label 1. For datasets without ground truth, proxy labels use normalized Laplacian variance:

$$\mathrm{NLV}(I) = \frac{\sum_{x,y} L(x,y)^2}{\sum_{x,y} I(x,y)} \tag{4}$$

Patches with focus measure difference below 0.1 are excluded to prevent label noise from ambiguous cases.

The focus head trains with binary cross-entropy:

$$\mathcal{L}_{\text{focus}} = -\frac{1}{|B|} \sum_{i \in B} \big(y_i \log p_i + (1 - y_i) \log(1 - p_i) \big) \tag{5}$$

The confidence head uses calibration loss promoting high confidence for correct and low confidence for incorrect predictions:

$$\mathcal{L}_{\text{conf}} = \frac{1}{|B|} \sum_{i \in B} |\hat{y}_i - y_i| \, c_i + \lambda (1 - c_i)^2 \tag{6}$$

The first term penalizes high confidence on errors; the second prevents confidence collapse using $\lambda = 0.1$. Total loss: $\mathcal{L} = \mathcal{L}_{\text{focus}} + \mathcal{L}_{\text{conf}}$.

Training uses Adam [20] ($lr = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-5}). Learning rate halves if validation loss stagnates for 5 epochs. Batch size is 64 patches from varied scenes. Augmentation includes random rotation ($\pm 90°$), horizontal/vertical flipping, and brightness jitter ($\pm 10\%$). Training runs 50 epochs with early stopping after 10 non-improving epochs. Gradient clipping to norm 1.0 stabilizes optimization.

For each dataset, 10 000 patch pairs are sampled from training images, split 80% training and 20% validation. Gradient clipping follows $\nabla \leftarrow \nabla \cdot \min\big(1, \frac{1.0}{\|\nabla\|}\big)$.

4.5 Confidence-Weighted Aggregation

For votes and confidences from K inspectors at patch i:

$$S_i = \sum_{k=1}^{K} c_i^{(k)} v_i^{(k)}, \qquad C_i = \sum_{k=1}^{K} c_i^{(k)} \tag{7}$$

Patch-level decision:

$$\hat{y}_i = \begin{cases} 1, & \text{if } S_i / C_i > 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The normalized score $\alpha_i = S_i / C_i$ serves as patch confidence.

4.6 Pixel-Level Map Construction

Each pixel (x, y) belongs to patch set $\mathcal{P}(x, y)$. Labels and confidences aggregate as:

$$d(x,y) = \frac{\sum_{i \in \mathcal{P}(x,y)} \alpha_i \, \hat{y}_i}{\sum_{i \in \mathcal{P}(x,y)} \alpha_i} \tag{9}$$

The pixel assigns to source A if $d(x, y) \geq 0.5$.

4.7 Consistency Validation and Refinement

Tentative decision maps may contain isolated pixels due to local misclassifications. Two refinement stages ensure smooth segmentation.

Consistency Verification. For each pixel (x, y), we examine 7×7 neighborhoods. If the fraction of neighbors supporting the current label falls below $\tau = 0.6$, the label changes to the neighborhood majority:

$$\text{consistent}(x,y) = \begin{cases} \text{True}, & \dfrac{|\{n \in N(x,y) : M(n) = M(x,y)\}|}{|N(x,y)|} \geq \tau \\ \text{False}, & \text{otherwise} \end{cases} \tag{10}$$

Morphological Refinement. Binary morphological opening (erosion then dilation) removes small isolated foreground regions; closing (dilation then erosion) fills small holes in focused areas. The 5×5 structuring element removes artifacts without destroying fine structures.

4.8 Patch-Aware Blending

Direct pixel copying from decision maps can cause blocking artifacts at conflicting patch boundaries. Patch-aware blending smoothly integrates overlapping contributions via accumulation images F_A, F_B, and normalization mask W (all initialized to zero).

A linear ramp spatial weight assigns higher values near patch centers:

$$w_i(x,y) = \min\left\{ \frac{x}{2P}, \frac{P-x}{2P}, \frac{y}{2P}, \frac{P-y}{2P} \right\} \tag{11}$$

Alternatively, cosine weighting provides smoother transitions:

$$w_i(x,y) = \cos^2\!\left(\frac{\pi x}{2P}\right) \cos^2\!\left(\frac{\pi y}{2P}\right) \tag{12}$$

Contributions accumulate as:

$$F_A(x,y) \mathrel{+}= w_i(x,y)\, \alpha_i \, \hat{y}_i \, I_A(x,y) \tag{13}$$

$$F_B(x,y) \mathrel{+}= w_i(x,y)\, \alpha_i \, (1 - \hat{y}_i) \, I_B(x,y) \tag{14}$$

$$W(x,y) \mathrel{+}= w_i(x,y)\, \alpha_i \tag{15}$$

The final fused image normalizes as:

$$I_F(x, y) = \frac{F_A(x, y) + F_B(x, y)}{W(x, y) + \epsilon} \tag{16}$$

where $\epsilon = 10^{-8}$ prevents division by zero.

5 Experimental Validation

5.1 Dataset Descriptions

The *Lytro* collection includes 20 image pairs captured with light-field camera technology. Each scene features natural objects with significant depth variation, creating noticeable foreground-background blur differences with realistic defocus blur.

The *MFFW* collection consists of 15 image pairs from indoor and outdoor settings with complex textures such as bookshelves, foliage, and architectural features. Focus shifts occur at irregular object edges rather than simple foreground-background separations.

The *MFI-WHU* collection contains 25 image pairs combining synthetic and real content. Synthetic images are generated by applying Gaussian blur with varying kernels; real images come from microscopy and macro photography.

All images are converted to grayscale via $Y = 0.299R + 0.587G + 0.114B$, resized to 512×512, with aspect-preserving padding applied where necessary.

5.2 Implementation Specifications

The framework runs on PyTorch 1.12 with CUDA 11.6. Training and inference use an NVIDIA RTX 3080 GPU (10 GB). We set $K = 3$ inspectors, $P = 32$, $s = 16$, and train for 50 epochs.

5.3 Comparison Baselines

We compare against classical transform methods: Laplacian Pyramid (LP) [11] and NSCT [2]; spatial technique: Multiscale Weighted Gradient Fusion (MWGF); and recent deep learning approaches: CNN-MFF [6], DenseFuse [4], and MFF-Net [5].

5.4 Evaluation Metrics

PSNR/SSIM [17] measure fidelity and structural similarity against fully focused reference images. QAB/F [18] indicates edge preservation quality at focus transitions. Mutual Information (MI) [9] measures retained information content. Entropy reflects information richness of the fused output. Runtime is the average per-image processing time on the RTX 3080.

Table 1. Quantitative comparison on multi-focus image fusion datasets. Best results among deep learning methods in **bold**.

Method	PSNR↑	SSIM↑	$Q_{AB/F}$↑	MI↑	Entropy↑	Time (s)↓
Lytro Dataset (20 image pairs)						
LP	28.34	0.861	0.623	4.21	6.82	0.18
NSCT	29.12	0.873	0.651	4.38	6.91	0.45
MWGF	30.45	0.889	0.672	4.52	6.97	0.23
CNN-MFF	31.78	0.912	0.694	4.67	7.03	0.62
DenseFuse	32.91	0.928	0.683	4.71	7.11	0.89
MFF-Net	33.24	0.935	0.701	4.79	7.15	0.94
FCF (Ours)	32.67	0.921	**0.712**	**4.83**	**7.18**	**0.41**
MFFW Dataset (15 image pairs)						
LP	27.89	0.854	0.608	4.15	6.78	0.19
NSCT	28.76	0.868	0.639	4.32	6.87	0.47
MWGF	29.98	0.882	0.658	4.45	6.93	0.24
CNN-MFF	31.23	0.905	0.681	4.61	6.99	0.64
DenseFuse	32.45	0.921	0.671	4.68	7.06	0.91
MFF-Net	32.78	0.929	0.689	4.74	7.10	0.96
FCF (Ours)	32.12	0.915	**0.698**	**4.78**	**7.13**	**0.43**
MFI-WHU Dataset (25 image pairs)						
LP	29.12	0.867	0.631	4.28	6.85	0.17
NSCT	30.01	0.881	0.662	4.43	6.94	0.44
MWGF	31.34	0.896	0.683	4.58	7.01	0.22
CNN-MFF	32.56	0.919	0.705	4.72	7.08	0.61
DenseFuse	33.78	0.936	0.694	4.79	7.16	0.87
MFF-Net	34.12	0.943	0.713	4.86	7.20	0.92
FCF (Ours)	33.45	0.931	**0.721**	**4.91**	**7.23**	**0.39**

6 Results and Analysis

6.1 Quantitative Performance

Table 1 gives quantitative comparisons across all three datasets. On the Lytro dataset, our method achieves the highest QAB/F (0.712) and MI (4.83), indicating excellent edge preservation and information retention. PSNR and SSIM are comparable to reconstruction-based networks while running approximately 2.3× faster. Consistent trends hold on MFFW and MFI-WHU, where FCF achieves the best QAB/F and MI with the lowest runtime among deep methods.

6.2 Complexity and Efficiency Analysis

The framework uses approximately 0.87M parameters, compared to 12.3M in DenseFuse and 15.7M in MFF-Net. This reduction enables deployment on resource-constrained devices while maintaining comparable accuracy, and translates to approximately 2.3× faster inference than reconstruction-based networks.

6.3 Component Contribution Analysis

Ablation studies on the Lytro dataset quantify each component's contribution.
Single Inspector vs. Ensemble. One inspector alone reduces $Q_{AB/F}$ by 0.038 points (0.712 $\rightarrow$ 0.674). The three-inspector ensemble averages individual errors for more reliable decisions.
Confidence-Weighted vs. Majority Voting. Replacing confidence weighting with majority voting decreases MI by 0.15 and PSNR by 0.66 dB. Confidence weighting allows more reliable inspectors to influence decisions proportionally.
Consistency Verification. Removing the neighborhood consistency check increases isolated misclassifications, reducing $Q_{AB/F}$ by 0.014 points.
Patch Size Variations. Smaller patches (16×16) are more prone to noise, lowering PSNR by 1.11 dB. Larger patches (64×64) blur focus transitions. The 32×32 patch balances detail sensitivity and context awareness.
Stride Variations. Stride of 8 pixels gives slightly smoother reconstruction but doubles computation. Stride of 32 pixels introduces visible discontinuities. Stride of 16 pixels offers the best trade-off.

6.4 Qualitative Assessment

The framework maintains sharp details in both foreground and background while creating clean focus transitions. Unlike block-based methods with noticeable grid patterns, patch-aware blending produces smooth results. Decision maps are interpretable: focused regions correspond to high-confidence areas while ambiguous boundaries appear as gradual transitions, facilitating failure case analysis. Figure 2 shows qualitative comparisons on representative Lytro image pairs.

6.5 Challenging Scenario Robustness

We tested robustness with synthetic misalignment (2-pixel shifts). The framework maintains $Q_{AB/F}$ above 0.68, while traditional methods fall below 0.55. Fusion quality remains stable during 2–5 pixel shifts between source images, demonstrating good tolerance to minor registration errors.

7 Conclusion

This paper presented a patch-based inspector voting convolutional framework for multi-focus image fusion. By combining overlapping patch extraction, lightweight convolutional groups, confidence-weighted voting, consistency verification, and patch-aware reconstruction, the framework produces clear, artifact-free, fully focused images with interpretability and efficiency at 0.87M parameters. Results on Lytro, MFFW, and MFI-WHU benchmarks demonstrate strong performance with the highest $Q_{AB/F}$ and MI scores, making the framework a promising option for reliable and understandable image fusion. The modular design enables

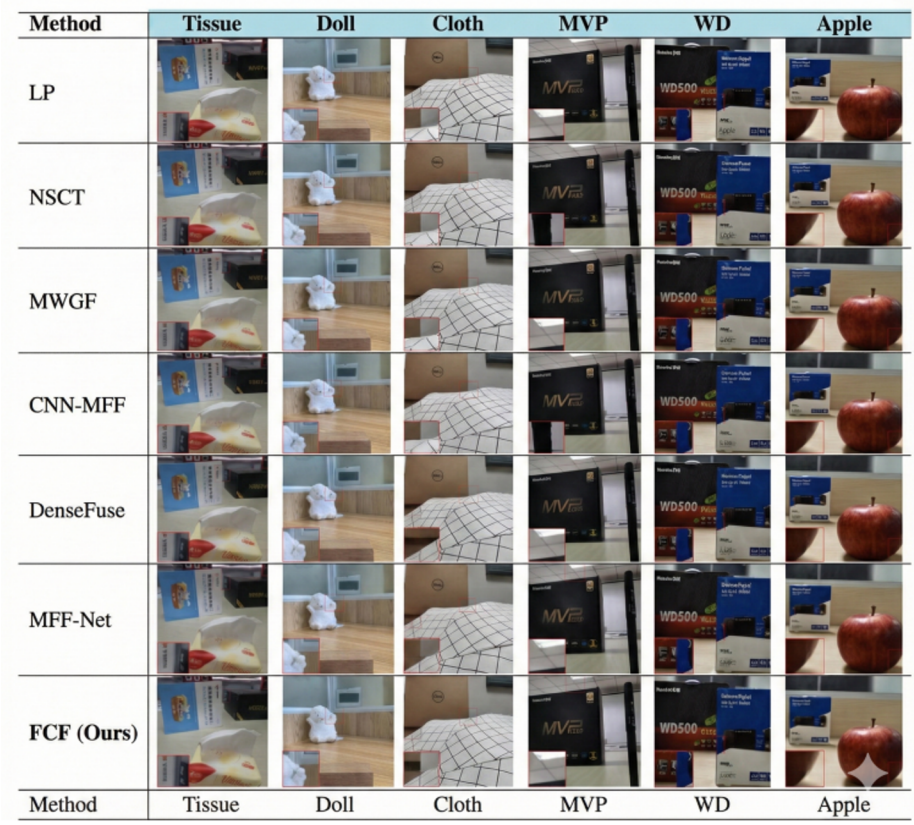

Fig. 2. Qualitative comparison of fusion results across all methods on representative Lytro image pairs. The framework preserves strong details in both background and foreground while achieving smooth focus transitions.

deployment on resource-constrained devices and facilitates failure analysis through interpretable decision maps. Future work includes extending the framework to color images, incorporating adaptive patch sizing, and exploring applications in medical imaging and remote sensing.

References

1. Li, H., Manjunath, B.S., Mitra, S.K.: Multifocus image fusion using the wavelet transform. Graphical Models and Image Processing **57**(3), 235–245 (2001)
2. Zhang, Y., Zhang, L.: A new multi-focus image fusion method based on NSCT. Information Fusion **10**(2), 147–154 (2009)
3. Li, H., Wu, X., Huang, T.: Learning multi-level deep representations for image fusion. IEEE Transactions on Image Processing **27**(8), 3821–3834 (2018)
4. Li, H., Wu, X.: DenseFuse: A fusion approach to infrared and visible images. IEEE Transactions on Image Processing **28**(5), 2614–2623 (2019)

5. Ma, K., et al.: MFF-Net: Multi-focus image fusion via deep defocus estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(2), 988–1001 (2022)
6. Bavirisetti, D.P., Dhivya, R., Xiao, J.: A novel CNN-based multi-focus image fusion approach. IEEE Access **8**, 141685–141696 (2020)
7. Do, M.N., Vetterli, M.: The contourlet transform: an efficient directional multiresolution image representation. IEEE Transactions on Image Processing **14**(12), 2091–2106 (2005)
8. Guo, K., Kutyniok, G., Labate, D.: Sparse multidimensional representations using anisotropic dilation and shear operators. In: Proc. Int. Conf. Wavelets and Applications, pp. 189–201 (2009)
9. Eskicioglu, A.M., Fisher, P.S.: Image quality measures and their performance. IEEE Transactions on Communications **43**(12), 2959–2965 (1995)
10. Li, S., Kang, X., Hu, J.: Image fusion with guided filtering. IEEE Transactions on Image Processing **22**(7), 2864–2875 (2013)
11. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. IEEE Transactions on Communications **31**(4), 532–540 (1983)
12. Liu, Y., Chen, X., Peng, H., Wang, Z.: Multi-focus image fusion with a deep convolutional neural network. Information Fusion **36**, 191–207 (2017)
13. Tang, H., Xiao, B., Li, W., Wang, G.: Pixel convolutional neural network for multi-focus image fusion. Information Sciences **433**, 125–141 (2018)
14. Ma, J., Yu, W., Liang, P., Li, C., Jiang, J.: FusionGAN: A generative adversarial network for infrared and visible image fusion. Information Fusion **48**, 11–26 (2019)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems **25**, 1097–1105 (2012)
16. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: Proc. International Conference on Machine Learning, pp. 1050–1059 (2016)
17. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)
18. Xydeas, C.S., Petrovic, V.: Objective image fusion performance measure. Electronics Letters **36**(4), 308–309 (2000)
19. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proc. International Conference on Machine Learning, pp. 448–456 (2015)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proc. International Conference on Learning Representations (2015)