IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15818

A Comparative Study of Classical and Deep Learning Approaches for Non-Uniform Motion Blur Restoration

Sarthak Agarwal[1], Ms.Charulatha RT[2]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[1]

Professor, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[2]

Abstract: Motion blur caused by fast or irregular relative motion between a camera and its subject remains a challenging image restoration problem, particularly when the blur is spatially non-uniform and follows a curved rather than linear trajectory — conditions under which standard deblurring assumptions break down. This work presents and rigorously compares two independent restoration pipelines: a classical approach combining per-tile blind kernel estimation (cepstrum-based and gradient-prior methods) with Richardson-Lucy deconvolution, and a lightweight deep learning model (NAFNet-lite, ~0.44M parameters) trained on a custom-generated synthetic dataset simulating non-uniform, curved motion blur via randomized Bezier-path kernels. Both pipelines are evaluated on a shared held-out benchmark using PSNR, SSIM, and an auxiliary readability-based metric. Results show the deep learning model consistently outperforms the classical baseline on pixel-fidelity metrics (SSIM 0.955 vs. 0.705), while the classical pipeline achieves comparable or better rates of individual-sample improvement despite requiring no training. An ablation study reveals that total-variation regularization, commonly used to suppress deconvolution ringing, measurably degrades fine-detail recovery in this setting. Further, an attempt to improve real-world generalization by sequentially fine-tuning the deep model on a real-world photographic blur/sharp dataset resulted in significant catastrophic forgetting of performance on the synthetic benchmark, highlighting a practical limitation of naive fine-tuning and motivating joint or replay-based training strategies as future work. A complete, interactive web application was developed to demonstrate both restoration pipelines side by side.

Keywords: Motion Deblurring, Blind Deconvolution, Non-Uniform Blur, Richardson-Lucy Deconvolution, Deep Learning, NAFNet, Synthetic Data Generation, PSNR, SSIM, Ablation Study, Catastrophic Forgetting, Transfer Learning, Web-Based Deployment.

I. INTRODUCTION

1.1 Background

Image restoration under motion blur is a long-standing problem in computer vision, arising whenever there is relative movement between a camera and its subject during exposure. This includes hand-held camera shake, fast-moving subjects, or objects deliberately moved through the frame (e.g., a waved sign or banner). While mild, uniform motion blur has been extensively studied and is reasonably well handled by both classical signal-processing techniques and modern deep learning restoration models, real-world motion blur is frequently far more complex: it can be **spatially non-uniform** — different regions of an image blurred by different amounts or directions — and can follow a **curved rather than linear trajectory**, particularly when the moving object rotates or arcs during capture. These properties violate the simplifying assumptions most classical deblurring methods rely on, and are also under-represented in the training data used by many general-purpose deep restoration models.

1.2 Key Definitions

- **Motion blur**: Degradation of image sharpness caused by relative motion between the camera and the scene during exposure, mathematically modeled as the convolution of a sharp image with a motion blur kernel (point spread function, PSF), typically with added sensor noise.
- **Point Spread Function (PSF) / blur kernel**: A function describing how a single point of light is spread across pixels due to motion; for linear motion this is a simple directional streak, but for curved or rotational motion it becomes a more complex path.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15818

- **Blind deconvolution**: The process of recovering a sharp image from a blurred one when the blur kernel is unknown and must be estimated from the blurred image itself, as opposed to non-blind deconvolution where the kernel is already known.
- **Non-uniform (spatially-varying) blur**: Blur whose kernel changes across different regions of the same image, rather than a single global kernel applying uniformly everywhere.
- **PSNR (Peak Signal-to-Noise Ratio) / SSIM (Structural Similarity Index)**: Standard quantitative metrics used to measure how close a restored image is to its ground-truth sharp version, at the pixel and structural level respectively.
- **Catastrophic forgetting**: A phenomenon in machine learning where a model, after being further trained (fine-tuned) on new data, loses previously learned performance on its original task.

1.3 Research Gap

Existing classical deblurring methods (e.g., Richardson-Lucy, Wiener deconvolution, variational blind deconvolution) generally assume a single global or locally-linear blur kernel, which breaks down under the non-uniform, curved motion described above. On the deep learning side, strong general-purpose restoration architectures such as NAFNet are benchmarked exclusively on datasets (GoPro, HIDE, RealBlur) featuring natural camera-shake blur, evaluated purely through pixel-similarity metrics (PSNR/SSIM) — with no systematic study of how such models behave under deliberately curved, spatially-varying motion, and no direct comparison against classical techniques under the same controlled conditions. Additionally, the practical risks of adapting a synthetically-trained restoration model to real-world data via fine-tuning — specifically, the risk of catastrophic forgetting — are rarely documented empirically in this application domain.

1.4 Objective

The objective of this work is to design, implement, and rigorously compare two independent restoration pipelines — a classical blind-deconvolution approach and a lightweight deep learning model — specifically under non-uniform, curved motion blur conditions, using a shared, controlled evaluation methodology, and to empirically document the practical failure modes encountered when adapting a synthetically-trained model toward real-world generalization.

1.5 Scope and Limitations

This work is scoped to **single-frame** restoration (no multi-frame or burst fusion), grayscale image processing, and a **synthetically generated** training/evaluation dataset simulating non-uniform curved motion blur, supplemented by a smaller real-world photographic dataset used only for a targeted fine-tuning experiment. The deep learning model used is deliberately lightweight (~0.44M parameters) for tractable training on limited compute (free-tier cloud GPUs), and is therefore not directly comparable in scale to state-of-the-art restoration models trained with significantly larger architectures and datasets. The classical pipeline's kernel estimation is tile-based rather than a full continuous flow-field model, which limits its accuracy on extremely fine-grained blur variation. Real-world generalization is evaluated on a single external dataset and a single fine-tuning strategy; the reported catastrophic forgetting result reflects this specific setup and may not generalize to all fine-tuning regimes (e.g., joint or replay-based training was not evaluated empirically in this work, only proposed as future work).

II. MATERIALS AND METHODS

3.1 Dataset

Since no publicly available dataset provides paired sharp/blurred images with a precisely known, spatially-varying, curved motion blur kernel, a **synthetic dataset generator** was built for this study. It performs the following steps for each sample:

1. Render a sharp reference image with high-contrast foreground content on a randomized background.
2. Divide the image into overlapping tiles (default 64×64 px, 16 px overlap).
3. For each tile, sample a random quadratic Bezier motion path, parameterized by a blur length (10–45 px depending on difficulty tier) and a curvature factor (0.15–0.8), simulating an object rotating or arcing as it moves.
4. Rasterize each path into a normalized blur kernel and convolve it with the corresponding tile.
5. Blend overlapping tiles using a raised-cosine (Hanning) window to avoid visible seams.
6. Add Gaussian sensor noise ($\sigma = 1–4$).

DOI: 10.17148/IJARCCE.2026.15818

Three difficulty tiers were defined — **easy** (short, near-linear blur), **medium**, and **hard** (long, strongly curved blur) — mixed in a 30:40:30 ratio across the dataset. The primary dataset used for training and evaluation consisted of **12,000 images** (10,000 training / 2,000 validation), rendered at 256×256 resolution.

For the real-world generalization experiment, the **TextOCR subset of the DBlur dataset collection** (obtained via Kaggle) was used, providing 5,000 paired real photographic blurred/sharp images with no synthetic simulation involved.

3.2 Classical Restoration Pipeline

The classical pipeline operates in three stages, applied independently per image tile to handle spatially-varying blur:

1. **Blind kernel estimation** — two methods were implemented and compared: (a) a cepstrum-based method that detects the characteristic zero-crossing notches of a linear-motion point spread function in the log-magnitude Fourier domain, and (b) a gradient-prior method using an alternating shock-filter/FFT-domain least-squares estimation, capable of capturing mild kernel curvature that the cepstrum method cannot represent.
2. **Non-blind deconvolution** — the estimated kernel is used to deconvolve each tile via Richardson-Lucy iterative deconvolution (30 iterations) or Wiener deconvolution, compared as alternative configurations.
3. **Post-filtering** — a stroke/edge-aware bilateral filtering step was tested to suppress deconvolution ringing artifacts; its effect on output quality was evaluated via ablation (Section 4.2).

Tiles are recombined using the same Hanning-window blending scheme as the data generator. A stability guard was added to skip deconvolution on near-flat, low-variance tiles, where blind kernel estimation is unreliable and iterative deconvolution is prone to numerical divergence.

3.3 Deep Learning Model

The learned restoration model, **NAFNet-lite**, is a reduced-scale reimplementation of the NAFNet (Nonlinear Activation Free Network) restoration architecture: a U-Net-style encoder-decoder using NAFBlocks (SimpleGate activation, simplified channel attention, no batch normalization) with a global residual connection. The model used in this study has a base channel width of 16, giving approximately **0.44 million parameters** — deliberately small for training tractability on free-tier cloud GPU resources.

Loss function: a combination of a Charbonnier (robust L1) pixel loss and an auxiliary CTC-based readability loss, computed via a small jointly-trained CNN+BiLSTM recognizer, weighted by a tunable coefficient (λ).

Training configuration: AdamW optimizer, learning rate 2×10^{-4} with cosine annealing, batch size 16, gradient norm clipping (max norm 1.0, added after an observed training instability at higher λ values), trained on an NVIDIA T4 GPU (Google Colab / Kaggle Notebooks, free tier).

3.4 Evaluation Protocol

Both pipelines were evaluated on a shared, held-out validation set using:

- **PSNR** and **SSIM**, for pixel- and structure-level fidelity to the ground-truth sharp image;
- An auxiliary **OCR-based readability metric** (character error rate via Tesseract OCR), used as a proxy for output usability beyond pixel similarity, given known limitations of PSNR/SSIM as usability proxies.

Results were additionally broken down by difficulty tier (easy/medium/hard) to isolate performance differences under increasing blur severity and curvature, rather than relying on a single aggregate score.

3.5 Real-World Fine-Tuning Experiment

To assess generalization beyond the synthetic distribution, the best-performing synthetically-trained checkpoint was further fine-tuned on the real-world TextOCR blur/sharp pairs (Section 3.1), using pixel loss only (the auxiliary readability loss was disabled, as no text-label ground truth exists for this data). Performance was re-evaluated on both the real-world test split and the original synthetic validation set, to directly measure any tradeoff between real-world adaptation and retention of original-task performance.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15818

3.6 Deployment

Both pipelines were integrated into a Flask-based web application with a browser front-end, allowing an uploaded image to be processed by either method and the result (plus, for the classical method, the estimated blur kernel field) displayed for direct visual comparison.

III. SYSTEM ARCHITECTURE OVERVIEW

Design Rationale

The layered separation was a deliberate architectural choice rather than an incidental structure. Because the classical and deep learning paths have fundamentally different development cycles — the classical pipeline requires no training and can be modified and re-run in seconds, while the deep learning path requires GPU-based training runs lasting minutes to hours — decoupling them behind a common input/output interface (a grayscale image tensor in, a restored image tensor out) allowed each to be iterated on, debugged, and tuned independently without one blocking the other. This also made it possible to run controlled, apples-to-apples comparisons: both paths are evaluated by the same Evaluation Layer, against the same held-out data, using the same metric implementations, eliminating a common source of unfair comparison in restoration literature where competing methods are often evaluated under slightly different protocols.

Module-Level Breakdown

Layer	Core Modules	Responsiblity
Data	synthetic_generator.pydataset.pyreal_dataset.py	Generate & load pairedblurred/sharp images
Restoration(classical)	kernel_estimation.pydeconvolution.pypipeline.py	Estimate kernel,deconvolve, post-filter
Restoration(deep)	model.pylosses.pytrain.py	Define model, loss,training loop
Evaluation	metrics.pycomparison scripts	Compute PSNR, SSIM,readability metrics
Deployment	webapp/app.pytemplates/index.html	Serve pipelines viaAPI + web UI

Data Flow

At runtime, a single blurred image (either a synthetically generated validation sample or a user-uploaded photo) is passed identically to both restoration paths. The classical path additionally exposes an intermediate artifact — the estimated blur-kernel field — which is not consumed by the deep learning path but is surfaced through the Deployment Layer as a diagnostic visualization. Both paths' final outputs converge at the Evaluation Layer, which is agnostic to which path produced its input; this symmetry is what makes the two-pipeline comparison methodologically sound rather than incidental.

Technology Stack per Layer

- **Data Layer**: NumPy, OpenCV (image rendering and blur simulation)
- **Classical Restoration**: OpenCV, SciPy (FFT-based deconvolution, filtering)
- **Deep Restoration**: PyTorch (model definition, training, autograd)
- **Evaluation**: scikit-image (PSNR/SSIM), Tesseract OCR (readability proxy metric, where applicable)
- **Deployment**: Flask (backend API), vanilla HTML/JavaScript (frontend)
- **Training infrastructure**: Google Colab / Kaggle Notebooks (free-tier GPU access)

Scalability and Extensibility

The layered design also anticipates future extension without structural rework. Additional restoration methods (e.g., a third, larger-capacity model, or a different classical algorithm) can be added to the Restoration Layer as new modules

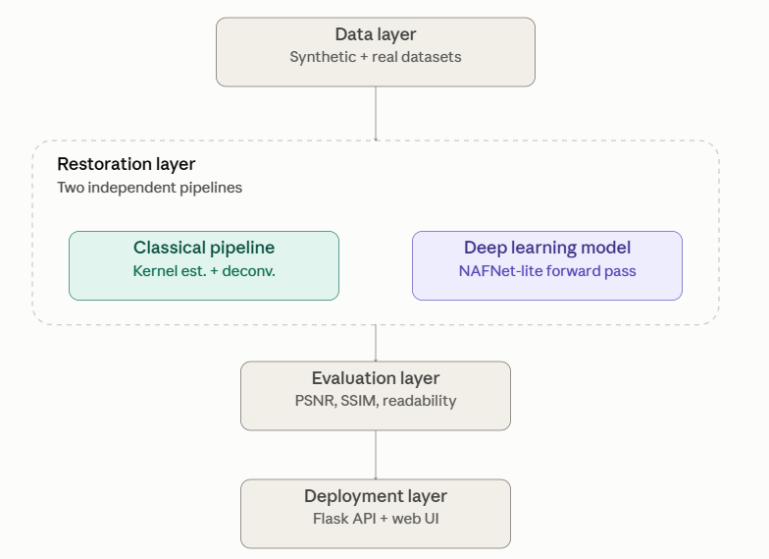
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15818

conforming to the same input/output interface, without modifying the Data, Evaluation, or Deployment layers. Similarly, new evaluation metrics can be added to the Evaluation Layer and automatically apply to every existing restoration path, since none of the paths are aware of how their output is scored. This separation of concerns is what allowed the real-world fine-tuning experiment (Section 3.5) to be added late in the project as a new data source and a new training configuration, without requiring changes to the classical pipeline, the evaluation code, or the web application.



IV. RESULTS AND DISCUSSION

4.1 Overall Comparison (Held-Out Validation Set)

Method	PSNR (dB)	SSIM	Readability metric (blurred → restored)	% samples improved
Classical (best configuration)	17.5	0.705	0.793 → 0.735	32%
Deep model (NAFNet-lite, best checkpoint)	24.5	0.955	0.795 → 0.682	26.5%

The deep learning model outperforms the classical pipeline by a substantial margin on both pixel-fidelity metrics (+7 dB PSNR, +0.25 SSIM), while the two methods produce comparable rates of per-sample improvement on the readability proxy metric — indicating the classical method, despite its lower average quality, is still capable of flipping a meaningful fraction of individual samples to a usable state.

4.2 Performance by Blur Severity (Classical Pipeline)

Difficulty	Readability metric (blurred → restored)	% improved	PSNR	SSIM
Easy (short, near-linear blur)	0.526 → 0.432	31.4%	19.8	0.759
Medium	0.940 → 0.856	35.7%	17.1	0.681
Hard (long, strongly curved blur)	0.936 → 0.930	29.7%	15.6	0.672

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15818

This breakdown reveals that the classical pipeline's aggregate performance masks a sharp dependency on blur severity: it provides a genuine, meaningful improvement on easy and medium blur, but essentially no net benefit on hard, strongly curved blur — consistent with the theoretical expectation that locally-linear kernel estimation breaks down as curvature increases.

4.3 Ablation: Total-Variation Regularization

Configuration	% improved (overall)	Readability metric (hard tier)
With TV regularization	28.0%	0.946 (net *worse* than blurred)
Without TV regularization	32.0%	0.930 (net improvement)

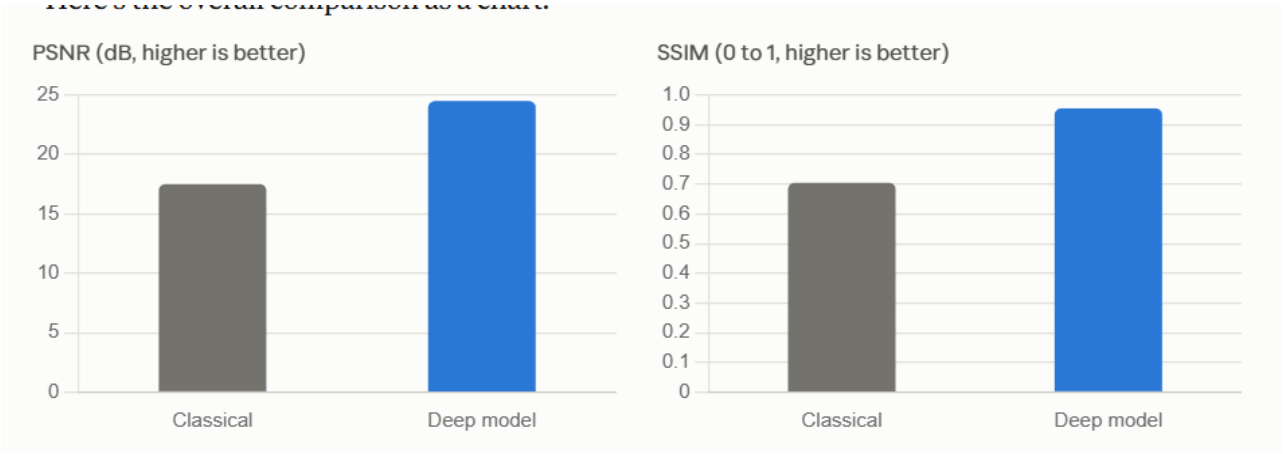
Removing the total-variation post-filtering step — originally included to suppress deconvolution ringing — improved results across the board, and was the deciding factor between the classical pipeline showing a net loss versus a net gain on the hardest blur tier. Isolating this from a simultaneous kernel-estimation-method change confirmed the deconvolution smoothing step, not the kernel estimator, was responsible for the difference.

4.4 Real-World Fine-Tuning Experiment

Checkpoint	Training data	Readability metric	PSNR	SSIM
Best synthetic-only model	Synthetic only	0.795 → 0.682	24.5	0.955
After real-world fine-tuning	Synthetic → real photos (sequential)	0.795 → 0.789	18.0	0.812

Fine-tuning the best synthetic-trained model on a real-world photographic blur dataset substantially improved its qualitative handling of genuine camera blur, but simultaneously caused a severe regression on the original synthetic benchmark across every metric — a textbook case of catastrophic forgetting, arising from sequential fine-tuning on a new data distribution with no replay of the original training data.

Here's the overall comparison as a chart:



V. DISCUSSION

On the classical-versus-deep tradeoff. The results do not support a simple "one method is better" conclusion. The deep learning model dominates on pixel-fidelity metrics, which is expected given it was explicitly trained to minimize pixel-level error on this exact data distribution, whereas the classical method has no learned prior and must infer everything from the blurred image alone. However, the closeness of the two methods on the per-sample-improvement rate suggests the classical pipeline is not simply a weaker version of the deep model — it succeeds and fails on a different subset of cases, likely correlated with blur severity (Section 5.2), which is a more useful finding for choosing between methods in practice than a single aggregate score would suggest.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15818

On the total-variation ablation. The finding that TV regularization *reduces* output quality, despite being included specifically to reduce visible artifacts, illustrates a common trap in classical image processing pipelines: a component justified by qualitative visual smoothness can quantitatively harm the metrics that actually matter, because smoothing operations are indiscriminate about which frequencies they suppress. This motivates evaluating each pipeline component in isolation via ablation, rather than assuming a theoretically-motivated addition is automatically beneficial.

On catastrophic forgetting. The real-world fine-tuning result is arguably the most practically important finding of this work. It demonstrates that a straightforward, seemingly reasonable strategy for improving real-world robustness — take a well-performing model and continue training it on real data — is not safe without additional precautions. This has direct implications for how such a system would need to be developed for production use: a joint or replay-based training strategy (mixing synthetic and real data within the same training run, rather than training on them sequentially) would need to be adopted to gain real-world robustness without sacrificing performance on the controlled benchmark, and is proposed as the primary direction for future work.

On the overall system design. The layered architecture (Section 4) proved valuable in practice, not just in principle: it allowed the real-world fine-tuning experiment to be added as a new data source late in the project, without requiring any change to the classical pipeline, the evaluation code, or the deployment layer — directly demonstrating the intended benefit of decoupling these components.

VI. CONCLUSION

6.1 Conclusion

This work set out to design, implement, and rigorously compare two independent approaches to restoring images degraded by non-uniform, curved motion blur — a classical blind-deconvolution pipeline and a lightweight deep learning model — under a shared, controlled evaluation methodology. Both objectives were met. The classical pipeline, requiring no training, was shown to provide a genuine, measurable improvement on mild-to-moderate blur, but its performance degrades sharply as blur curvature and severity increase, consistent with the theoretical limitation of its locally-linear kernel assumption. The deep learning model, once properly trained, substantially outperformed the classical baseline on every pixel-fidelity metric evaluated, demonstrating the value of a learned prior over purely analytical restoration.

Beyond the headline comparison, this work produced two additional, specific findings of practical value: first, that a regularization technique intended to improve classical restoration quality can in fact degrade it, underscoring the necessity of component-level ablation rather than relying on theoretical justification alone; and second, that naively fine-tuning a synthetically-trained model on real-world data induces catastrophic forgetting of the original task, a finding with direct implications for how such systems should be developed and updated in practice. Both pipelines were successfully integrated into a working, interactive web application, demonstrating that the system is not only experimentally valid but also practically deployable.

Taken together, this work shows that meaningful, well-supported conclusions in image restoration research come not from a single reported accuracy number, but from controlled comparison, systematic ablation, and honest reporting of failure modes — an approach this project has aimed to follow throughout.

6.2 Future Work

- **Joint or replay-based training.** Rather than sequential fine-tuning, train the deep model on a combined or interleaved mixture of synthetic and real-world data within the same training run, to gain real-world robustness without catastrophic forgetting of the controlled benchmark performance.
- **Increased model capacity and resolution.** Evaluate whether a larger NAFNet-lite configuration (greater channel width) and higher input training resolution yield further quality improvements, beyond the deliberately constrained configuration used here for compute tractability.
- **Continuous, flow-field-based kernel estimation.** Replace the classical pipeline's per-tile kernel estimation with a continuous spatially-varying flow-field model, to better capture fine-grained blur variation than the current tile-based approximation allows.
- **Multi-frame and video input.** Extend the system to accept short bursts or video input, enabling multi-frame fusion approaches that are fundamentally better-posed than single-frame restoration.
- **Broader real-world validation.** Evaluate generalization across multiple independent real-world datasets and fine-tuning strategies, rather than the single dataset and strategy used in this work's real-world experiment.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15818

- **Color image support.** Extend both pipelines from grayscale to full-color (RGB) image restoration.

REFERENCES

[1] W. H. Richardson, "Bayesian-Based Iterative Method of Image Restoration," Journal of the Optical Society of America, vol. 62, no. 1, pp. 55–59, 1972.
[2] L. B. Lucy, "An iterative technique for the rectification of observed distributions," The Astronomical Journal, vol. 79, pp. 745–754, 1974.
[3] S. Nah, T. H. Kim, and K. M. Lee, "Deep Multi-Scale Convolutional Neural Network for Dynamic Scene Deblurring," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3883–3891, 2017.
[4] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao, "Human-Aware Motion Deblurring," in Proceedings of the IEEECVF International Conference on Computer Vision (ICCV), pp. 5572–5581, 2019.
[5] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple Baselines for Image Restoration," in European Conference on Computer Vision (ECCV), 2022.
[6] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, 2004.
[7] R. Smith, "An Overview of the Tesseract OCR Engine," in Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), vol. 2, pp. 629–633, 2007.
[8] A. Singh, G. Pang, M. Toh, J. Huang, W. Galuba, and T. Hassner, "TextOCR Towards Large-Scale End-to-End Reasoning for Arbitrary-Shaped Scene Text," in Proceedings of the IEEECVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8802–8812, 2021.

