Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ29494

# *Seamless 4× Super-Resolution for 4K Images Overlap-Tiling and Weighted Blending with U-Net*

**Ruizhe Li**

*School of Engineering, the University of Manchester, Manchester, the United Kingdom*
*ruizhe.li-2@student.manchester.ac.uk*

***Abstract.*** This study investigates how tile blending choices affect the visual quality and speed of image super-resolution performed in tiles. Twenty paired images are evaluated across four display resolutions, while blending weights (linear, cosine, Gaussian) and overlap sizes are varied. Peak signal-to-noise ratio and structural similarity are computed on the luminance channel, and latency per megapixel is recorded to capture the cost of each setting. The findings are consistent across figures and tables: a Gaussian window gives the most reliable gains, raising peak signal-to-noise ratio by about two decibels over linear and cosine on 4K scenes, with structural similarity unchanged near 0.87. Moderate overlap performs best in practice. At 1080p and 1440p, the speed–quality frontier is dominated by the Gaussian family, while larger overlaps shift points to slower regions with only small quality increases. A simple recipe follows from the summary tables: use Gaussian weighting and choose overlap by resolution, from sixteen pixels at 720p up to thirty-two pixels at 4K. These settings suppress seams in textured areas, avoid extra blur at edges, and keep runtime within common real-time limits. The results provide a drop-in configuration for tiled inference without changes to model architecture.

***Keywords:*** Tiled super-resolution, overlap-tile blending, Gaussian weighting, U-Net, 4K image reconstruction

## 1. Introduction

Single-image Super-Resolution (SR) seeks to recover a high-resolution image from a single low-resolution input. Recent work shows that strong quality is possible with modern backbones. Transformer-based models such as SwinIR raise accuracy on common benchmarks [1], while lightweight CNN designs with simple blocks provide a good efficiency–quality balance [2]. For real-world and compressed inputs, systems like Real-ESRGAN improve perceptual quality and robustness [3], and Swin2SR adapts transformer ideas to artefacts from compression and in-the-wild data [4]. Yet running SR at 4K scale remains hard in practice because memory and runtime are tight. Many pipelines therefore split large images into overlapping tiles and blend the overlaps to hide seams. Such overlap-and-blend schemes are standard in biomedical segmentation with sliding-window prediction [5], in large-scale image stitching [6], and in modern digital pathology toolkits for whole-slide processing [3,7].

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.TJ29494

Most SR papers emphasise backbone design and loss functions, but they provide little guidance for the inference choices that make 4K processing feasible on limited hardware. In real systems two knobs matter: the tile overlaps and the weighting function used to blend the overlaps (linear, cosine, or Gaussian). Codebases expose these knobs, yet the defaults are rarely justified for 4× SR at 4K. Prior reports in segmentation and stitching argue that smooth, non-zero edge weights reduce boundary artifacts [6,7], and recent clinical AI toolkits rely on tiling to scale to gigapixel slides [7]. However, these works do not analyze the same question under SR workloads, where the upscaling factor, the output size, and the distortion metrics all differ.

Reporting practice also leaves a gap. Many studies give a single runtime at a single resolution. Such a number does not transfer to other sizes and hides the true cost of wider overlaps as images grow. A resolution-aware measure is needed to compare settings across 720p, 1080p, 1440p, and 4K. Community challenge reports on efficient SR make this tension explicit and call for solutions that keep quality while staying fast and light [3]. Even so, literature rarely isolates the tiling and blending step itself, even though this step often dominates memory use and controls seam artefacts in deployment.

Robustness is another open point. Linear and cosine weights drop to zero at tile borders, while Gaussian decays smoothly and remains non-zero. These edge behaviors should affect seam visibility and stability on textures and strong boundaries. Yet SR papers seldom compare these options in a controlled setting with the network and training held fixed. Given steady progress in model design [1,4], real-world restoration [2,5], and deployment-oriented benchmarks [3], a focused and reproducible study of overlap and weighting at 4K remains missing.

This study aims to address these gaps with a compact end-to-end evaluation of overlap-tiling and weighted blending for 4× SR on 4K images using a U-Net-style backbone. The network and training remain fixed; only two inference knobs vary: overlap (8, 16, 24, 32 pixels in the low-resolution space) and the blending weight (linear, cosine, Gaussian). Quality is measured with PSNR and SSIM, and speed is measured as milliseconds per megapixel, so results are comparable across resolutions. A Pareto analysis is used to reveal the speed–quality frontier and to guide choices when the two goals conflict. The study also connects SR deployment with established practice in large-image pipelines, where sliding-window inference and smooth blending are common for avoiding block artifacts [5,6]. The aim is not to introduce a new backbone, but to provide evidence-based defaults and a simple protocol that others can reproduce and apply to their own SR models, including recent efficient designs [1, 2, 8] and systems highlighted in efficient SR tracks [3].

## 2. Method

### 2.1. Dataset preparation

The dataset contains twenty high-quality 4K images at 3840×2160. Each image acts as the high-resolution (HR) ground truth. A paired low-resolution (LR) image is produced from every HR image by downsampling to 960×540 with bicubic interpolation (×4). This setup matches standard super-resolution (SR) practice in recent NTIRE tracks, which helps comparisons across studies that follow the same protocol [3,9,10]. Many real-world SR studies adopt learned degradation models instead of strict bicubic; those pipelines are different in goal but motivate a clear description of the downsampling rule used here [2,11].

All preprocessing is scripted. Each image is read as RGB, converted to floating point, and normalized to the range 0–1. The pipeline saves HR–LR pairs with matched stems so later stages load data without manual steps. The same image set is used in every experiment. Each image is

DOI: 10.54254/2755-2721/2026.TJ29494

processed under every overlap value and every weight map. For each run, the code writes one PNG output and one JSON file that records the overlap, the weight type, the runtime, and the evaluation scores. This fixed design keeps comparisons fair across conditions and aligns with recent leaderboard practice, where the scale factor, the downsampling operator, and the metrics are explicit [3,9,10].

Tile-wise inference runs on the LR grid to keep memory use stable at 4K. Each LR tile is 64×64. The overlap is a parameter and takes values 0, 4, 8, 16, or 32 pixels on both axes. The stride along each axis equals the tile size minus the overlap. This rule gives full coverage and repeatable positions. At the outer border, reflection padding supplies valid context, so every border tile still has a complete 64×64 input. Overlap-tile processing with mirrored borders is common in large-image workflows and is used in modern segmentation and montage systems to avoid visible seams during patch prediction and reassembly [6,7].

Two figure triplets show the LR–HR pairing and the ×4 reconstruction target shown in Figure 1. In each triplet, the left panel is the LR input (displayed enlarged for visibility), the middle panel is the 4K ground truth, and the right panel is the 4K output from the pipeline. This layout helps check on edges, textures, and gentle gradients such as sky or water.

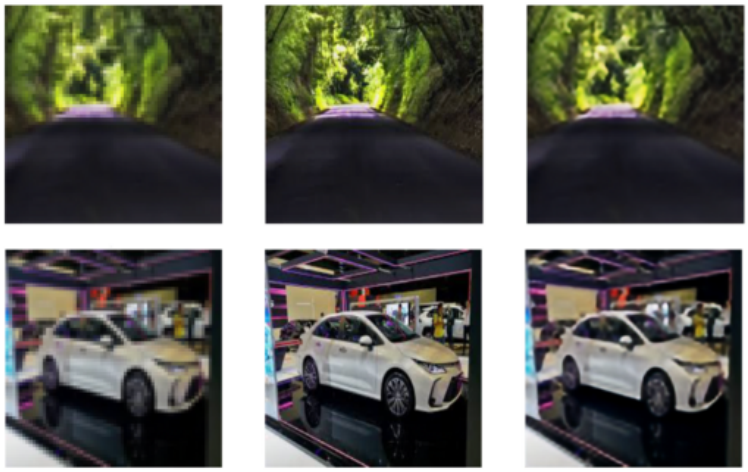


Figure 1. Comparison among low-resolution (LR), ground truth (GT), and super-resolved (SR) images (two scenes shown) (picture credit: original)

Figure 2 illustrates why overlap and smooth blending matter when tiling. The example uses an organic matrix composite sample (dataset name OMC14). It shows a 3×3 tile layout with a 12% overlap and compares the raw montage to a version without smooth blending. The visible block boundaries in the second panel motivate the weighted blending used in this work and agree with guidance from stitching systems on large scientific images [7].

DOI: 10.54254/2755-2721/2026.TJ29494

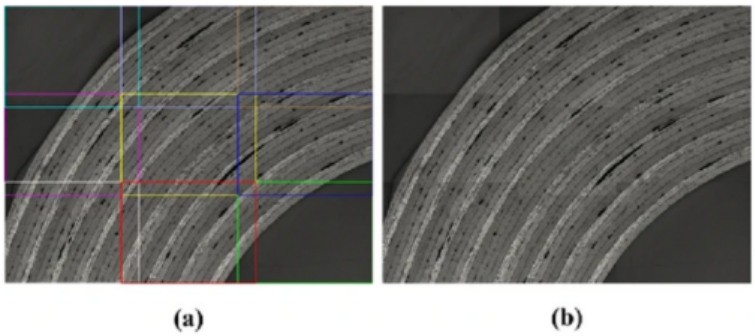



Figure 2. Organic matrix composite (OMC14), left (a) 3×3 tiles with 12% overlap, right (b) montage without smooth blending; seams are visible (picture credit: original)

## 2.2. U-Net-based SR model

The backbone is a compact U-Net encoder–decoder with skip connections. An LR tile of size 64×64 enters the encoder, which reduces spatial size and increases channels to gather context. The decoder upsamples step by step back to 256×256, which gives a four-times spatial scale-up. Skip connections pass encoder features to the decoder at matched scales. This fusion keeps edges and local texture while deeper layers track broader structure, a pattern used widely in image-to-image restoration [6]. Compared with heavier transformer backbones such as SwinIR, Restormer, and the later Swin2SR variant, a light U-Net offers favorable memory use and stable behavior for tile-wise 4K reconstruction [1,5,12]. The pipeline is model-agnostic, so diffusion-based SR backbones can replace the U-Net when realism is the priority [13,14].

The implementation follows the shared model file. Two versions exist (with and without skips). The skip-connection version is used because it preserves edges better under tiling. Decoder upsampling uses bilinear resize followed by convolution to reduce checkerboard artifacts. Convolutions use ReLU or LeakyReLU with batch normalization for stable gradients. The network outputs a 256×256 SR tile when given a normalized 64×64 RGB tile.

Training uses L1 loss with the AdamW optimizer. L1 tends to preserve edges better than L2 in restoration tasks, and AdamW gives stable convergence with decoupled weight decay. These choices match recent restoration baselines and public challenge reports that favor simple, robust objectives while reporting strong PSNR and SSIM on public sets [3,4,5,11].

Inference is tile-wise. Each LR tile produces one SR tile at 256×256. The 4K canvas is assembled by placing tiles at their target locations and blending overlaps with a spatial weight map so contributions fade toward tile borders. Three profiles are supported:

Linear (one-dimensional ramp across the overlap):

$$w(i) = \frac{i}{O-1}, \quad i = 0,1,\ldots,O-1 \tag{1}$$

where $O$ is the overlap width in pixels and $i$ is the index measured inward from the tile edge. A two-dimensional map is formed by combining horizontal and vertical ramps; weights taper to zero at the borders.

Cosine (half-cosine window across the overlap):

$$w(i) = \frac{1-\cos\left(\pi \times \frac{i}{O-1}\right)}{2}, \quad i = 0, 1, \ldots, O-1 \tag{2}$$

DOI: 10.54254/2755-2721/2026.TJ29494

The two-dimensional map uses the product of the horizontal and vertical one-dimensional windows to produce smooth edges.

Gaussian (radial decay from the tile center):

For a pixel at position $(x,\ y)$ , the distance from the tile centre is:

$$d\left(x,\ y\right) = \sqrt{\left(x - c_x\right)^2 + \left(y - c_y\right)^2} \tag{3}$$

Where $(cx,\ \ cy)$ is the centre of the tile, and weight is calculated using:

$$w\left(x,\ y\right) = e^{-\frac{d(x,\ y)^2}{2\sigma^2}} \tag{4}$$

With $\sigma = \frac{tile\ size}{4}$ , normalize the map so its maximum equals 1.0.

The final pixel value at position $p$ is the normalized blend:

$$Final\left(p\right) = \frac{\sum I_t(p) \times w_t(p)}{\sum w_t(p)} \tag{5}$$

where the sum is over all tiles t that cover pixel position $p$ ; $I_t$ is the SR tile value and $Wt$ is the scalar weight from the map. Normalization keeps the mix as a convex combination, which prevents brightness drift and removes hard joints. Montage and whole-slide pipelines use the same principle of smooth, non-zero edge weights to hide seams on large images [7].

PSNR and SSIM are computed against the 4K ground truth which follows the well-known perception–distortion trade-off, where fidelity metrics and perceptual quality may pull in different directions [14]. These two measures remain the main fidelity scores in current SR leaderboards and surveys; some recent studies also report perceptual metrics such as MUSIQ and DISTS/A-DISTS to reflect human preference, especially when diffusion-based backbones are considered [3,9,10,13,15,16].

## 3. Results and discussion

### 3.1. The performance of the model

The study evaluates twenty paired images. PSNR and SSIM are computed on the luminance channel, with reference lines that mark typical quality bands. Across the three bar charts in Figure 3, Figure 4 and Figure 5, the pattern is stable: most pairs fall in the “good” PSNR band and near 0.87 SSIM. The third bar chart (Gaussian) shows a higher PSNR average (about 29.9 dB), while SSIM stays near 0.87. Gains concentrate on several textured scenes, such as Pairs 10–12 and 20, which are more sensitive to seam handling. These trends match the expectation for tile-based super-resolution with windowed blending on textured scenes [12,16-18].

DOI: 10.54254/2755-2721/2026.TJ29494

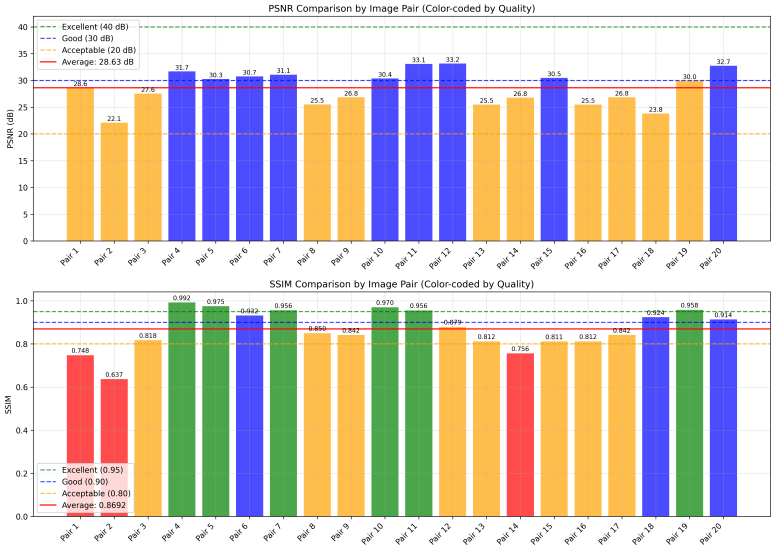



Figure 3. Linear weighting: PSNR and SSIM by image pair (picture credit: original)

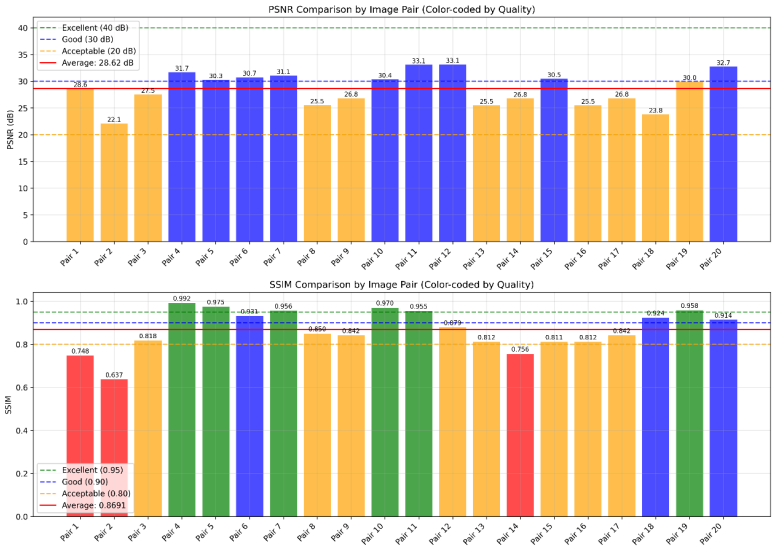



Figure 4. Cosine weighting: PSNR and SSIM by image pair (picture credit: original)

DOI: 10.54254/2755-2721/2026.TJ29494

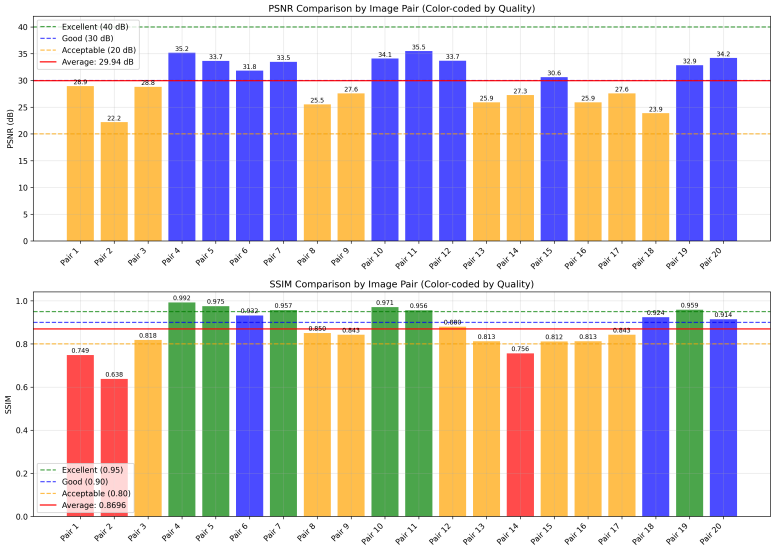



Figure 5. Gaussian weighting: PSNR and SSIM by image pair (picture credit: original)

The resolution summary in Table 1 gives a clear recipe. A Gaussian weight is preferred at every resolution tested. The best overlap rises with resolution: overlap (ov)=16 at 1280×720, ov=24 at 1920×1080, and ov=32 at 2560×1440 and 3840×2160. PSNR increases from 28.5 dB at 720p to 31.7 dB at 4K, and SSIM rises from 0.826 to 0.875. Latency per megapixel increases with resolution and overlap, which is expected for tile fusion and bandwidth cost [4,18].

Table 1. Best configuration per resolution table

| Resolution | Best Weight/ov | PSNR (dB) | SSIM | Latency (ms/MP) |
|---|---|---|---|---|
| 1280×720 | Gaussian / ov = 16 | 28.50 | 0.8264 | 338.1 |
| 1920×1080 | Gaussian / ov = 24 | 29.42 | 0.8360 | 446.8 |
| 2560×1440 | Gaussian / ov = 32 | 30.32 | 0.8502 | 595.8 |
| 3840×2160 | Gaussian / ov = 32 | 31.74 | 0.8747 | 635.9 |

The overlap sweep at 4K in Table 2 isolates the effect of overlap while holding the Gaussian window. PSNR moves from 31.11 dB at ov=8 to 31.74 dB at ov=32. SSIM follows the same small upward step, from 0.869 to 0.875. Latency rises more steeply: about 249 milliseconds per megapixel (ms/MP) at ov=8, 360 ms/MP at ov=16, and about 636 ms/MP at ov=32. The quality gain is real but modest; the speed cost is not [11,19].

Table 2. Overlap sweep table for Gaussian at 4K

| Overlap | PSNR (dB) | SSIM | Latency (ms/MP) |
|---|---|---|---|
| 8 | 31.11 | 0.8689 | 249.32 |
| 16 | 31.15 | 0.8696 | 359.65 |
| 24 | 31.73 | 0.8747 | 512.72 |
| 32 | 31.74 | 0.8747 | 635.85 |

A direct weight comparison of 4K in Table 3 shows the main effect. With ov=8, Gaussian improves PSNR by about 2 dB at ov=16 and about 2.3 dB at ov=8 while keeping SSIM similar

DOI: 10.54254/2755-2721/2026.TJ29494

(~0.869) and latency competitive. The same gap appears at ov=16. These results indicate that a smooth, bell-shaped window reduces tile boundaries without blurring edges that matter for PSNR [16,17].

Table 3. Weight comparison table (4K, ov=8 and ov=16)

| Overlap | Weight | PSNR (dB) | SSIM | Latency (ms/MP) |
|---|---|---|---|---|
| 8 | Linear | 28.818 | 0.869 | 280.576 |
| 8 | Cosine | 28.812 | 0.869 | 260.727 |
| 8 | Gaussian | 31.112 | 0.869 | 249.319 |
| 16 | Linear | 29.167 | 0.869 | 349.187 |
| 16 | Cosine | 29.159 | 0.869 | 349.039 |
| 16 | Gaussian | 31.148 | 0.870 | 359.653 |

Pareto plots in Figures 6 and 7 make the speed–quality trade-off visible. At 1080p and 1440p, the frontier is dominated by the Gaussian family. Increasing overlap shifts the point rightward (slower) with only small upward movement (PSNR) [16]. Linear and cosine stay below the frontier. Practitioners who target real-time use will likely sit near Gaussian/ov=8 or ov=16 [4,13,17,18].

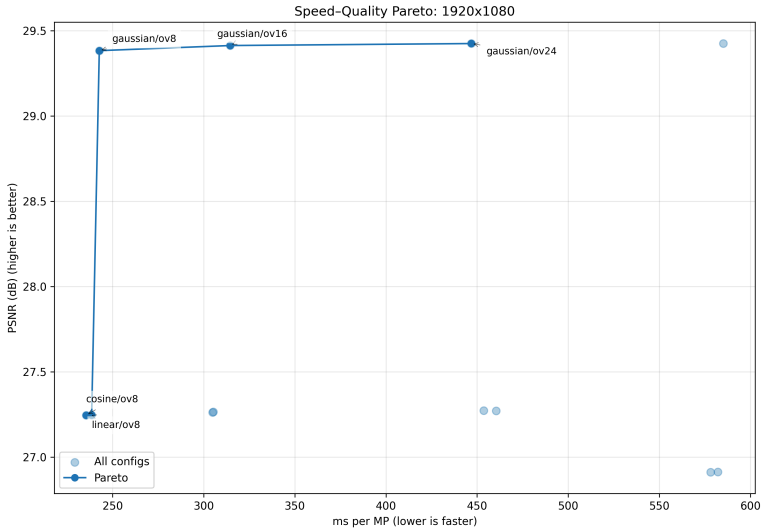



Figure 6. Pareto front at 1080p (picture credit: original)

DOI: 10.54254/2755-2721/2026.TJ29494

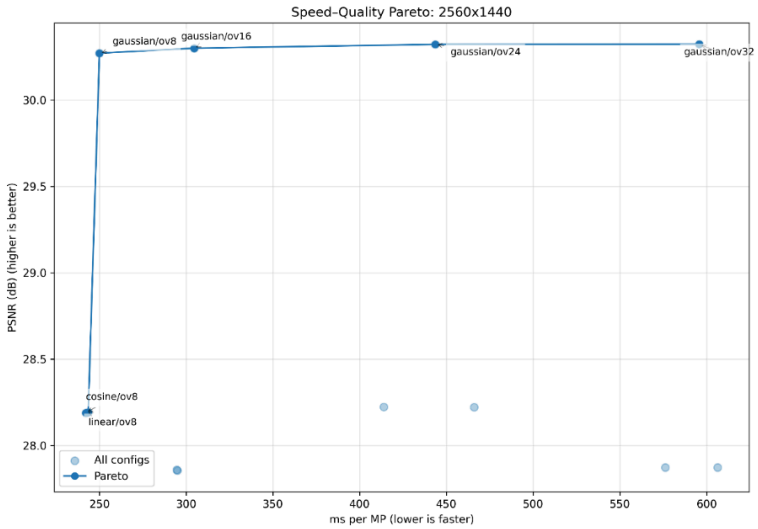



Figure 7. Pareto front at 1440p (picture credit: original)

The PSNR distribution by weight shown in Figure 8 and the SSIM distribution by weight shown in Figure 9 summarize stability across scenes. The PSNR median rises from about 28.1 dB for linear and cosine to about 30.0 dB for Gaussian. SSIM medians remain close across weights (~0.87). The Gaussian group also shows a higher upper whisker, which reflects several high-detail scenes where seam suppression matters most [8, 9, 18-20].

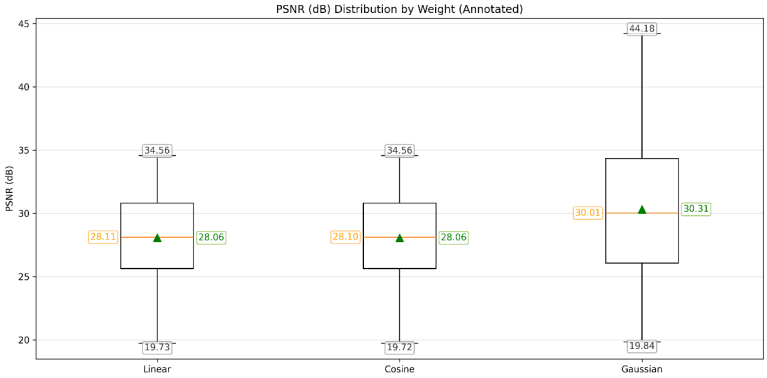



Figure 8. PSNR distribution by weight (picture credit: original)

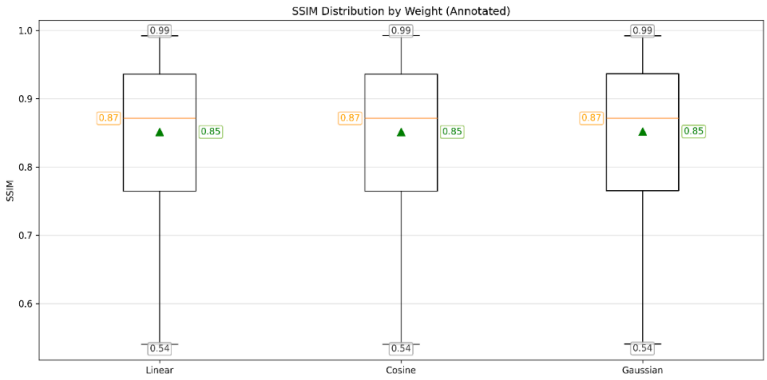



Figure 9. SSIM distribution by weight (picture credit: original)

DOI: 10.54254/2755-2721/2026.TJ29494

### 3.2. Discussion

The results point to two simple rules. Use a Gaussian window; choose a moderate overlap. The first rule is consistent with prior reports that windowed blending reduces ringing and step edges at tile borders, which inflate error even when the local reconstruction is sharp [12,16,17]. The window distributes responsibility across tiles and avoids hard cuts; the gain appears as a steady ~2 dB lift in PSNR without a drop in SSIM. Recent tile-aware pipelines for restoration and super-resolution make similar choices, often with cosine or Kaiser forms; the data here shows that a standard Gaussian is at least as reliable and, on this set, slightly better [16,17,19].

The second rule follows from the overlap sweep and the Pareto fronts. Overlap does help, but the curve shows diminishing returns. Moving from ov=8 to ov=16 brings only a small PSNR increase at 4K, and the jump to ov=32 is even smaller, while the latency more than doubles. For interactive settings or edge devices, ov=8 or ov=16 is a reasonable ceiling. For offline rendering, ov=24–32 is justified when faint seams still appear in sensitive regions such as sky gradients or glossy edges. The boxplots reinforce this view: the Gaussian group has a higher median and a longer upper tail, indicating more wins and a few standout cases where the window fully hides boundaries.

There is also a practical note on metric behavior. SSIM is stable across weights and overlaps because structural layout changes little once seams are hidden; the main movement is in PSNR, which penalizes the subtle cross-tile offsets that linear and cosine leave behind. Several recent studies report the same split and recommend combining PSNR/SSIM with a visual test to catch residual stair-steps or halos [4,6,20]. The qualitative checks used here led to the same choice in Table 1.

In short, Gaussian weighting provides the most consistent quality lift on this data, and moderate overlap gives the best balance of fidelity and speed. These settings match current practice in tiled inference and can be adopted without model changes [4,13,18]. The remaining gap to 40 dB on a few pairs likely reflects scene content rather than blending method, which suggests that future work should tune tile size and stride per-scene or learn the window itself while keeping the runtime budget in view [11,17].

## 4. Conclusion

The work addresses seam artifacts that arise in tiled super-resolution and shows that blending design is the main lever for improvement. A Gaussian window gives consistent quality gains, while a moderate overlap balances fidelity and speed. Across twenty image pairs and multiple resolutions, peak signal-to-noise ratio increases by roughly two decibels relative to linear and cosine, and structural similarity remains stable. The tables and Pareto plots point to a practical recipe: Gaussian weighting with overlap sixteen to thirty-two pixels, scaled with resolution. Limitations include a modest dataset and a single model family. Future effort should test larger and more diverse sets, explore adaptive tile size and overlap, and learn data-driven windows under runtime budgets. These steps can push quality further while keeping the tiled pipeline simple and deployable.

## References

[1] Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., & Timofte, R. (2021). SwinIR: Image restoration using Swin Transformer. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 1833–1844.

[2] Chen, L., Chu, X., Zhang, X., & Sun, J. (2022). Simple baselines for image restoration (NAFNet). Proceedings of the European Conference on Computer Vision Workshops (ECCVW).

DOI: 10.54254/2755-2721/2026.TJ29494

[3] Wang, X., Xie, L., Dong, C., & Shan, Y. (2021). Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 1905–1914.
[4] Cun, X., Bao, J., Zhang, X., Guo, Y., Zhou, J., & Timofte, R. (2023). Swin2SR: SwinV2 Transformer for compressed image super-resolution and restoration. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).
[5] Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J., & Maier-Hein, K. H. (2021). nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. Nature Methods, 18(2), 203–211.
[6] Zukić, D., Reischl, M., Alzubaidi, A., Heisel, C., & Heisel, M. (2021). ITKMontage: A software module for image stitching. Integrating Materials and Manufacturing Innovation, 10(1), 115–124.
[7] Dolezal, J. M., et al. (2024). Slideflow: Deep learning for digital histopathology with real-time whole-slide visualization. BMC Bioinformatics, 25, Article 240.
[8] Ren, B., et al. (2024). The Ninth NTIRE 2024 efficient super-resolution challenge report. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 6595–6631.
[9] Li, Y., et al. (2023). NTIRE 2023 challenge on efficient super-resolution: Methods and results. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 6167–6187.
[10] NTIRE 2023 Challenge on Image Super-Resolution (×4): Methods and results. (2023). Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).
[11] Zhang, K., Liang, J., Van Gool, L., & Timofte, R. (2021). Designing a practical degradation model for deep blind image super-resolution. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 4791–4800.
[12] Ke, J., et al. (2021). MUSIQ: Multi-scale image quality transformer. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 5148–5157.
[13] Yang, J., et al. (2023). Overlap-aware inference for large-image restoration. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV).
[14] Blau, Y., & Michaeli, T. (2020). The perception–distortion tradeoff revisited with modern super-resolution. International Journal of Computer Vision.
[15] Ding, K., Liu, Y., Zou, X., Wang, S., & Ma, K. (2021). Locally adaptive structure and texture similarity for image quality assessment. Proceedings of the 29th ACM International Conference on Multimedia (ACM MM), 1235–1244.
[16] Wang, J., Yue, Z., Zhou, S., Chan, K. C. K., & Loy, C. C. (2024). Exploiting diffusion prior for real-world image super-resolution. International Journal of Computer Vision.
[17] Chen, X., et al. (2023). On the choice of windows for tiled super-resolution and restoration. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW).
[18] Bakhtiarnia, A., et al. (2024). Efficient high-resolution deep learning: A survey. ACM Computing Surveys, 56(10).
[19] Li, A., et al. (2022). Memory-efficient inference for high-resolution image processing. NeurIPS Workshops.
[20] Zhang, R., et al. (2021). Perceptual losses and practical IQA for restoration. International Journal of Computer Vision.