Every Pixel Has its Moments: Ultra-High-Resolution Unpaired Image-to-Image Translation via Dense Normalization (Supplementary Material)

Ming-Yang Ho[1], Che-Ming Wu[2], Min-Sheng Wu[3], and Yufeng Jane Tseng[1]

[1]National Taiwan University [2]Amazon Web Services [3]aetherAI
kaminyou@cmdm.csie.ntu.edu.tw, unowu@amazon.com, vincentwu@aetherai.com, yjtseng@csie.ntu.edu.tw

S1 Implementation details

We implemented our Dense Normalization (DN) layer using PyTorch 1.13.0 and Python 3.9.5. All experiments were conducted on an Ubuntu 20.04.5 LTS operating system, equipped with an NVIDIA RTX 3090 GPU. For all experiments involving KIN, we used a constant kernel size of 5, as suggested by the authors of KIN. To reduce distortion in the margins during translation, each patch is initially reflectively padded, and then unpadded post-translation.

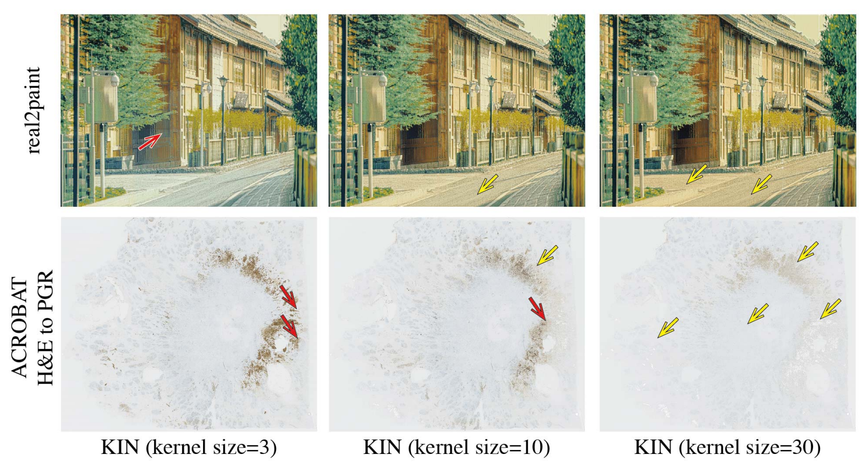


Fig. S1: The impact of varying kernel size in KIN [1]. Careful selection of kernel size is critical for balancing the removal of tiling artifacts and preservation of color and hue details to enhance the quality of translated images generated by KIN. Red arrows (↘) indicate tiling artifacts, and yellow arrows (↘) indicate over/under-colorizing.

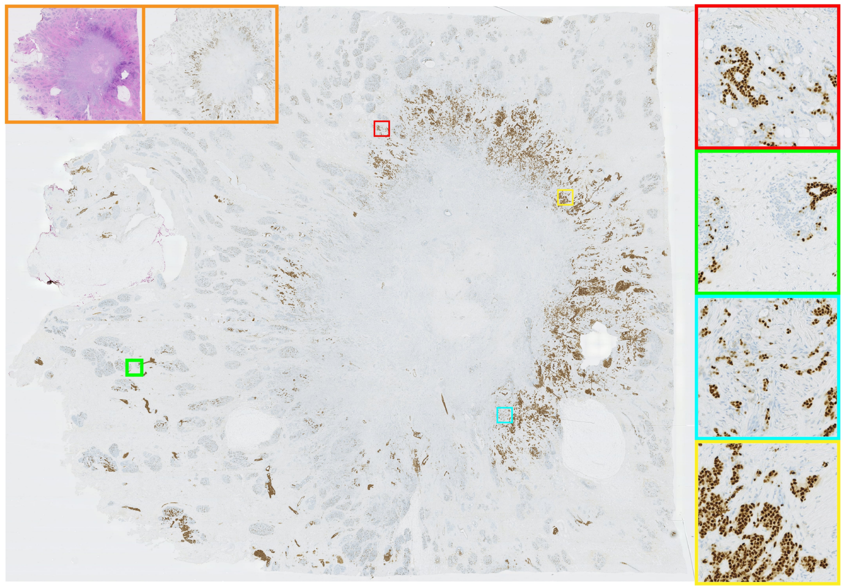
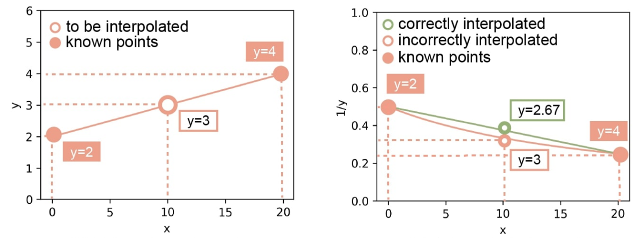


Fig. S2: A translated result of an ultra-high-resolution image (24,000×28,000 pixels) generated using our Dense Normalization (DN). Our DN is able to transform a whole slide image stained with hematoxylin and eosin (H&E) into a progesterone receptor (PGR) stain without any discernible tiling artifacts, while also retaining all color and hue details. The right side of the image contains four close-up boxes for closer examination.



(a) Interpolate the value of y for x=10.

(b) Interpolate the value of 1/y for x=10.

Fig. S3: Reciprocal-based interpolation. These figures illustrate the challenges involved in computing interpolation on the reciprocal. In (a), two points (0, 2) and (20, 4) are given in the Cartesian coordinate system, and the interpolated value of y at $x = 10$ is obtained by taking the mean of 2 and 4, resulting in 3. In (b), when the interpolated results are transformed into the reciprocal form, they exhibit a hyperbolic function (orange line). However, it is preferable for the reciprocal of the interpolated standard deviation to be a linear function (green line) to prevent any nonlinear transformations from occurring during normalization of the image.

two-stage DN

single-pass DN

Fig. S4: Comparison of two-stage and single-pass DN. A naïve implementation of DN might resemble KIN, operating in two stages. However, our dispatcher design and prefetching strategy enable the prefetching branch to run in parallel with the inference branch across most neural network (NN) layers, and to execute asynchronously in the DN layer, effectively hiding the runtime of the prefetching branch.

Table S1: Comparison of runtime and GPU memory usage. Using an NVIDIA RTX 3090 GPU, we benchmarked the runtime and GPU VRAM usage for a 4,302 × 3,024 image. One-stage DN, despite involving substantial operations on statistical moments, runs faster than KIN.

	IN*	TIN	KIN	DN	DN
Statistics type	patch-level	image-level	patch-level	pixel-level	pixel-level
# of pipeline stage	1	1	2	2	1
Operations on statistics			✓	✓	✓
Runtime (s)	2.46	2.62	4.42	5.51	4.35
GPU VRAM usage (mb)	2951	3335	3145	3161	4157

IN*: patch-wise IN

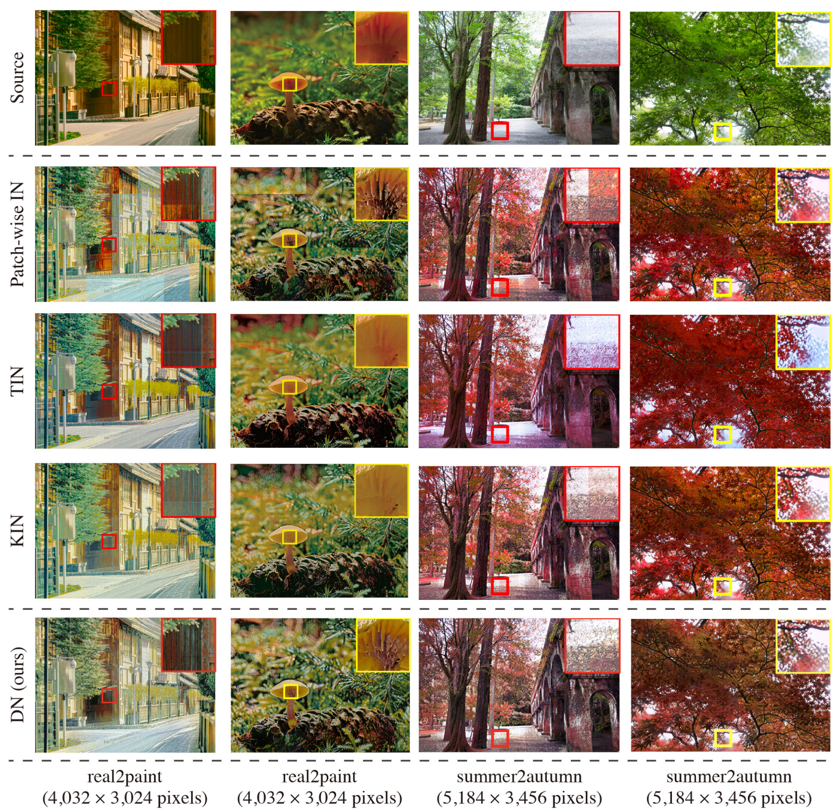


Fig. S5: Results of translation on natural images. The figure compares the transformation results on UHR images using four normalization methods: patch-wise IN [2], TIN [3], KIN [1], and DN with a CUT [4] framework. Red close-up boxes highlight both tiling artifact comparisons, while yellow close-up boxes focus on evaluating over/under-colorizing and local hue preservation. DN shows the best performance overall. For a better view, please zoom in.

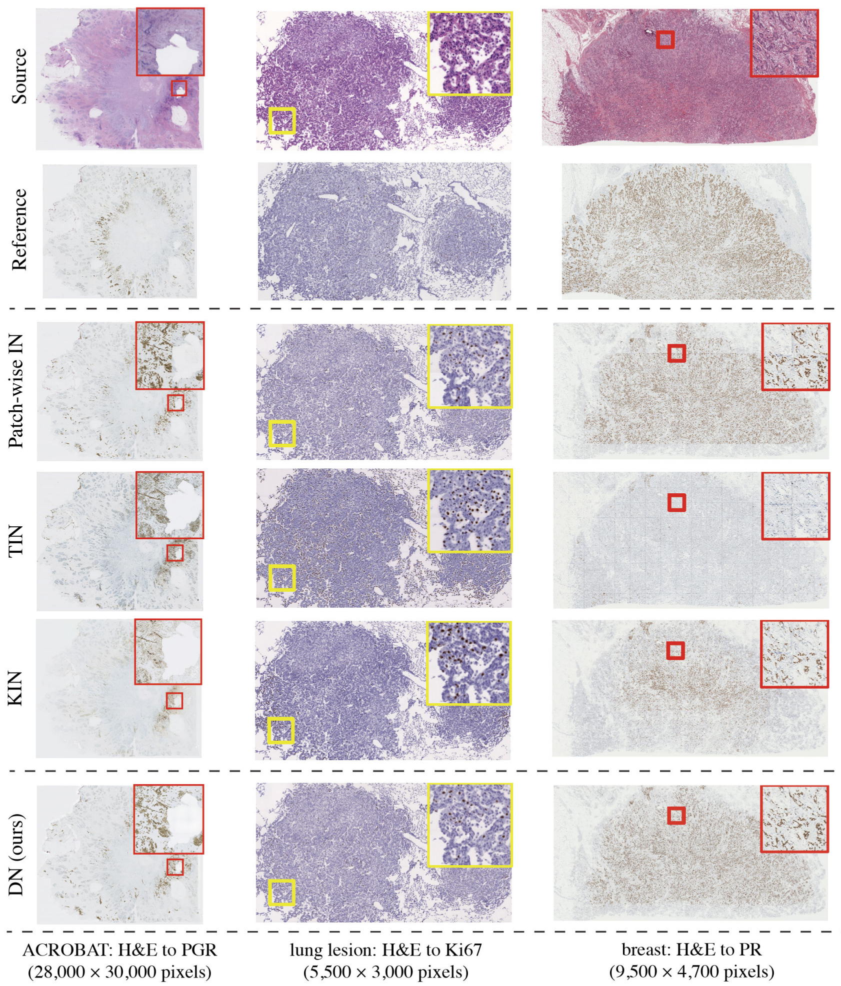


Fig. S6: Results of translation on pathological whole slide images. The figure compares the stain transformation results on UHR whole slide images using four normalization methods: patch-wise IN [2], TIN [3], KIN [1], and DN with a CUT [4] framework. Red close-up boxes highlight both tiling artifact comparisons, while yellow close-up boxes focus on evaluating over/under-colorizing and local hue preservation. DN shows the best performance overall. For a better view, please zoom in.

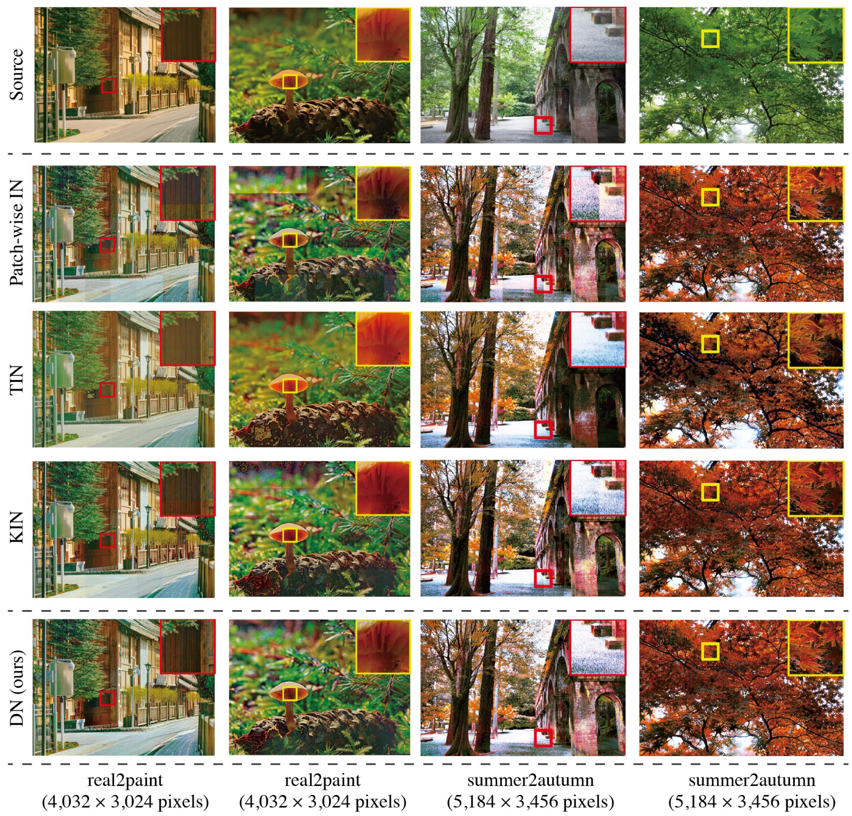


Fig. S7: Results of translation on natural images with a CycleGAN framework. The figure compares the translation results on ultra-high-resolution images using four normalization methods: patch-wise IN [2], TIN [3], KIN [1], and DN with a CycleGAN framework. Red close-up boxes highlight both tiling artifact comparisons, while yellow close-up boxes focus on evaluating over/under-colorizing and local hue preservation. DN shows the best performance overall. For a better view, please zoom in.

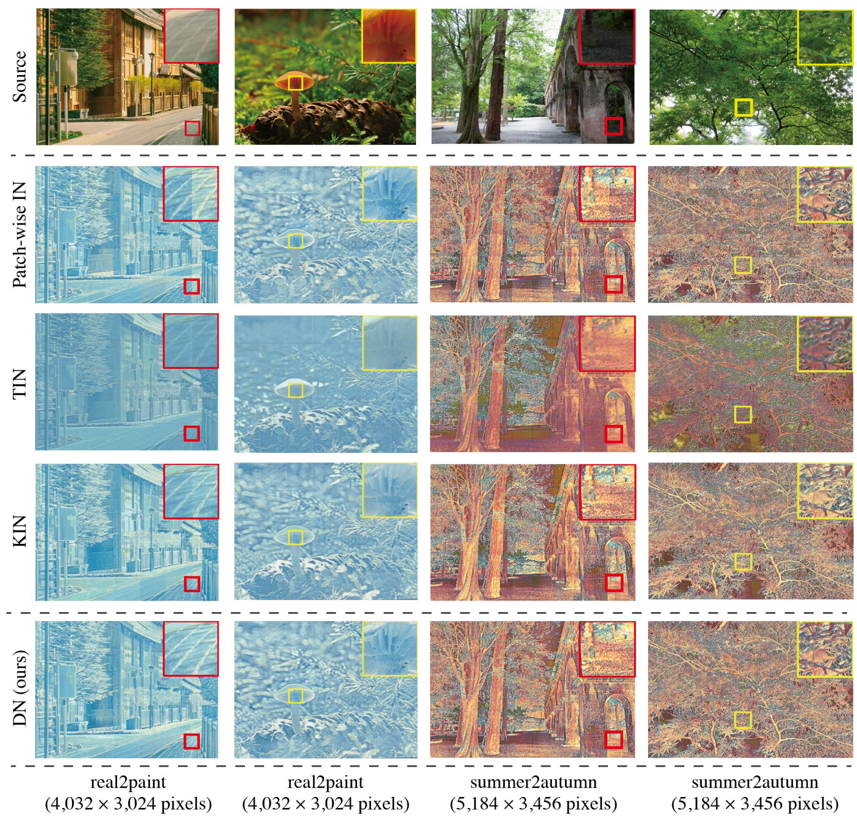


Fig. S8: Results of translation on natural images with an L-LSeSim framework. The figure compares the translation results on ultra-high-resolution images using four normalization methods: patch-wise IN [2], TIN [3], KIN [1], and DN with an L-LSeSim framework. Despite the limitations of the L-LSeSim framework in effective translation, DN is still capable of removing tiling artifacts and maintaining color details. Red close-up boxes highlight both tiling artifact comparisons, while yellow close-up boxes focus on evaluating over/under-colorizing and local hue preservation. DN shows the best performance overall. For a better view, please zoom in.

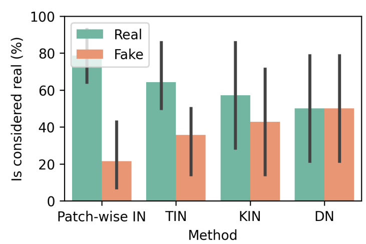


Fig. S9: Fidelity evaluation. Images generated by DN are nearly indistinguishable from real pathological images.

References

1. Ming-Yang Ho, Min-Sheng Wu, and Che-Ming Wu. Ultra-high-resolution unpaired stain transformation via kernelized instance normalization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXI*, pages 490–505. Springer, 2022.
2. Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.
3. Zhe Chen, Wenhai Wang, Enze Xie, Tong Lu, and Ping Luo. Towards ultra-resolution neural style transfer via thumbnail instance normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.
4. Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, pages 319–345. Springer, 2020.