INRetouch: Context Aware Implicit Neural Representation for Photography Retouching

Supplementary Material

Omar Elezabi, Marcos V. Conde, Zongwei Wu*, Radu Timofte
Computer Vision Lab, CAIDAS & IFI, University of Würzburg
{omar.elezabi, marcos.conde, zongwei.wu, radu.timofte}@uni-wuerzburg.de
omaralezaby.github.io/inretouch/

We first kindly refer the readers to the project website omaralezaby.github.io/inretouch/ for video results, source code, and dataset.

In this supplementary material, we provide more implementation details of our work in 1. We also provide more ablation studies in 2.

As for visuals, we first provide a comparison on visual consistency in 4. In Section 5 we discuss the limitations of the proposed approach. More visuals on retouching transfer comparison can be found in Section 6. Finally, in Section 7, we show the variety of our presets applied to a natural image.

1. Implantation Details

Compared Methods For the compared method that requires pre-training on the dataset, we modified and adapted their architectures for our task. For the Deep Preset [5] method, we modified the reference branch to take a 6-channel input. We provide the image pair before and after editing as a reference by stacking them together. For Neural Preset, we modified the architecture to generate an editing mask with the same size as the input instead of just a modification vector to allow for local modification. Similarly, we use the pair of before and after editing stacked together as the reference to the model. For the Style GAN [6] based method, we used the Domain Alignment Module proposed in [3]. This module was proposed to apply color changes to an image, based on a provided reference. We modified the module to take the stacked pair of before and after editing as a reference. We emphasize that all the compared methods take the same input information (reference before and after editing).

For the other methods that require no pre-training on our dataset (Image Analogies [4, 8] and In-Context learning [1, 9] methods), we used the open-source models provided by the authors.

Evaluation Dataset Lightroom preset system suffers from visual inconsistency. As we see in Fig. D same preset can produce different styles when applied to different images. For an accurate evaluation process, we need to make sure the chosen reference visual style matches the style of the GT. We achieve that by choosing a reference that has the same color distribution as the input image, as it is more likely to generate the same style when applying the preset. We calculate the 3D color histogram of each reference image before editing, and we compare it with the 3D color histogram of the input image. We choose the reference image with the closest color histogram to the input images as a reference. For a fair comparison, we used the same reference in all compared methods.

Dataset Disclaimer All the images used in our dataset were obtained from the open-source MIT5k dataset [2], and all the presets used are open-access licensed under Creative Commons (CC 4.0). All dataset creation processes and components are checked to avoid any violations or misuse and to ensure ethical conduct.

2. More Ablations

INR Components Ablation : In Fig. F we show the incremental performance improvement over the baseline MLP INR architecture. Context awareness enables recognizing textures, objects, and edges, producing fewer artifacts and better editing.

Sampling Window Size In Fig. A (left), we can notice some improvement when increasing the size of the sampled window. This can be attributed to the model processing a

*Corresponding Auther

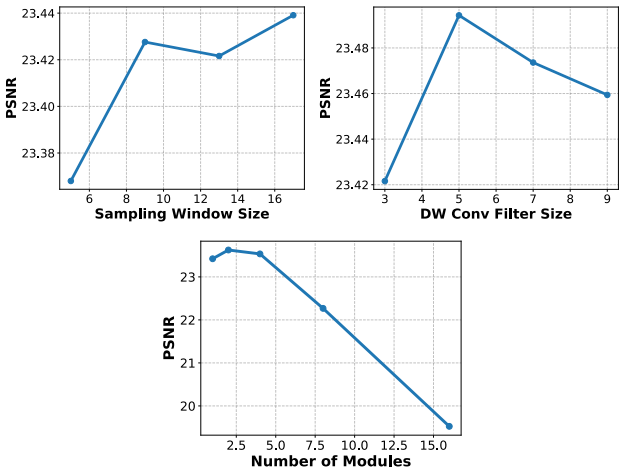

Figure A. Ablation for Sampling Window Size, Depth-Wise Conv Filter Size, and the Depth of the INR architecture. From the results, we can notice that model simplicity is crucial for good performance and for better generalization.

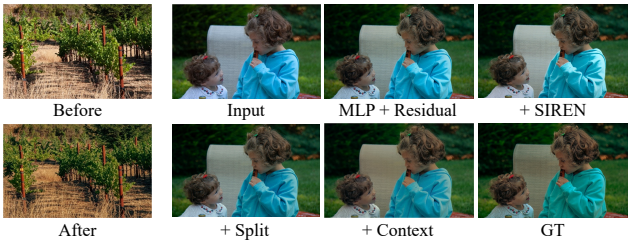

Figure B. The effect of different components in our architecture on the output.

bigger, coherent area to learn more about update smoothness. But after some degree, we see no noticeable improvement. For our experiments, we chose a sampling Window size of 13 for the best trade-off between cohesion and memory footprint during training.

DW CNN Filter Size Fig. A (right) shows that increasing filter size can improve performance as it considers more information from neighboring pixels. However, increasing the filter size introduces more parameters that require more time to optimize and can result in overfitting issues with a drop in performance. We choose the filter size of 3 for fast optimization and as less parameters increase as possible.

Depth of the INR architecture Fig. A (down), we notice that increasing the size of the INR by adding more layers reduces the INR performance. When adding more parameters, the network tends to overfit on the reference, failing to generalize to new images.

3. Testing on Professionally Edited Images

In FigC we show additional qualitative results of various methods tested on the professionally edited images collected online. We can appreciate our method's ability to adaptively learn and apply a variety of edits, including local and region-specific edits. Additionally, we can notice superior visual consistency between the output produced by our method and the reference in comparison with other methods, including Lightroom presets. Please refer to "Professional Test Data" for the full 100 pairs test data, including all the outputs produced by the compared methods.

4. Visual Style Consistency

In Fig. D, we compare the editing consistency of our proposed method with that of the widely-used Lightroom preset. Our method demonstrates the ability to produce more realistic outputs that better adhere to the reference edits, validating its effectiveness and highlighting its superior visual consistency.

Lightroom presets work by saving the Lightroom edits applied to an image. These presets are usually created to process RAW images. These edits consist of image processing pipeline operations like color correction, hue adjustment, and exposure correction. These operations affect every image differently depending on the image details, like the sensor of the camera, lighting conditions, and color distribution. This limits the visual reproducibility of the edits to similar images. Additionally, saving edits in formats like presets is software-specific, so they require the same software to use them. Our method provides a more visually consistent way to transfer edits between images without being software-dependent.

Comparison with Style Transfer The current style transfer methods use a single reference image to represent the style of the desired output. As we see in Fig. E, these methods fail in the task of photo retouching. The task of photo retouching requires fine edits and specific color changes based on location and context. It is not feasible to capture these edits using only a single reference image. We tested different style transfer methods developed for photo editing based on a reference. We can notice artifacts in the output because of undesirable changes. Additionally, they fail to recognize the fine details of the style, limited to reference ambiguity. For a quality output, we notice these methods are limited to reference images with similar characteristics to the input image (nature, portrait) or with general and noticeable aesthetics (color filter, day-night images, etc). Our proposed approach allows the use of available references with much less limitation for high-quality output.

5. Limitations

INRs are naturally noise-suppressing because of their inductive bias towards low frequencies [7]. So, our method struggles with transferring high noise and grains. A separate noise addition module will be useful for better transfer.

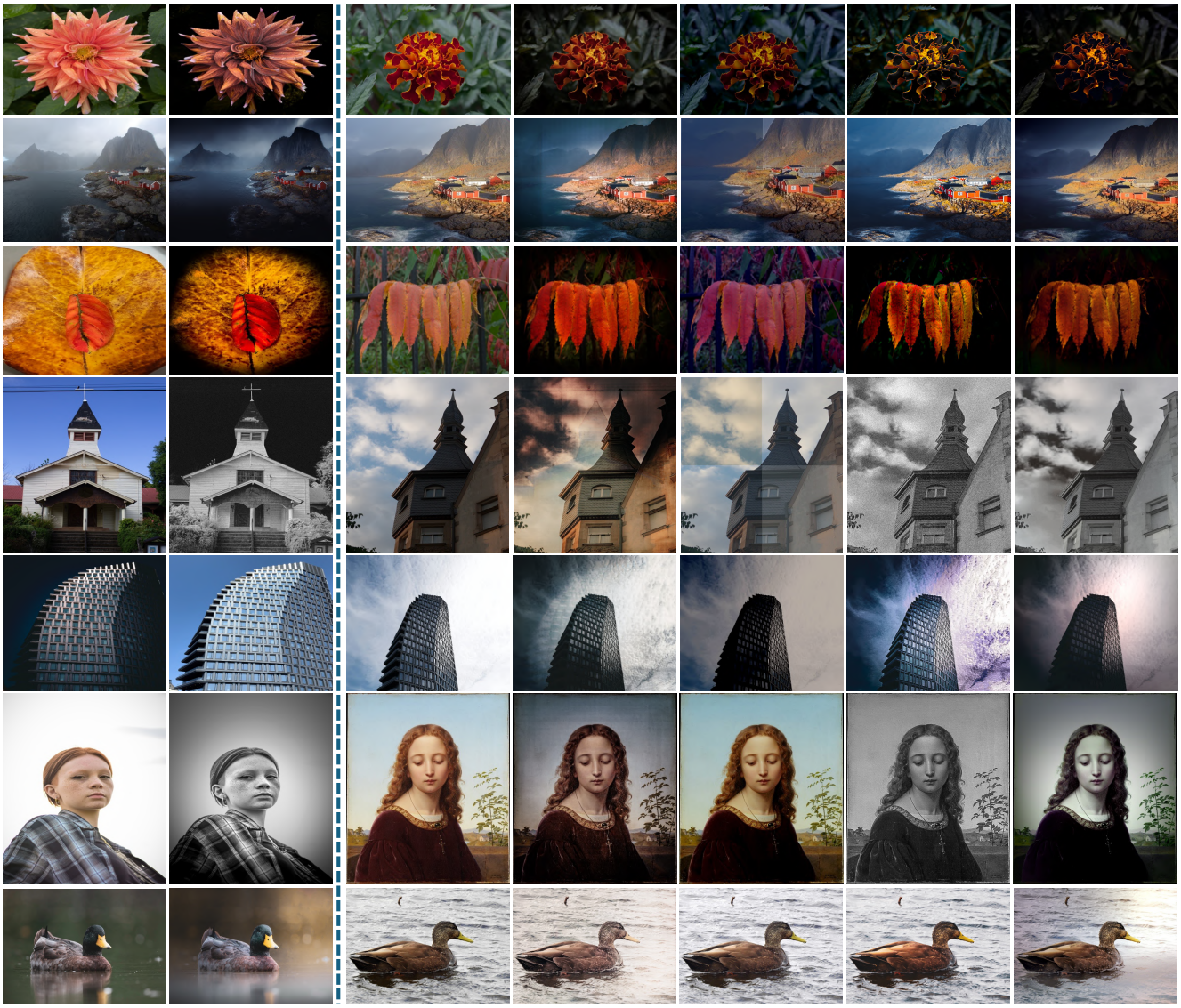

Figure C. Additional examples of **the comparison between different methods on professionally edited images.** Qualitative Comparisons. We can appreciate the visual consistency between the reference and our method output. Our method is able to recognize objects and apply local and region-specific edits accurately.

Additionally, since our method learn from one reference, the closer the reference to the input, the better the produced results. Even though our method can learn from a reference that is totally different from the input image, as we showed in our evaluation, having a reference with minimal variety (uniform colors) and no resemblance to the input image can produce sub-optimal results. However, this use case is not common as humans usually tend to utilize visual clues, choosing references that resemble some similarity to the input images.

6. More Visual Results

We show in Fig. F the qualitative results of various methods for the retouching transfer task. Our approach excels in accurately learning the edits from before-and-after image pairs, producing outputs that are not only more realistic but also better aligned with the intended edits. In contrast, other methods struggle to achieve similar fidelity, often resulting in noticeable artifacts and inconsistencies. This highlights the effectiveness and robustness of our method in capturing and applying complex retouching transformations.

Figure D. Additional examples of the issue of **visual editing consistency** in presets. Some presets produce a different style when applied to different images. We can appreciate our method providing a more visually consistent edit.

7. Dataset Presets

To ensure the versatility and robustness of our dataset, we curated a diverse collection of varying presets, designed to simulate a wide range of editing styles and conditions. As shown in Fig. G, we apply some of these presets to a single natural image, showcasing the richness and variety inherent in the dataset. This comprehensive coverage not only highlights the adaptability of our approach to diverse editing scenarios but also establishes our dataset as a valuable resource for developing and evaluating methods capable of handling complex retouching tasks. Such diversity enables generalizing effectively across different styles.

References

[1] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35: 25005–25017, 2022. 1

[2] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1

[3] Ruicheng Feng, Chongyi Li, Huaijin Chen, Shuai Li, Jinwei Gu, and Chen Change Loy. Generating aligned pseudo-supervision from non-aligned data for image restoration in under-display camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5013–5022, 2023. 1

[4] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 557–570. 2023. 1

[5] Man M Ho and Jinjia Zhou. Deep preset: Blending and retouching photos with color style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2113–2121, 2021. 1

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1

[7] Chaewon Kim, Jaeho Lee, and Jinwoo Shin. Zero-shot blind image denoising via implicit neural representations. *arXiv preprint arXiv:2204.02405*, 2022. 2

[8] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017. 1

[9] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6830–6839, 2023. 1

Figure E. Comparison between using a **style reference (Style Transfer), and a Before-After editing reference (Retouch Transfer)**. Having the reference before and after editing provides more control over the edits applied, producing a higher fidelity output with fewer artifacts.

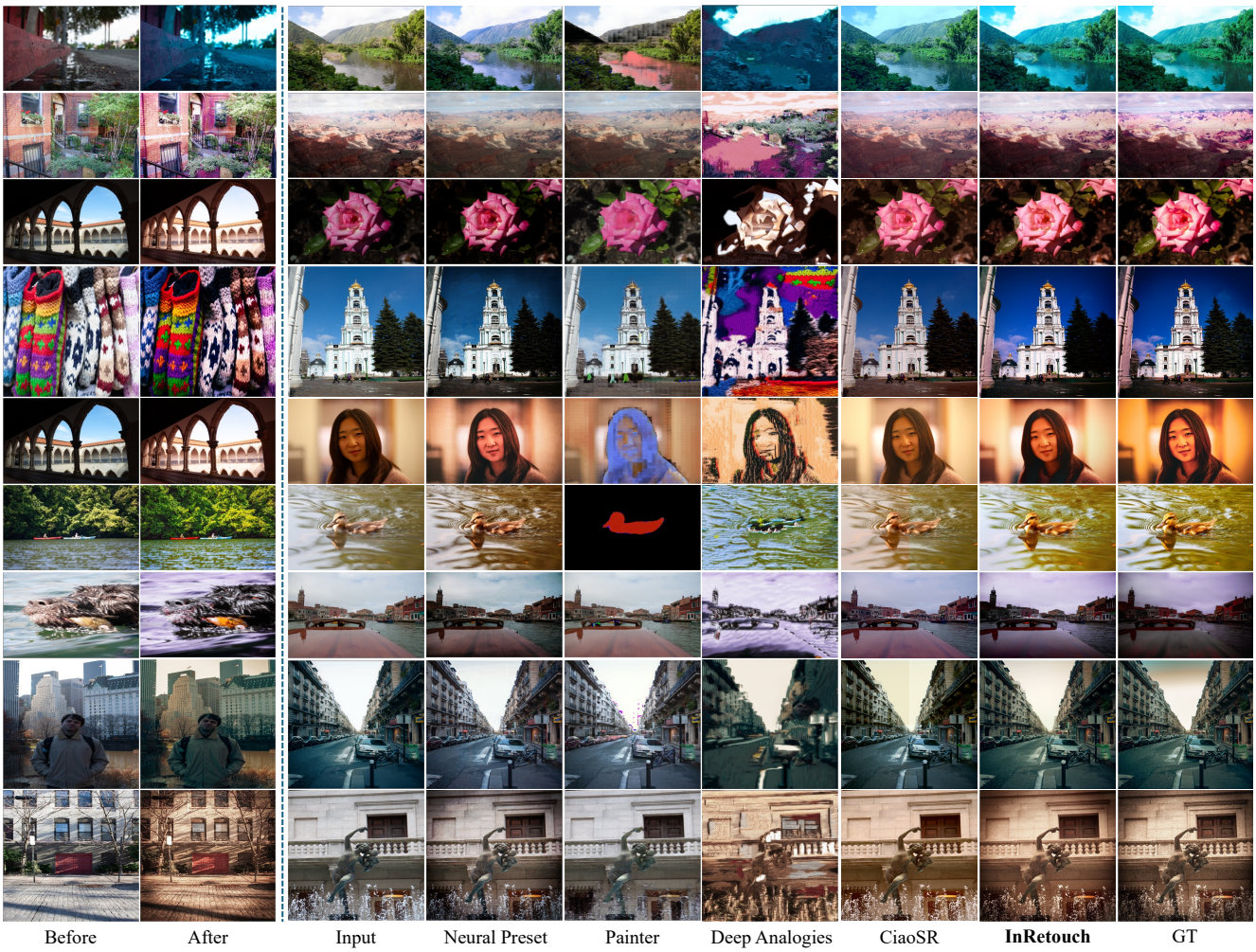

Figure F. Additional examples of **the comparison between different methods on retouching transfer task.** Our method learns the edits effectively from a single sample, generalizing to a wide variety of edits, and has the most consistent output with the GT. We can appreciate the ability of our method to learn and adapt to complex edits like vignetting and local modification.

Figure G. Visualization of the **variety of edits in the used presets**. Images are produced by applying different presets to a natural image (highlighted top-left).