One-shot Detail Retouching with Patch Space Neural Transformation Blending

Fazilet Gokbudak and Cengiz Oztireli
Department of Computer Science and Technology, University of Cambridge
{fg405, aco41}@cam.ac.uk

Abstract—Photo retouching is a difficult task for novice users as it requires expert knowledge and advanced tools. Photographers often spend a great deal of time generating high-quality retouched photos with intricate details. In this paper, we introduce a one-shot learning based technique to automatically retouch details of an input image based on just a single pair of before and after example images. Our approach provides accurate and generalizable detail edit transfer to new images. We achieve these by proposing a new representation for image to image maps. Specifically, we propose neural field based transformation blending in the patch space for defining patch to patch transformations for each frequency band. This parametrization of the map with anchor transformations and associated weights, and spatio-spectral localized patches, allows us to capture details well while staying generalizable. We evaluate our technique both on known ground truth filters and artist retouching edits. Our method accurately transfers complex detail retouching edits.

Index Terms—Computational Photography

1 Introduction

Photo retouching is often desirable as it improves the aesthetic quality of photographs by eliminating imperfections and highlighting subjects of interest. Even with significant progress in digital photography owing to advancements in camera sensors and image processing algorithms, professional retouches via manual adjustments are still needed to achieve a desired look. These artistic edits require considerable manual effort as they consist of global adjustments, such as brightening and contrast enhancement, as well as fine edits applied to local regions. Professionals spend a great deal of time to generate such edits y, which motivates us to automatically mimic a specific style or type of retouch.

The development of automatic photo retouching tools can be helpful for both novice users and experts as it offers a basis for a professional retouching style. However, automating detailed edits of professionals is challenging as their editing pipelines are spatially varying, context-aware, and highly nonlinear, containing per-pixel adjustments. Recent learning-based methods address this complexity in image-to-image translation by proposing local context-aware methods, such as pixel-adaptive neural network architectures [1, 2], learning parameters of local filters [3], or multi-stream models to extract global and local features separately [4]. However, these data-driven methods require a large dataset of matching example image pairs. Even then, the mappings are sensitive to segmentation errors, unseen semantic regions, and image content [5].

Motivated by the gap between manual and automatic enhancement, we propose a novel photo retouching technique that can learn global and local adjustments from just *a single example image pair*. Our method thus sidesteps the need for large datasets, which are very difficult to obtain for the detail retouching task. We allow users to choose one example *before-after* pair from which our technique learns the underlying retouching style. Subsequently, we can apply the retouching edit to a different input image.

We assume that example and input images share similar local content. The user can thus decide on the semantics of the example and input photos and the structural changes to be transferred. This is easy for humans and practical for many scenarios, e.g. face edits transferred to faces. Our method then handles the difficult part for humans: capturing how fine details change in an edit and applying those automatically to a new image. The method can further be combined with brushes if fully automatic transfers are not desired.

We achieve these by defining the retouching problem as a map that is given by a *spatio-spectral patch-space neural field based transformation blending*. This representation is primarily inspired by professional detail retouching pipelines as we elaborate on in Section 3. Our map representation is composed of learned patch maps at multiple scales, i.e. frequency bands. Each of these maps is represented by a number of *transformation matrices* blended with *patch-adaptive weights* that are represented as neural fields. We jointly optimize the transformation matrices and corresponding weights for each band. This representation captures edits to details better than any previous techniques while staying generalizable to new images. It is also simple enough to be extended in many different ways in future works.

In summary, there are two main contributions of this work:

- **A novel patch-space image map representation** as a blending of transformation matrices with neural fields.
- **A one-shot detail retouching algorithm** that allows transfer of edits to details to new images based on a single before-after image pair.

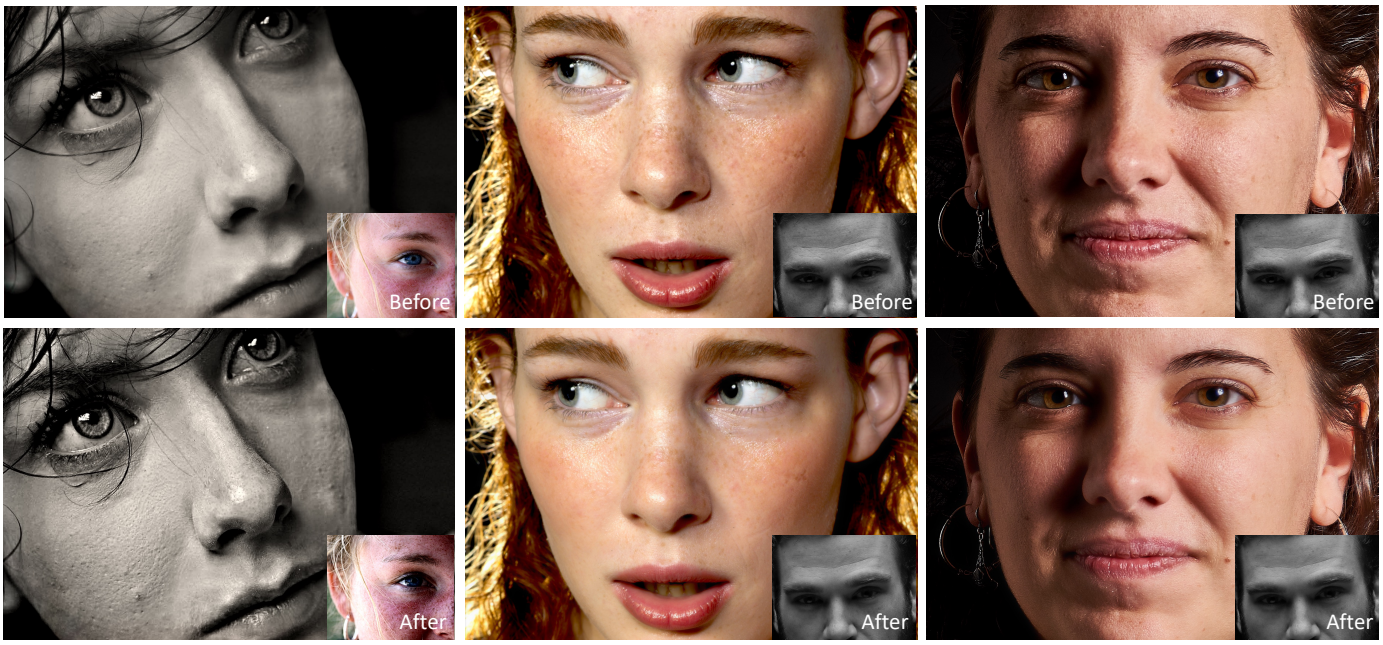

Fig. 1: Our technique automatically transfers retouching edits to new images by learning the desired edits from one example *before-after* pair (insets). The transferred edits accurately capture intricate details such as wrinkles, dark spots, strands of hair, or eyelashes, as shown in the input (top) and retouched (bottom) pairs.

2 Related Work

Photo retouching has been explored in image processing and computer vision communities under different domains, such as photo enhancement and image-to-image translation. Below we first discuss recent methods on photo enhancement and then image to image map definitions with the main focus on learning-based methods.

2.1 Digital Photo Enhancement

Global image enhancement. Color and tone transfer has been considered a very effective technique to improve the perceptual quality of photos with pre-defined rules or examples [6]. Earlier methods typically apply global changes and adjust image statistics [7, 8, 9, 10, 11, 12, 13, 14], e.g., mean and standard deviation, without considering image content and local variations [15]. These methods generally transfer color changes, ignoring edits in fine details. On the other hand, our method learns a mapping per frequency band, capturing transfers even in high frequencies. Bychkovsky et al. [7] collected the MIT-Adobe FiveK dataset of 5,000 photographs and their retouched versions by five artists. The authors propose a regression model to learn artists' retouching styles from before-retouched pairs. Chen et al. [16] introduce a fully-convolutional neural network model to learn global image processing operators, such as photographic style, nonlocal dehazing, and pencil drawing. In [17], a photo retouching pipeline for various post-processing operations is presented, where global adjustment curves are approximated. The authors suggest a deep reinforcement learning approach to model users' edit preferences from a given photo collection.

Nevertheless, global transfers cannot capture local and regional variations in a photo [15].They may result in artifacts when the local target regions of the example and input images do not match. We adapt our mappings to each image patch separately, thus accurately capturing local edits in intricate details.

Local context-aware image enhancement. To capture local variations, different methods have been proposed, such as learning local representative color transform [18], estimating an image-to-illumination mapping with a local feature extractor [19], local histogram matching [20], segmentation [21, 22], combining and learning pre-defined filters [23, 24, 25, 26, 27] or with further user guidance [28, 29, 30], detection or learning of image semantics and context [4, 31, 32, 33, 34, 35], matching [36], or precise alignment [37, 38].

Furthermore, recent work has focused on learning global and local adjustments via spatially-varying filters [1, 2, 3, 4, 39]. Chen et al. [39] introduce a global feature extraction layer along with per-pixel adjustments to enhance photos. Bilateral guided joint upsampling [40] also allows for local and global image processing with an encoder-decoder approach. HDRNet [4] learn content-aware, global, and local adjustments via a two-stream convolutional architecture, which extracts local and global features separately to fit local affine transformations and encode the high-level description of images, respectively. Also, Moran et al. [3] propose to learn the parameters of three different spatially local filters to automatically enhance photos.

Local color and tone adjustments might still be insufficient to capture intricate details [8]. Transfer of such details, in general, requires a dense matching [36] or alignment between example and input images [41]. To achieve either dense matching or alignment, methods constrain their datasets to contain very similar example and input images, for example, faces with similar characteristics and views [41]. On the other hand, our method does not require

dense correspondences between input and example images but still transfers intricate details. It accurately represents such complex mappings with an operator summing the effects of various transformations multiplied with corresponding patch-adaptive weights, applied at multiple frequency bands.

Differentiable image processing pipelines. To have more flexibility and control over the rendering process, methods based on image signal processors (ISP)s have been proposed to enhance photos. In both [42, 43], hyperparameters of an ISP are optimized. Different from [42], which only applies to a fixed pipeline, [43] can explore different ISP architectures. Furthermore, [44] model a commercial raw processing pipeline with a series of neural networks to render sRGB images from raw inputs. As we assume example and input images to be processed RGB images rather than raw data, we refrain from comparing our method with such ISP-based methods.

2.2 Defining Maps between Images

Unsupervised methods. Some learning-based techniques only require one or more examples of retouched photos without their before examples to learn the transfer. Such unsupervised methods capture a certain style by decomposing images into a reflection map and an illumination map [45], extracting and recomposing band representations of training images [46], regularizing unpaired training using information extracted from the input [47], segmenting the image into semantic regions [48], adaptive image regions [49], learning semantic and global features [24], progressively translating image from coarse to fine via pyramids of generative models [50], or utilizing artistic principles and pre-defined filters [17, 51]. These methods transfer pre-defined elements of the desired style, or global color and tone. Defining the desired style and the content of the input image that is to remain is challenging. Hence, these methods typically assume prior knowledge of the type of desired adjustments. Even then, capturing the retouching edits in details remains out of scope since these methods are typically designed for domain transfer, working on high level features of images.

Supervised methods. For a conceivable representation, many supervised transfer methods require a large dataset of well-aligned example image pairs whose contents are very similar [18, 19]. However, finding or generating such a dataset is difficult as the content of images can change dramatically. Even with such a dataset, segmentation errors, unseen semantic regions, or image content can still change the results significantly [52]. In contrast, our method allows users to choose the example pairs from which the desired style is learned, hence sidestepping the challenging semantics problem. Similar content and structures between example and input images lead to more natural transfers.

Convolutional neural networks (CNNs) are the de-facto model for image processing with supervised learning methods. While CNNs present state-of-the-art results in computer vision tasks, they are not required [53]. MLP-based architectures have recently gained popularity in image classification and image-to-image translation. Cazenavette and De Guevara [54] propose the MLP-Mixer architecture that only uses simple MLP blocks to learn image classification. Cazenavette and De Guevara [54] also show an application of an MLP-based architecture for image synthesis. They adapt the MLP-Mixer architecture [53] to perform unpaired image-to-image translation. Our observation that MLP-based architectures attain competitive results in challenging vision tasks motivated us to explore the use of an MLP block as an alternative to CNNs in the context of photo retouching.

3 Overview and Motivations

Given a pair of example images X and Y, we aim to learn a map M such that $Y = M(X)$. The learned map can then be applied to a new input image I to obtain the retouched output $O = M(I)$.

To define this map, we first decompose the example images into multiple feature maps X_l, Y_l capturing details at different scales, such as coefficients at different bands of a Laplacian pyramid. We then define a separate mapping for each X_l, Y_l pair in the patch space as a blending of transformation matrices with neural field based weights, all learned jointly. We illustrate the overall map representation in Figure 2.

For transfer of edits, the M_l are computed and applied to each patch of the decomposition I_l of an input image I to obtain the corresponding output patch of O_l. The patches are finally placed at their spatial locations and averaged to reconstruct each image O_l , which are summed to get the output image O.

Our motivation behind designing such a map representation with frequency decomposition and transformation blending comes from studying the nature of retouching edits. First, artists often decompose images into different frequency bands to have better control over structural and textural edits to details. Second, image patches of similar content, e.g. skin or hair, are retouched similarly. This means similar patches in the patch space translate into similar edits. Our representation leads to a different transformation for patches of differing content. Third, these edits are typically applied via brushes for smooth transitions. Our neural field based blending allows for such smooth interpolation, mimicking such brush strokes.

Although we are inspired by professional artist pipelines, we illustrate in the next sections that this new image to image mapping representation can replicate the effect of and transfer edits for many filters.

4 One-shot Retouching

4.1 Frequency Decomposition

We first decompose example and input images into different frequency bands by constructing a Laplacian pyramid to capture details at multiple scales. In principle, it is possible to utilize any multiscale image decomposition method. However, we observed that a basic Laplacian pyramid helped us capture more accurate and generalizable results compared to a guided or bilateral pyramid. Therefore, we decompose images by

$$X_l = L_l(X) = \begin{cases} X - G(2) * X & l = 0 \\ G(2^l) * X - G(2^{l+1}) * X & l > 0, \end{cases} \quad (1)$$

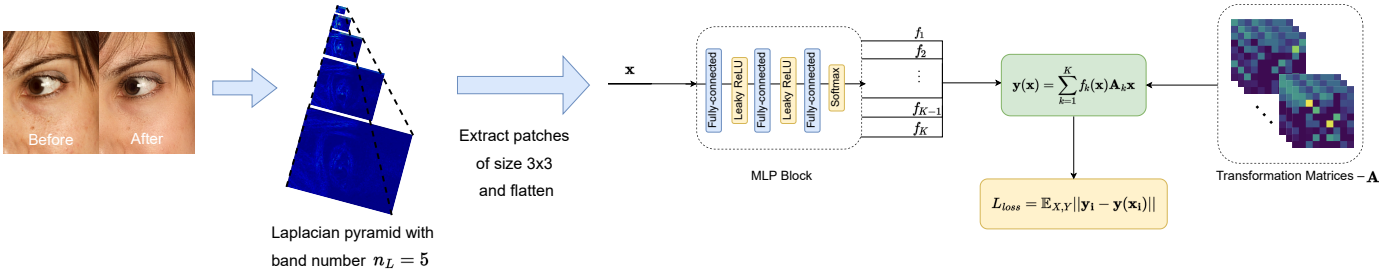

Fig. 2: Our technique learns a separate mapping per frequency band by decomposing images into five different bands with a Laplacian pyramid. At each band l, we define a mapping between flattened patches $\mathbf{x}_i$, $\mathbf{y}_i$ extracted from before-after bands X_l, Y_l. Our field based method (MLP block) adapts transformations to input patches, providing local context-aware adjustments. All transformation matrices and MLP parameters are learned jointly from scratch per band for each before-after pair.

where $G(\sigma)$ is the normalized Gaussian kernel, and $*$ denotes convolution. We also store the low-pass filtered image $S(X)$ such that $X = S(X) + \sum_{l=0}^{n_L} L_l(X)$. We then downsample each $L_l(X)$ and $S(X)$ according to the maximum frequency present at that band. This allows us to use small 3×3 patches at each band. In our experiments, we used $n_L = 5$ bands for the Laplacian pyramid.

Since each band is processed independently, we explain the steps of our technique below for two generic images X and Y.

4.2 Transformation Blending

The mapping is defined between patches $\mathbf{x} \in \mathbb{R}^{d_X}$ to $\mathbf{y} \in \mathbb{R}^{d_Y}$ extracted from X and Y, respectively, where we denote the patches with vectors stacking the pixel values and define the patch spaces as $\mathbb{R}^{d_X}$ and $\mathbb{R}^{d_Y}$. For all results in this work, we work with 3×3 patches and thus $d_X = d_Y = 9$.

Our mapping takes the form of a weighted average of learned transformation matrices, where each transformation matrix is first multiplied with its corresponding blending weight:

$$\mathbf{y}(\mathbf{x}) = \sum_{k=1}^{K} f_k(\mathbf{x})\mathbf{A}_k\mathbf{x}, \tag{2}$$

Here, K is the number of transformation matrices, and f_k are the blending weights, learned by an MLP block of output size K. The $\mathbf{A}_k$'s and f_k's are jointly learned by minimizing the following loss on patches extracted from the before and after images.

$$L_{loss} = \mathbb{E}_{X,Y}||\mathbf{y_i} - \mathbf{y}(\mathbf{x_i})|| \tag{3}$$

Each $\mathbf{A}_k$ corresponds to a different type of transformation and the $f_k(\mathbf{x})$'s, represented with the MLP, allow for a smooth transition between different transformations. The form of f_k's is relatively simple with three fully-connected layers and nonlinear activation functions applied after each layer. This blending forms a simple but expressive transformation as we illustrate in the Section 5.

4.3 Retouching an Input Image

We process the input image I the same way as the before-after pair. First, we decompose the input into its Laplacian layers and then extract its patches per layer. After applying the learned mappings M_l to the patches of the corresponding layers $L_l(I)$ independently, we then reconstruct the Laplacian bands of the output image O_l by placing the patches at their spatial locations and averaging over the overlapping regions. Later, we obtain the final output image O by summing the outputs O_l and the residual of the input image:

$$O = S(I) + \sum_{l=0}^{n_L} M_l(L_l(I)) \,. \tag{4}$$

Fig. 3: Our algorithm accurately represents simple algorithmic filters, such as Gaussian (top) and unsharp masking (bottom), learned from a *before-after* pair. The *filtered* images obtained by applying the learned mapping to the *input* images have very small difference to the ground truth images obtained by direct application of algorithmS.(PSNRs: 45.59 and 39.79 dB, respectively.)

4.4 Implementation

Patch size and stride.: In order to capture each frequency band at the right level of detail, we do not upsample the images $L_l(X)$ and use a small 3×3 patch size (with stride 1). We experimented with larger patch sizes. However, this turned out to be counterproductive for the detail

level we target, as details are blurred in larger patches. They also lead to overfitting and are harder to optimize for in general. We used a stride of 1, and hence patches overlap on the image plane. The overlapping patches are averaged while reconstructing the image.

Detail and color modifications.: We aim to capture intricate details present in highly detailed retouches and a wide range of image processing operators. Based on the observation that various operators can edit materials in the image space using the luminance component [55], we focus on learning changes in luminance while preserving the input chrominance channels. In case desired retouching edits also involve color changes, we learn the mapping independently for luminance and chrominance channels to keep the dimensionality of the patch space low. We achieved satisfactory results with mild color changes (see Figure 11).

Evaluation metrics.: To quantitatively compare our method with state-of-the-art methods, we used PSNR and SSIM metrics. This is only possible if the before-after image pair was processed with a known, reproducible operator (see Section 5.3 for details).

Training details.: We train different mappings with the same structure, defined in Equation 2, for each frequency band of the Laplacian pyramid. Each mapping consists of one MLP block and K number of transformation matrices, which are learned jointly per frequency band from scratch for each before-after pair. The MLP block employed in our experiments consists of three fully-connected layers and non-linearities applied after each layer. The output size of the last layer is the same as the number of transformation matrices.

To normalize the weights, we chose the last activation function to be Softmax, while for the first two layers, we apply Leaky ReLU. Each transformation matrix is randomly initialized with uniform distribution in the range $[0, 1]$. All experiments use the Adam optimizer with a learning rate of 10^{-2}, which exponentially decays with a decay rate of 0.96. We use l_1 loss function in all our experiments.

5 Results

5.1 Ablation Study

The success of our learned mappings relies on two key components: patch-adaptive retouching and transformation blending. We thus conduct experiments to illustrate the significance of these.

Transformation Matrices.: We compared transformation matrices of size 9×9 with scalar values. The method still remained spatially-varying, since we left the MLP the same, and used $K = 256$ scalar weights. We tested both methods on 100 images and computed average PSNR values. We observed that our technique with matrices performed better than scalar values even in simple algorithmic filters, such as Gaussian and Unsharping Masking (around 2 dB and 3 dB higher PSNRs, respectively).

As the complexity of a retouching style depends on multiple factors, such as artists' design choices, user preferences, or the artist toolbox, it is challenging to analyze such effects on retouching examples quantitatively. For simple algorithmic filters, such as a Gaussian filter or unsharp masking, $K = 1$ can sufficiently reproduce the filter. In contrast, more complex algorithms, such as a bilateral filter, require more matrices to capture the algorithmic edits accurately (Figure 4). Since retouching edits combine the effect of multiple operators and are highly non-linear, we empirically chose $K = 256$ for our retouching examples.

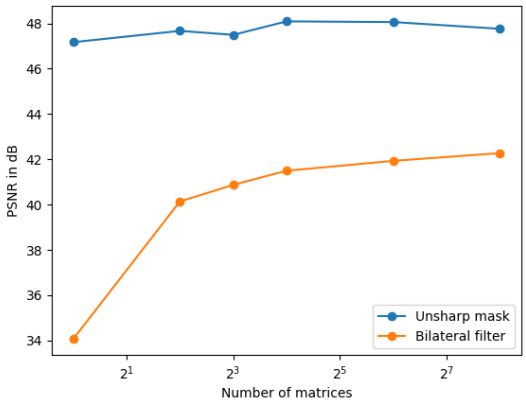

Fig. 4: The higher the complexity of the learned algorithm, the more transformation matrices our technique requires to capture the effects on local regions accurately. While $K = 1$ can be sufficient for our model to capture unsharp masking, it requires more matrices to represent bilateral filtering precisely.

Patch-adaptive Transformation Blending.: We also compared our patch-adaptive mapping to an MLP regressor on the extracted patches. This directly learns the mapping from the decomposition of example before-after images instead of utilizing blended transformations. The MLP regressor follows a similar architecture as our MLP block (Figure 2), with the only difference being the last activation function. We used Leaky ReLU here, since the Softmax function outputs pseudo-probabilities and is unsuitable for regression. Not explicitly handling the spatially-varying structure of the mapping and directly regressing limits the expressiveness of the model. This results in blurry results as shown in Figure 5 because such a model cannot capture edits in intricate details, such as highlights around eyes and hair or brightening of the skin. We also tried increasing the capacity of the MLP regressor but did not observe much improvement in performance.

5.2 Qualitative Results

We tested our technique on a diverse range of before-after pairs, including face images from the **FFHQ** dataset [56]. We focus on human portraits and face retouching in our experiments as they are arguably the most common and prioritized types of photos for retouching. We also illustrate that our technique provides visually pleasing results in different types of images, such as materials, rooms or landscape, and accurately captures image processing filters.

Human faces pose a particular challenge for our technique. However, our model can still capture highly non-linear retouching edits and generalizes well to different types of faces, view directions, and lighting conditions, as illustrated in Figures 1, 3, 6, and 8.

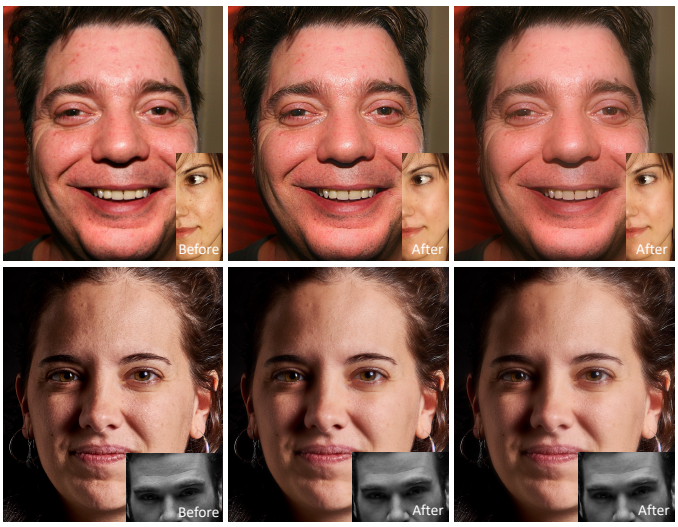

Fig. 5: An MLP regressor cannot capture local edits, resulting in inaccurate retouching edits, such as blurring on the skin or around the eyes. Images in each row have the following order: input, filtered with our map design, filtered with an MLP regressor.

Fig. 6: The reproduced retouching style from the example pair (inset) improves skin texture without affecting fine details, such as eyes and hair, for a visually improved portrait. Moreover, our technique generalizes well to faces with different lighting conditions and accurately reproduces the example retouching style.

The example pairs in Figures 1, 6 and 8 were generated by brushing onto the skin with artist created brushes, eye sharpening (sharpening example in Figure 1), and further brightness/contrast adjustments. These brushes first decompose the skin into a detail and base layer, typically with frequency decomposition, alter the detail layer and blend it with the base layer. They differ in how (1) they decompose the skin into the layers, i.e., what frequencies are in each layer, and (2) they edit and blend each layer with different opacity values. This variation creates retouching nuances, as shown in Figure 8. Our method can still accurately capture such slight differences in styles.

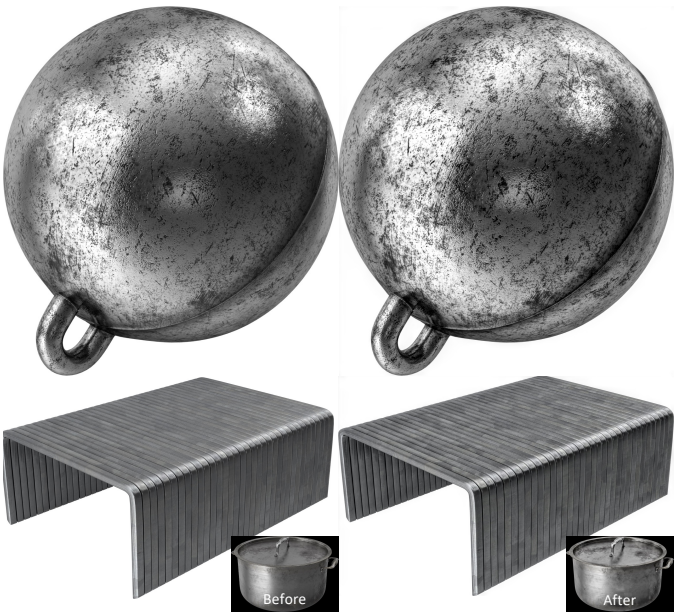

Fig. 7: Material editing results on photos (left), and rendered images (right), based on the before-after pair (inset).

Fig. 8: Our patch-adaptive technique can precisely reproduce nuances of retouching styles. The top and bottom rows show the example pairs and their corresponding retouched photos, respectively.

In all our experiments, intricate details of the desired retouching, such as small-scale texture, eye, facial hair or material details, and global features, such as overall light-

Comparison results (PSNR in dB / SSIM)				
Filter Type	**ASAPNet Generator**	**Deep Edge-aware**	**UNet**	**Ours**
Gaussian	39.42 / 0.985	36.11 / 0.962	40.60 / 0.981	**41.36 / 0.986**
Unsharp Mask	29.79 / 0.891	30.23 / 0.902	32.01 / 0.923	**33.81 / 0.939**
Bilateral Filter	33.79 / 0.938	33.09 / 0.935	34.15 / 0.941	**39.23 / 0.971**
Local Laplacian ($\alpha = 2, \sigma = 0.2$)	30.91 / 0.915	30.29 /0.924	31.65 / 0.931	**33.93 / 0.955**
Local Laplacian ($\alpha = 0.5, \sigma = 0.1$)	31.87 / 0.908	30.91 / 0.892	32.95 / 0.926	**35.79 / 0.932**

TABLE 1: Quantitative performance comparison for the reproduction of various image processing filters. Average PSNR and SSIM values are computed over 182 images of different types of images including faces, landscapes, materials, and rooms.

ing and tone, are accurately reproduced. It is interesting to observe that the *glamour* implied by, e.g., the example retouching in Figure 6 is transferred from the example pair very accurately without causing an artificial look. Zooming into the skin reveals that pores and wrinkles are minimized, and the blemishes and discoloring of the skin are eliminated. At the same time, depending on the retouching edit, details, such as eyes or material texture, are more highlighted or preserved, and delicate features such as hair are preserved well (Figures 6, 7 and 8).

In summary, our technique efficiently edits such intricate details, due to the significantly distinct local statistics of the texture at multiple scales, without affecting overlaying structures thanks to its spatially-varying nature and frequency decomposition.

5.3 Comparison with the state-of-the-art

Although there are various works related to automatic photo enhancement, to the best of our knowledge, none of them works with a single example pair for detail retouching. We thus compare our results with closely related automatic image-to-image translation methods, namely U-Net [57], ASAPNet generator [1], and Deep edge-aware filters [58].

We trained each network from scratch with one *before-after* pair. To train the U-Net architecture, we changed the activation function of its last layer to ReLU and used l_1 loss function with Adam optimizer (same as ours). Similar to our method, ASAPNet is also a spatially-adaptive network. However, it is instead designed to hallucinate new details. Therefore, we similarly trained their generator model to ours with $l1$ loss, removing the discriminator. We observed that bilinear downsampling in their model causes checkerboard artifacts. Hence, we also removed this operator and learned an MLP per pixel, which caused the model to be highly complex with too many parameters.

For a fair comparison with contemporary methods, we trained each network with the same example pair processed by four algorithmic filters: Gaussian, unsharp masking, Bilateral, and local Laplacian filters. As local Laplacian filters can perform a wide range of edge-aware operations, we apply two different versions of the filter, one for smoothing ($\alpha = 2, \sigma = 0.2$) and one for enhancing details ($\alpha = 0.5, \sigma = 0.1$). Each network is trained from scratch with the same example pair resized to 256×256 for the corresponding filter. To prove the generalizability of our technique, we tested the models on different types of images, namely face images (100 images that are randomly sampled from MIT-Adobe FiveK [7]), material images (22 images), room images (30), and landscape images (30). Each type was trained separately with its corresponding example pair. For instance, we trained an example pair of landscape images to test our model on landscape images. We evaluated the models using average PSNR and SSIM values. To generate the ground truths of the input images, we applied the same filter as applied to the before example image to obtain the after image. We trained each model in Y-channel after converting RGB images to their YCbCr versions and evaluated the results for Y-channel images. We duplicated the Y-channel in case the model requires three-channel images.

To obtain the UNet results for each type of images, we ran an additional experiment in which we changed the number of trainable parameters by removing some layers and trained the network from scratch for unsharp masking and bilatering filter. The number of parameters we chose were 0.1M (with a few convolutional layers), 1.8M, 10M and 30M. For material images, we observed that 10M performed the best in terms of PSNR and SSIM values, while for other types of images 30M performed best. We tested the trained models on the images of the corresponding types and computed average PSNR values. Later, we chose the model with the best-performing parameters for each type of image for the quantitative comparison (Table 1).

Overall, our method can outperform all architectures for each considered filter in terms of both PSNR and SSIM values. UNet shows the closest performance to our method, but their network capacity is significantly higher than ours (0.16M). Our technique proves more generalizable in learning different image processing operators from a single example pair with a lower model capacity.

5.4 Limitations and Future Work

A primary limitation of our work is its dependence on local patches at different scales, disregarding their spatial location. Hence, our method is most useful when details are retouched based on local and repeated characteristics of an image. Non-repeating spatially-dependent strong effects, e.g., tattoos or portrait stylizations with spatially varying lighting [41], cannot be handled by the current technique (see Figure 9). We leave this as future work.

Since we rely on a single example image pair, transferring filters applied to arbitrary images [34] is out of the scope of our current work. We require example and input images to have similar semantics for predictable transfer. Extending the technique to more than one pair of example images will require us to have consistently retouched details

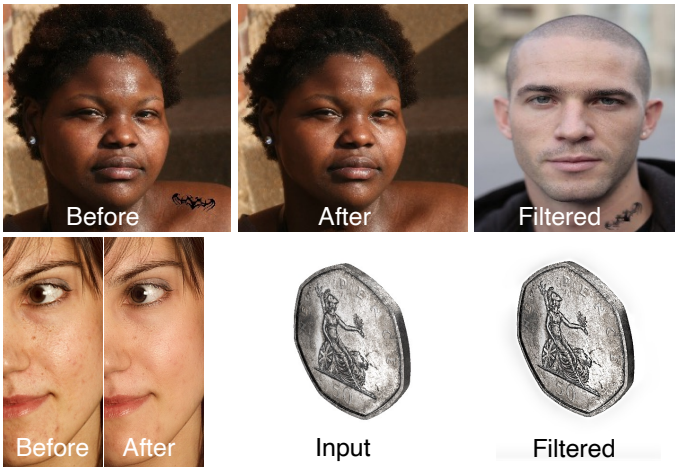

Fig. 9: Our technique cannot accurately handle extreme non-repeating local effects such as tattoos (top), and when example and input images are of very different semantics (bottom).

on all those example images. Finally, we require the example before and after images to be perfectly aligned. This requirement can be alleviated by incorporating an ICP [59]-like approach into the optimization in Section 4.

Although our main focus in this paper is on artist-driven subjective retouching edits, the proposed technique is general. It can be applied to summarize and transfer arbitrary image transformations, significantly where details are modified. We are thus planning to investigate our technique further as a general transfer method for image-to-image translation. The patch-adaptive nature of our mappings makes them amenable to analysis.

6 Conclusions

We presented a neural field based technique for example-based automatic retouching of images. By formulating the transfer problem in the patch space, we showed that blending multiple transformation matrices with patch-adaptive weights can be utilized to learn an accurate and generalizable map. This allowed us to use images of different scenes, people, views, and environmental conditions as the example pair and input. We illustrated the technique's utility on various retouching examples. We believe that our image map representation can be helpful in many other image processing tasks.

References

[1] T. R. Shaham, M. Gharbi, R. Zhang, E. Shechtman, and T. Michaeli, "Spatially-adaptive pixelwise networks for fast image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 882–14 891.

[2] W. Li, K. Zhou, L. Qi, N. Jiang, J. Lu, and J. Jia, "Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 343–20 355, 2020.

[3] S. Moran, P. Marza, S. McDonagh, S. Parisot, and G. Slabaugh, "Deeplpf: Deep local parametric filters for image enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 826–12 835.

[4] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 118:1–118:12, Jul. 2017.

[5] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu, "Automatic photo adjustment using deep neural networks," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 2, pp. 1–15, 2016.

[6] H. S. Faridul, T. Pouli, C. Chamaret, J. Stauder, A. Tremeau, and E. Reinhard, "A Survey of Color Mapping and its Applications," in *Eurographics 2014 - State of the Art Reports*, S. Lefebvre and M. Spagnuolo, Eds. The Eurographics Association, 2014.

[7] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 97–104.

[8] S. Bae, S. Paris, and F. Durand, "Two-scale tone management for photographic look," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 637–645, Jul. 2006.

[9] F. Pitie, A. Kokaram, and R. Dahyot, "N-dimensional probability density function transfer and its application to color transfer," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, Oct. 2005, pp. 1434–1439 Vol. 2.

[10] F. Pitié, A. C. Kokaram, and R. Dahyot, "Automated colour grading using colour distribution transfer," *Comput. Vis. Image Underst.*, vol. 107, no. 1-2, pp. 123–137, Jul. 2007.

[11] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *Computer Graphics and Applications, IEEE*, vol. 21, no. 5, pp. 34–41, Sep. 2001.

[12] K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister, "Multi-scale image harmonization," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 125:1–125:10, Jul. 2010.

[13] J. He, Y. Liu, Y. Qiao, and C. Dong, "Conditional sequential modulation for efficient global image retouching," in *European Conference on Computer Vision*. Springer, 2020, pp. 679–695.

[14] J. Park, J.-Y. Lee, D. Yoo, and I. S. Kweon, "Distort-and-recover: Color enhancement using deep reinforcement learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5928–5936.

[15] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu, "Color harmonization," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 624–630, Jul. 2006.

[16] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2497–2506.

[17] Y. Hu, H. He, C. Xu, B. Wang, and S. Lin, "Exposure: A white-box photo post-processing framework," *ACM Trans. Graph.*, vol. 37, no. 2, pp. 26:1–26:17, May

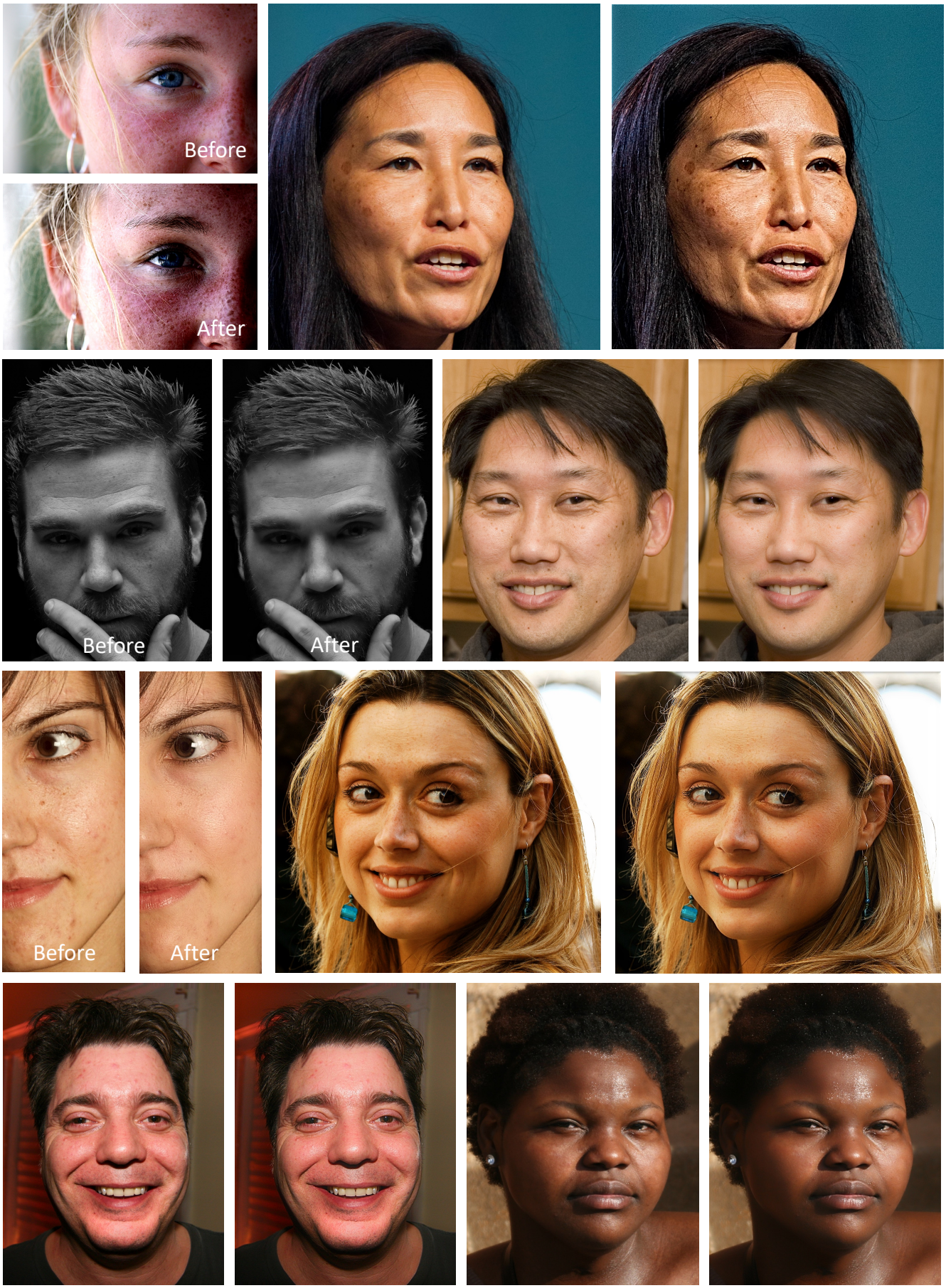

Fig. 10: Retouches reproduced by our algorithm based on single before-after pairs.

Fig. 11: Material editing on photos (left), and rendered images (right), based on the examples provided (insets). Here, chrominance channels are also learned (see Section 4.4).

Fig. 12: Our technique can consistently capture filters applied to examples of various scenes.

2018. [Online]. Available: http://doi.acm.org/10.1145/3181974

[18] H. Kim, S.-M. Choi, C.-S. Kim, and Y. J. Koh, "Representative color transform for image enhancement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4459–4468.

[19] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia, "Underexposed photo enhancement using deep illumination estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6849–6857.

[20] D. Shapira, S. Avidan, and Y. Hel-Or, "Multiple histogram matching," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, Sep. 2013, pp. 2269–2273.

[21] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays, "Transient attributes for high-level understanding and editing of outdoor scenes," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 149:1–149:11, Jul. 2014.

[22] Y. W. Tai, J. Jia, and C. K. Tang, "Soft color segmentation and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1520–1537, Sep. 2007.

[23] F. Berthouzoz, W. Li, M. Dontcheva, and M. Agrawala, "A framework for content-adaptive photo manipulation macros: Application to face, landscape, and global manipulations," *ACM Trans. Graph.*, vol. 30, no. 5, pp. 120:1–120:14, Oct. 2011. [Online]. Available: http://doi.acm.org/10.1145/2019627.2019639

[24] Y.-S. Chen, Y.-C. Wang, M.-H. Kao, and Y.-Y. Chuang, "Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[25] S.-S. Huang, G.-X. Zhang, Y.-K. Lai, J. Kopf, D. Cohen-Or, and S.-M. Hu, "Parametric meta-filter modeling from a single example pair," *Vis. Comput.*, vol. 30, no. 6-8, pp. 673–684, Jun. 2014.

[26] M. Omiya, E. Simo-Serra, S. Iizuka, and H. Ishikawa, "Learning Photo Enhancement by Black-Box Model Optimization Data Generation," in *SIGGRAPH Asia 2018 Technical Briefs*, 2018.

[27] A. Saeedi, M. D. Hoffman, S. J. DiVerdi, A. Ghandeharioun, M. J. Johnson, and R. P. Adams, "Multimodal prediction and personalization of photo edits with deep generative models," in *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018, arXiv:1704.04997 [stat.ML]. [Online]. Available: http://www.cs.princeton.edu/~rpa/pubs/saeedi2018multimodal.pdf

[28] X. An and F. Pellacini, "User-controllable color transfer," *Computer Graphics Forum*, vol. 29, no. 2, pp. 263–271, 2010.

[29] T. Pouli and E. Reinhard, "Progressive color transfer for images of arbitrary dynamic range," *Computers & Graphics*, vol. 35, no. 1, pp. 67 – 80, 2011, extended Papers from Non-Photorealistic Animation and Rendering (NPAR) 2010.

[30] Y.-W. Tai, J. Jia, and C.-K. Tang, "Local color transfer via probabilistic segmentation by expectation-

maximization," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, Jun. 2005, pp. 747–754 vol. 1.

[31] S. J. Hwang, A. Kapoor, and S. B. Kang, "Context-based automatic local image enhancement," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part I*, ser. ECCV 2012. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 569–582.

[32] L. Kaufman, D. Lischinski, and M. Werman, "Content-aware automatic photo enhancement," *Comput. Graph. Forum*, vol. 31, no. 8, pp. 2528–2540, Dec. 2012.

[33] S. Nam and S. J. Kim, "Deep semantics-aware photo adjustment," *CoRR*, vol. abs/1706.08260, 2017.

[34] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu, "Automatic photo adjustment using deep learning," *CoRR*, vol. abs/1412.7725, 2014.

[35] F. Zhu and Y. Yu, "Automatic image stylization using deep fully convolutional networks," *CoRR*, vol. abs/1811.10872, 2018.

[36] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 70:1–70:10, Jul. 2011.

[37] S. Kagarlitsky, Y. Moses, and Y. Hel-Or, "Piecewise-consistent color mappings of images acquired under various conditions," in *Computer Vision, 2009 IEEE 12th International Conference on*, Sep. 2009, pp. 2311–2318.

[38] Y. Shih, S. Paris, F. Durand, and W. T. Freeman, "Data-driven hallucination of different times of day from a single outdoor photo," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 200:1–200:11, Nov. 2013.

[39] Y.-S. Chen, Y.-C. Wang, M.-H. Kao, and Y.-Y. Chuang, "Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6306–6314.

[40] J. Chen, A. Adams, N. Wadhwa, and S. W. Hasinoff, "Bilateral guided upsampling," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–8, 2016.

[41] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand, "Style transfer for headshot portraits," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 148:1–148:14, Jul. 2014.

[42] E. Tseng, F. Yu, Y. Yang, F. Mannan, K. S. Arnaud, D. Nowrouzezahrai, J.-F. Lalonde, and F. Heide, "Hyperparameter optimization in black-box image processing using differentiable proxies." *ACM Trans. Graph.*, vol. 38, no. 4, pp. 27–1, 2019.

[43] K. Yu, Z. Li, Y. Peng, C. C. Loy, and J. Gu, "Reconfigisp: Reconfigurable camera image processing pipeline," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4248–4257.

[44] E. Tseng, Y. Zhang, L. Jebe, X. Zhang, Z. Xia, Y. Fan, F. Heide, and J. Chen, "Neural photo-finishing," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–15, 2022.

[45] T. Ma, M. Guo, Z. Yu, Y. Chen, X. Ren, R. Xi, Y. Li, and X. Zhou, "Retinexgan: Unsupervised low-light enhancement with two-layer convolutional decomposition networks," *IEEE Access*, vol. 9, pp. 56 539–56 550, 2021.

[46] W. Yang, S. Wang, Y. Fang, Y. Wang, and J. Liu, "From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3063–3072.

[47] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang, "Enlightengan: Deep light enhancement without paired supervision," *IEEE Transactions on Image Processing*, vol. 30, pp. 2340–2349, 2021.

[48] S. Liu, X. Ou, R. Qian, W. Wang, and X. Cao, "Makeup like a superstar: Deep Localized Makeup Transfer Network," *ArXiv e-prints*, Apr. 2016.

[49] O. Frigo, N. Sabater, J. Delon, and P. Hellier, "Split and Match: Example-based Adaptive Patch Sampling for Unsupervised Style Transfer," Mar. 2016, peer-reviewed paper accepted to be presented at IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2016, Las Vegas, United States.

[50] J. Lin, Y. Pang, Y. Xia, Z. Chen, and J. Luo, "Tuigan: Learning versatile image-to-image translation with two unpaired images," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 2020, pp. 18–35.

[51] W. Zhang, C. Cao, S. Chen, J. Liu, and X. Tang, "Style transfer via image component analysis," *IEEE Transactions on Multimedia*, vol. 15, no. 7, pp. 1594–1601, Nov. 2013.

[52] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu, "Automatic photo adjustment using deep neural networks," *ACM Trans. Graph.*, vol. 35, no. 2, Feb. 2016. [Online]. Available: https://doi.org/10.1145/2790296

[53] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 261–24 272, 2021.

[54] G. Cazenavette and M. L. De Guevara, "Mixergan: An mlp-based architecture for unpaired image-to-image translation," *arXiv preprint arXiv:2105.14110*, 2021.

[55] I. Boyadzhiev, K. Bala, S. Paris, and E. Adelson, "Band-sifting decomposition for image-based material editing," *ACM Trans. Graph.*, vol. 34, no. 5, pp. 163:1–163:16, Nov. 2015.

[56] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *CoRR*, vol. abs/1812.04948, 2018. [Online]. Available: http://arxiv.org/abs/1812.04948

[57] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[58] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia, "Deep edge-aware filters," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1669–1678.

[59] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.