WISE: Whitebox Image Stylization by Example-based Learning

Winfried Lötzsch[1⋆], Max Reimann[1⋆], Martin Büssemeyer[1], Amir Semmo[2], Jürgen Döllner[1], and Matthias Trapp[1]

[1] Hasso Plattner Institute, University of Potsdam, Germany
[2] Digital Masterpieces GmbH, Germany

Abstract. Image-based artistic rendering can synthesize a variety of expressive styles using algorithmic image filtering. In contrast to deep learning-based methods, these heuristics-based filtering techniques can operate on high-resolution images, are interpretable, and can be parameterized according to various design aspects. However, adapting or extending these techniques to produce new styles is often a tedious and error-prone task that requires expert knowledge. We propose a new paradigm to alleviate this problem: implementing algorithmic image filtering techniques as differentiable operations that can learn parametrizations aligned to certain reference styles. To this end, we present WISE, an example-based image-processing system that can handle a multitude of stylization techniques, such as watercolor, oil or cartoon stylization, within a common framework. By training parameter prediction networks for global and local filter parameterizations, we can simultaneously adapt effects to reference styles and image content, e.g., to enhance facial features. Our method can be optimized in a style-transfer framework or learned in a generative-adversarial setting for image-to-image translation. We demonstrate that jointly training an XDoG filter and a CNN for postprocessing can achieve comparable results to a state-of-the-art GAN-based method. https://github.com/winfried-loetzsch/wise

1 Introduction

Image stylization has become a major part of visual communication, with millions of edited and stylized photos shared every day. At this, a large body of research in Non-photorealistic Rendering (NPR) has been dedicated to imitating hand-drawn artistic styles [31,47]. Traditionally, such *heuristics-based algorithms* [51] for image-based artistic rendering emulate a certain artistic style using a series of specifically developed *algorithmic* image processing operations. Thus, creating new styles is often a time-consuming process that requires the knowledge of domain experts.

Recently, deep learning-based techniques for stylization and image-to-image translation have gained popularity by enabling the learning of stylistic abstractions from example data. In particular, Neural Style Transfer (NST) [22,21] that

⋆ Both authors contributed equally to this work.

(a) Content/Style (b) *WISE* / without edits (c) *WISE* / with local edits

Fig. 1: **Example-based effect adjustment.** *WISE* optimizes effect parameters, e.g., of a watercolor stylization effect, to match stylized outputs to a reference style a. The results b can then be interactively adjusted by tuning the obtained parameters globally and locally for increased artistic control c[3].

transfers the artistic style of a reference image and Generative Adversarial Network (GAN)-based [11,19] methods for fitting style distributions have achieved impressive results and are increasingly used in commercial applications [1].

Classical heuristics-based filters and filter-based image stylization pipelines, such as the eXtended difference-of-Gaussians (XDoG) filter [59], cartoon effect [60], or watercolor effect [2,57], expose a range of parameters to the user that enable fine-grained global and local control over artistic aspects of the stylized output. By contrast, learning-based techniques are commonly limited in their modes of control, i.e., NST [6] only offers control over a general content-style tradeoff. Furthermore, their learned representations are generally not interpretable as a set of design aspects and configurations. Thus, these approaches often do not meet the requirements of interactive image editing tasks that go beyond one-shot global stylizations towards editing with high-level and low-level artistic control [18,14,10]. Additionally, deep network-based methods are often computationally expensive in both training and inference on high image resolutions [6,24,25]. This further limits their applicability in interactive or mobile applications [9] and their capability to simulate fine-grained (pigment-based) local effects and phenomena of artistic media such as watercolor and oil paint.

[3] View examples of editing in our supplemental video: https://youtu.be/wIndN7cr0PE

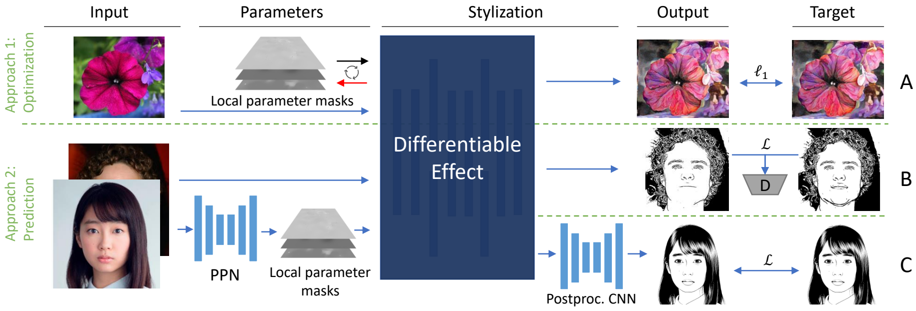


Fig. 2: **Overview of *WISE***. Differentiable effects can be adapted to example data, demonstrated for three use-cases: *parametric style transfer* (**A**) optimizes parameter masks to match a hand-drawn or synthesized stylization target (e.g., from NST as in Fig. 1) which enables style transfer results to remain editable and resolution-independent. *Local parameter prediction* (**B**) trains PPNs to predict parameter masks to adapt the effect to the content (e.g., for facial structure enhancement as shown here). Combined with a postprocessing CNN, local parameter prediction can learn sophisticated *image-to-image translation* tasks such as learning hand-drawn sketch-styles (**C**).

To counterbalance these limitations, this work aims to combine the strengths of *heuristics-based* and *learning-based* image stylization by implementing algorithmic effects as *differentiable* operations that can be trained to learn filter-based parameters aligning to certain reference styles. The goal is to enable (1) the creation of complex, example-based stylizations using lightweight algorithmic approaches that remain interpretable and can operate on very high image resolutions, and enable (2) the editing with artistic control on a fine-granular level according to design aspects. To this end, we present *WISE*, a *whitebox* system for example-based image processing that can handle a multitude of stylization techniques in a common framework. Our system integrates existing algorithmic effects such as XDoG-based stylization [59], cartoon stylization [60], watercolor effects [2,57], and oilpaint effects [50], by creating a library of differentiable image filters that match their shader kernel-based counterparts. We show that the majority of filters (e.g., bilateral filtering) can be transformed into auto-differentiable formulations, while for the remaining filters, gradients can be approximated (e.g., for color quantization). Using our framework, effects can be adapted to reference styles using popular, deep network-based image-to-image translation losses. We train exemplary effects using both NST and GAN-based losses and show qualitatively and quantitatively that the results are comparable to state-of-the-art deep networks while retaining the advantages of filter-based stylization. To summarize, this paper contributes the following:

1. It provides an end-to-end framework for example-based image stylization using differentiable algorithmic filters. Fig. 2 shows an overview of the system.
2. It demonstrates the applicability of style transfer-losses to adapt stylization effects to a reference style (Fig. 1 and Fig. 2 A). The results remain editable and resolution-independent.

3. It shows that both global and local parametrizations of stylization effects can be optimized as well as predicted by a parameter prediction network (PPN). The latter can be trained on content-adaptive tasks (Fig. 2 B).
4. It demonstrates that filters can be trained in combination with CNNs for improved generalization on image-to-image translation tasks. Combining the XDoG effect with a simple post-processing Convolutional Neural Network (CNN) (Fig. 2 C) can achieve comparable results to state-of-the-art GAN-based image stylization for hand-drawn sketch styles, but at much lower system complexity.

2 Related Work

Heuristics-based Stylization. In NPR, image-based artistic rendering deals with emulating traditional artistic styles, using a pipeline of rendering stages [31,47,51]. Commonly, edge detection and content abstraction are important parts of such pipelines. The XDoG filter [59,60] is an extended version of the Difference-of-Gaussians (DoG) band-pass filter, and can be used to create smooth edge stylizations. Furthermore, edge-aware smoothing filters such as bilateral filtering [56] or Kuwahara filtering [30] can abstract image contents, and can be combined with image flow to adapt the results to local image structures [32,34]. These techniques can be found in heuristics-based effects such as cartoon stylization [60], oil-paint abstraction [52] and image watercolorization [2,57], each consisting of a series of rendering stages such as image blending, wobbling, pigment dispersion and wet-in-wet stylization. For a comprehensive taxonomy of techniques, the interested reader is referred to the survey by Kyprianidis *et al.* [31].

These effects are typically parameterized globally, and can be further adjusted within pre-defined parameter ranges, or locally on a per-pixel level using parameter masks [52]. In this work, we implement variants of the XDoG, cartoon filtering, and watercolor pipeline in our framework using auto-differentiable formulations of each rendering stage. At runtime, users choose between one of these different effect pipelines; and results are generally achieved by optimizing the exact chain of filters as introduced in [59,60,2,57,50].

Deep Learning-based Methods. With the advent of deep learning, CNNs for image generation and transformation have led to a range of impressive results.

In NST, first introduced by Gatys *et al.* [6], the stylistic characteristics of a reference image are transferred to a content image by matching deep feature statistics using an optimization process. Fast feed-forward networks have been trained to reproduce a single [22] or even arbitrarily many styles [17,12,41]. Furthermore, there have been efforts to increase controllability of NSTs, e.g., by control over color [7], sub-styles [45] or strokes [20,44], however, the representations are not interpretable. Recently, style transfer has been formulated as a neurally-guided stroke rendering optimization approach [29], that retains interpretability, however, is slow to optimize. We show that our method can obtain comparable results to a state-of-the-art NST [27], while retaining interactive editing control.

GANs, first introduced by Goodfellow *et al*. [11], learn powerful generative networks that model the input distribution. They have been widely used for conditional image generation tasks, with both paired [19] and unpaired [65] training data. In the stylization domain, it has found applications for collection style transfer [3,49], cartoon generation [4,55], and sketch styles [62]. However, domain-specific applications often require sophisticated losses and multiple networks to prevent artifacts [4,63]. We show that our differentiable implementation of the XDoG filter can be trained as a generator network in a GAN framework and can produce comparable sketches to state-of-the-art (CNN-based) GANs [62,63].

Learnable Filters. While the previous end-to-end CNNs deliver impressive results, they are limited in their output resolution. A few recent methods have proposed training fast algorithmic filters to operate efficiently at high resolutions. Getreuer *et al*. [8] introduce learnable approximations of algorithmic image filters, such as of the XDoG filter. At run-time, a linear filter is selected per image pixel according to the local structure tensor; filters can be combined in pipelines for image enhancement [5]. Gharbi *et al*. [9] train a CNN to predict affine transformations for bilateral image enhancement, e.g., to approximate edge-aware image filters or tone adjustments. The transform filters are predicted at a low resolution and then applied in full resolution to the image. "Exposure" framework [16] combines learning linear image filters with reinforcement learning, where an actor-critic model decides which filters to include to achieve a desired photo enhancement effect.

These methods have in common that they learn several simple, linear functions to approximate image processing operations [8,61,9,16,39]. Our framework consists of pipelines of differentiable filters as well. However, in contrast to previous work, we make a variety of heuristics-based stylization operators differentiable and learn to predict their parameterizations. Thereby, sophisticated stylization effects (e.g., those found in stylization applications) can be ported and directly used in our framework.

3 Differentiable Image Filters

Heuristics-based stylization effects consist of pipelines of image filtering operations. To compute gradients for effect input parameters, all image operations within the pipeline are required to be differentiable with respect to their parameters and the image input. Gradients throughout the pipeline can then be obtained by applying the chain rule. Fig. 3 outlines the gradient flow from a loss function to the effect parameters by the cartoon pipeline example, gradients for parameters can be obtained both globally as well as locally using per-pixel parameter masks.

In previous works, individual operations in such pipelines are typically implemented as shader kernels for fast GPU-based processing. To achieve an end-to-end gradient flow, we implement these operations in an auto-grad enabled

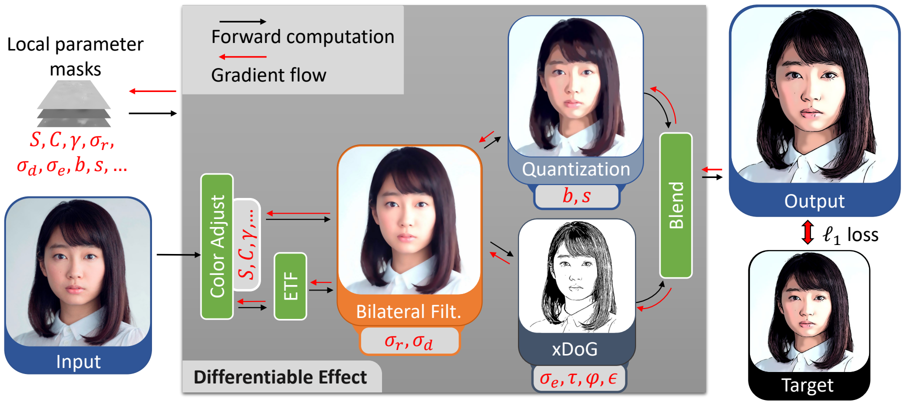


Fig. 3: **Exemplary differentiable effect pipeline.** Shown for the cartoon effect proposed by Winnemöller *et al.* [60]. Gradients are backpropagated from the loss to the filter parameter masks. In the effect, colors are first adjusted based on parameters such as saturation S, contrast C or gamma γ, and then, using an Edge Tangent Flow (ETF) [23], orientation-aligned bilateral filtering [32], and XDoG [59] are computed. Additionally, the image is quantized with respect to the number of bins b and softness s.

framework, the implemented filtering stages are listed in Tab. 1. Point-based and fixed-neighbourhood operations such as color space conversions, structure tensor computation [32], or DoG [59] can generally be converted into differentiable filters by transforming any kernelized function into a sequence of its constituent auto-differentiable transformations. The exception to this are functions which are inherently not differentiable, such as color quantization, and which require the approximation of a numeric gradient. An example for numeric gradient approximation of color quantization is shown in the supplemental material. Structure-adaptive neighborhood operations, such as the orientation-aligned bilateral filter and flow-based Gaussian smoothing filter, iteratively determine sampling locations based on the structure of the content (often oriented along a flow field). To preserve gradients and make use of the in-built fixed-neighborhood functions of auto-grad enabled frameworks, per-pixel iteration is transformed into a grid-sampling operation where neighborhood values are accumulated into a new dimension with size D which represents the expected maximum kernel neighbourhood. A structure-adaptive filter transformation by example of the orientation-aligned bilateral filter is shown in the supplementary material.

Implementation Aspects. We implement differentiable filters in PyTorch and create reference implementations of the same effects using OpenGL shaders. The learnability of each effect parameter is validated in a functional benchmark (shown in supplemental material) by optimizing the differentiable effect to match reference effects with randomized parameters. At inference time, OpenGL shaders can be interchangeably used with the differentiable effect, and can be efficiently executed on high-resolution images using parameters predicted on low-

Table 1: Differentiable filters by type and effect. Filters can be classified by their sampling approach, which is either point-based (PB) or in a fixed neighborhood (FN) or structure-adaptive neighborhood (SN). Some filters are non-differentiable and require numerical gradient (NG) approximations for training.

Filtering operation	Differentiable Filter Type	Car-toon	Water-color	Oil-paint
Anisotropic Kuwahara [34]	SN		✓	
Bilateral [56]	FN		✓	✓
Bump Mapping / Phong Shading [42]	FN			✓
Color Adjustment	PB	✓	✓	✓
Color Quantization [60]	PB,NG	✓		
Flow-based Gaussian Smoothing [32]	SN	✓	✓	✓
Gaps [38]	FN		✓	
Joint Bilateral Upsampling [28]	SN			✓
Flow-based Laplacian of Gaussian [33]	SN			✓
Image Composition [43]	PB		✓	
Orientation-aligned Bilateral [32]	SN	✓		✓
Warping / Wobbling [2]	FN		✓	
Wet-in-Wet [57]	SN		✓	
XDoG [59]	SN	✓	✓	✓

resolution images. At training time, memory usage of differentiable filters can be reduced by controlling their kernel-size (shown in supplemental material).

4 Parameter Prediction

With the introduced differentiable filter pipelines, parameters can be optimized using image-based losses. To generalize to unseen data, we explore Parameter Prediction Networks (PPNs) that are trained to predict global parameters or spatially varying (local) parameters.

4.1 Parameter Prediction Networks

Global Parameter Prediction. We construct a PPN that predicts the effect parameters of a stylized example image, given both the stylized and source image. Thereby, the network is trained to effectively reverse-engineer the stylization effect. During training, gradients are back-propagated through the effect, the parameters, and finally to the PPN. Formally, let I denote the input image, T the target image, $O(\cdot)$ the differentiable effect, $P_G(\cdot)$ the PPN network. The loss for the global PPN is computed using an ℓ_1 image space-based loss as:

$$\mathcal{L}_{\text{global}} = \|O(I, P_G(I))) - T\|_1 \tag{1}$$

Our global PPN architecture consists of a VGG backbone [53] that extracts features of the input and stylized image and computes layer-wise Gram matrices

Table 2: **Global PPN loss functions.** The PPNs are trained with different loss functions on the NPR benchmark [48]. Networks trained in parameter space use reference parameters as the loss signal directly. We measure the Structural Similarity Index Measure (SSIM) and Peak Signal-to-Noise Ratio (PSNR)

Loss domain	SSIM	PSNR	Parameter loss
Parameter space ℓ_1/ℓ_2	0.738/0.737	12.530/12.927	**0.158**/0.162
Image space ℓ_1/ℓ_2	**0.780**/0.764	**13.875**/13.286	0.183/0.190

to encode important style information [6]. The accumulated features are passed to a multi-head module to predict the final global parameters. We found this network architecture to perform superior against other common architecture variants, please refer to the supplementary material for an ablation study and details on the architecture.

Local Parameter Prediction. While the global PPN can predict settings of similar algorithmic effects, real-world, hand-drawn images often vary significantly based on the local content, which cannot be modeled by a global parameterization. Therefore, we construct a PPN to predict local *parameter masks*. We use a U-Net architecture [46] for mask prediction at input resolution, where each output channel represents a parameter. Gradients are back-propagated to the PPN through the differentiable effects for all parameter masks.

For training, a paired data GAN approach is used, where a patch-based Pix2Pix discriminator [19] matches the distribution of patches in the reference image and an additional weighted ℓ_1 image space loss enforces a more strict pixel-wise similarity. Formally, let $D(\cdot)$ denote the discriminator, $\mathcal{L}_{TV}(\cdot)$ the total variation regularizer [22] to enforce smooth parameter masks, and $P_L(\cdot)$ the local PPN which acts as the generator network. The final loss $\mathcal{L}$ for the PPN generator is computed as:

$$\begin{aligned}\mathcal{L} = \mathbb{E}\big[&\alpha \log(1 - D(I, O(I, P_L(I)))) \\ &+ \beta \|O(I, P_L(I)) - T\|_1 + \gamma \mathcal{L}_{TV}(P_L(I))\big]\end{aligned} \tag{2}$$

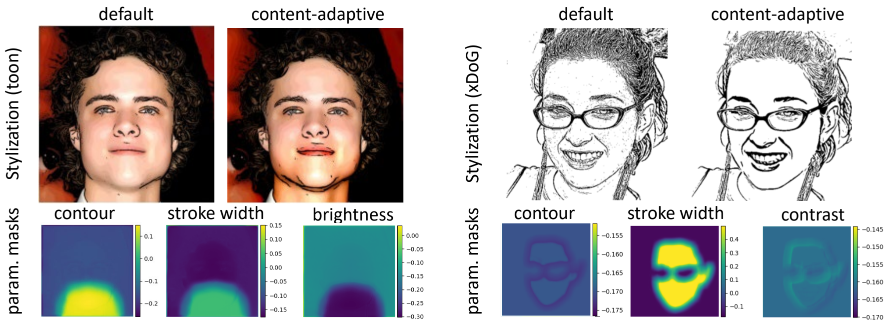
4.2 PPN Experiments

We conduct several experiments to validate our approach for global and local parameter prediction.

Global Parameter Prediction. We compare the loss function and loss space of global PPNs in (Tab. 2). We find that while directly predicting in parameter space (without obtaining gradients from the effect) yields closer parameter values, the highest visual accuracy is achieved using an ℓ_1 image space-based loss. This validates the usefulness of the differentiable effect being part of the training pipeline. Global PPNs can accurately match reference stylizations created by the same effect, as shown in Fig. 4b. Furthermore, they can approximate similar hand-drawn styles, albeit with significant local deviations, as shown in Fig. 4e.

(a) Source (b) Predicted (c) Reference (d) Source (e) Predicted (f) Reference

Fig. 4: **Predictions using the global PPN for XDoG.** The stylized reference c is synthetic (generated using the reference XDoG implementation), while the reference f is hand-drawn, taken from APDrawing [62].



(a) Chin contour enhancement (b) Selective detail reduction

Fig. 5: **Local PPN results.** Networks are trained on CelebAMask-HQ [35] to generate selective enhancements by predicting parameter masks. The default result is created by a global parameter configuration, while the shown predicted parameter masks create the content-adaptive result.

Content-adaptive Effects. Using the previously described approach for local PPN training, we demonstrate its applicability to several common problems that are often present in purely algorithmic image-stylization techniques. We consider three tasks to improve stylization quality: (1) highlighting facial features, for example by increasing contours at low-contrast edges such as the chin (Fig. 5a), (2) selectively reducing details such as small wrinkles in the face (Fig. 5b), and (3) background removal (refer to supplemental material). We use the CelebAMask-HQ dataset [35] for training, which consists of 30,000 high-resolution face images and segmentation masks for all parts of the face. For the above tasks, we each create a synthetic training dataset by stylizing images using a reference effect and adjusting its parameters for certain parts of the face (obtained from dataset annotations) according to the task, e.g., increasing the amount of contours in the chin area. In Fig. 5 trained PPN networks are evaluated by plotting the predicted local parameter masks together with the generated stylizations. It shows that the networks learn to predict parameter masks for the relevant regions accurately solely by observing pixels without additional supervision.

(a) Content image (b) Style image (c) Stylized result (d) Locally edited

Fig. 6: **Parametric style transfer.** Effect parameters are optimized (watercolor effect in top row, oilpaint effect in bottom row) to match the stylistic reference image b. Users can then interactively edit the result c by adjusting resulting parameter masks. In the top row, the saturation parameter mask is adjusted to highlight foreground objects d. In the bottom row, the oilpaint-specific bump scale and flow-smoothing are adjusted in d to make the background appear to be painted wet-in-wet with long brushstrokes.

5 Applications

Using our framework for global and local parameter prediction for differentiable algorithmic pipelines, example-based stylization with closely related references is made possible. To adapt to real-world, more diverse example data, our framework can be integrated with existing stylization paradigms. In the following, we demonstrate our approach for the task of (statistics-based) style transfer reconstruction and GAN-based image-to-image translation based on the APDrawing dataset [62].

5.1 Style Transfer

We investigate the combination of iterative style transfer and algorithmic effects. We use Style Transfer by Relaxed Optimal Transport and Self-Similarity (STROTSS) by Kolkin *et al.* [27] to create stylized references for our effect. We subsequently try to recreate the style transfer result with our algorithmic effects by optimizing parameter masks (Fig. 6). For this, a ℓ_1 loss in image space is again used to match effect output and reference image.

Optimization. We run the style transfer algorithm [27] for 200 steps to create a stylized reference with a resolution of 1024 $\times$ 1024 pixels. The local parameter masks are then optimized using 1000 iterations of Adam [26] and a learning rate of 0.1. The learning rate is decreased by a factor of 0.98 every 5 iterations starting from iteration 50. To avoid the generation of artifacts in parameter masks, we smooth all masks at increasing iterations (10, 25, 50, 100, 250, 500) using a

Table 3: Comparison of our method to STROTSS [27] for style loss $\mathcal{L}_S$, content loss $\mathcal{L}_C$, and the ℓ_1 difference between respective results. We test against in-domain style images and against a set of common (arbitrary domain) NST styles. In each case, results are averaged over 10 styles and NPRB [48] as content[4]

		XDoG	Cartoon	Watercolor		Oilpaint	
	style domain:	bw-drawing	cartoon	watercolor	common	oilpaint	common
$\mathcal{L}_S$	STROTSS	0.340	0.289	0.351	0.39	0.209	0.39
	Our results	0.246	0.406	0.359	0.42	0.384	0.52
$\mathcal{L}_C$	STROTSS	0.099	0.094	0.081	0.036	0.037	0.036
	Our results	0.172	0.148	0.092	0.034	0.023	0.033
	ℓ_1 Difference	0.188	0.136	0.007	0.021	0.036	0.039

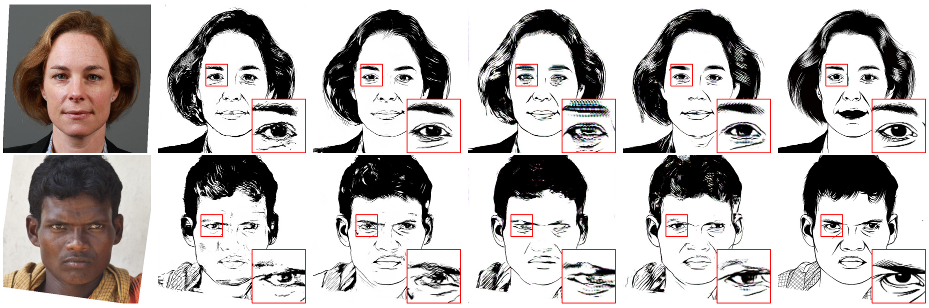
Gaussian filter. Alternatively to image-space matching, parameters can also be directly optimized using NST [6] losses $\mathcal{L}_S + \mathcal{L}_C$, however this often fails to transfer more complex stylistic elements (shown in the supplemental material).

Results. As Tab. 3 shows, parametric style transfer works better for effects that have a high expressivity and are closer to hand-drawn styles, such as the watercolor or oilpaint, compared to more restricted effects such as XDoG or cartoon. After the generation of local parameter masks, the parameters can be refined by the user as shown in Fig. 6d. By optimizing parameters, we obtain an interpretable "whitebox" representation of a style that, in contrast to current pixel-optimizing NSTs [6,22,27], retains controllability according to artistic design aspects. Furthermore, our method is resolution independent, i.e., parameter masks can be optimized at lower resolutions and then scaled up to high resolutions for editing. In Fig. 6d (bottom row) the effect is applied at 4096 $\times$ 4096 pixels, while current style transfers are mostly memory-limited to much lower resolutions. We further compare matching performance of in-domain styles with a set of common NST styles and observe that the ℓ_1 difference (Tab. 3) is only marginally higher for the latter. Thus, highly parameterized effects such as watercolor or oilpaint can emulate any out-of-domain style by per-pixel optimization of parameter combinations reasonably well. However, as this creates highly fragmented parameter masks, a tradeoff between generalizability and interpretability of masks can be made using a weighted total variation-loss.

5.2 GAN-based image-to-image translation with PPNs

For learning a style distribution, i.e., the characteristics of an artistic style over a larger collection of artworks, GAN-based approaches have achieved impressive results [4,58,62]. We investigate training PPNs in a GAN-setting for image-to-image translation. While the global and local PPNs discussed in Sec. 4 can match in-domain styles very well (Sec. 4.2), they cannot produce local image structures that are not synthesizable by their constituent image filters. Artistic reference

[4] Exemplary style images and results in suppl. material.



(a) Input (b) APDraw. -GAN [62] (c) APDraw-ing++ [63] (d) Ours - CNN-only (e) Ours - XDoG+CNN (f) Ground truth

Fig. 7: **Results on APDrawing.** While APDrawing GAN b and APDrawing++ GAN c can produce inconsistent lines, our proposed method e generally produces flow-consistent lines. The differentiable filter in our approach is important for consistent quality, as solely using an image translation CNN [22] often produces local dithering artefacts (upper row) and patchy features (lower row) d.

styles, however, often contain stylistic elements that have not been modeled in the heuristics-based filter - this holds especially true for more simple effects such as xDoG. For reference styles that are stylistically close to such effects (e.g., xDoG), such as line-drawings, we hypothesize that combining our filter pipeline with a lightweight CNN-based post-processing operation and learning them end-to-end can close the domain gap while retaining the positive properties of the filter parametrization and beeing computationally efficient.

Dataset. For our experiment, we select the APDrawing [62] dataset which consists of closely matching photos and their hand-drawn stylistic counterparts. Its hand-drawn images are reasonably similar to the XDoG results, while still containing many stylistic abstractions that cannot be emulated solely by the XDoG. We hypothesize, that re-creating such an effect entails both edge detection and content abstraction, which could be performed by our differential XDoG pipeline combined with a separate convolution network for content abstraction.

APDrawing contains a set of 140 portrait photographs along with paired drawings of these portraits. The GAN-based local PPN approach introduced in Sec. 4 is used to train on this paired dataset, where solely the generator is extended using a CNN $N(\cdot)$ to post-process the XDoG output, i.e., following Eq. (2) our generator now combines PPN, effect and CNN: $N(O(I, P_L(I)))$.

Architecture. We investigate the efficacy of each component in our proposed approach for APDrawing in Tab. 4. Following Yi *et al.* [63], we measure the Fréchet Inception Distance (FID) score [15] and Learned Perceptual Image Patch Similarity (LPIPS) [64] to the test set. We train for 200 epochs and otherwise use the same hyperparameters as Pix2Pix [19]. We observed that using the ResNet-based architecture for image translation introduced by Johnson *et al.* [22] works best for the post-processing CNN. Furthermore integrating the XDoG in the pipeline improves the results vs. a convolutional-only pipeline. Note that this combination of algorithmic effects and CNNs in a training pipeline is only made possible by our introduced approach for end-to-end differentiable filter pipelines and PPNs. Omitting the PPN and using fixed parameters for XDoG significantly degrades the results, which validates the integrated training of filter and CNN. Further, we observe that training with XDoG as a postprocessing instead of as a preprocessing step does not converge. All architecture choices are extensively evaluated in an ablation study, please refer to the supplemental material.

Results. While the CNN alone (without XDoG) already achieves good FID and LPIPS scores, we show in Fig. 7 that it creates major artifacts especially around eyebrows and eyes, which are not detected by those metrics. Compared to the APDrawing GAN approach by Yi *et al.* [62], our model improves the FID score (Tab. 4). The state-of-the-art APDrawing++ [63] improves on these metrics and quantitatively performs better than our model, however qualitatively it can suffer from artifacts in small structures such as the eyes (Fig. 7c) whereas our approach leads to more consistent lines. We note that their approach consists of a sophisticated combination of several losses and task-specific discriminators that require facial landmarks to train multiple local generator networks for facial features such as eyes, nose, and mouth separately. This limits their generalizability to other datasets, while our approach, on the other hand, represents a general setup for image-to-image translation consisting of a globally trained CNN and a simple effect, making it applicable to any paired training data without further annotation requirements.

Table 4: Our results on APDrawing [62]

PPN	XDoG	CNN	FID	LPIPS
✗	✗	U-Net	71.26	0.322
✗	✗	ResNet	62.44	**0.275**
✗[1]	✓	U-Net	75.40	0.329
✗[1]	✓	ResNet	71.56	0.305
✓	✓	U-Net	89.93	0.366
✓	✓	ResNet	**60.55**	0.285
APDrawing GAN			62.14[2]	0.291[2]
APDrawing++			**54.40**	**0.258**[2]
Train vs. Test			49.72	-

[1] a fixed parameter preset is used
[2] results obtained from [62][63]

6 Discussion

Applicability. In the previous sections, we have demonstrated the applicability of differentiable filters to several example-based stylization tasks using four established heuristics-based filter pipelines. Their constituent image filters (Tab. 1) form a common basis of many image-based artistic rendering approaches [31]. We expect that other filtering-based effects, such as pencil-hatching [37], or stippling [54], can be transferred to our framework with relative ease due to their pipeline-based, GPU-optimized formulations. Stroke-based rendering approaches, on the other hand, are typically optimized globally [40] or locally [13], and are thus challenging to transform into differentiable formulations. However, a recent approach by Liu *et al*. [36] has shown that strokes can be predicted in a single feedforward pass of a CNN, which could be regarded as a complementary approach.

Limitations. Our PPN-based approaches make use of a paired data training regime. While paired data can be synthetically generated for content-adaptive effects aiming at solving filter-specific problems, datasets with paired real-world paintings are subject to limited availability. As our training approach follows Pix2Pix GAN [19], future work extending the method to train with unpaired training losses, such as cycle-consistency losses [65], could alleviate this limitation. An inherent limitation of predicting parameters in comparison to directly predicting pixels (as with convolutional GANs), remains the constraint of only being able to produce styles that lie in the manifold of achievable effects of the underlying image filters. While this can be mitigated using a post-processing CNN, this represents a trade-off with respect to interpretability and range of low-level control (we examine this aspect in the supplemental material). On the other hand, our parametric style transfer is able to match arbitrary styles when optimizing highly parameterized effects such as watercolor. Training a PPN with such an effect on a large dataset, e.g., using unpaired training, could similarly already have sufficient representation capability without postprocessing CNNs.

7 Conclusions

In this work, we propose the combination of algorithmic stylization effects and example-based learning by implementing heuristics-based stylization effects as differentiable operations and learning their parametrizations. The results show that both optimization of parameters, e.g., to achieve style transfers, and their global and local prediction, e.g., for content-adaptive effects, are viable approaches for example-based algorithmic stylizations. Our experiments demonstrate that our approach is especially suitable for applications that require fast adaptation to new styles while retaining full artistic control and low computation times for high image resolutions. Furthermore, stylizations beyond the filters' abstraction capabilities are achieved by adding convolutional post-processing. This approach can generate results on-par with state-of-the-art CNN-based methods. For future research, learning the composition of filters as building blocks of a generic algorithmic effect pipeline would allow for seamless integration of user control and example-based stylization without the limitation to a specific stylization technique.

Acknowledgments. We thank Sumit Shekhar and Sebastian Pasewaldt for comments and discussions. This work was partially funded by the BMBF through grants 01IS18092 ("mdViPro") and 01IS19006 ("KI-LAB-ITSE").

References

1. Adobe. Photoshop neural filters overview. https://helpx.adobe.com/photoshop/using/neural-filters.html, 2021. 2
2. Adrien Bousseau, Matt Kaplan, Joëlle Thollot, and François X Sillion. Interactive watercolor rendering with temporal coherence and abstraction. In *Proc. International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 141–149, 2006. 2, 3, 4, 7
3. Xinyuan Chen, Chang Xu, Xiaokang Yang, Li Song, and Dacheng Tao. Gated-GAN: Adversarial gated networks for multi-collection style transfer. *IEEE Transactions on Image Processing*, 28(2):546–560, 2018. 5
4. Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. CartoonGAN: Generative Adversarial Networks for Photo Cartoonization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9465–9474, 2018. 5, 11
5. Ignacio Garcia-Dorado, Pascal Getreuer, Bartlomiej Wronski, and Peyman Milanfar. Image stylisation: from predefined to personalised. *IET Computer Vision*, 14(6):291–303, 2020. 5
6. Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. 2, 4, 8, 11
7. Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling Perceptual Factors in Neural Style Transfer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3730–3738, 2017. 4
8. Pascal Getreuer, Ignacio Garcia-Dorado, John Isidoro, Sungjoon Choi, Frank Ong, and Peyman Milanfar. BLADE: Filter learning for general purpose computational photography. In *Proc. IEEE International Conference on Computational Photography (ICCP)*, pages 1–11, 2018. 5
9. Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep Bilateral Learning for Real-Time Image Enhancement. *ACM Transactions on Graphics*, 36(4):1–12, 2017. 2, 5
10. Amy A. Gooch, Jeremy Long, Li Ji, Anthony Estey, and Bruce S. Gooch. Viewing Progress in Non-photorealistic Rendering through Heinlein's Lens. In *Proc. International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 165–171, 2010. 2
11. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2, 5
12. Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. Arbitrary Style Transfer with Deep Feature Reshuffle. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8222–8231, 2018. 4
13. Aaron Hertzmann. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In *Proc. of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 453–460, 1998. 14
14. Aaron Hertzmann. Computers do not make art, people do. *Communications of the ACM*, 63(5):45–48, 2020. 2
15. Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 12

16. Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. Exposure: A White-Box Photo Post-Processing Framework. *ACM Transactions on Graphics*, 37(2):1–17, 2018. 5
17. Xun Huang and Serge Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1501–1510, 2017. 4
18. Tobias Isenberg. Interactive NPAR: What Type of Tools Should We Create? In *Proc. Non-Photorealistic Animation and Rendering (NPAR)*, 2016. 2
19. Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017. 2, 5, 8, 12, 14
20. Yongcheng Jing, Yang Liu, Yezhou Yang, Zunlei Feng, Yizhou Yu, Dacheng Tao, and Mingli Song. Stroke Controllable Fast Style Transfer with Adaptive Receptive Fields. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 4
21. Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural Style Transfer: A Review. *IEEE Transactions on Visualization and Computer Graphics*, 26(11):3365–3385, 2019. 1
22. Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Proc. European Conference on Computer Vision (ECCV)*, pages 694–711, 2016. 1, 4, 8, 11, 12
23. Henry Kang, Seungyong Lee, and Charles K. Chui. Flow-Based Image Abstraction. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):62–76, 2009. 6
24. Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 2
25. Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. 2
26. Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2015. 10
27. Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style Transfer by Relaxed Optimal Transport and Self-Similarity. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10051–10060, 2019. 4, 10, 11
28. Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. Joint Bilateral Upsampling. *ACM Transactions on Graphics (ToG)*, 26(3):96–es, 2007. 7
29. Dmytro Kotovenko, Matthias Wright, Arthur Heimbrecht, and Björn Ommer. Rethinking Style Transfer: From Pixels to Parameterized Brushstrokes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12196–12205, 2021. 4
30. Michiyoshi Kuwahara, Kozaburo Hachimura, S Eiho, and Masato Kinoshita. Processing of RI-Angiocardiographic Images. In *Digital Processing of Biomedical Images*, pages 187–202. 1976. 4
31. Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg. State of the "Art": A Taxonomy of Artistic Stylization Techniques for Images and Video. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):866–885, 2012. 1, 4, 13
32. Jan Eric Kyprianidis and Jürgen Döllner. Image Abstraction by Structure Adaptive Filtering. In *Proc. EG UK Theory and Practice of Computer Graphics (TPCG)*, pages 51–58, 2008. 4, 6, 7

33. Jan Eric Kyprianidis and Henry Kang. Image and Video Abstraction by Coherence-Enhancing Filtering. In *Computer Graphics Forum*, volume 30, pages 593–602. Wiley Online Library, 2011. 7
34. Jan Eric Kyprianidis, Henry Kang, and Jürgen Döllner. Image and Video Abstraction by Anisotropic Kuwahara Filtering. *Computer Graphics Forum*, 28(7):1955–1963, 2009. 4, 7
35. Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5549–5558, 2020. 9
36. Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. Paint Transformer: Feed Forward Neural Painting with Stroke Prediction. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 6598–6607, 2021. 14
37. Cewu Lu, Li Xu, and Jiaya Jia. Combining Sketch and Tone for Pencil Drawing Production. In *Proc. International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, 2012. 13
38. Santiago E Montesdeoca, Hock Soon Seah, Pierre Bénard, Romain Vergne, Joëlle Thollot, Hans-Martin Rall, and Davide Benvenuti. Edge- and substrate-based effects for watercolor stylization. In *Proc. International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 1–10, 2017. 7
39. Sean Moran, Pierre Marza, Steven McDonagh, Sarah Parisot, and Gregory Slabaugh. DeepLPF: Deep Local Parametric Filters for Image Enhancement. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12826–12835, 2020. 5
40. Reiichiro Nakano. Neural Painters: A learned differentiable constraint for generating brushstroke paintings. *arXiv preprint arXiv:1904.08410*, 2019. 14
41. Dae Young Park and Kwang Hee Lee. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5880–5888, 2019. 4
42. Bui Tuong Phong. Illumination for Computer Generated Pictures. *Communications of the ACM*, 18(6):311–317, 1975. 7
43. Thomas Porter and Tom Duff. Compositing Digital Images. *SIGGRAPH Comput. Graph.*, 18(3):253–259, 1984. 7
44. Max Reimann, Benito Buchheim, Amir Semmo, Jürgen Döllner, and Matthias Trapp. Controlling strokes in fast neural style transfer using content transforms. *The Visual Computer*, pages 1–15, 2022. 4
45. Max Reimann, Mandy Klingbeil, Sebastian Pasewaldt, Amir Semmo, Matthias Trapp, and Jürgen Döllner. Locally controllable neural style transfer on mobile devices. *The Visual Computer*, 35(11):1531–1547, 2019. 4
46. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proc. International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241, 2015. 8
47. Paul Rosin and John Collomosse. *Image and video-based artistic stylisation*, volume 42. 2012. 1, 4
48. Paul L. Rosin, Yu-Kun Lai, David Mould, Ran Yi, Itamar Berger, Lars Doyle, Seungyong Lee, Chuan Li, Yong-Jin Liu, Amir Semmo, Ariel Shamir, Minjung Son, and Holger Winnemöller. NPRportrait 1.0: A three-level benchmark for non-photorealistic rendering of portraits . *Computational Visual Media*, 8(3):445–465, 2022. 8, 11

49. Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Björn Ommer. A Style-Aware Content Loss for Real-time HD Style Transfer. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 5
50. Amir Semmo, Tobias Dürschmid, Matthias Trapp, Mandy Klingbeil, Jürgen Döllner, and Sebastian Pasewaldt. Interactive Image Filtering with Multiple Levels-of-Control on Mobile Devices. Proc. SIGGRAPH ASIA Mobile Graphics and Interactive Applications (MGIA), pages 2:1–2:8, 2016. 3, 4
51. Amir Semmo, Tobias Isenberg, and Jürgen Döllner. Neural Style Transfer: A Paradigm Shift for Image-based Artistic Rendering? In *Proc. International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 1–13, 2017. 1, 4
52. Amir Semmo, Daniel Limberger, Jan Eric Kyprianidis, and Jürgen Döllner. Image Stylization by Interactive Oil Paint Filtering. *Computers & Graphics*, 55:157–171, 2016. 4
53. Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proc. International Conference on Learning Representations (ICLR)*, 2015. 7
54. Minjung Son, Yunjin Lee, Henry Kang, and Seungyong Lee. Structure grid for directional stippling. *Graphical Models*, 73(3):74–87, 2011. 13
55. Hao Su, Jianwei Niu, Xuefeng Liu, Qingfeng Li, Jiahe Cui, and Ji Wan. Manga-GAN: Unpaired Photo-to-Manga Translation Based on The Methodology of Manga Drawing. In *Proc. AAAI Conference on Artificial Intelligence*, pages 2611–2619, 2021. 5
56. Carlo Tomasi and Roberto Manduchi. Bilateral Filtering for Gray and Color Images. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 839–846, 1998. 4, 7
57. Miaoyi Wang, Bin Wang, Yun Fei, Kanglai Qian, Wenping Wang, Jiating Chen, and Jun-Hai Yong. Towards Photo Watercolorization with Artistic Verisimilitude. *IEEE Transactions on Visualization and Computer Graphics*, 20(10):1451–1460, 2014. 2, 3, 4, 7
58. Xinrui Wang and Jinze Yu. Learning to Cartoonize Using White-box Cartoon Representations. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8090–8099, 2020. 11
59. Holger Winnemöller. XDoG: Advanced Image Stylization with eXtended Difference-of-Gaussians. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 147–156, 2011. 2, 3, 4, 6, 7
60. Holger Winnemöller, Sven C Olsen, and Bruce Gooch. Real-Time Video Abstraction. *ACM Transactions On Graphics (TOG)*, 25(3):1221–1226, 2006. 2, 3, 4, 6, 7
61. Zhicheng Yan, Hao Zhang, Baoyuan Wang, Sylvain Paris, and Yizhou Yu. Automatic Photo Adjustment Using Deep Neural Networks. *ACM Transactions on Graphics (TOG)*, 35(2):1–15, 2016. 5
62. Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. APDrawingGAN: Generating Artistic Portrait Drawings from Face Photos with Hierarchical GANs. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10743–10752, 2019. 5, 9, 10, 11, 12, 13
63. Ran Yi, Mengfei Xia, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Line Drawings for Face Portraits from Photos using Global and Local Structure based GANs. In *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 5, 12, 13

64. Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 12
65. Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *Proc. IEEE international Conference on Computer Vision (ICCV)*, pages 2223–2232, 2017. 5, 14