Time-of-Day Neural Style Transfer for Architectural Photographs

Yingshu Chen, Tuan-Anh Vu, Ka-Chun Shum, Binh-Son Hua, and Sai-Kit Yeung

Abstract—Architectural photography is a genre of photography that focuses on capturing a building or structure in the foreground with dramatic lighting in the background. Inspired by recent successes in image-to-image translation methods, we aim to perform style transfer for architectural photographs. However, the special composition in architectural photography poses great challenges for style transfer in this type of photographs. Existing neural style transfer methods treat the architectural images as a single entity, which would generate mismatched chrominance and destroy geometric features of the original architecture, yielding unrealistic lighting, wrong color rendition, and visual artifacts such as ghosting, appearance distortion, or color mismatching. In this paper, we specialize a neural style transfer method for architectural photography. Our method addresses the composition of the foreground and background in an architectural photograph in a two-branch neural network that separately considers the style transfer of the foreground and the background, respectively. Our method comprises a segmentation module, a learning-based image-to-image translation module, and an image blending optimization module. We trained our image-to-image translation neural network with a new dataset of unconstrained outdoor architectural photographs captured at different magic times of a day, utilizing additional semantic information for better chrominance matching and geometry preservation. Our experiments show that our method can produce photorealistic lighting and color rendition on both the foreground and background, and outperforms general image-to-image translation and arbitrary style transfer baselines quantitatively and qualitatively. Our code and data are available at https://github.com/hkust-vgd/architectural_style_transfer.

Index Terms—Computational Photography, Image-to-Image Translations, Style Transfer

1 Introduction

Artificial intelligence has been revolutionizing photography with high-fidelity image synthesis using generative modeling techniques, which has led to a wide range of new applications for image manipulation and editing. In this direction, style transfer is a special visual task that aims at generating aesthetically pleasant images in the style of a reference image. It has been well known that style transfer techniques can successfully extract artistic styles from famous paintings and seamlessly blend the styles into real photographs, generating novel images.

Architectural photography is a form of photography that captures subjects such as buildings into pictures with into visually pleasing pictures. In general, an architectural photograph often has a building in the foreground and a sky background captured at a specific time of a day that exhibits dramatic lighting. Taking architectural photographs has been so far a challenging task, requiring both skills and aesthetic senses of a professional photographer.

The availability of neural networks has led to a new possibility: let a machine learn to generate realistic architectural photographs with new styles. In this paper, we realize time-of-day style transfer for architectural photographs using neural networks. Specifically, we consider styles of outdoor architecture photographs with dramatic lighting at magic times of a day. Magic times of a day refer to golden hours at sunset when the sun is falling close to the horizon, blue hours at twilight when the sun is below the horizon and nighttime after sunset and before sunrise without any sunlight. We call this problem the *architectural style transfer* problem.

Fig. 1: Architectural style transfer at magic times of a day.

Given the special composition of the foreground and background in architectural photographs, performing style transfer among these photographs is a challenging task. In many cases, the style of the foreground and the background

- *Yingshu Chen, Tuan-Anh Vu, Ka-Chun Shum, and Sai-Kit Yeung are with the Hong Kong University of Science and Technology.*
- *Binh-Son Hua is with VinAI Research, Vietnam.*

cannot be represented by a single latent space, e.g., the lighting on the building and the color and texture of the sky are very different. This causes existing image-to-image translation methods [1], [2], [3], [4], [5], [6] to fail because these methods treat the input image as a single entity; they are only good for global style transfer but fail to preserve image details or tend to exhibit visual artifacts when being used for architectural photographs. In the literature, there exist a few methods that consider style transfer for outdoor images, but they merely address natural landscapes [7], [8] or time-lapse videos [9], [10], which do not fit well to the composition in architectural photographs.

To overcome such challenges, we propose a new style transfer framework for architectural photography. We semantically disentangle the image content in an architectural photograph so that the style transfer can be done on the foreground (e.g., buildings) and the background (e.g., sky) respectively. For the foreground, we keep the geometry of static objects intact and transfer sufficient and appropriate style details. For the background, we transfer overall style including color and texture. To realize this approach, we devise a two-branch neural network that handles the transfer for the foreground and background, respectively. We devise a set of loss functions for preserving the image details in the transfer. To support training and testing, we also collect a new dataset of architectural photographs captured at different times of day. Our method is validated with qualitative and quantitative comparisons with state-of-the-art image-to-image translation and arbitrary neural style transfer methods, which demonstrates the robustness of our method.

In summary, our main contributions are:

- A new problem setting for style transfer: photorealistic style transfer for architectural photographs of different times of day;
- An image-to-image translation framework with disentanglement representation that separately considers style transfer for image foreground and background respectively, accompanied with simple but effective geometry losses designed for image content preservation.
- A new dataset of architectural photographs and an extensive benchmark for architectural style transfer.

2 Related Work

Our architectural style transfer problem is related to a large body of works in modern computer vision, namely style transfer, image-to-image translation, image relighting, and timelapse translation methods, which we discuss below.

2.1 Style Transfer

Image style transfer has a long history in computer vision. Early techniques take user input as guidance, e.g., colorization methods [11], [12] take user strokes to provide color mapping or segmentation cues. By using a pair of a reference (target) style image and an input (source) image, automatic methods aim at transferring the global style of the target image to the source image while preserving details of the source. This can be formulated as parametric algorithms such as color tone transfer [13], data-driven search for style transfer mapping [14], multi-level style feature transforms [15], progressive stylization [16], image-optimization-based approaches [17], [18], [19], [20]. However, many of these methods usually require an iterative process or slow post-processing that is computationally expensive.

Prior works also explore style transfer using visual cues from semantic segmentation [10], [21], [22], [23], improving the photorealism based on the matting Laplacian [18], [24], screened Poisson equation [25], or photorealistic smoothing [15] as a post-processing step. Our method belongs to the family of techniques in which segmentation information is explicitly used to address the style transfer problem.

With deep learning, neural networks can be trained and perform instant style or color transfer at inference time [26], [27], [28], [29], [30], [31], [32], [33]. Extended from the seminal work of arbitrary global style transfer with adaptive instance normalization (AdaIN) [28], recent networks such as SANet [31] and AdaAttN [33] further introduce attention-aware mechanism to cover image local features and preserve better content appearance. Our network utilizes AdaIN [28] for style fusion; we empirically found AdaIN more robust for style transfer and attention-based networks like SANet or AdaAttN does not preserve photorealistic geometry.

2.2 Image-to-image Translation

Image-to-image translation is a task that aims to transfer an image from a source to a target domain, preserving its original content while having the characteristics of the target domain. Compared to traditional style transfer, this class of methods learns to perform the mapping from data from both domains. Image-to-image translation is, therefore, more general and can also be used for solving image colorization or style transfer problems. Typical methods of image-to-image translation include supervised methods such as pix2pix [1], unsupervised methods such as CycleGAN [2] and UNIT [34], multi-modal translation methods such as MUNIT [4], DIRT [3], DSMAP [6], and more generic translators such as StarGAN v2 [35] and FUNIT [5] for handling multiple image classes. These methods are effective at translating style elements such as color, texture at a speedy inference time.

An important factor to perform effective translation is to disentangle the content and style representation. In an encoder-decoder framework, recent methods [3], [4], [5], [6], [10] inject both content code and style code generated from content encoder and style encoder respectively into the content generator to get the final transferred images with input content appearance but target style. Park *et al.* [36] designed an autoencoder to swap textures between two images by disentangling structure and texture, which changes input content geometry and does not suit our task.

In this work, we build our method upon the framework of image-to-image translation with disentanglement of content and style representation. Moreover, the separation of luminance and chrominance of images in image-to-image translation is widely used in colorization [37] or color style transfer [38] tasks, and is beneficial to content integrity with luminance information intact. Therefore, we design losses that preserve high spatial frequency details of the image by considering the gradients of the luminance channel, so that the translation network can retain geometric details with high fidelity.

2.3 Image Relighting and Timelapse Translation

Our architectural style transfer problem is also relevant to some traditional image relighting methods. Particularly, Shih *et al.* [14] proposed an approach with a global to local mapping and local affine transformation for time-lapse appearance style transfer from video to photographs. It achieves an elegant color appearance change of an outdoor scene via rich appearance information from abundant time-lapse videos. However, this method requires a matched time-lapse video as a transfer reference and thus does not support arbitrary style transfer. More prior works such as Liu *et al.* [39], Yu *et al.* [40], Laffont *et al.* [41] and Duchêne *et al.* [42] are tailored for image decomposition to achieve outdoor scene relighting. They require multi-view images [40], [41], [42] or illumination-varying images [39] to achieve intrinsic decomposition. These methods can relight scenes with general lighting conditions such as sunlight and shadow, but cannot transfer semantic color information. Later, Laffont *et al.* [43] further explored more outdoor scene attribute transients and proposed a high-level example-based appearance transfer system.

In recent years, timelapse translation for images or videos can be done with deep learning on datasets from videos, landscapes, and street photos. Karacan *et al.* [7] realize style transfer with specific attributes of natural scenes via generated style references produced by a deep scene generator. With different specific training data such as landscape photos [8], photos in different time of sky [44], [45], Google street views [46], [47], timelapse transition can demonstrate in diverse ways, e.g., color and texture [8], sunlight estimation [44], [45], lighting and shadow conditions in street views [39]. Animation video synthesis from a single outdoor photo is achieved by Endo *et al.* [48] predicting motion and appearance via convolutional neural network (CNN) models. Some latest works address time-lapse video synthesis, for examples, Nam *et al.* [9] and Cheng *et al.* [10] trained end-to-end supervised models from time-lapse videos to synthesize time-lapse video from a single image.

3 Background and Motivation

The primary emphasis of an architectural photograph is the harmonious rendition of a building structure in the foreground on an aesthetically pleasing background. Given an input architectural image (source) and a style architectural image (target), we aim to transfer the characteristics of the target to the source while preserving the content and structure of the source image. The transferred characteristics include the color and texture rendition in both the foreground and the background of the target. An example of the source, target, and the transferred result can be seen in Figure 1. We call this problem the architectural style transfer problem.

The foreground and background in architectural photographs pose unique challenges in this domain-specific style transfer problem. At first, we are tempted to directly use recent methods for generic neural style transfer and image-to-image translation to solve this problem, but we soon realized that these methods barely work up to our expectations, and there remain the following challenges. Existing image-to-image translation methods treat the image as a single entity without knowing the foreground and background. The lack of this inductive bias in architectural photography makes these methods perform not as effectively. We found that while these methods are efficient in transferring the global style, they tend to operate poorly in preserving geometry details of the foreground, having visual artifacts, or producing unfaithful results with mismatching color (e.g., Fig. 5 (c)). It is, therefore, necessary to derive a specific method for architectural style transfer that takes all such issues into account.

4 Proposed Method

4.1 System Overview

We design a specific method for image-to-image translations of architectural photographs. We take the inductive bias from architectural photography into account by having the neural networks learn to transfer the foreground and background styles, respectively. This learning bias, while seemingly trivial, provides a strong constraint for architectural style transfer as it allows us to define the semantic correspondences between the input and the style image, effectively model the characteristics of architectural photographs because the style of the foreground and background can be vastly different and diverse.

Our framework has three main modules: image segmentation, image translation, and image blending. Given an input image and a style image, we first segment their background and foreground for data preprocessing, and then train the image translations separately for the foreground and background. To train the neural networks, we propose a new geometry loss to preserve structural details that are vital in architectural images. Particularly, we aim to preserve the geometry contour (*Image Gradient loss*) and spatial luminance distribution (*Spatial Luminance KL-divergence loss*) for image foreground translation. With the illumination density constraint, empirically, the appearance information is well retained.

To produce the final result, we blend the predicted foreground and background using the original high-fidelity input source as a geometric constraint. We intentionally let this step be fixed and not trainable as we found that such a post-processing step can already provide satisfactory results. An overview of our framework is presented in Figure 2.

4.2 Semantic Correspondences

We determine the foreground and background of the architectural images by a semantic segmentation model. Particularly, we segment the input and the style images into the background (i.e., sky) and foreground (other elements such as buildings, trees, rivers, etc.). This allows us to build high-level semantic correspondences between the input and the style image to perform the transfer on each correspondence, respectively. Our semantic segmentation is built upon an encoder-decoder model as follows. We use ResNet-50-dilated [49] for the encoder, and the pyramid pooling module with loss optimization from PSPNet [50] for the decoder. We use the official pretrained model trained on the ADE20K dataset [51].

Given the pretrained semantic segmentation model, we preprocess our training images by applying the model to

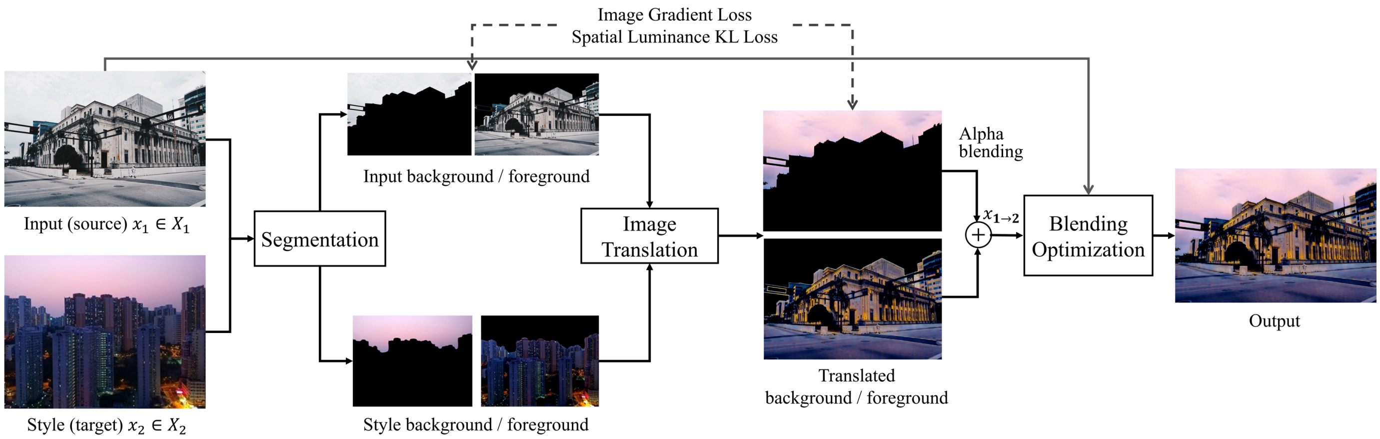


Fig. 2: Our architectural style transfer framework has three main modules: segmentation, image translation and blending optimization. Only style transfer in one direction ($X_1 \to X_2$) is illustrated. x_1 and x_2 are images from two domains (X_1 and X_2), x_1 as high-fidelity source input, x_2 as target style reference. After segmentation, foreground and background images are fed into the translation network respectively. The translated image $x_{1\to2}$ (or c_{style}) and the input source x_1 can be further post-processed by the blending optimization module. More details are in Section 4.3, 4.4, and 4.5.

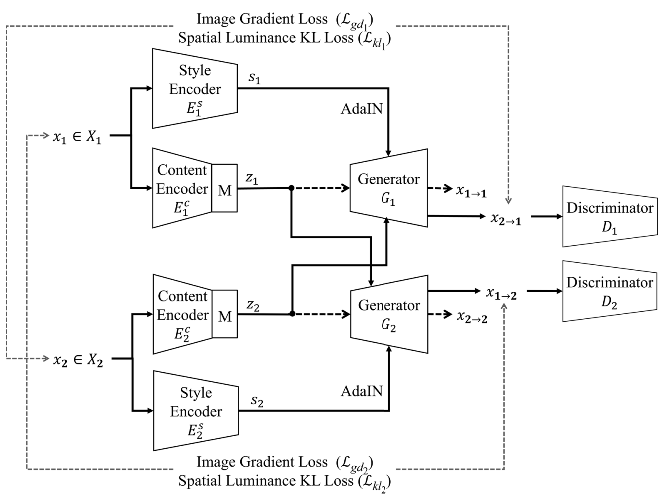


Fig. 3: The common image-to-image translation network architecture. x_1 and x_2 are images from two domains (X_1, X_2). $x_{1\to2}$ and $x_{2\to1}$ are style transferred outputs between two domains. $x_{1\to1}$ and $x_{2\to2}$ are reconstruction outputs. We use the same architecture for foreground and background translation.

predict the masks that indicate the foreground and background. We store all the masks and use them to separate the foreground and background for our training. At inference, we have the flexibility to allow users to use the pretrained segmentation model to automatically segment the inputs or manually provide their masks. We empirically observed that our training tolerates segmentation imperfection at a certain extent. We discuss some failure case due to imperfect segmentation in Sec. 5.6 and Fig. 11.

4.3 Neural Network Architecture

Given the source domain X_1, the target domain X_2, and a pair of image $x_1 \in X_1, x_2 \in X_2$, our goal is to develop an image-to-image translation network to transform x_1 to the target domain, i.e., the translated image should resemble in style of x_2 while preserving the details in x_1. Our method is based on the disentanglement of style and content under the same separation domains assumption as [3], [4], [6]. It assumes that each image belongs to a shared domain-invariant content latent space $\mathcal{D}^c$ but a different domain-specific style latent space $\mathcal{D}^s$. An overview of our method is shown in Fig. 3.

Particularly, to transfer style from source domain X_1 to target domain X_2, we employ two encoders E_1^c, E_2^s to embed content and style features respectively into latent spaces $\mathcal{D}^c$ and $\mathcal{D}^s$, and a decoder G_2 to generate results with specified content and style. We further employ a mapping module M followed the design of DSMAP [6] to map the content latent space $\mathcal{D}^c$, which is domain invariant, to become domain-specific for better generation in the target domain.

Let $c_1 = E_1^c(x_1) \in \mathcal{D}^c$ be the domain-invariant content code of image x_1, and $z_1 = M(c_1)$ be the domain-specific content code after mapping. The style code can be extracted as $s_2 = E_2^s(x_2) \in \mathcal{D}^s$. In the generator G_2, we apply adaptive instance normalization (i.e., AdaIN [28]) for style transfer. Finally, the latent content and style codes are fed into the generator to obtain a new image $x_{1\to2} = G_2(z_1, s_2) \in X_2$ that has the content of X_1 and the style of X_2. Likewise, the translated image $x_{2\to1}$ from domain X_2 to X_1 can be obtained via $x_{2\to1} = G_1(z_2, s_1)$.

Accordingly, we have two discriminators D_1 and D_2 to discriminate the real images and generated images in each domain X_1 and X_2, respectively. Similar to Huang *et al.* [4], we employ the multi-scale discriminator architecture. The details of our network architecture are illustrated in the supplementary.

We separately apply the same network architecture to perform style transfer for both the foreground and the background, respectively. We empirically found that such a separation is necessary and robust because the styles of the foreground and background are vastly different. Both network branches for the foreground and the background share the same set of training objectives. We leave the investigation of joint training both branches as future work.

4.4 Training Objectives

We design a set of training objectives that can work for both the foreground and the background. We train both translation networks using unpaired data with the reconstruction loss, cycle-consistency loss, and adversarial loss. Particularly, to preserve high-frequency geometry information of foreground, we presume the luminance of an image contains both geometry and illumination information, and devise the geometry losses (i.e., *Image Gradient loss and Spatial Luminance KL-divergence loss*) to guide the generator to reproduce high-frequency content of the source.

Here we detail the losses by assuming the transfer direction to be from domain X_1 to domain X_2.

Image Gradient Loss. Image gradient can well represent edges of objects in an image. Preserving the good gradient attribute of an image to some extent guarantees the photorealism and fidelity [25]. Our image gradient loss for $x_{1\to 2}$ is:

$$\mathcal{L}_{gd_1} = \mathbb{E}_{x_1,x_2}\left[\|\nabla(Y(x_{1\to 2})) - \nabla(Y(x_1))\|_1\right], \tag{1}$$

where $\mathbb{E}[.]$ is the expectation operator, $\nabla(\cdot)$ is the image gradient, $Y(x)$ is luminance of image x. Following ITU-R BT.601 conversion standard [52] to get Y channel values from RGB channels, we define the luminance by

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B. \tag{2}$$

Here R, G and B are image values in RGB channels.

Spatial Luminance KL Divergence Loss. The relative entropy or the so-called Kullback-Leibler divergence (KL divergence), is a useful distance measure for continuous distributions. To constrain the geometry luminance distribution to the source input image, we apply KL divergence loss on the luminance channel of output $x_{1\to 2}$ and input x_1:

$$\mathcal{L}_{kl_1} = \mathbb{E}_{x_1,x_2}\left[\mathrm{KL}(Y(x_{1\to 2}) \parallel Y(x_1))\right] \tag{3}$$

where each value of luminance $Y(x)$ is normalized to $[0,1]$ when calculating the loss; $\mathrm{KL}(p \parallel q) = \sum_x p(x)\log[p(x)/q(x)]$ measures the KL divergence between a distribution p and a reference distribution q. This loss constrains the model to generate image geometry in illumination distribution of input domain (X_1).

We call the total of image gradient loss and the KL divergence loss the *geometry loss* as it can improve the geometry quality while transferring the illumination faithfully.

Reconstruction Loss. We utilize the same concept of bidirectional reconstruction loss in [4] for image reconstruction loss, which involves image self-reconstruction loss $\mathcal{L}_x$, content latent code reconstruction loss $\mathcal{L}_c$ and style latent code reconstruction loss $\mathcal{L}_s$. And we have reconstruction loss of domain-specific content latent code $\mathcal{L}_z$ same as that in [6]:

$$\mathcal{L}_{x_1} = \mathbb{E}_{x_1}[\| x_{1\to 1} - x_1 \|_1], \tag{4}$$

$$\mathcal{L}_{c_1} = \mathbb{E}_{x_1,x_2}[\| E_1^c(x_{1\to 2}) - E_1^c(x_1) \|_1], \tag{5}$$

$$\mathcal{L}_{s_2} = \mathbb{E}_{z_1,r}[\| E_2^s(G_2(z_1,r))) - r \|_1], \tag{6}$$

$$\mathcal{L}_{z_1} = \mathbb{E}_{x_1,x_2}[\| M(E_1^c(x_{1\to 2})) - M(E_1^c(x_1))) \|_1] \tag{7}$$

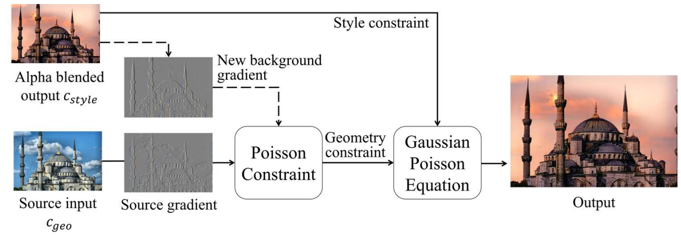


Fig. 4: Blending optimization. Alpha blended image c_{style} (e.g., $x_{1\to 2}$) works as style constraint, or meanwhile as background geometry constraint. The high-fidelity source input c_{geo} (e.g., x_1) constrains the precise image contour, enhancing overall photorealism.

where $x_{1\to 1} = G_1(z_1, s_1)$, $G_2(z_1, r)$ is generation with appearance of x_1 and random style r in style space of X_2, r is a random value drawn from a Gaussian distribution [3] to ensure diversity of the style embedded codes and multi-modal translations.

On one side, we consider the reconstruction of the input source image; on the other side, we take into account the reconstructions of domain-specific content latent code, style latent code, and domain-invariant content latent code, respectively from the content encoder E_1^c and style encoder E_2^s and content space mapping M.

Cycle Consistent Loss. For unsupervised GANs, Zhu *et al.* [2] proposed the cycle-consistent loss, which is widely accepted in unsupervised image translation tasks. We compute the cross-cycle consistency loss $\mathcal{L}_{cycle}$ between input x_1 and the reconstruction from domain X_1 to domain X_2 and back to X_1 again (i.e., $X_{1\to 2\to 1}$):

$$\mathcal{L}_{cycle_1} = \mathbb{E}_{x_1,x_2}\left[\| G_{1\to 2\to 1}(x_1, x_2), -x_1 \|_1\right]. \tag{8}$$

Adversarial Loss. We also adopt the adversarial loss L_{adv} of LSGAN [53] between discriminators and generators:

$$\mathcal{L}_{adv_1}^{D_2} = \mathbb{E}_{x_1,x_2}\left[\frac{1}{2}(D_2(x_1) - 1)^2 + \frac{1}{2}D_2(G_2(z_1, s_x))^2\right], \tag{9}$$

$$\mathcal{L}_{adv_1}^{G_2} = \mathbb{E}_{x_1,x_2}\left[\frac{1}{2}(D_2(G_2(z_1, s_x) - 1)^2\right], \tag{10}$$

where D_2 is the discriminator for images in target domain X_2, $s_x = \{s_2, r\}$.

Total Loss. All encoders, generators and discriminators are trained simultaneously with bidirections. We get the final total loss for the generator as

$$\begin{aligned}\mathcal{L}_{total} &= \lambda_x\mathcal{L}_x + \lambda_c\mathcal{L}_c + \lambda_s\mathcal{L}_s + \lambda_z\mathcal{L}_z \\ &+ \lambda_{cycle}\mathcal{L}_{cycle} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{gd}\mathcal{L}_{gd} + \lambda_{kl}\mathcal{L}_{kl},\end{aligned} \tag{11}$$

where λ's are hyperparameters to balance the losses. Each loss contains losses in both directions

$$\mathcal{L}_* = \mathcal{L}_{*_1} + \mathcal{L}_{*_2} \tag{12}$$

where $* \in \{x, c, s, cs, cycle, adv, gd, kl\}$. For training with background we set $\lambda_{gd} = \lambda_{kl} = 0$.

Hyperparameter Settings In training, we adapt the Adam Optimizer with an initial learning rate of 1×10^{-4}, $\beta_1 = 0.5$, $\beta_2 = 0.999$. For foreground and background training, we set different hyperparameters for geometry loss, remaining weights the same for all our models. Empirically, for foreground training, we set the weight of image gradient loss $\lambda_{gd} = 5$, weight of spatial luminance KL divergence loss $\lambda_{kl} = 5$, while for background training, we set both of them zero to let the generator learn to change the background texture. For other losses, we empirically set $\lambda_x = 10$, $\lambda_c = 2$, $\lambda_{cs} = 2$, $\lambda_s = 10$, $\lambda_{cc} = 5$, $\lambda_{adv} = 1$. We set the batch size to 2 for training. Each model for foreground and background was trained for 200k iterations, respectively.

4.5 Image Blending Optimization

After image translation, we get two generated images (foreground and background). Foreground and background generated images are integrated again with the segmentation mask from the segmentation module using alpha blending.

We apply a similar strategy to [55] for blending optimization (see Fig.4), which helps restore the original gradient. Instead of training a new GAN to generate a relatively low-resolution color constraint image as described in [55], we apply our translated image as the style constraint ($c_{style} = x_{1\rightarrow 2}$). With realistic low-resolution style constraint (c_{style}) and the high-fidelity source image ($c_{geo} = x_1$) with perfect geometry, we iteratively optimize the Gaussian Poisson Equation [55] and finally retrieve source geometry while the transferred style is preserved. Differently, to retain novel generated background textures (e.g., new cloud texture), the style constraint image is used to extract a new background gradient for blending optimization. Empirically, 1 or 2 iterations are enough for high fidelity restoration.

5 Experiments

In this section, we first introduce the dataset used for training and evaluation, and then the baselines and evaluation metrics. Next, quantitative and qualitative comparisons are reported and discussed. In addition, the ablation study results are illustrated to validate the effectiveness of our framework design. Finally, we show the comparisons of the proposed method (deep learning base) with traditional methods (non-deep learning base).

5.1 Time-lapse Architectural Dataset

To better achieve style transition of different times in the day for architectural images, we collected 21,291 high-resolution exterior architectural photos for training. The training photos include 16,908 unpaired landmark photos in the wild and 4,383 extracted frames from 110 time-lapse videos of outdoor scenes from [14]. The evaluation set consists of 1,003 photos of high fidelity collected from public domains [56], [57], [58]. We manually filtered out low-resolution or unattractive images, labeled images into four classes (day, golden, blue, night), and pre-processed images with proper cropping and resizing before training. All photographs of the sky and main architecture(s) have different geometry, luminance, and chrominance changes. The sky has a dynamic appearance as time elapses, while the architectures keep static geometry. Therefore, an outstanding time-of-day architectural style transfer should accomplish matched style transfer semantically.

More details can be checked in the supplementary. We will distribute our training dataset upon request for research purposes and release our evaluation set publicly.

5.2 Experiment Setup

Baselines. We qualitatively and quantitatively compare our results to state-of-the-art image-to-image translation baselines, i.e., DRIT++ [3], [54], MUNIT [4], FUNIT [5] and DSMAP [6], and StarGANv2 [35]; neural style transfer methods including AdaIN [28], LST [32], SANet [31], AdaAttN [33]. For ablation study, our method also compare to our variants without proposed geometry loss, and our model trained without segmentation.

Implementation Details. We trained all models until the total generation loss did not decrease noticeably. All baselines are trained using whole images, while our work uses segmented images with different hyperparameter settings for foreground and background training.

At the training time of baselines and ours, all images are resized into 286×286 and randomly cropped to 256×256 with random horizontal flipping. At inference time, photos are resized with the shortest side to 256 pixels or 512 pixels to generate transferred results. All models are trained in a resolution of 256×256. Our networks are fully convolutional and support input images of arbitrary resolutions. For quantitative evaluation, results are evaluated in 256× at smaller side. For qualitative comparison, we display results of 512× at smaller side. The teaser Fig 1 show results of 1024× at smaller side.

All methods use a single NVIDIA GeForce RTX 2080 Ti GPU for training except that FUNIT utilizes 4 GPUs. FUNIT, StarGANv2 and all neural style transfer approaches trained with daytime images as source and all four classes together as targets. We use the default training configuration for all baselines except that we increase the weight of content loss in AdaIN (content : style = 3 : 10) and SANet (content : style = 2 : 1) and the weight of local feature loss in AdaAttN (local : global = 1 : 2) to enhance the geometry of the stylized images.

The unseen evaluation set apart from the training set is used for result inference. The main exploratory experiments are three types of style transfers, i.e., daytime to golden hours, daytime to blue hours, and daytime to nighttime. In the evaluation set, all daytime images get transferred with every style reference image in each target-style set (i.e., golden, blue, and night), and we got totally over 200k results from each method for quantitative evaluation.

Performance Metrics. The quantitative evaluation involves generation accuracy and diversity, geometry and appearance preservation, and semantic style transfer.

We trained an InceptionV3 [59] classifier using our dataset with three target domain labels (i.e., golden, blue, and night). We evaluate the generation top-1 accuracy and Inception Score (IS) [60] which indicate how realistic and diverse the generation is.

To quantitatively evaluate geometry and appearance similarity, we utilize Structural Similarity Index (SSIM) [61]. Similar to [16], we calculate *Edge Conditioned SSIM* (edge-SSIM), which computes image structural similarity (SSIM) between Canny edge detected maps of images [62], alleviating luminance influence on geometry.

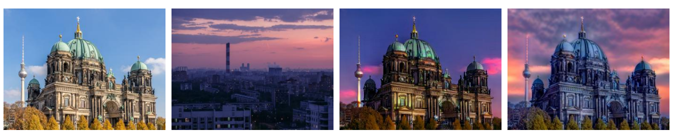
Intersection over Union (IoU) between input and output is used to evaluate structure preservation and visual recognizability in foreground and background. We use the same segmentation model used in our framework.

	DRIT++ [3], [54]	MUNIT [4]	FUNIT [5]	DSMAP [6]	StarGANv2 [35]	AdaIN [28]	SANet [31]	AdaAttN [33]	LST [32]	Ours
e-SSIM↑	0.5214	<u>0.5653</u>	0.4959	0.4790	0.4778	0.4962	0.4854	0.5194	0.4903	**0.6359**
Acc↑	0.8903	0.8678	0.77.14	<u>0.9106</u>	0.8788	0.7352	0.6193	0.6443	0.7071	**0.9486**
IS↑	2.6160	2.5916	2.5903	<u>2.6580</u>	2.6088	2.4082	2.1062	2.0928	1.7299	**2.7290**
IoU↑	0.6915	**0.7382**	0.5473	0.4975	0.4100	0.6642	0.7183	0.6532	0.6264	<u>0.7257</u>

TABLE 1: Evaluation results of Daytime to Golden, Blue and Nighttime Hour translations. **Bold** and <u>underlined</u> text indicates the best and 2nd best result, respectively.

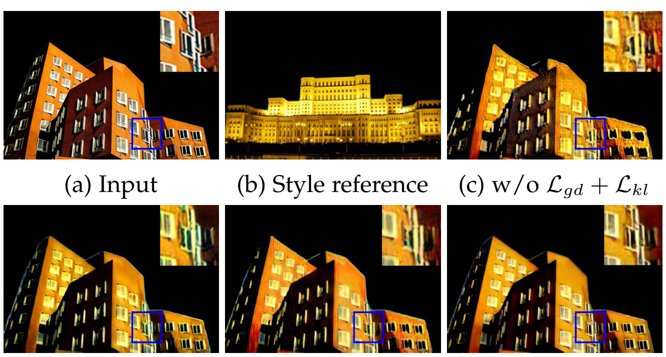
	DRIT++ [3], [54]	MUNIT [4]	FUNIT [5]	DSMAP [6]	StarGANv2 [35]	AdaIN [28]	SANet [31]	AdaAttN [33]	LST [32]	Ours-opt
e-SSIM↑	0.4939	<u>0.5074</u>	0.5003	0.4911	0.4935	0.4863	0.4921	0.4706	0.4815	**0.8094**
Acc↑	0.8586	0.8507	0.7532	<u>0.8721</u>	0.8391	0.7019	0.5332	0.5973	0.6623	**0.9007**
IS↑	2.5528	2.5236	2.5227	<u>2.5572</u>	2.5087	2.2646	1.8310	1.9978	2.1587	**2.6127**
IoU↑	0.6997	0.7228	0.7143	0.6972	0.6902	0.7083	<u>0.7369</u>	0.7023	0.7053	**0.7715**

TABLE 2: Evaluation results of Daytime to Golden, Blue and Nighttime Hour translations with blending optimization applied to all methods. **Bold** and <u>underlined</u> text indicates the best and 2nd best result, respectively.



(a) Input (b) Style (c) Whole (d) Segmented

Fig. 5: Qualitative comparison on segmentation condition. (a) Input image. (b) Style reference. (c) Style transfer result from our network trained on whole images. (d) Result from our network trained on segmented images.



(a) Input (b) Style reference (c) w/o $\mathcal{L}_{gd} + \mathcal{L}_{kl}$

(d) w/o $\mathcal{L}_{kl}$ (e) w/o $\mathcal{L}_{gd}$ (f) $\mathcal{L}_{total}$

Fig. 6: Qualitative comparison on geometry losses.

5.3 Quantitative Results

We carried out metric evaluations on three style translations, i.e., daytime to golden hour, daytime to blue hour, and daytime to nighttime, and reported mean scores in Tables 1 and 2. Our translation results and optimized results (Ours and Ours-opt) outperform others or have competitive performance on style transfer accuracy and quality (Accuracy, IS, IoU), structure and perceptual similarity (edge-SSIM, IoU). It validates the effectiveness of geometry losses, and our blending optimization (Ours-opt) can somehow restore primal scene contour information (edge-SSIM) and improve perceptual similarity (IoU). Overall, our models have better style transfer in terms of photorealism and style diversity.

	e-SSIM↑	Acc↑	IS↑	IoU↑
Ours-whole	0.6838	0.8282	2.5240	0.7410
Ours	0.6359	0.9486	2.7290	0.7257
Ours-opt	0.8094	0.9007	2.6127	0.7715

TABLE 3: Ablation study of segmentation. Ours-whole indicates our full model trained with whole images, while Ours (or Ours-opt) is the full model trained with segmented images.

	w/o $\mathcal{L}_{kl} + \mathcal{L}_{gd}$	w/o $\mathcal{L}_{kl}$	w/o $\mathcal{L}_{gd}$	$\mathcal{L}_{total}$
e-SSIM↑	0.4800	0.5539	0.5159	**0.6359**
Acc↑	0.8934	0.9201	0.9265	**0.9486**
IS↑	2.6858	2.7183	2.7241	**2.7290**
IoU↑	0.6056	0.6536	0.6612	**0.7257**

TABLE 4: Ablation study of different geometry losses for foreground models. **Bold** text indicates the best result.

Ablation Study

Does segmentation help?

We conducted ablation experiments on models trained with and without segmentation. On the one hand, segmentation provides explicit semantic color distribution information. As seen in Figure 5, some cases in which the model trained with whole images cannot semantically transfer color from buildings and sky, respectively. Our trained model using segmented foreground and background can successfully convey the correct color style semantically. On the other hand, with different hyperparameters of geometry losses for foreground and background models, our result Fig. 5(d) generates a dense cloud texture similar to style image Fig. 5(b) and keeps foreground appearance unchanged. But the result in Fig. 5(c) preserves both unchanged foreground and background texture and geometry.

The evaluated metric results are illustrated in Table 3. Compared to *Ours*, *Ours-whole* has better geometry preservation (e-SSIM, IoU) but worse stylization performance (Accuracy, IS). It might be because *Ours* has different background from source input while *Ours-whole* keeps whole image geometry information intact. The evaluation results in Table 1

Fig. 7: Comparisons among image-to-image translation baselines and our proposed method. Our results have plausible colors from foreground and background, and preserve the geometry in different style transfer cases. (Please see our interactive viewer in the supplementary for a detailed comparison.)

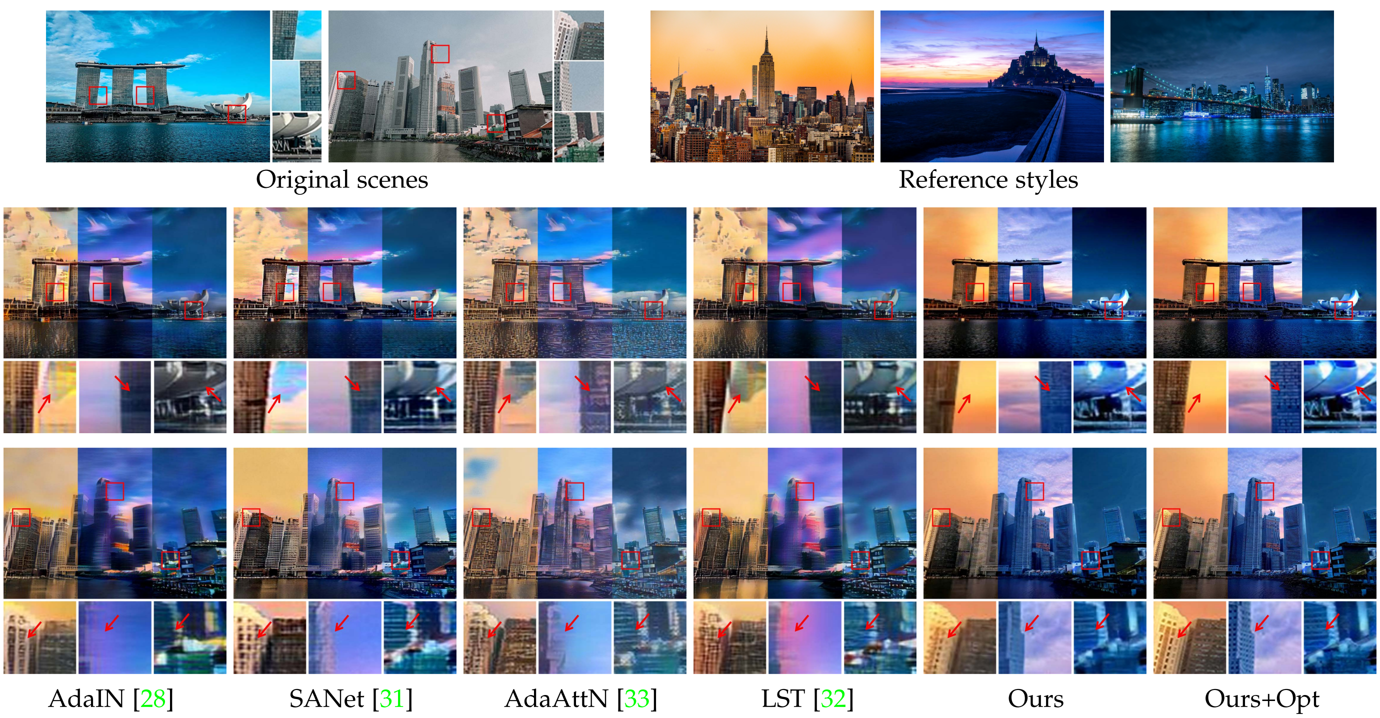


Fig. 8: Comparisons among neural style transfer baselines and proposed method. While neural style transfer methods tend to have visual artifacts, our results have matched colors from foreground and background respectively, and preserve the geometry of the foreground while generating diverse cloud textures in the background. (Please see our interactive viewer in the supplementary for a detailed comparison.)

(IoU) and Figure 9 (Structure & Style) also show that our segmented training strategy has better performance than baselines with respect to semantic style transfer.

Are the geometry losses effective?

To validate the proposed geometry losses (*Image Gradient loss* and *Spatial Luminance KL Divergence loss*), we train

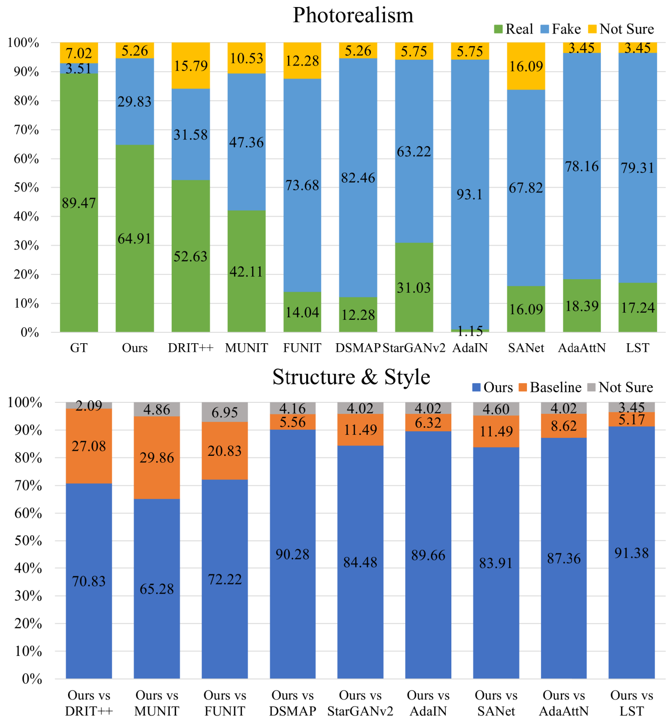


Fig. 9: Perceptual preference in terms of photorealism, structure and style.

Fig. 10: Additional comparisons to traditional methods.

foreground models with different settings: no geometry loss, without KL divergence loss, without image gradient loss, and with full geometry losses.

According to qualitative results in Figure 6 and quantitative results in Table 4, a model with either only KL divergence loss or only gradient loss can to some extent help preserve geometry but does not perform better than a model with both geometry losses. Some previous work [25] tried applying gradient loss in style transfer network but failed. We conjecture that the generator attains insufficient geometry transfer information from only gradient loss, e.g., with unconstrained luminance distribution, so infinite geometry changes have similar or even equal gradient losses. With the limit of spatial luminance distribution ($\mathcal{L}_{kl}$), gradient loss can protect primal geometry with geometry luminance density preservation while transferring sufficient style from source domain to target domain. In general, gradient loss plus spatial luminance KL divergence loss can largely improve the correctness of geometry luminance transfer and keep geometry unchanged.

User Study

To validate our results, we also conducted a perceptual user study (Fig. 9) covering three aspects, i.e., image photorealism and semantic structure similarity plus semantic style consistency. The photorealism score contains the percentage of images that look real or fake. Structure and style score is obtained from pair comparisons (ours versus other baselines). Semantic structure similarity illustrates how well generated images keep foreground geometry intact and transfer target background texture. Semantic style consistency shows how much style transfers correctly for foreground and background. The results in Fig. 9 show that the proposed method outperforms previous works in terms of image fidelity and semantic style matching, indicating that our method can achieve more photorealistic style transfer than others. Particularly, neural style transfer methods generate much more non-photorealistic images than most image-to-image translation approaches. Please refer to supplementary materials for the complete quantitative results and the user study details.

5.4 Qualitative Results

Results from some baselines for style transfers from daytime to golden hour, blue hour, and nighttime are selected to display in Figure 7 and Figure 8. More visual results can be checked in the supplementary.

In general, all baselines tend to have inaccurate semantic color matching. As can be seen, for golden or blue style transfer, all baselines treat some parts of the building as sky, so it leaks sky color (DRIT++, MUNIT, FUNIT, DSMAP, AdaIN, SANet, AdaAttN) or texture (FUNIT, DSMAP, AdaIN, LST) to the foreground.

FUNIT, and DSMAP tend to generate noisy artifacts, e.g., golden hour and nighttime style transfer. DSMAP and FUNIT sometimes merge part of the building into the background, hence appearing the ghosting artifacts and building distortion. By contrast, our approach can transfer a more semantically matched color style while retaining the foreground appearance. Sky in our results has both style and texture in line with the target style images, and our foreground has plausible new lighting style and geometry quality.

From visual results in Fig.8, neural style transfer methods (AdaIN, SANet, AdaAttN, and LST) in some way preserve global geometry because of the effect of content related losses. Nonetheless, these style transfer methods transfer

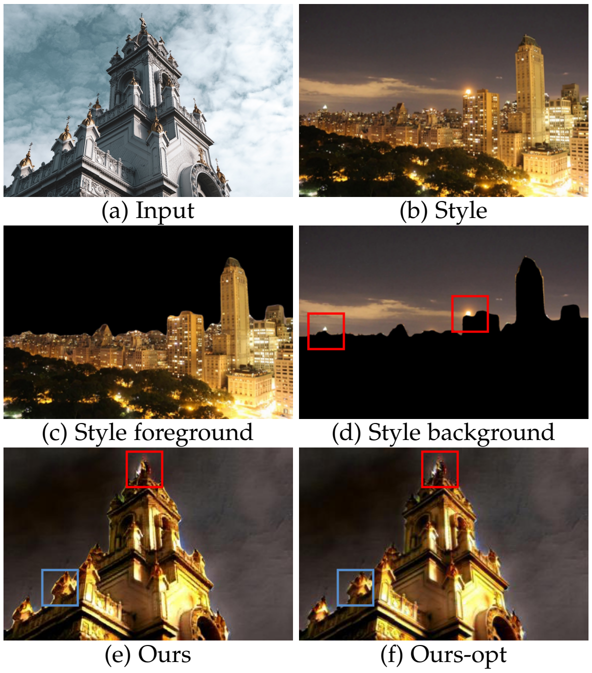


Fig. 11: Failure cases. (Zoom in for a best view.)

texture strokes for both sky and foreground (e.g. sky in results in the first row, building texture in results in the second row). Ours tends to produce different sky textures according to the reference styles, such as removing clouds for golden style and adding clouds for blue style in Fig.8. Particularly, AdaAttN has sharper appearance features since it constrains both local (high-frequency) and global (low-frequency) content features during training. But AdaattN fails to produce photorealistic appearance. By contrast, our proposed geometry losses are based on high-frequency spatial information, thus helping with geometry preservation. Regarding stylization performance, AdaIN, SANet, and LST have visually more aesthetic style effects than AdaAttN and globally matched color in terms of reference styles, though the transferred images look like paintings. In general, our results are more photorealistic, while neural style transfer baselines are more artistic.

5.5 Comparisons to Traditional Methods

We provide additional comparisons to non-deep learning methods including color transfer by Shih *et al.* [14], by Pouli and Reinhard [63], and an intrinsic decomposition for relighting method by Laffont *et al.* [41] in Figure 10. Shih et al.'s locally affine model is solved from a pair of frames of a similar scene retrieved from their video database. The intrinsic decomposition-based method by Laffont et al. [41] requires paired images that capture a typical scene under varying lighting for illumination transfer. Our method instead learns the transfer with unpaired images.

For night-style images, Shih *et al.* fail to handle the background properly, while our result has a natural sky. Our method also has more consistent styles with better color saturation than the method by Laffont *et al.* and better highlights on rocks than the method by Pouli and Reinhard.

5.6 Limitation

As our approach largely relies on segmentation, sometimes deficient segmentation (e.g., nighttime style background and daytime imperfect foreground in Fig. 11) will lead to wrong color rendition or artifacts. For example, in Figure 11, the results generate shiny spot artifacts (red boxes in (e) and (f)) on the top of building due to wrong style background, and wrong bright yellow boundary (blue boxes in (e) and (f)) on background due to input foreground enclosing background portion.

6 Conclusion

We realize a photorealistic neural style transfer system to solve the architectural style transfer problem, which transfers an outdoor architectural photograph from daytime style to a target style at different magical times in a day (i.e., golden hour, blue hour, and nighttime). An ideal architectural style transfer is able to achieve color and illuminance transfer for foreground buildings, roads, etc., and realize novel target sky background color and texture transfer. With foreground and background segmentation for training respectively and proposed geometry losses, our image-to-image translation models successfully transfer semantically matched styles and effectively preserve content information of foreground.

Our proposed method is not limited to architectural photographs and can be generalized to transfer other types of images as long as they have foreground and background in different lighting and texture properties, e.g. headshot portrait transfer, group photos, natural landscape images with animals. Exploring these domains is interesting future work once sufficient data are collected. Additionally, developing a more robust segmentation technique for complex outdoor scenes, and training the proposed networks in an end-to-end fashion would be interesting research avenue.

Acknowledgments

This paper was partially supported by an internal grant from HKUST (R9429) and the HKUST-WeBank Joint Lab.

References

[1] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134. 2

[2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232. 2, 5

[3] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 35–51. 2, 4, 5, 6, 7, 8

[4] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 172–189. 2, 4, 5, 6, 7, 8

[5] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 551–10 560. 2, 6, 7, 8

[6] H.-Y. Chang, Z. Wang, and Y.-Y. Chuang, "Domain-specific mappings for generative adversarial style transfer," in *European Conference on Computer Vision*. Springer, 2020, pp. 573–589. 2, 4, 5, 6, 7, 8

[7] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Manipulating attributes of natural scenes via hallucination," *ACM Transactions on Graphics*, vol. 39, no. 1, Nov. 2019. 2, 3

[8] I. Anokhin, P. Solovev, D. Korzhenkov, A. Kharlamov, T. Khakhulin, A. Silvestrov, S. Nikolenko, V. Lempitsky, and G. Sterkin, "High-resolution daytime translation without domain labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7488–7497. 2, 3

[9] S. Nam, C. Ma, M. Chai, W. Brendel, N. Xu, and S. J. Kim, "End-to-end time-lapse video synthesis from a single outdoor image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 1409–1418. 2, 3

[10] C.-C. Cheng, H.-Y. Chen, and W.-C. Chiu, "Time flies: Animating a still image with time-lapse video as reference," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5641–5650. 2, 3

[11] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, "Natural image colorization," in *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, ser. EGSR'07. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007, pp. 309–320. [Online]. Available: http://dx.doi.org/10.2312/EGWR/EGSR07/309-320 2

[12] R. Ironi, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Rendering Techniques*. Citeseer, 2005, pp. 201–210. 2

[13] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer graphics and applications*, vol. 21, no. 5, pp. 34–41, 2001. 2

[14] Y. Shih, S. Paris, F. Durand, and W. T. Freeman, "Data-driven hallucination of different times of day from a single outdoor photo," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 200, 2013. 2, 3, 6, 9, 10

[15] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, "A closed-form solution to photorealistic image stylization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 453–468. 2

[16] J. Yoo, Y. Uh, S. Chun, B. Kang, and J.-W. Ha, "Photorealistic style transfer via wavelet transforms," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9036–9045. 2, 6

[17] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423. 2

[18] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4990–4998. 2

[19] S. Penhouët and P. Sanzenbacher, "Automated deep photo style transfer," *arXiv preprint arXiv:1901.03915*, 2019, source code on GitHub: https://github.com/Spenhouet/ automated-deep-photo-style-transfer. 2

[20] P. Wang, Y. Li, and N. Vasconcelos, "Rethinking and improving the robustness of image style transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 124–133. 2

[21] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand, "Style transfer for headshot portraits," *ACM Trans. Graph.*, 2014. 2

[22] M. Lu, H. Zhao, A. Yao, F. Xu, Y. Chen, and L. Zhang, "Decoder network over lightweight reconstructed feature for fast semantic style transfer," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2469–2477. 2

[23] Y.-S. Liao and C.-R. Huang, "Semantic context-aware image style transfer," *IEEE Transactions on Image Processing*, vol. 31, pp. 1911–1923, 2022. 2

[24] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 228–242, 2007. 2

[25] R. Mechrez, E. Shechtman, and L. Zelnik-Manor, "Photorealistic style transfer with screened poisson equation," *BMVC*, 2017. 2, 5, 9

[26] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," *Advances in neural information processing systems*, vol. 30, 2017. 2

[27] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 110:1–110:11, Jul. 2016. [Online]. Available: http://doi.acm.org/10.1145/2897824.2925974 2

[28] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510. 2, 4, 6, 7, 8

[29] B. Zhang, M. He, J. Liao, P. V. Sander, L. Yuan, A. Bermak, and D. Chen, "Deep exemplar-based video colorization," 2019. 2

[30] M. He, J. Liao, D. Chen, L. Yuan, and P. V. Sander, "Progressive color transfer with dense semantic correspondences," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 2, p. 13, 2019. 2

[31] D. Y. Park and K. H. Lee, "Arbitrary style transfer with style-attentional networks," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5880–5888. 2, 6, 7, 8

[32] X. Li, S. Liu, J. Kautz, and M.-H. Yang, "Learning linear transformations for fast arbitrary style transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 6, 7, 8

[33] S. Liu, T. Lin, D. He, F. Li, M. Wang, X. Li, Z. Sun, Q. Li, and E. Ding, "Adaattn: Revisit attention mechanism in arbitrary neural style transfer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6649–6658. 2, 6, 7, 8

[34] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in neural information processing systems*, 2017, pp. 700–708. 2

[35] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8188–8197. 2, 6, 7

[36] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. A. Efros, and R. Zhang, "Swapping autoencoder for deep image manipulation," in *Advances in Neural Information Processing Systems*, 2020. 2

[37] M. M. Ho, L. Zhang, and J. Zhou, "Semantic-driven colorization," *arXiv preprint arXiv:2006.07587*, 2020. 2

[38] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," *arXiv preprint arXiv:2008.02268*, 2020. 2

[39] A. Liu, S. Ginosar, T. Zhou, A. A. Efros, and N. Snavely, "Learning to factorize and relight a city," in *European Conference on Computer Vision*. Springer, 2020. 3

[40] Y. Yu, A. Meka, M. Elgharib, H.-P. Seidel, C. Theobalt, and W. A. Smith, "Self-supervised outdoor scene relighting," in *European Conference on Computer Vision*. Springer, 2020, pp. 84–101. 3

[41] P.-Y. Laffont, A. Bousseau, S. Paris, F. Durand, and G. Drettakis, "Coherent intrinsic images from photo collections," *ACM Transactions on Graphics (TOG)*, 2012. 3, 9, 10

[42] S. Duchêne, C. Riant, G. Chaurasia, J. Lopez-Moreno, P.-Y. Laffont, S. Popov, A. Bousseau, and G. Drettakis, "Multi-view intrinsic images of outdoors scenes with an application to relighting," *ACM Transactions on Graphics (TOG)*, p. 16, 2015. 3

[43] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays, "Transient attributes for high-level understanding and editing of outdoor scenes," *ACM Transactions on graphics (TOG)*, vol. 33, no. 4, pp. 1–11, 2014. 3

[44] Y. Hold-Geoffroy, A. Athawale, and J.-F. Lalonde, "Deep sky modeling for single image outdoor lighting estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6927–6935. 3

[45] W. Xiong, W. Luo, L. Ma, W. Liu, and J. Luo, "Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2364–2373. 3

[46] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *Computer*, vol. 43, no. 6, pp. 32–38, 2010. 3

[47] A. R. Zamir and M. Shah, "Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 8, pp. 1546–1558, 2014. 3

[48] Y. Endo, Y. Kanamori, and S. Kuriyama, "Animating landscape: self-supervised learning of decoupled motion and appearance for single-image video synthesis," *arXiv preprint arXiv:1910.07192*, 2019. 3

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. 3

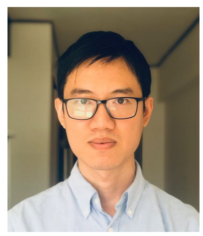
[50] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890. 3

[51] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal on Computer Vision*, 2018. 3
[52] "Ycbcr - wikipedia," Jun 2022. [Online]. Available: https://en.wikipedia.org/wiki/YCbCr 5
[53] X. Mao, Q. Li, H. Xie, R. Lau, and Z. Wang, "Least squares generative adversarial networks. cite," *arXiv preprint arxiv:1611.04076*, vol. 4, 2016. 5
[54] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, "Drit++: Diverse image-to-image translation via disentangled representations," *International Journal of Computer Vision*, vol. 128, no. 10, pp. 2402–2417, 2020. 6, 7, 8
[55] H. Wu, S. Zheng, J. Zhang, and K. Huang, "Gp-gan: Towards realistic high-resolution image blending," in *Proceedings of the 27th ACM international conference on multimedia*, 2019, pp. 2487–2495. 6
[56] "Pexels," accessed November, 2020. [Online]. Available: https://www.pexels.com/ 6
[57] "Pikwizard," accessed November, 2020. [Online]. Available: https://pikwizard.com/ 6
[58] "Unsplash," accessed November, 2020. [Online]. Available: https://unsplash.com/ 6
[59] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826. 6
[60] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, pp. 2234–2242, 2016. 6
[61] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004. 6
[62] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986. 6
[63] T. Pouli and E. Reinhard, "Progressive color transfer for images of arbitrary dynamic range," *Computers & Graphics*, vol. 35, no. 1, pp. 67–80, 2011. 9, 10

Yingshu Chen is a Ph.D. student in the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology (HKUST). She received her Bachelor of Engineer in Digital Media Technology at Zhejiang University. Her research interests cover computer vision and computer graphics for computational design such as 2D and 3D style transfer.

Tuan-Anh Vu is a third-year Ph.D. student at Hong Kong University of Science and Technology (HKUST). He received an engineering degree from International University, Vietnam National University, Ho Chi Minh City, Vietnam in 2016. His current research interests include Deep Learning, Unsupervised Learning, 3D Reconstruction, Scene Understanding, and Shape Analysis.

Ka-Chun Shum is an M.Phil. student in the department of Computer Science and Engineering of the Hong Kong University of Science and Technology (HKUST). He received a BSc double degree in Computer Science and Mathematics from HKUST. His research interests are in the computer vision field, including deep learning on images and content generation.



Binh-Son Hua is a research scientist at VinAI Research, Vietnam. His research interests are at the intersection of computer graphics, computer vision, and machine learning with the focus on 3D deep learning and realistic image synthesis. He received his PhD in Computer Science from National University of Singapore in 2015, and spent his postdoctoral research at Singapore University of Technology and Design and The University of Tokyo. He received the Best Paper Honorable Mention at 3DV 2016.

Dr. Sai-Kit Yeung is an Associate Professor at the Division of Integrative Systems and Design (ISD) and the Department of Computer Science and Engineering (CSE) at the Hong Kong University of Science and Technology (HKUST). His research interests include 3D vision and graphics, content generation, fabrication, and novel computational techniques and integrative systems for marine-related problems. He has served as a Senior Program Committee member in IJCAI and AAAI, and as a Course Chair for SIGGRAPH Asia 2019. In addition, he regularly serves as a Technical Papers Committee member for SIGGPAPH & SIGGRAPH Asia and is currently an Associate Editor of the ACM Transactions on Graphics (TOG).