Stylization-Based Architecture for Fast Deep Exemplar Colorization

Zhongyou Xu[†], Tingting Wang[†], Faming Fang[†*], Yun Sheng[§], Guixu Zhang[†]
[†]Shanghai Key Laboratory of Multidimensional Information Processing,
and the School of Computer Science and Technology, East China Normal University
[§]Liverpool John Moores University

Abstract

Exemplar-based colorization aims to add colors to a grayscale image guided by a content related reference image. Existing methods are either sensitive to the selection of reference images (content, position) or extremely time and resource consuming, which limits their practical application. To tackle these problems, we propose a deep exemplar colorization architecture inspired by the characteristics of stylization in feature extracting and blending. Our coarse-to-fine architecture consists of two parts: a fast transfer sub-net and a robust colorization sub-net. The transfer sub-net obtains a coarse chrominance map via matching basic feature statistics of the input pairs in a progressive way. The colorization sub-net refines the map to generate the final results. The proposed end-to-end network can jointly learn faithful colorization with a related reference and plausible color prediction with unrelated reference. Extensive experimental validation demonstrates that our approach outperforms the state-of-the-art methods in less time whether in exemplar-based colorization or image stylization tasks.

1. Introduction

Colorization is a classic task in computer vision which aims at adding colors to a gray image. It is of great significance since color image is visually plausible and perceptually meaningful, and has enormous potential in practical applications such as painting creation, black and white movie rendering, 3D modeling, etc. It is worth noting that there is no “correct ” solution for colorization (*e.g.,* flowers in white, red, or purple are common in nature). The ambiguity and diversity make the colorization task challenging.

In order to achieve a convincing result, human intervention often plays an important role in the previous colorization works. Some papers [18, 39, 25] propose to manually add color scribbles over a grayscale image carefully, then propagate these known colors to the whole image, which is, however a challenging task for an untrained user who lacks professional skills and artistic sensitivity. Other researches [23, 3] try to alleviate such empirical problem by replacing the color scribbles with a closely related reference image. Credible correspondences are established between the target gray image and the reference image for propagating and coloring directly. The fragile model usually generates an inferior result for the dissimilarity caused by lighting, viewpoints and content differences. In recent years, deep learning-based colorization techniques [17, 2, 40] have achieved remarkable results. The colorization network is trained on a large number of image dataset to learn the relationship between grayscale image and its corresponding color version. Once the net parameters are determined, the colorization result can be easily obtained. Although such automatic colorization models make a huge success, the colorized result is sort of uncontrollable without any user intervention. More recent works [31, 41, 8] attempt to combine the controllability from interaction and robustness from learning to achieve more promising colorization.

In this paper, we propose a novel deep learning approach for fast exemplar-based colorization inspired by stylization networks [9, 20]. The proposed architecture consists of two parts: transfer sub-net and colorization sub-net. The transfer sub-net is an arbitrary fast photorealistic image stylization network which can solve the distoration problem and alleviate artifacts compared with current stylization methods. It is designed to obtain a coarse chrominance map for the target by fusing its features with the reference in a progressive way. Then the map is refined by the colorization sub-net to get the final colorization result. Experiments show that our proposed network outperforms state-of-the-art exemplar-based colorization methods in less time. Besides, the transfer sub-net presents amazing effects in photorealistic image stylization tasks.

The main contributions of this work are threefold.

[*] Corresponding Author. Email address: fmfang@cs.ecnu.edu.cn. This work was sponsored by the Key Project of the National Natural Science Foundation of China under Grant 61731009 and the National Natural Science Foundation of China under Grant 61871185, and ”Chenguang Program” supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission under Grant 17CG25.

- We propose a novel two sub-nets architecture to jointly learn faithful colorization with a related reference and plausible color prediction with an unrelated one.
- We achieve better colorization in less time compared with other examplar-based methods by using the AdaIN operation for feature matching and blending.
- We extend the transfer sub-set to photorealistic image stylization without any additional modification.

2. Related Work

Current colorization methods can be roughly divided into four categories: scribble-based, exemplar-based, learning-based and hybrid colorization. In this section we provide an overview of the related works of each category as well as photorealistic image stylization.

2.1. Colorization

Scribble-based colorization. These interactive methods propagate initial strokes or color points to the whole grayscale image. The propagation is based on some low-level similarity metircs, such as spatial offset and intensity difference. Levin *et al.* [18] solve a Markov Random Field for propagating scribble colors based on the assumption that adjacent pixels with similar intensity should have similar color. Later methods [28, 25] aim to present advanced similarity metrics. An *et al.* [1] use an energy-optimization formulation to propagate the initial coarse edits to refined ones by enforcing the policy that similar edits are applied to spatially-close regions with similar appearance. These methods are capable of providing plausible colorization results when given good prior colors. However, it is still a challenging task for an untrained user who lacks professional skills and artistic sensitivity.

Exemplar-based colorization. This kind of methods present an intuitive way to colorize a gray image guided by a reference one. There are two main categories: global transfer method [29, 27, 4] and local transfer method [16, 32, 37]. The former one transfers colors from reference to target by matching global color statics like mean, variance and histogram [38]. These approaches yield unrealistic results since they ignore spatial pixel information. The later one usually considers different levels of correspondences, such as pixel level [23], superpixel level [3, 7] and segmented region level [13, 33]. These traditional methods based on hand-crafted similarity metrics are susceptible to generate terrible result when two images have different appearances but perceptually similar semantic structures.

Learning-based colorization. These methods have achieved automatic colorization by training an end-to-end network to reconstruct an image by predicting every pixel of the target image. Researchers intend to adapt well-designed loss function for better experimental results, such as L_2 loss [11], classification loss [17] and L_1+GAN [14]. All of these studies learn the network parameters from huge image datasets automatically without any user intervention. However, most of them only produce a single plausible result for each grayscale image, with no thought for multi-model uncertainty of colorization. Besides, the colorization result is uncontrollable without any user interaction.

Hybrid colorization. More recently, some methods attempt to combine the controllability from interaction and robustness from learning to achieve more promising colorization. Zhang *et al.* [41] and Sangkloy *et al.* [31] combine scribble-based methods with learning-based methods by provided color points or strokes. To reduce the skill requirement for users, Xiao *et al.* [38] utilize histgram of reference image to guide the cilorization by pyramid structure network. He *et al.* [8] introduce a similarity network to build bidirectional mapping. They retrain a gray extractor for calculating semantic similarity of the target and reference images and add a perceptual branch for processing appearance differences. Although this method ensures proper colorization even with improper reference, it requires heavy computation and loses creativity in terms of unseen objects.

2.2. Photorealistic image stylization

Photorealistic image stylization is evolved from traditional style transfer [5, 6]. The most discrepancy is that the output of the former should still maintain the original edge structure clearly. To suppress distortions appearing in stylization results, DSPT [24] adds a regularization term to the loss function of the neural style algorithm. Li *et al.* [20] present a modified Whitening and Coloring Transforms (PhotoWCT) model that utilizes unpooling layers and additional smooth operations. These methods have poor robustness and cause artifacts and blurriness in the result images, thus can not be directly applied to colorization. Inspired from these methods, we propose a transfer sub-net which can not only applied to image stylization but also generate an appropriate chrominance map for gray input.

3. Stylazation-based Colorization

The major problem in classic exemplar-based colorization lies in the selection of reference image. It is difficult to find the reference matching all the objects in the target image. Besides, the processing time is too long to be applied to real-time colorization. Our exemplar-based approach aims to generate plausible colorization result in real time whether the two input images are semantically related or not. We denote the target image as T and reference image as R and operate mainly in the CIELab space. The structure of the proposed end-to-end network is shown in Fig. 1, which consists of two sub-nets. The transfer sub-net generates an initial ab map (T_{ab}) for the target gray image by matching its basic

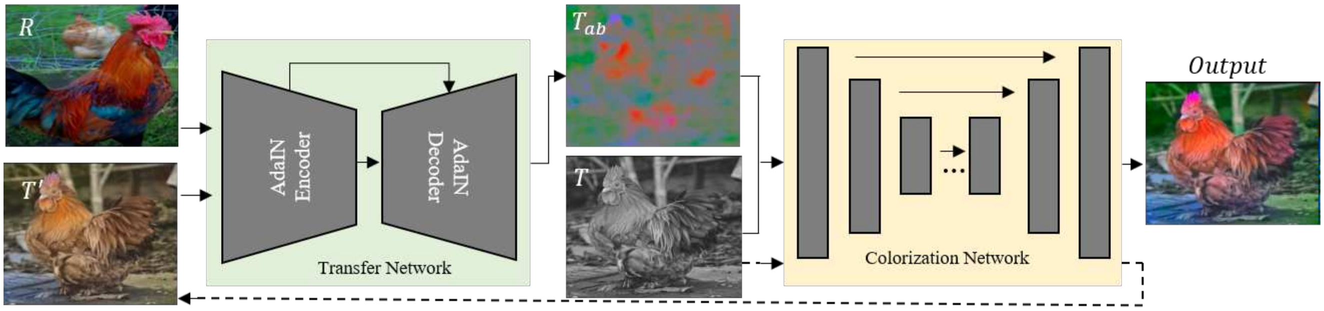


Figure 1. **System pipline.** Our model consists of two sub-nets: the transfer sub-net aims at generating coarse T_{ab} for target image that blends the color information of the reference; the colorization sub-net outputs the final result by refining T_{ab} and refers meaningful colors of the objects that are unrelated with the reference.

feature statistics with reference image in different layers directly. The colorization sub-net takes T_{ab} and known T as the input to refine the initial ab map and refer the discrepant objects with meaningful colors. Next we will introduce details of the two sub-nets.

3.1. Transfer sub-net

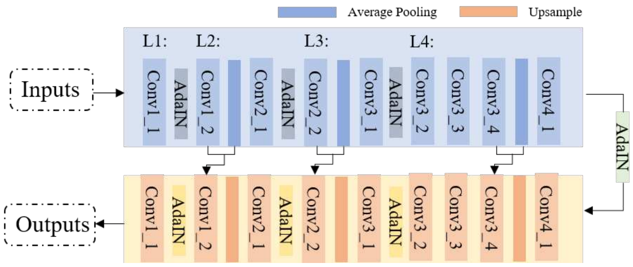


Figure 2. Detailed architecture of transfer sub-net.

The transfer sub-net is inspired by recent observations that arbitrary style transfer in real time becomes possible and photorealistic image stylization performs well on feature extracting, feature blending and content-consistency maintaining. We intend to obtain coarse chrominance information for the target gray image by matching and blending features of the reference with those of the target. As shown in Fig. 1, an encoder-decoder architecture is utilized for the transfer sub-net, which takes the target-reference image pairs as the input and outputs initial T_{ab}.

Detailed architecture of the transfer sub-net is shown in Fig. 2. We adopt the pretrained VGG19 modules (from $conv1\_1$ layer to $conv4\_1$ layer) as the encoder and a symmetrical decoder for image reconstruction. Since multi-decoders used in stylization methods [9] may destroy the property of VGG19 network and amplify the artifacts [20], we utilize a single pass encoder-decoder network instead of multi-decoders.

Specifically, we feed the two input images into the pretrained VGG19 and take the intermediate output of $conv\{i\}\_1$ layer as our feature representation. To match the features, WCT [19] calculates singular value decomposition (SVD) to project the content feaures to the eigenspace of style feaures. When the input images are with large size, the cumbersome computation is intolerant for a regular computer. To accelerate feature matching and blending, the fast Adaptive Instance Normalization (AdaIN) [9] is utilized.

AdaIN is a variant of IN [34] that is proposed to replace the batch normalization (BN) [12] layers in the stylization network. The success of IN benefits from its invariance to the contrast of the content image [35] and the affine parameters in IN can completely change the style of the output image [9]. AdaIN extends IN by adaptively computing the mean and variance independently for each channel and each sample. Given a content input x and a style input y, AdaIN is defined as

$$AdaIN(x,y) = \sigma(y)\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \mu(y) \qquad (1),$$

where $\sigma(\cdot)$ and $\mu(\cdot)$ are standard deviation and mean, respectively. Compared with BN and IN, AdaIN has no learnable affine parameters. We introduce the AdaIN operation after each $conv\{i\}\_1$ layer in the transfer sub-net, which can deal with arbitrary reference images and generate reliable ab map fast for the subsequent colorization sub-net.

However, the simple symmetrical architecture tends to cause artifacts and distortions during reconstruction process. It is intuitive to transmit more useful information in the previous layers to help the reconstruction. We concatenate the upsampled pooling feature and its previous convolution feature in each level of the encoder net for reconstruction in symmetrical module of the decoder by using a skip connection, as shown in Fig. 2. Besides, we replace every max-pooling layer with average-pooling layer in the encoder modules, which is proved to give better results in stylization task [5].

There is still a critical problem to be solved: how to extract features of the two input images using the same encoder net, since the target image is grayscale while the reference is color. One way is to train a gray VGG19 only

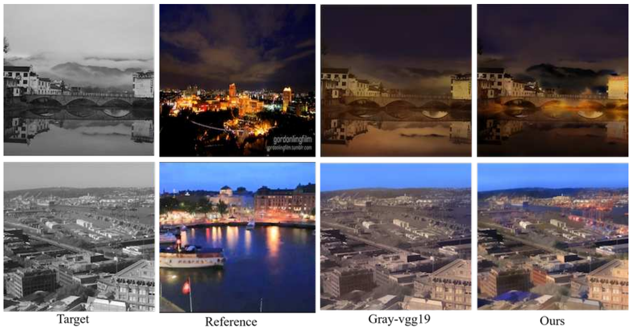


Figure 3. Intermediate outputs of transfer sub-net with a gray VGG19 extractor and our pre-colorization strategy, respectively.

using the luminance channel of an image [8]. Although the gray VGG19 achieves acceptable top-5 accuracy, it discards the chrominance information, which leads to poor effects on feature extracting. To ensure more reliable feature extraction, we still use the original color VGG-19 by utilizing a pre-trained colorization network to give the gray target image pre-color. Although the pre-color may not be very accurate, previous studies have observed obvious improvement in classification accuracy on recolored datasets compared with gray datasets [40]. We compare intermediate outputs of transfer sub-net of the two ways in Fig. 3. Our method obtains a more saturated result than gray VGG19 extractor, which will help the subsequent colorization sub-net refine the initial T_{ab}. In fact, it is unnecessary to train an extra automatic colorization model. We can reuse the trained colorization sub-net with fixed parameters, which will be detailed in the colorization sub-net.

3.2. Colorization sub-net

The T_{ab} obtained by transfer sub-net is inaccurate and has some artifacts, especially when given semantically unrelated reference. To refine it, we propose a colorization sub-net which takes known luminance T along with initial chrominance T_{ab} as input. However, it is not easy to train such a colorization sub-net to meet our requirements. On the one hand, there is no exact ground truth in exemplar-based colorization. On the other hand, we expect the sub-net can not only propagate "right" colors to "right" regions based on semantic features, but also refer meaningful colors of the objects when a reliable reference is unavailable.

Inspired by the idea of "peek", we take color images with randomly sampled ab channels as input and expect the network to learn their complete ab information. Such design can alleviate the learning pressure and make the network more likely to propagate known chrominance information to semantically related areas. Similar as the design strategy of [41], our colorization sub-net adopts an analogous U-Net [30] architecture which is formed by ten feature blocks and one output block. To avoid averaging problem, Huber loss L_h [10] is used to evaluate how close the output and the ground truth are,

$$L_c = L_h((1+\lambda M) \odot F_c(x), (1+\lambda M) \odot y) \qquad (2),$$

where F_c is the colorization sub-net, x and y represent the input and output respectively, M is the binary mask indicating the location of sampled ab channels, $\odot$ is the dot product. We pay more attention to locations with meaningful ab values by using a non-negative weighting parameter λ. The trained network can be used for automatic colorization by providing a gray image with empty ab map, which is used for pre-colorizing the input target image as mentioned before.

4. Experiments

This section describes the implementation details and presents a various of experimental results.

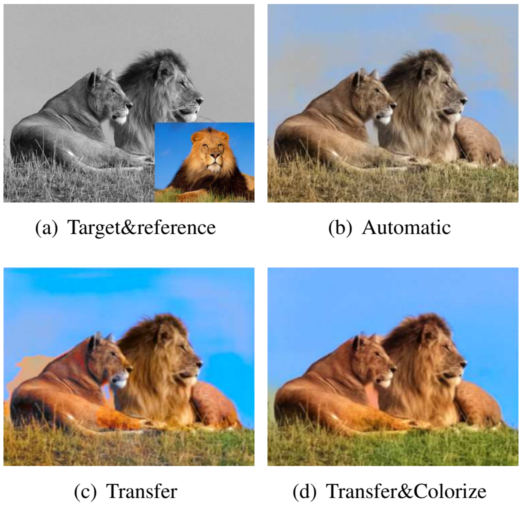
4.1. Implementation details

The decoder of transfer sub-net is trained on the Microsoft COCO dataset [22] by minimizing the sum of the L_2 reconstruction loss with weight 0.8 and perceptual loss [15] with weight 0.2. We use ADAM optimizer to minimize the loss, the learning rate of which is initially set to 0.0001 and then decreased by a factor of 2 every 5 epochs. Similar to state-of-the-art methods [6, 20], our transfer sub-net can achieve more accurate feature matching and color transferring by leveraging semantic label maps in AdaIN operation when they are available.

The colorization sub-net is trained on the ImageNet dataset. The weight coefficient λ is set to 10 in our experiments. For the first four blocks, we fine-tune the pre-trained weights from [40] to help training and the whole colorization sub-net is trained with ADAM optimizer, the learning rate of which is set to 0.00001 and then decreased by 10 after 10 epoch. In order to reuse the network for pre-colorizing the target image, we also train 5 epochs providing input images with empty ab map. Note that the coarse ab map obtained by transfer sub-net is randomly sampled to fit the trained colorization sub-net.

4.2. Ablation study

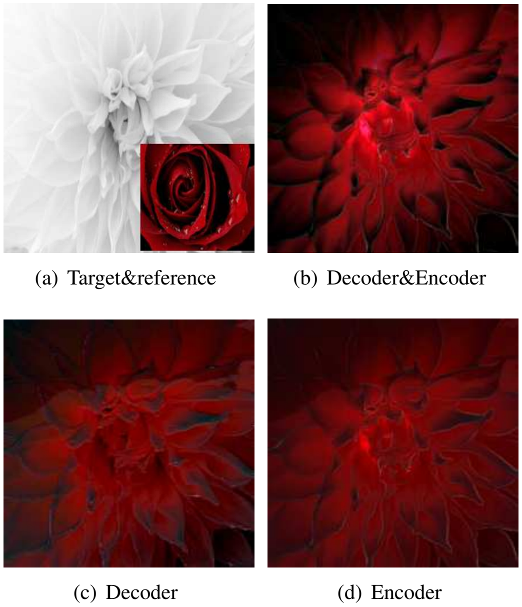
To illustrate the importance of the two sub-nets in our model, we show the colorization results generated by each single sub-net in Fig. 4. First we feed the target image with an empty ab map to the colorzation sub-net and obtain the pre-colorized result in Fig. 4(b). Since there is no initial color to be propagated, the sub-net only produces an undersaturated result. Then, we feed the pre-colorized target image along with the reference to the transfer sub-net and show coarse colorization result in Fig. 4(c). There appear obvious improper colorized patches. The final result is



(a) Target&reference (b) Automatic

(c) Transfer (d) Transfer&Colorize

Figure 4. Ablation study about our model architecture.



(a) Target&reference (b) Decoder&Encoder

(c) Decoder (d) Encoder

Figure 5. Results of AdaIN operations added in different parts.

shown in Fig. 4(d) by using the two sub-nets. It can be seen that the result via the entire networks has saturated color and few artifacts, which demonstrates the effectiveness of two cascaded sub-net architecture.

We further discuss the effect of AdaIN layer that added to the transfer sub-net modules. As we say, AdaIN layers are added after $conv\{i\}\_1$ of encoder and decoder net to match the features of reference and target images. The colorization result is shown in Fig. 5(b). We also show the result obtained by only using AdaIN in encoder or decoder net in Fig. 5(c) and (d), respectively. Among the three result images, Fig. 5(b) are more saturated and with less artifacts, which is sufficient to verify the importance that AdaIN has played in the feature matching and blending process.

4.3. Comparison with colorization methods

We compare our proposed method with five existing exemplar-based colorization methods [36, 26, 7, 8, 38] and present visual comparison and user study to evaluate these methods. The runtime is also compared since efficiency is one of the most important factors in the pratical application. To ensure a fair comparison, all results are obtained by running available codes that authors provide.

Visual comparison. We run all six models on 68 pairs of images collected from previous papers and show several representative results in Fig. 6. We can see that deep learning based methods, i.e., ours and [8, 38], yield perceptually more appealing colorization results than conventional methods, i.e., [36, 7, 26].

We further compare our method with [8] since we both take unrelated images pairs into consideration. First we show a challenging colorization case in Fig. 7. Most of the image objects are matched correctly with these two methods respectively, such as the mask of the man in the green box. Although the trees in the target image have no matched objects in the reference image, they are colorized reasonably. Comparing the two results carefully, it can be seen that some image details in result of [8] still remain gray while they are well colorized in our result (see e.g., the scarf in the yellow box and the sky in the black box).

Another comparison case is shown in Fig. 8, where we give five reference images to guide the colorization of a car. The first row of Fig. 8 shows five different reference images. Corresponding colorization results of [8] and the proposed method are shown in the second and third rows, respectively. When the references also contain cars (the first four ones), both the two methods can correctly match them in the two input images in spite of different colors or shapes. The color of our results looks more saturated and vivid. The last reference is a cock that is totally unrelated to the target. Our method can still predict reasonable color for the target image since the colorization sub-net is trained through large-scale datasets directly without reference images. While the color of the car in result of [8] looks unnatural because their network must borrow colors from the reference, even though there is no appropriate color.

User evaluation. Since the exemplar-based colorization is a highly subjective task, there is no metric that is applicable to measure the result. We thus conduct a user study to quantitatively evaluate our method against other methods. The benchmark dataset we use consists of 15 target-reference image pairs from previous papers and includes many kinds of image contents (i.e.,Landscape, buildings, human beings, animals). We invite 50 college students from different ages and majors to participate and ask them to choose the one which most closely matches the

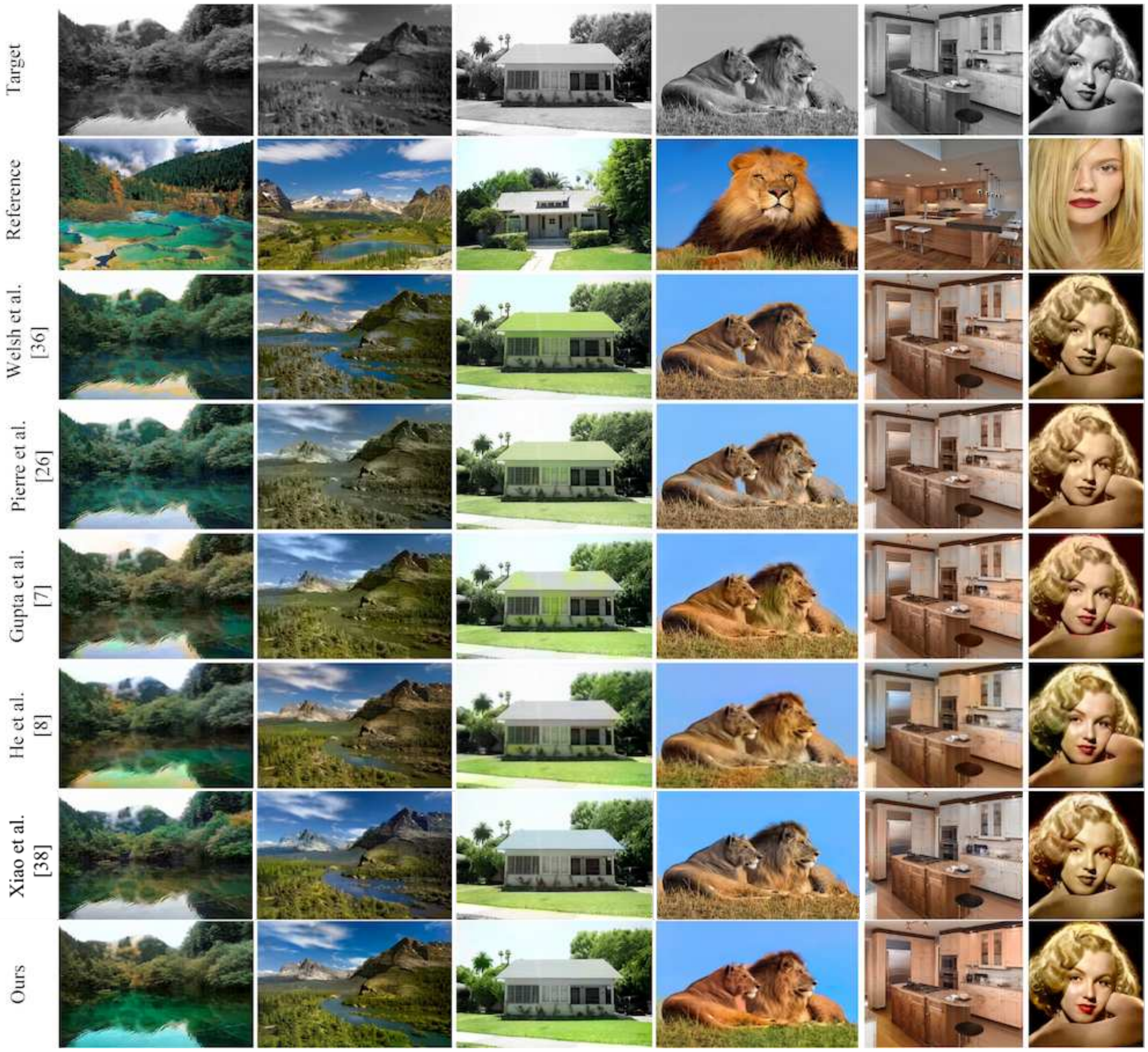
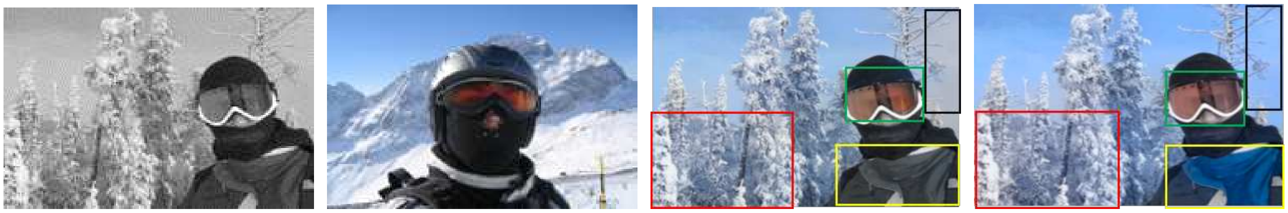


Figure 6. Comparison with example-based colorization methods. The first two rows are input target-reference image pairs. The last six rows are corresponding colorization results generated by [36, 26, 7, 8, 38] and the proposed method, respectively.



(a) Target (b) Reference (c) He *et al.* [8] (d) Ours

Figure 7. Detailed comparison between He *et al.* [8] and ours.

reference and has less artifacts from colorization results of [36, 26, 7, 8] and the proposed method. To avoid bias, five result images are shown in a random order. For each image pair and each method, we count the total number of user preferences (clicks), the statistic result of which is shown in Fig. 9. The highest clicks show that our colorization result is mostly preferred by the users.

Runtime comparison. Table 1 shows the runtime comparison with five aforementioned colorization methods [36, 7, 26, 8, 38]. The three conventional methods [36, 7, 26]

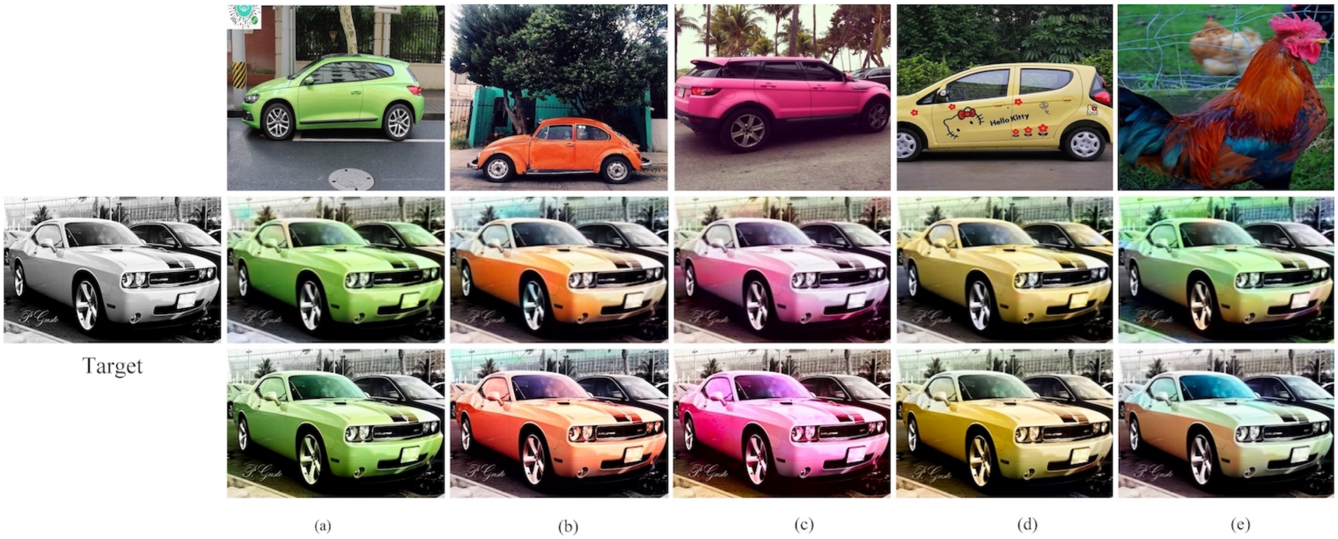


Figure 8. Results of colorization with different reference images. The first row shows five different references. The second and third row are their correspinding colorization results by He *et al.* [8] and the proposed method, respectively.

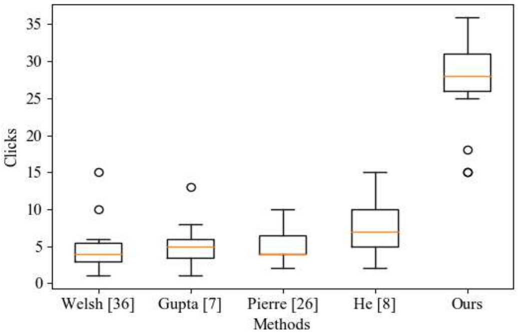


Figure 9. Boxplots of user preferences for different methods, showing the mean (yellow line), quartiles, and extremes (black lines) of the distributions.

are implemented using MATLAB 2015a on a laptop with 2.6GHZ Intel core i7-4720 CPU and 16G RAM. The code of [8, 38] and ours are implemented on a PC with an Intel E5 2.5GHz CPU and a single NVIDIA 1080Ti GPU. Welsh *et al.*'s method [36] is based on simple low-level feature matching, thus runs fastest among the three conventional methods. Gupta *et al.*'s method [7] costs lots of time on feature matching since features they used are more complicated and with high dimension. Due to the the use of Deep Image Analogy [21] (DIA), He *et al.*'s method [8] runs slowly and has comparable time consumption with Pierre's method [26]. Note that when the input image size is 1024×1024, He *et al.*'s method [8] gets Out Of Memory (OOM) exceptions. Pyramid structure in Xiao *et al.* [38] costs the most time. That is, we can achieve the fastest colorization with common configurations.

Table 1. Runtime comparison on colorization tasks.

Image Size	256 x 256	512 x 512	1024 x 1024
Welsh *et al.* [36]	3.51	13.52	53.14
Pierre *et al.* [26]	3.69	14.12	66.38
Gupta *et al.* [7]	114.51	446.47	1784.58
He *et al.* [8]	7.25+1.14	33.69+1.30	49.96+OMM
Xiao *et al.* [38]	0.48	1.4	3.96
Ours	0.04+0.08x2	0.14+0.12x2	0.59+0.28x2

4.4. Comparison with Photorealistic Image Stylization Methods

We also evaluate our colorization method on photo dataset used in photorealistic image stylization. Results of the proposed method as well as PhotoWCT [20] are shown in Fig. 10. The first two rows show the input image pairs and the last two rows show the result obtained by PhotoWCT [20] and our method, respectively. Note that PhotoWCT takes one content image and one style image as inputs for photorealistic stylization, while our network regards the gray version of the content image as the target to be colorized and style image as the reference, respectively. Although PhotoWCT exhibits good stylization effects, its results loss many image details due to the smoothing step. Our colorization results look more photorealistic, despite of inputting no chrominance information.

4.5. Comparison with Stylization Methods

Image stylization is the process of rendering a content image in the style of another image. There are some problems occuring in current state-of-the-art approaches. DSPT [24] is time consuming and its result always suffers from serious distortions. PhotoWCT [20] utilizes smooth operation to alleviate artifacts at the cost of unclear im-

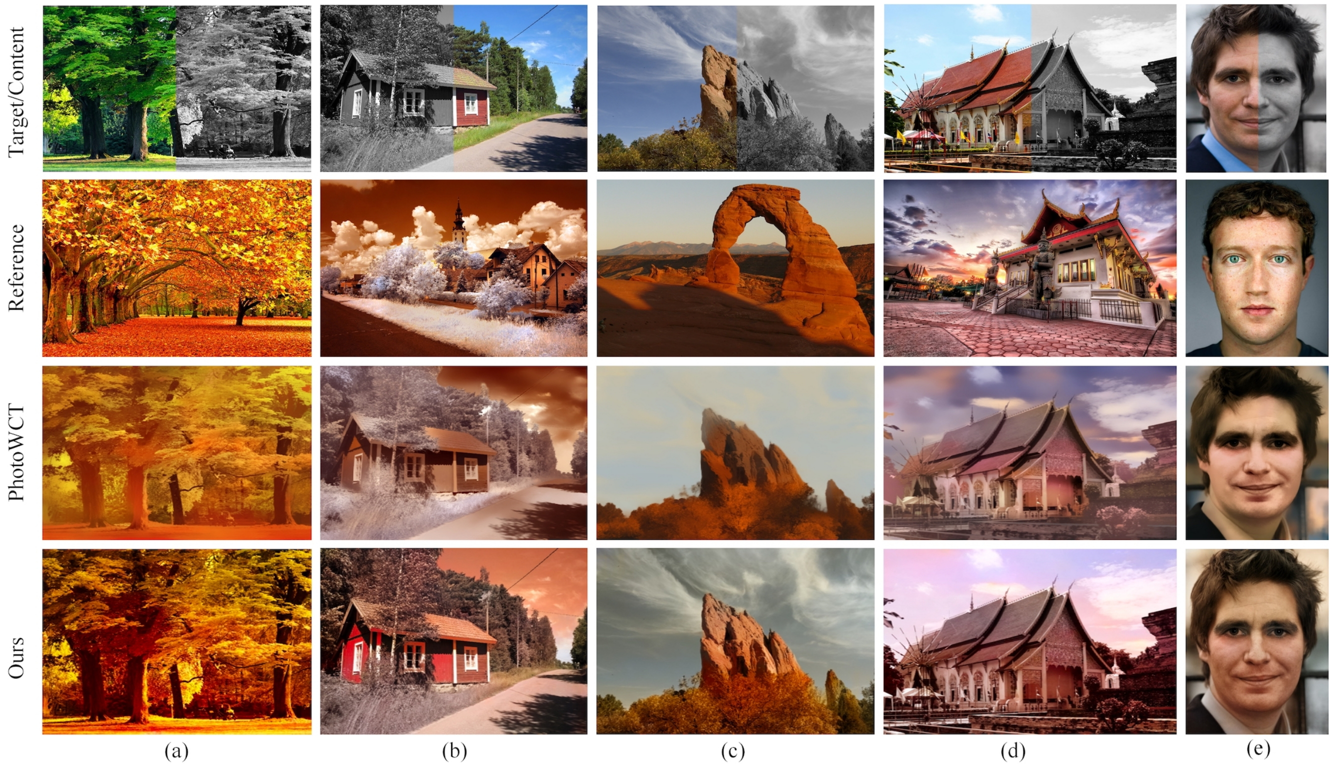
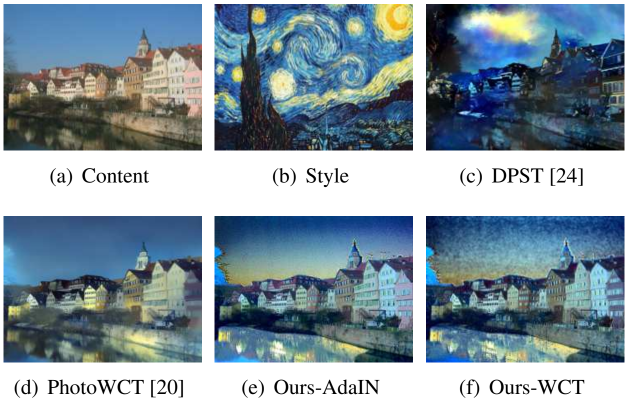


Figure 10. Comparison with PhotoWCT [20] on photo dataset used in photorealistic image stylization. The first two rows show the input image pairs and the last two rows show the result of the method, respectively.



(a) Content (b) Style (c) DPST [24]

(d) PhotoWCT [20] (e) Ours-AdaIN (f) Ours-WCT

Figure 11. Results of stylization methods [24, 20] and our transfer sub-net with WCT and AdaIN respectively.

age details. Our transfer sub-net is inspired from stylization methods and has the same ability. We compare it with DSPT [24] and PhotoWCT [20], the result of which is shown in Fig. 11. One can see that the result of DPST [24] has obvious artifacts. PhotoWCT [20] generates over-smooth result. The result obtained by our transfer sub-net has the best visual effect with no visible artifact. To show the superiority of AdaIN, we replace it with WCT in our transfer sub-net and show the result in Fig. 11(f). Compared with Fig. 11(e), it has noticeable noise.

5. Conclusions and Discussion

In this paper we present a fast stylization-based colorization architecture which consists of two sub-nets. The transfer sub-net aims at obtaining a coarse ab map for the target gray image by matching features of its pre-colorized version and the reference image. The colorization sub-net trained on large-scale image dataset is used to refine the coarse ab map as well as provide pre-color for the target image. Its unique design of inputting sampled ab map avoids the difficulty of building training dataset for exemplar-based colorization and make the network more likely to propagate known chrominance information to semantically related areas. Extensive experiments show that our method works well even given an unrelated reference in less time.

We believe that there's still plenty of room for improvement. For example, the AdaIN operation only aligns low-level feature statistics, i.e., mean and variance, which influences the matching accuracy. Besides, some colors in the coarse ab map are hardly propagated by the colorization sub-net. The reason may lies in that images in the training dataset are unevenly distributed. In the future, we will introduce higher level features for feature matching and explore more advanced network to solve these problems.

References

[1] Xiaobo An and Fabio Pellacini. Appprop: all-pairs appearance-space edit propagation. 27(3):40, 2008.

[2] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *IEEE ICCV*, pages 415–423, 2015.

[3] Alex Yong-Sang Chia, Shaojie Zhuo, Raj Kumar Gupta, Yu-Wing Tai, Siu-Yeung Cho, Ping Tan, and Stephen Lin. Semantic colorization with internet images. 30(6):156, 2011.

[4] Daniel Freedman and Pavel Kisilev. Object-to-object color transfer: Optimal flows and smsp transformations. In *IEEE CVPR*, pages 287–294. IEEE, 2010.

[5] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE CVPR*, pages 2414–2423, 2016.

[6] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *IEEE CVPR*, pages 3985–3993, 2017.

[7] Raj Kumar Gupta, Alex Yong-Sang Chia, Deepu Rajan, Ee Sin Ng, and Huang Zhiyong. Image colorization using similar images. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 369–378. ACM, 2012.

[8] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4):47, 2018.

[9] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE ICCV*, pages 1501–1510, 2017.

[10] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[11] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (TOG)*, 35(4):110, 2016.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[13] Revital Ironi, Daniel Cohen-Or, and Dani Lischinski. Colorization by example. In *Rendering Techniques*, pages 201–210. Citeseer, 2005.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE CVPR*, pages 1125–1134, 2017.

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *IEEE ECCV*, pages 694–711. Springer, 2016.

[16] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (TOG)*, 33(4):149, 2014.

[17] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *IEEE ECCV*, pages 577–593. Springer, 2016.

[18] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. 23(3):689–694, 2004.

[19] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in neural information processing systems*, pages 386–396, 2017.

[20] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *IEEE ECCV*, pages 453–468, 2018.

[21] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics (TOG)*, 36(4):1–15.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *IEEE ECCV*, pages 740–755. Springer, 2014.

[23] Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng. Intrinsic colorization. 27(5):152, 2008.

[24] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *IEEE CVPR*, pages 4990–4998, 2017.

[25] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320. Eurographics Association, 2007.

[26] Fabien Pierre, J-F Aujol, Aurélie Bugeau, Nicolas Papadakis, and V-T Ta. Luminance-chrominance model for image colorization. *SIAM Journal on Imaging Sciences*, 8(1):536–563, 2015.

[27] Francois Pitie, Anil C Kokaram, and Rozenn Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1434–1439. IEEE, 2005.

[28] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. 25(3):1214–1220, 2006.

[29] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[31] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE CVPR*, pages 5400–5409, 2017.

[32] YiChang Shih, Sylvain Paris, Connelly Barnes, William T Freeman, and Frédo Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics (TOG)*, 33(4):148, 2014.

[33] Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *IEEE CVPR*, volume 1, pages 747–754. IEEE, 2005.

[34] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[35] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *IEEE CVPR*, pages 6924–6932, 2017.

[36] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. 21(3):277–280, 2002.

[37] Fuzhang Wu, Weiming Dong, Yan Kong, Xing Mei, Jean-Claude Paul, and Xiaopeng Zhang. Content-based colour transfer. 32(1):190–203, 2013.

[38] Chufeng Xiao, Chu Han, Zhuming Zhang, Jing Qin, and Shengfeng He. Example based colourization via dense encoding pyramids. *Computer Graphics Forum*, (12), 2019.

[39] Liron Yatziv and Guillermo Sapiro. Fast image and video colorization using chrominance blending. *IEEE transactions on image processing*, 15(5):1120–1129, 2006.

[40] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666. Springer, 2016.

[41] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics*, 36(4):1–11.