HISTOGRAM-GUIDED SEMANTIC-AWARE COLORIZATION

Jie Zhang[a], Yi Xiao[a], Guo Chen[a], Qingping Sun[a], Fangqiang Xu[a], Chi-Sing Leung[b]*

[a]Hunan University, Changsha, Hunan Province, China
[b]City University of Hong Kong, Kowloon, Hong Kong SAR

ABSTRACT

User-guided colorization can predict the colors of a grayscale image according to user inputs, including exemplar images, local inputs and global inputs. Global inputs-based methods are probably the easiest ones to use, but are hard to distribute the input colors into correct regions, due to the lack of color-semantic correspondences. In this paper, we propose a novel histogram-guided semantic-aware colorization method, which explicitly builds the correspondences between global colors and local features with an attention mechanism and uses a differentiable histogram loss to impose the histogram of the results. Our method starts with a semantic-aware subnetwork to build the color-semantic correspondences, followed by a colorization subnetwork to reconstruct the color channels. Experiments demonstrate that our method can effectively control the results with the input histogram. Extensive visual, numerical and user study comparisons show that our method outperforms other global input-based state-of-the-art methods in color naturalness and consistency.

***Index Terms*—** Colorization, Histogram, Semantic-Aware, Deep Convolutional Neural Network

1. INTRODUCTION

The purpose of image colorization is to predict the color of each pixel of a grayscale image and preserve the semantic information. As a classic task in computer vision, colorization technology can generate images with high fidelity, so it has practical applications in many fields such as image compression, painting creation, black and white photo recovery, etc. However, colorization is known as a multi-modal problem with no unique solutions(e.g., the same kind of flowers may have diverse colors). The diversity can make deep learning based automatic colorization method [1, 2, 3, 4, 5] hard to satisfy user requirements.

Instead, user-guided colorization can generate colors of a grayscale image according to user inputs, such as exemplar images, local inputs (e.g. scribbles), and global inputs (e.g. histograms and color themes). Early exemplar-based methods [6, 7, 8, 9, 10] proposed to colorize a grayscale image by learning the color hints from one or several exemplar color images with similar semantics. Recently, He et al. [11, 12] propose to learn the semantic correspondences between the gray image and exemplar image by matching the VGG features [13], and then generate the color channels using deep neural networks. Instead of relying on only semantic correspondences, [14, 15] try to fuse both semantic colors and global color distribution in the exemplar image to accomplish colorization. Exemplar-based methods can produce plausible images with suitable images. Unfortunately, it's sometimes hard to find a suitable exemplar image with similar semantics. Moreover, pixel-level semantic matching is still a difficult problem to get accurate results.

Local inputs-based methods [16, 17, 18, 19, 20, 21, 22] rely on user inputs such as color points or scribbles to control the color of the entire image. The local inputs are propagated to the whole image, based on either traditional optimization method [16, 17, 18, 19] or deep learning [20, 21, 22]. These methods can accurately control the generated colors, but require a large number of inputs for images with complex textures.

Global input-based methods, driven by the power of deep neural networks, propose to use global information such as histogram [20], color theme/platte [21, 22, 14], or text [23] to control the results. Compared with other methods, global inputs-based methods are probably the easiest ones to use. For example, users can directly assign a histogram, or use the color statistics of an image which may does not share semantic similarity with the gray image. However, they are hard to distribute the input colors into correct regions, due to the lack of color-semantic correspondences.

Motivation: Our purpose is to keep the usability of global inputs and improve the performance of the global input-based colorization methods. Inspired by the attention mechanism [24] and the fact that histogram contains a lot of semantic information and can be differentiable to control the generated image [25, 26], we propose a novel histogram-based semantic-aware colorization framework. The key idea is to explicitly build the correspondences between global histogram colors and local features with the attention mechanism and uses a differentiable histogram loss to impose the histogram of the results.

The main contributions of our work are as follows:

1) We propose a novel histogram-guided colorization method, which consists of a semantic-aware subnetwork and a colorization subnetwork, supervised by a ground truth loss, a gradient loss and a differential histogram loss to enable high-fidelity colorization;

2) We propose a new histogram representation and an attention mechanism based semantic-aware subnetwork to build the color-semantic correspondences.

2. OUR METHOD

2.1. Overall framework with a novel input form

Given a grayscale image $T_L \in \mathbb{R}^{H\times W\times 1}$ (the L channel in CIE Lab space) and a reference ab-channel histogram $R_H \in \mathbb{R}^{B\times B\times 1}$, our histogram-guided colorization aims to find a function to predict the ab channels of the grayscale image in CIE Lab color space. Since the histogram only implicitly contains the color information, we propose to add a color palette $R_P \in \mathbb{R}^{B\times B\times 2}$ that explicitly indicates the ab values of each bins on the a-axis and b-axis, to ease the learning process. Therefore, the proposed representation of histogram $R_I =$

*Corresponding author: yixiao1984@gmail.com. The work is supported by the National Key R&D Program of China (2021YFF0900604), NSFC from PRC (61872137), the science and technology innovation Program of Hunan Province (2021RC3064) and Hunan NSF (2020JJ4009)

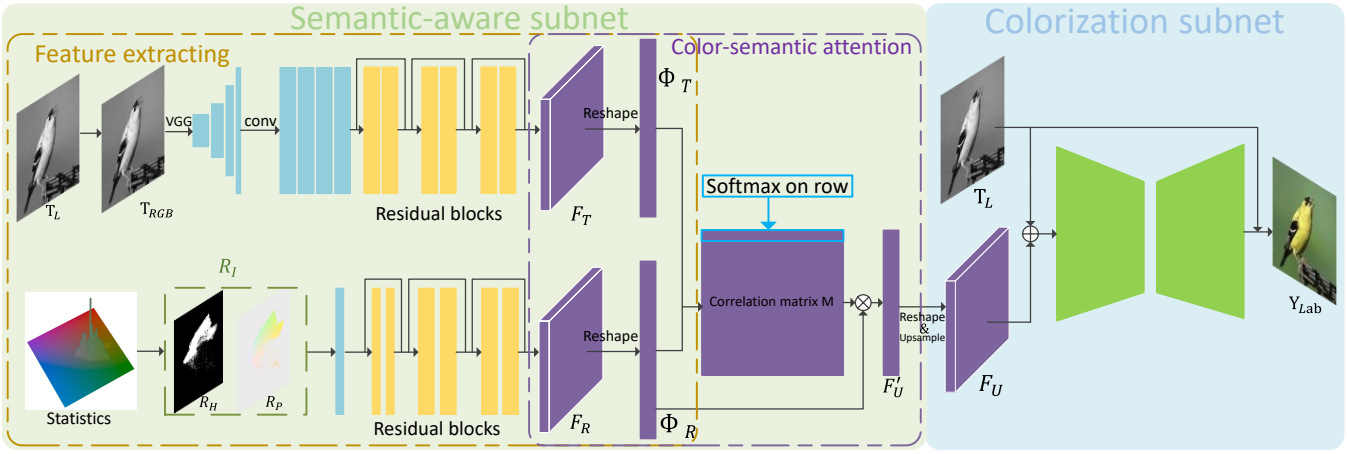


Fig. 1. The overview of our network model which consists of two subnets: a semantic-aware network and a colorization network. The notation $\oplus$ is concatenating operation and $\otimes$ is defined by Equation 2.

$[R_H, R_P]$ explicitly stores a color on the palette and the probability of this color. This is shown to improve the image quality in the ablation study (Table 1).

The pipeline for our histogram-guided colorization is shown in Fig. 1. The proposed architecture is a two-stage network, which consists of two subnets: a semantic-aware network $\mathcal{S}$ and a colorization network $\mathcal{C}$. In the first stage, $\mathcal{S}$ uses a feature extracting module to extract the features of the L channel T_L and the histogram representation R_I, and define an attention module to build the color-semantic correspondences, resulting in the semantic-aware features $F_U \in \mathbb{R}^{H\times W\times C}$. In the second stage, $\mathcal{C}$ uses T_L along with F_U to generate a colorized image $Y_{Lab} \in \mathbb{R}^{H\times W\times 3}$. Formally, we formulate the histogram-guided colorization for T_L as:

$$Y_{Lab} = \mathcal{C}(T_L, \mathcal{S}(T_L, R_I)). \tag{1}$$

2.2. Network architecture

As shown in Fig. 1, the semantic-aware subnetwork $\mathcal{S}$ is composed of a feature extracting module and a color-semantic attention module. The feature extracting module contains two branches. The first branch starts with a pretrained VGG [13] to extract the feature maps from layers of $relu2\_2$, $relu3\_2$, $relu4\_2$ and $relu5\_2$. Since VGG accepts images in RGB space, we convert the L channel T_L together with zero ab channels to T_{RGB} in the RGB color space. The VGG features are then passed to 4 groups of convolutional blocks containing 2 convolutional layers with 3×3 kernels to form a feature map $\in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$. We then use 3 residual blocks to generate the feature maps $F_T \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$, whose spatial resolution is set to $\frac{H}{4}\times\frac{W}{4}$ to save GPU memory. In the second branch, R_I are processed by a 64-channel convolutional layer with 3×3 kernels and 3 residual blocks, generating a feature map $F_R \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$.

Subsequently, the color-semantic attention module reshapes and normalizes F_T and F_R to Φ_T and $\Phi_R \in \mathbb{R}^{\frac{HW}{16}\times C}$ respectively to generate a correlation matrix $M \in \mathbb{R}^{\frac{HW}{16}\times\frac{HW}{16}}$ by the matrix multiplication of Φ_T and the transpose of Φ_R. The correlation matrix M learns how much each color is related to each feature. To extract the color-semantic mapping information for each location of the gray image, we propose to calculate the weighted sum of Φ_R,

$$F'_U(i) = \sum_j \underset{j}{\text{softmax}}(M(i,j)) \cdot \Phi_R(j,\cdot) \tag{2}$$

The softmax function acts on the row vector of M to apply different weights on Φ_R and then generate a semantic-aware feature map $F'_U \in \mathbb{R}^{\frac{HW}{16}\times C}$, which is then reshaped and linearly upsampled to $F_U \in \mathbb{R}^{H\times W\times C}$.

Finally, the colorization subnetwork $\mathcal{C}$ concatenates T_L and F_U to generate the colorized image Y_{Lab}. Since the U-Net structure has already shown promising results in colorization [20, 11], we also use a similar network structure as [20], except that we replace all the BatchNorm layers with InstanceNorm layers [27], and modify the the number of channels to fit our input.

2.3. Loss function

To maintain the local details of the gray images and meanwhile impose the input histograms on the colorized images, we define a loss function consisting of three components, given by

$$\mathcal{L} = \mathcal{L}_{GT} + \lambda_1 * \mathcal{L}_{Grad} + \lambda_2 * \mathcal{L}_H, \tag{3}$$

where $\mathcal{L}_{GT}$, $\mathcal{L}_{Grad}$, $\mathcal{L}_H$ are the ground truth loss, gradient loss and histogram loss, respectively. λ_1 and λ_2 are the parameters to balance the influences of three losses. In our experiment, we empirically set λ_1 and λ_2 to 1 and 10, respectively.

The ground truth loss is defined by Hubber loss as it generates results of relative high contrast as shown in [20, 21, 22]. The ground truth loss is given by

$$\mathcal{L}_{GT} = \sum_p \mathcal{L}_{Huber}(T_{ab}(p), Y_{ab}(p)) \tag{4}$$

where $\mathcal{L}_{Huber}(x,y) = \frac{1}{2}(x-y)^2$ for $|x-y| < \delta$; $\mathcal{L}_{Huber}(x,y) = \delta\,|x-y| - \frac{1}{2}\delta^2$ for else, and we set $\delta = 0.01$ in our experiments.

The gradient loss $\mathcal{L}_{Grad}$ is defined to maintain the local details and remove the color artifacts in the background, given by

$$\mathcal{L}_{Grad} = \sum_p \mathcal{L}_{Huber}(\text{Sobel}(T_{ab}(p)), \text{Sobel}(Y_{ab}(p))) \tag{5}$$

where Sobel denotes the Sobel gradient operator.

Finally, a differentiable histogram loss is defined to enforce that T_{ab} has a similar histogram T_H with the reference histogram R_H. It is given by

$$\mathcal{L}_H = \sum \mathcal{L}_{Huber}(T_H, R_H), \tag{6}$$

where T_H and R_H are the differentiable histograms calculated from the convolution layers [25].

3. EXPERIMENT

3.1. Experimental setting

Dataset: We generate a training dataset based on the ImageNet dataset [28] by removing all the grayscale images and images smaller than 3 KB from the original training split. We test our network on the widely used testing dataset ctest10k [1], which contains 10k images from the testing split of the ImageNet dataset. All the image pairs in our experiments are randomly sampled from the original testing split. To visualize the ab-space histograms, all the shown histograms use the average value of the target gray image as the L channel.

Implementation details: We implement our model with the PyTorch framework and train it using the Adam optimizer [29] on a machine with two NIVIDIA RTX2080Ti GPUs. We separate the training set into two parts: 10,000 images are randomly selected as a tiny set D_t, and the remaining part forms a large set D_l. To reduce training time, we set a default learning rate of 0.0001 to train our model for 100 and 3 epochs on the two training sets D_t and D_l, respectively, using a resolution of 128×128. Subsequently, we use a resolution of 256×256 to train our model on dataset D_l for 3 epochs with the default learning rate decays by 0.1 per epoch. Codes and models will be available at https://github.com/huluwaXYZ/histGuidedColorization.

Table 1. Ablation study.

Method	PSNR	SSIM
Baseline	28.9036	0.9373
Baseline + R_P	29.0697	0.9485
Baseline + semantic-aware + R_H	29.2282	0.9377
Ours	**29.7488**	**0.9516**

3.2. Ablation study

We conduct a numerical ablation study on the widely used testing dataset ctest10k [1] to validate the effectiveness of our model in Table 1. We define a baseline method, which removes the R_P and the color-semantic attention module, and directly feeds the union of T_L and features from R_H into the colorization network $\mathcal{C}$. We adopt two commonly used experimental measures, Peak-Signal-to-Noise-Ratio (PSNR) and Structural Similarity(SSIM), to evaluate the colorization quality. As we can see in the table, both the novel input form R_P and our color-semantic attention module can improve the PSNRs and SSIMs. Therefore, the numerical ablation study proves that both parts are necessary to generate high-quality images.

3.3. Comparisons

Numerical Evaluation. We compare our method with five automatic methods [2, 3, 1, 4, 5] and three global input-guided methods [20, 21, 22] by PSNR and SSIM, as shown in Table 2. All the images in our testing set are scaled to 256×256, and all the added inputs, such as histograms and palettes, are extracted from the ground truth images. Then we compute the average PSNR and SSIM over the whole testing dataset. As we can see in Table 2, the fully automatic colorization methods achieve very low PSNR and SSIM values. The reason for this phenomenon is that the multi-modal property of colorization make automatic colorization methods hard to restore diverse colors. The latter three global input based methods significantly increase image qualities by fusing global inputs into the image features. With our semantic-aware subnetwork, our approach achieves the best PSNR and SSIM among all these methods.

Table 2. Numerical comparisons.

Method	Added Inputs	PSNR	SSIM
Iizuka et al. [2]	automatic	22.938	0.884
Zhang et al. [3]	automatic	21.317	0.861
Larsson et al. [1]	automatic	24.295	0.898
Su et al. [4]	automatic	25.991	0.925
Xia et al. [5]	automatic	26.320	0.909
Zhang et al. [20]	+ global hist	28.207	0.947
Xiao et al. [21]	+ global palette	27.854	0.945
Xiao et al. [22]	+ global palette	28.179	0.947
Ours	+ global hist	**29.749**	**0.952**

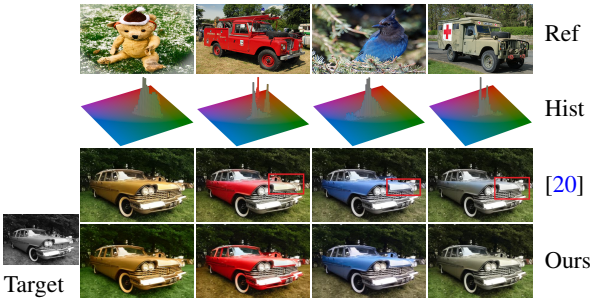
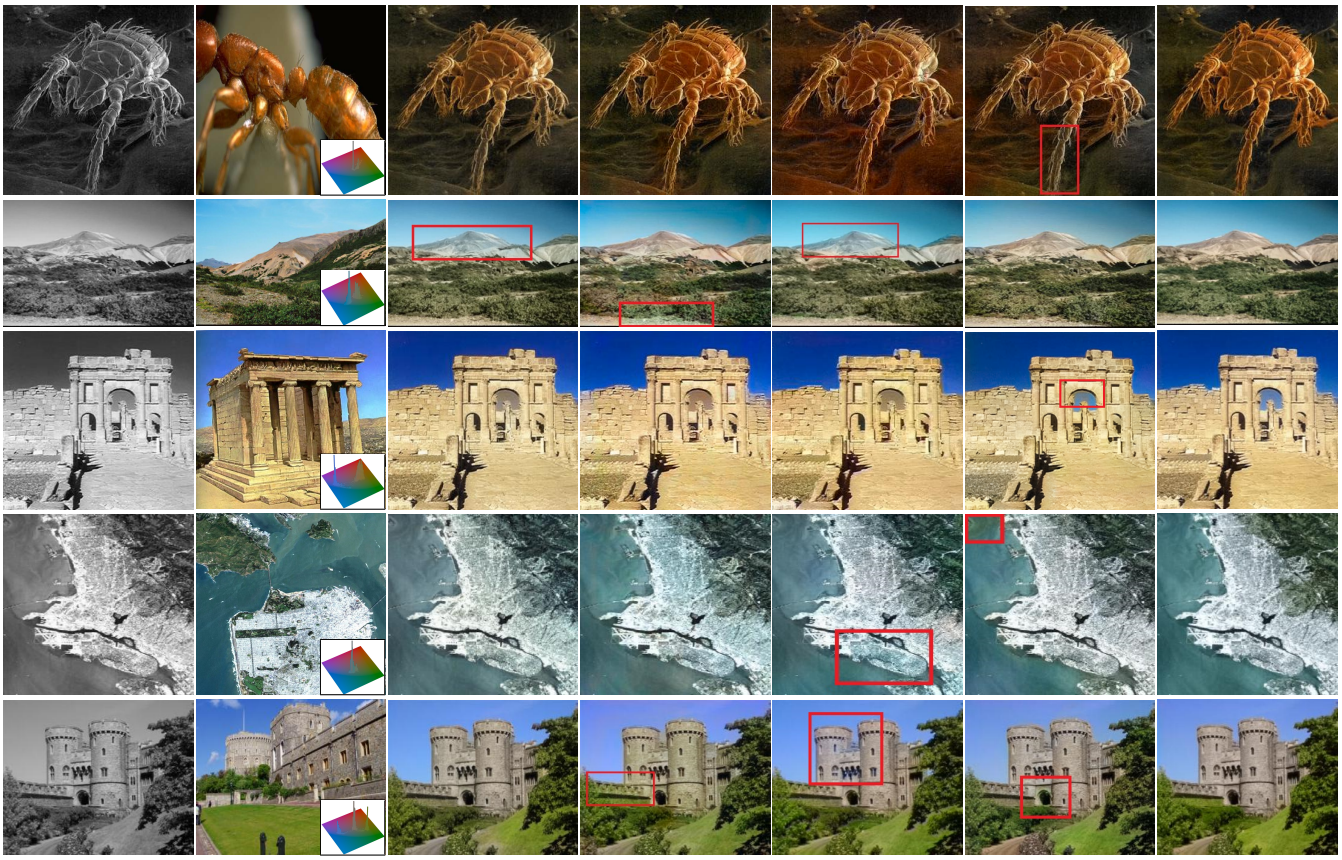


Fig. 2. Results of colorization with different histograms.

Visual Evaluation. We first show how different histograms in our method affect the generated images. We also show the results of [20], which also uses histograms as global inputs. As we can see in Fig. 2, the color styles of the generated images are more consistent with the input histograms, which verifies that we can faithfully use the histogram to control the results.

We further compare our method with a histogram-guided method [20] and three exemplar-based colorization methods [15, 14, 11] in Fig. 3. [20] takes a global histogram as additional input and directly fuses the features extracted from the histogram into the network. [15] takes histogram and reference image as input simultaneously, and [14] extracts both semantic and global features at different scales of the exemplar image. All the target images and exemplar images are from [11]. As shown in the figure, [20] tends to generate dim and unsaturated images (column (c) of Fig. 3). Except our method, all other methods suffer color bleeding artifacts, or inconsistent colors for the same kind of objects (see the regions highlighted by the red rectangles in Fig. 3). [20] fails to distribute colors in the histogram into image regions, due to the possible lack of color-semantic correspondences and histogram supervision. The exemplar-based methods [15, 14, 11] can make wrong semantic matches, even if the exemplars share similar semantics with the



(a) Target (b) Reference (c) [20] (d) [15] (e) [14] (f) [11] (g) Ours

Fig. 3. Visual comparisons with the start-of-the-art methods. All the target images and reference images are from [11]. Our method has remarkable improvements in visual quality. Please zoom in for the best view.

gray images, generating artifacts in certain regions. In contrast, our method builds color-semantic correspondences with the semantic-aware subnetwork. Therefore, our method can better reflect the color of the exemplar image and suppress color bleedings and artifacts.

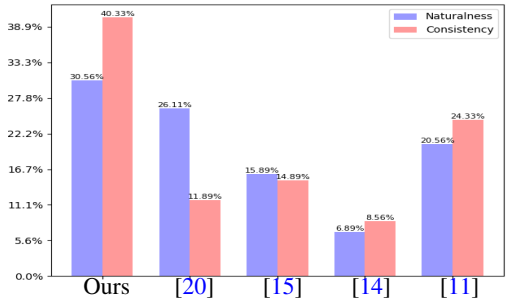


Fig. 4. Results of two user studies. Each bar indicates the percentage of participants that voted for this method. Our method achieves the best performance.

User Study. We further conduct two user studies for a comparison. We randomly select 30 pairs of target images and exemplar images from the testing split of the ImageNet dataset, and perform colorization with the proposed method and other methods [20, 15, 14, 11] to generate 30 groups of results. 30 volunteers are invited to answer two questions for images in each group: *Which image is more natural?* and *Which image is more consistent with the exemplar image in color distribution?* Results are reported in Fig. 4. In the first user study, our method achieves the highest click-through rate (30.56%), and its performance is close to that of Zhang et al. [20]. That is to say, both our method and [20] can generate plausible images. It is worth noting that in the second user study, our method significantly outperforms the other four methods, and gets the highest click-through rate (40.33%). The result reveals that our method can better transfer color information from the histogram. To summarize, the proposed method outperforms all other methods in the numerical comparison, visual comparison, and the user studies.

4. CONCLUSION

In this paper, we propose a novel semantic-aware network to improve the histogram-guided colorization. With the proposed histogram representation and the color-semantic attention module, our network can build color-semantic correspondences and distribute global colors into proper regions. Extensive experiments have shown the superiority of our method against other start-of-the-art methods. In the future, we would like to apply our idea to interactive image colorization and video colorization.

5. REFERENCES

[1] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich, "Learning representations for automatic colorization," in *European conference on computer vision*, 2016, pp. 577–593.

[2] 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 Trans. Graph.*, vol. 35, no. 4, July 2016.

[3] Richard Zhang, Phillip Isola, and Alexei A Efros, "Colorful image colorization," in *ECCV*, 2016, pp. 649–666.

[4] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang, "Instance-aware image colorization," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7965–7974.

[5] Jun Xia, Guanghua Tan, Yi Xiao, Fangqiang Xu, and Chi-Sing Leung, "Edge-aware multi-scale progressive colorization," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1655–1659.

[6] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller, "Transferring color to greyscale images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 277280, July 2002.

[7] Youngha Chang, Suguru Saito, Keiji Uchikawa, and Masayuki Nakajima, "Example-based color stylization of images," *ACM Trans. Appl. Percept.*, vol. 2, no. 3, pp. 322345, July 2005.

[8] Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng, "Intrinsic colorization," *ACM Transactions on Graphics (SIGGRAPH Asia 2008 issue)*, vol. 27, no. 5, pp. 152:1–152:9, December 2008.

[9] 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," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6, pp. 1–8, 2011.

[10] 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*, 2012, pp. 369–378.

[11] Mingming He, Dongdong Chen, Jing Liao, Pedro V. Sander, and Lu Yuan, "Deep exemplar-based colorization," *ACM Trans. Graph.*, vol. 37, no. 4, July 2018.

[12] Bo Zhang, Mingming He, Jing Liao, Pedro V. Sander, Lu Yuan, Amine Bermak, and Dong Chen, "Deep exemplar-based video colorization," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8044–8053.

[13] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] Chufeng Xiao, Chu Han, Zhuming Zhang, Jing Qin, Tien-Tsin Wong, Guoqiang Han, and Shengfeng He, "Example-based colourization via dense encoding pyramids," *Computer Graphics Forum*, vol. 39, no. 1, pp. 20–33, 2020.

[15] Peng Lu, Jinbei Yu, Xujun Peng, Zhaoran Zhao, and Xiaojie Wang, "Gray2colornet: Transfer more colors from reference image," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3210–3218.

[16] Anat Levin, Dani Lischinski, and Yair Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689694, Aug. 2004.

[17] Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, 2005, p. 351354.

[18] Liron Yatziv, Liron Yatziv, Guillermo Sapiro, and Guillermo Sapiro, "Fast image and video colorization using chrominance blending," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 15, pp. 2006, 2004.

[19] Michal Kawulok and Bogdan Smolka, "Competitive image colorization," in *2010 IEEE International Conference on Image Processing*, 2010, pp. 405–408.

[20] 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 Trans. Graph.*, vol. 36, no. 4, July 2017.

[21] Yi Xiao, Peiyao Zhou, Yan Zheng, and Chi-Sing Leung, "Interactive deep colorization using simultaneous global and local inputs," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 1887–1891.

[22] Yi Xiao, Jin Wu, Jie Zhang, Peiyao Zhou, Yan Zheng, Chi-Sing Leung, and Ladislav Kavan, "Interactive deep colorization and its application for image compression," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2020.

[23] Hyojin Bahng, Seungjoo Yoo, Wonwoong Cho, David Keetae Park, Ziming Wu, Xiaojuan Ma, and Jaegul Choo, "Coloring with words: Guiding image colorization through text-based palette generation," in *Computer Vision – ECCV 2018*, 2018, pp. 443–459.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[25] Zhe Wang, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang, "Learnable histogram: Statistical context features for deep neural networks," in *European Conference on Computer Vision*, 2016, pp. 246–262.

[26] Mahmoud Afifi, Marcus A. Brubaker, and Michael S. Brown, "Histogan: Controlling colors of gan-generated and real images via color histograms," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7941–7950.

[27] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv e-prints*, pp. arXiv–1607, 2016.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[29] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.