# ColorFormer: Image Colorization via Color Memory Assisted Hybrid-Attention Transformer

Xiaozhong Ji[1], Boyuan Jiang[1], Donghao Luo[1], Guangpin Tao[1], Wenqing Chu[1], Zhifeng Xie[2], Chengjie Wang[1(✉)], and Ying Tai[1(✉)]

[1] Youtu Lab, Tencent, Shanghai, China
{xiaozhongji,byronjiang,michaelluo,guangpintao,wenqingchu,jasoncjwang, yingtai}@tencent.com
[2] Shanghai University, Shanghai, China
zhifeng_xie@shu.edu.cn

**Abstract.** Automatic image colorization is a challenging task that attracts a lot of research interest. Previous methods employing deep neural networks have produced impressive results. However, these colorization images are still unsatisfactory and far from practical applications. The reason is that semantic consistency and color richness are two key elements ignored by existing methods. In this work, we propose an automatic image colorization method via color memory assisted hybrid-attention transformer, namely ColorFormer. Our network consists of a transformer-based encoder and a color memory decoder. The core module of the encoder is our proposed global-local hybrid attention operation, which improves the ability to capture global receptive field dependencies. With the strong power to model contextual semantic information of grayscale image in different scenes, our network can produce semantic-consistent colorization results. In decoder part, we design a color memory module which stores various semantic-color mapping for image-adaptive queries. The queried color priors are used as reference to help the decoder produce more vivid and diverse results. Experimental results show that our method can generate more realistic and semantically matched color images compared with state-of-the-art methods. Moreover, owing to the proposed end-to-end architecture, the inference speed reaches 40 FPS on a V100 GPU, which meets the real-time requirement.

**Keywords:** Automatic colorization · Vision transformer · Global and local attention · Memory network

X. Ji and B. Jiang—Equal contribution.

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19787-1_2.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13676, pp. 20–36, 2022.
https://doi.org/10.1007/978-3-031-19787-1_2

# 1 Introduction

Colorization is a challenging task since there are many possible color images for a grayscale image. Recent colorization approaches can be divided into two categories: *reference-based* and *automatic*. Reference-based colorization methods require user assistance or colorful reference images to reduce uncertainty. In this paper, we focus on automatic colorization task, which requires no additional reference and is therefore more widely applicable. With the development of deep learning, automatic colorization is simply modeled as a learning task. However, it is very challenging to achieve reasonable and natural colorization in a fully automatic setting.

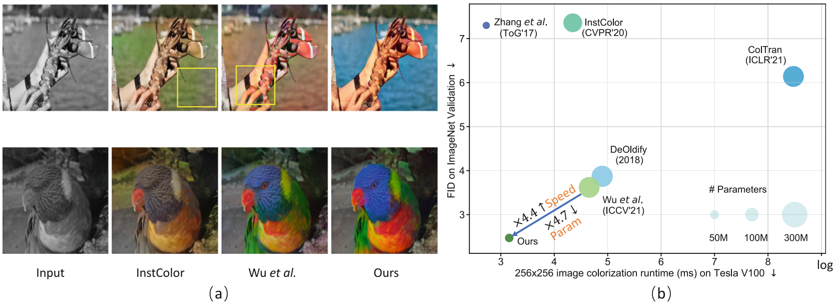

**Fig. 1. (a) Visual comparison.** The first row indicates InstColor [26] and Wu *et al.* [30] produce unreasonable colors in the water surface with inconsistent color and color-bleeding in the yellow rectangle areas. The second row shows our result is more vivid and colorful. **(b) Speed, parameters and FID comparison.** Our method can colorize images at 40 FPS with the best FID. (Color figure online)

Some classic methods [22,34] based on convolutional neural networks suffer from color confusion because of lacking effective semantic understanding. To locate and learn meaningful semantics, recent methods [26,28,36] combine other tasks (classification, detection, and segmentation) to enhance global or object-level semantic representation, but they still fail to build long-range visual dependencies, resulting in unreasonable colorization results such as green water surface with a lobster held by a person, as shown in the first row of Fig. 1(a).

Besides the limitations in semantic plausibility, these methods also fail to produce photo-realistic results. To improve the color richness, Wu *et al.* [30] combines image synthesis models and ref-based methods for automatic colorization, which is affected by the results of Generative Adversarial Network (GAN) inversion leading to a lack of vividness as shown in the second row of Fig. 1(a). Furthermore, these methods split colorization into multiple stages or branches, which affects the inference efficiency.

In summary, the automatic colorization methods mainly face two difficulties: 1) ***Semantic consistency***: The color of an object should be semantically consistent with itself as well as its environment. 2) ***Color richness***: The color of objects with different semantics should be diverse. To better address these challenges, we propose a novel colorization approach via hybrid attention and color memory, termed ColorFormer. ColorFormer is divided into two parts: transformer-based encoder to extract contextual semantic, and memory decoder for diverse color acquisition.

For the encoder part, we argue that capturing local and global visual dependencies through self-attention is crucial to reduce the uncertainty of semantic and produce natural results in different scenes. However, the global self-attention operation will bring challenges due to quadratic computation complexity. Therefore, we propose the Global-Local hybrid Multi-head Self-Attention (GL-MSA) operation to build a transformer-based encoder, which enjoys both efficient computation and global attention receptive field.

For the decoder part, we design a Color Memory (CM) module for semantic-level diverse color acquisition. Reference-based methods often produce better results than automatic due to the extra reference images or user guidance. However, searching for suitable reference images is time-consuming and difficult. Inspired by this, we propose a color memory module which stored multiple groups of semantic-color mapping. The decoder can adaptively query the semantic-related color priors from CM, used as reference information to help the decoder produce vivid results. As in Fig. 1, compared to state-of-the-art competitors, our network achieves more reasonable and colorful results, along with faster speed due to the more practical one-stage architecture design.

In general, our contributions can be summarized as follows:

- An effective transformer-based architecture for context-aware semantic extraction in automatic image colorization.
- A novel memory network for diverse color prior acquisition at semantic level.
- Comprehensive experiments demonstrate the effectiveness and efficiency of our method compared to state-of-the-art methods.

## 2 Related Work

Research on image colorization has developed rapidly in recent years, mainly due to the proposal of high-performance visual backbone network and the inspiration of semantic information for upstream vision tasks. In the following, we focus on the related work in reference-based and automatic colorization methods and briefly introduce existing progress in Vision Transformers.

**Reference-Based Colorization.** Reference-based colorization integrates the grayscale input with color knowledge transferred from a given reference. The earliest work [29] learns to transfer color by matching brightness and texture within the pixel's neighborhood, but the results are unsatisfactory due to the lack of spatial semantic consistency. To overcome this problem, recent works [12,31] employ deep neural networks to improve spatial correspondence and colorization results. These methods achieve remarkable results but are time-consuming and challenging for automatic retrieval systems [30]. Instead, we employ the color memory to automatically query color priors without reference images.

**Automatic Colorization.** Automatic colorization is inherently a highly ill-posed problem. The emergence of large-scale datasets makes it possible with deep learning in a data-driven manner. Cheng *et al.* [5] propose the first deep learning based image colorization method. Zhang *et al.* [34] learn the color distribution of every pixel. The network is trained with a multinomial cross-entropy loss with rebalanced rare classes allowing unusual colors to appear. Yoo *et al.* [32] present a memory-augmented colorization model along with threshold triplet loss that can produce high-quality colorization with limited data. Su *et al.* [26] propose instance-aware colorization, which leverages object detectors to crop and extract object-level features. Instance and full-image colorization share the same network, but are trained separately, and a fusion module is applied to predict the final colors. Wu *et al.* [30] propose to recover vivid colors by exploiting the rich and diverse color priors contained in pre-trained Generative Adversarial Networks (GANs). Specifically, matching features are first retrieved through a GAN encoder, and then incorporated into the colorization process through feature modulation. Different from these methods, we focus on integrating accurate semantic understanding and diverse color acquisition into a single network, which can produce reasonable and colorful results with a faster inference speed, and 72% improvement on FID as well, shown in the right part in Fig. 1.

**Vision Transformers.** Vision Transformers have achieved rapid development and are widely used in various high-and-low level tasks since Dosovitskiy *et al.* [8] successfully introduce Transformer from natural language processing to image recognition. Recently, Liu *et al.* [24] propose a hierarchical Transformer, namely Swin-Transformer, that capably serves as a general-purpose backbone for computer vision and achieves encouraging performance in many computer vision tasks. The representation is computed with shifted windows, which brings greater efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection. Kumar *et al.* [21] propose ColTran, which uses axial self-attention [15] to capture global receptive field attention to conditional produce a low-resolution coarse coloring of the grayscale image and then upsample the coarse colored low-resolution image into a finely colored high-resolution image. However, the inference speed of ColTran is quite slow. It takes about 4.5 s to colorize one image on a V100 GPU, which is unaccepted in real-time applications. Our work revisits the local window attention module by utilizing global information outside the local window, achieving ×180 faster than ColTran.

# 3 Method

## 3.1 Overview

Grayscale image colorization is to restore the missing $ab$ channel $x^{ab} \in \mathbb{R}^{H \times W \times 2}$ with only $l$ channel $x^{l} \in \mathbb{R}^{H \times W \times 1}$, where the $l, ab$ channels represent the lumi-

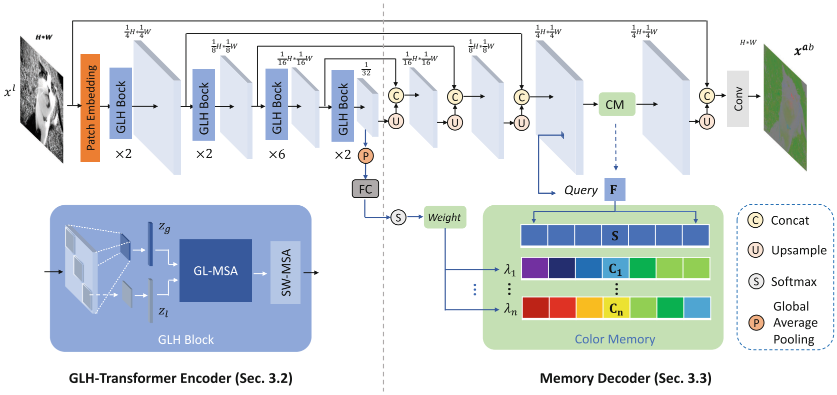
H*W
$x^l$
Patch Embedding
GLH Bock
×2
$\frac{1}{4}H*\frac{1}{4}W$
GLH Bock
×2
$\frac{1}{8}H*\frac{1}{8}W$
GLH Bock
×6
$\frac{1}{16}H*\frac{1}{16}W$
GLH Bock
×2
$\frac{1}{32}$
P
FC
S
Weight
C
U
$\frac{1}{16}H*\frac{1}{16}W$
$\frac{1}{8}H*\frac{1}{8}W$
$\frac{1}{4}H*\frac{1}{4}W$
CM
$\frac{1}{4}H*\frac{1}{4}W$
Query
F
Conv
H*W
$x^{ab}$
$z_g$
$z_l$
GL-MSA
SW-MSA
GLH Block
S
$\lambda_1$
$C_1$
$\lambda_n$
$C_n$
Color Memory
C Concat
U Upsample
S Softmax
P Global Average Pooling
GLH-Transformer Encoder (Sec. 3.2)
Memory Decoder (Sec. 3.3)

**Fig. 2. The framework of our approach.** Our network is divided into two parts: *GLH-Transformer Encoder* consisting of multiple GL-MSA modules to model contextual information from grayscale input, and *Memory Decoder* incorporated with CM module to generate colorful results.

nance and chrominance in CIELAB color space, respectively [2,16,23]. We construct an encoder-decoder network for automatic colorization. The encoder is to extract four hierarchical semantic information from the input gray image using stacked Global-Local Hybrid (GLH) Transformer blocks. Each block has the ability to capture global receptive field dependencies with the proposed Global-Local hybrid Multi-head Self-Attention (GL-MSA) module. The decoder consists of four upsampling stages with shortcuts from the corresponding stage of the encoder. Between the third and the last stage, we design a Color Memory (CM) module to introduce adaptive color priors providing relevant color reference for the decoder. As described in Fig. 2. The proposed model mainly consists of the GLH-Transformer encoder and the CM-incorporated decoder.

### 3.2 GLH-Transformer Encoder

Given a grayscale input image, the encoder first splits it into non-overlapping patches (tokens) and then a linear embedding layer is applied to project patch features to an arbitrary dimension (denoted as C). Here we use a patch size of 4 × 4 therefore the number of tokens is $\frac{H}{4} \times \frac{W}{4}$. After patch embedding, several GLH-Transformer blocks with global-local hybrid attention computation are applied on these patch tokens. To produce a hierarchical representation, we build four transformer stages. The number of tokens is reduced by a multiple of 2 × 2 (2× downsampling of spatial resolution) and the feature dimension is double by a patch merging layer between two stages. In the following we will detailly introduce how to design the GLH-Transformer block.

**GL-MSA.** The GL-MSA is designed for efficiently building long-range dependencies. As illustrated in Fig. 3(a), supposing the input feature map $\mathbf{z} \in \mathbb{R}^{h \times w \times c}$,

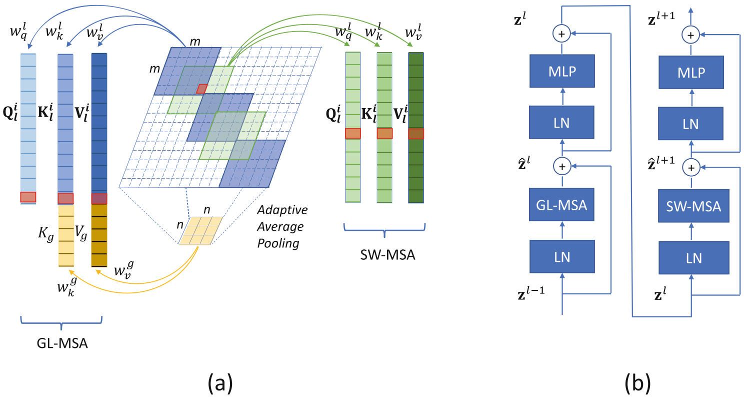

**Fig. 3. (a) Attention receptive field of GL-MSA and SW-MSA.** GL-MSA has global and local hybrid attention receptive field by fusing global and local information to key and value. **(b) Detailed architecture of the GLH-Transformer block.**

GL-MSA first divides $\mathbf{z}$ into non-overlapped $m \times m$ patches, producing the number of $L = \lceil \frac{h}{m} \rceil \times \lceil \frac{w}{m} \rceil$ patches. For patch $\mathbf{z}_l^i \in \mathbb{R}^{mm \times c}, i = 1 \ldots L$, we can obtain local query $\mathbf{Q}_l^i \in \mathbb{R}^{mm \times d}$, key $\mathbf{K}_l^i \in \mathbb{R}^{mm \times d}$ and value $\mathbf{V}_l^i \in \mathbb{R}^{mm \times d}$ with three projection layers. We then apply a $n \times n$ adaptive average pooling layer to the input feature map $\mathbf{z}$, resulting $\mathbf{z}_g \in \mathbb{R}^{nn \times c}$. $n \times n \ll h \times w$ and we set $n = 8, 4, 2, 1$ for four stages respectively. With $\mathbf{z}_g$ we can compute global key $\mathbf{K}_g \in \mathbb{R}^{nn \times d}$ and global value $\mathbf{V}_g \in \mathbb{R}^{nn \times d}$ with two projection layers. To compute global and local hybrid self-attention, we fuse local and global key and value by concatenation, which are formed as:

$$\begin{aligned}\mathbf{K}^i &= [\mathbf{K}_l^i, \mathbf{K}_g], \\ \mathbf{V}^i &= [\mathbf{V}_l^i, \mathbf{V}_g],\end{aligned} \tag{1}$$

where $\mathbf{K}^i \in \mathbb{R}^{(mm+nn) \times d}$ and $\mathbf{V}^i \in \mathbb{R}^{(mm+nn) \times d}$ are global and local hybrid key and value. We then calculate GL-MSA by:

$$\text{GL-MSA}(\mathbf{Q}_l^i, \mathbf{K}^i, \mathbf{V}^i) = \text{softmax}(\frac{\mathbf{Q}_l^i \mathbf{K}^{iT}}{\sqrt{d}})\mathbf{V}^i. \tag{2}$$

**GLH-Transformer Block.** Although GL-MSA is able to capture global and local hybrid dependencies, we experimentally find that equipped with Shift Window based Multi-head Self-Attention (SW-MSA) [24] for cross-window connection, our model can produce more semantic reasonable and consistent colorful images. As illustrated in Fig. 3(b), GLH-Transformer blocks are computed as:

$$\begin{aligned}\hat{\mathbf{z}}^l &= \text{GL-MSA}(\text{LN}(\mathbf{z}^{l-1})) + \mathbf{z}^{l-1}, \\ \mathbf{z}^l &= \text{MLP}(\text{LN}(\hat{\mathbf{z}}^l)) + \hat{\mathbf{z}}^l, \\ \hat{\mathbf{z}}^{l+1} &= \text{SW-MSA}(\text{LN}(\mathbf{z}^l)) + \mathbf{z}^l, \\ \mathbf{z}^{l+1} &= \text{MLP}(\text{LN}(\hat{\mathbf{z}}^{l+1})) + \hat{\mathbf{z}}^{l+1},\end{aligned} \tag{3}$$

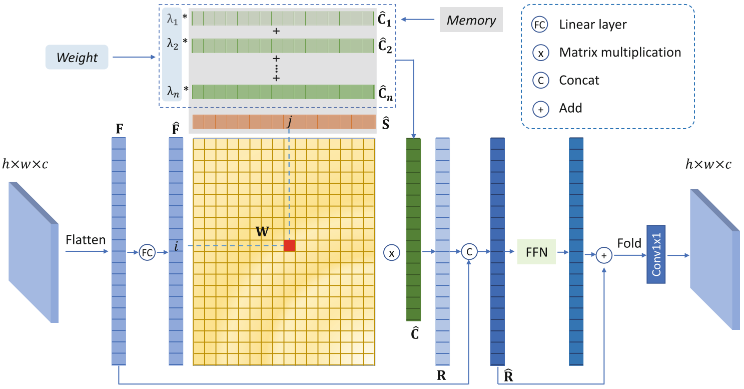

**Fig. 4. Detailed architecture of the CM module.** The input features $\hat{\mathbf{F}}$ query weighted *values* $\hat{\mathbf{C}}$ based on similarity to *keys* $\hat{\mathbf{S}}$.

where LN is layer normalization [3] and MLP is multi-layer perceptrons.

### 3.3 Memory Decoder

The memory decoder consists of four cascaded stages progressively enlarging the spatial resolution, where each stage is made up of an upsampling layer and a concatenation layer. In detail, the upsampling layer is implemented by convolution and pixel-shuffle, and the concatenation layer also combines a convolution to integrate the feature from the corresponding stage of the encoder by shortcut connections. Between the third and the last stage, the Color Memory (CM) module is applied to provide diverse color acquisition. We choose to calculate on feature maps of size $\frac{H}{4} \times \frac{W}{4}$ for the balance of computation complexity and representational capacity. At the end of the network are a residual block and a convolutional layer to get the final *ab* value. In the following part, we describe the detailed architectural design of the CM module.

**Architecture of CM Module.** For automatic image coloring, introducing various colors is the key to the diversity of the results. The CM module is used to provide the decoder with semantically matched color priors. We construct color memory to store two types of information: *keys* is the patch-level semantic representation $\mathbf{S} \in \mathbb{R}^{m \times k}$ of color images, and *values* store the corresponding color priors $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_n \in \mathbb{R}^{m \times 2}$, where $k$ is the dimension of semantic representation, $m$ is the number of semantic-color prior correspondence pairs, each color prior with two values (*i.e. ab* in CIELAB color space). We adopt multiple groups of color priors to enlarge the capacity of color memory and $n$ is the number of groups. We first describe the network structure here, and then describe how to obtain the *keys* and *values* in the next Subsect. 3.4. Due to the fact that one object may have different colors in different scenes, we set multiple color values for one semantic, the proportion of which is determined by global semantics. In our network, different color priors are fused together via the output weights from encoder. Note that Yoo *et al.* [32] proposes a memory-augmented colorization model, which stores the one-to-one mapping of the whole image-level spatial feature and color histogram for few-shot or zero-shot colorization. Different from this, our CM stores multiple groups of color prior and provide fine-grained guidance at feature-level. Next, we introduce the specific operation of the CM module illustrated in Fig. 4.

Given the input feature $\mathbf{F} \in \mathbb{R}^{h \times w \times c}$, we first flatten its spatial dimensions and use a linear function to transform it into $\hat{\mathbf{F}} \in \mathbb{R}^{hw \times d_1}$ as the *query*. Then, the semantic representation $\mathbf{S}$ is mapped to $\hat{\mathbf{S}} \in \mathbb{R}^{m \times d_1}$ as the *keys* to match the dimension of the *query*. The color priors $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_n$ are mapped to $\hat{\mathbf{C}}_1, \hat{\mathbf{C}}_2, \ldots, \hat{\mathbf{C}}_n \in \mathbb{R}^{m \times d_2}$ as the *values*, where $d_2$ is the dimension of color prior embedding. To integrate multiple color prior vectors, we propose an image-adaptive color priors fusion mechanism, which is based on the global semantic information of the input image. Specifically, we first apply a global average pooling layer, a linear layer, and a softmax layer to generate the fusion weights $\lambda$ at the end of the encoder, and then fuse color priors via the weights:

$$\hat{\mathbf{C}} = \sum_{l=1}^{n} \lambda_l \hat{\mathbf{C}}_l. \tag{4}$$

Then, we compute the attention weight between each query $\hat{\mathbf{F}}_i \in \mathbb{R}^{d_1}$ and each key $\hat{\mathbf{S}}_j \in \mathbb{R}^{d_1}$ and normalize them through a softmax layer along dimension $j$, which can be formulated as matrix multiplication:

$$\mathbf{W} = \text{softmax}(\hat{\mathbf{F}}\hat{\mathbf{S}}^T). \tag{5}$$

The weight $\mathbf{W} \in \mathbb{R}^{hw \times m}$ indicates the correspondence between query locations and stored semantic embeddings. The stored color prior is then transferred to query location according to the correspondence. Then we get the cross attention result $\mathbf{R} \in \mathbb{R}^{hw \times d_2}$, calculated as:

$$\mathbf{R} = \mathbf{W}\hat{\mathbf{C}}. \tag{6}$$

We then concatenate $\mathbf{R}$ with the input feature $\mathbf{F}$ at the channel dimension. The concatenated feature $\hat{\mathbf{R}} \in \mathbb{R}^{hw \times (c+d_2)}$ is regarded as the enhanced feature by color prior, which can help produce more diverse and vividly colorful images. To further enhance the generation ability of CM module, the concatenated result is then fed into an FFN [27] layer which consists of two linear transformations with a GELU [13] activation in between. Finally the output of CM is the addition of the output of the FFN layer and the original input together with a 1 × 1 convolution to the recover feature dimension to $c$:

$$\text{CM}(\mathbf{F}) = \text{Conv}_{1\times1}(\text{FFN}(\hat{\mathbf{R}}) + \hat{\mathbf{R}}). \tag{7}$$

### 3.4 Memory Build

In this subsection, we describe the detailed build process of the *keys* and *values* in the CM. To better provide semantic-aware colorization guidance, we propose to establish the correspondence between semantic representations (*i.e. keys*) and color priors (*i.e. values*) before network training. The construction process can be divided into two steps: semantic clustering and color clustering.

We first represent local regions of images as specific semantic embeddings and divide them into specific clusters. We randomly select $N = 10,000$ colorful images from ImageNet training set [7] to balance the semantic richness and computational complexity of clustering, and then use a pre-trained classification network (*e.g.*, GLH-Transformer) to extract semantic features. All input images are resized to a fixed size of 256, then we obtain feature maps of size $8 \times 8$ by the pre-trained network. In total, we collect $64N$ semantic features, each representing a local patch. In order to reduce the computational complexity of clustering while ensuring that the projected features are as dispersed as possible, we use the Principal Component Analysis (PCA) algorithm to reduce the feature space dimension to $k$. To represent these features sparsely, we use K-means clustering algorithm to divide them into $m$ categories, and regard the geometric center of each cluster as a semantic representation of image local information.

For regions with similar semantics, we also divide their multiple corresponding colors into $n$ categories. First, we scale the image to the same size as the feature map, and transform it from RGB space to CIELAB space. These divided local areas of the same semantic cluster have various $ab$ values which can be aggregated to form $n$ clusters. Similarly, the geometric centers are regarded as possible color candidates for the current semantic. Furthermore, the $n$ centers are ordered in a counter-clockwise order within the $ab$ plane to ensure all clusters are in the same order. Through the above two clustering steps, we establish a mapping relationship between the semantic embedding and the corresponding color priors, loaded by the CM as the *keys* and the *values*.

### 3.5 Objectives

During the training, we adopt three losses: *Content loss* to provide pixel-level supervision, *Perceptual loss* to align semantic feature, and *Adversarial loss* to improve authenticity.

**Content Loss.** The content loss is $L_1$ distance between the colorized image $\hat{y}$ and the ground-truth colorful image $y$:

$$\mathcal{L}_c = \|y - \hat{y}\|_1. \tag{8}$$

This loss encourages the generator to output similar color as the given images.

**Perceptual Loss.** To make generated images with better visual quality, we use pre-trained VGG16 network [25] to extract deep features of $\hat{y}$ and $y$ [18] and calculate the distance between them:

$$\mathcal{L}_p = \sum_{l=1}^{5} w_l \|\Phi_l(\hat{y}) - \Phi_l(y)\|_1, \tag{9}$$

where $\Phi_l(\cdot)$ denotes the layer *conv_l_1* of the VGG16 network, $w_l$ is the weight for the corresponding layer and set to $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, and 1.0, respectively.

**Adversarial Loss.** Our model is a GAN-based [9] network, where the generator G and the discriminator D are trained alternately. We use the popular PatchGAN discriminator [17], consisting of 4 convolutions with a stride of 2. The loss can be formulated as:

$$\begin{aligned} \mathcal{L}_d &= \|1 - \mathrm{D}(y)\|_1 + \|\mathrm{D}(\hat{y})\|_1, \\ \mathcal{L}_g &= \|1 - \mathrm{D}(\hat{y})\|_1. \end{aligned} \tag{10}$$

**Table 1. Quantitative results with SOTA methods on benchmark datasets.** $\Delta$CF is the absolute difference of CF between generated colorization images and ground-truth images. ↑ and ↓ mean higher or lower is desired.

| Method | ImageNet | | | | COCO-Stuff | | | | CelebA-HQ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FID↓ | CF↑ | $\Delta$CF↓ | PSNR↑ | FID↓ | CF↑ | $\Delta$CF↓ | PSNR↑ | FID↓ | CF↑ | $\Delta$CF↓ | PSNR↑ |
| CIC [34] | 19.17 | **43.92** | 4.83 | 20.86 | 27.88 | 33.84 | 3.01 | 22.73 | 14.97 | 38.21 | 4.54 | 24.54 |
| ChromaGAN [28] | 5.16 | 27.49 | 11.6 | 23.12 | 25.65 | 27.86 | 8.99 | 23.56 | 14.43 | **45.93** | 3.18 | 24.54 |
| ColTran [21] | 6.14 | 35.50 | 3.59 | 22.30 | 14.94 | 36.27 | 0.58 | 21.72 | 10.05 | 43.62 | 0.87 | 22.98 |
| Zhang *et al.* [35] | 7.30 | 27.23 | 11.86 | **24.13** | 17.43 | 25.95 | 10.9 | **24.66** | 11.81 | 36.98 | 5.77 | **26.25** |
| DeOldify [1] | 3.87 | 22.83 | 16.26 | 22.97 | 13.86 | 24.99 | 11.86 | 24.19 | 9.48 | 43.93 | 1.18 | 25.20 |
| InstColor [26] | 7.36 | 27.05 | 12.04 | 22.91 | 13.09 | 27.45 | 9.4 | 23.38 | 13.28 | 37.08 | 5.67 | 24.77 |
| Wu *et al.* [30] | 3.62 | 35.13 | 3.96 | 21.81 | – | – | – | – | – | – | – | – |
| Ours | **1.71** | 39.76 | **0.67** | 23.00 | **8.68** | **36.34** | **0.51** | 23.91 | **7.54** | 42.43 | **0.32** | 25.62 |

**Full Objectives.** Therefore the full objective for the generator is formed as:

$$\mathcal{L}(y, \hat{y}) = \lambda_c \mathcal{L}_c + \lambda_p \mathcal{L}_p + \lambda_g \mathcal{L}_g, \tag{11}$$

where $\lambda_c$, $\lambda_p$, and $\lambda_g$ represent weights for different terms, respectively.

# 4 Experiments

## 4.1 Datasets and Implementation Details

**Dataset.** We conduct experiments on datasets: ImageNet [7], COCO-Stuff [4] and CelebA-HQ [19]. We use the training part of ImageNet to train our method and evaluate it on the validation part. To show the generalization of our method, we also test on COCO-Stuff and CelebA-HQ validation sets without fine-tuning.

**Evaluation Metrics.** We mainly use Fréchet inception distance (FID) [14] and Colorfulness Score (CF) [10] to measure the performance of our method, where FID measures the distribution similarity between generated colorization images and ground truth color images, and CF reflects the vividness of generated colorization images. We also provide PSNR for reference. Although such pixel-wise measurements may not well reflect the actual performance [30].

**Implementation Details.** We train our network with Adam optimizer [20] and set $\beta_1 = 0.9$, $\beta_2 = 0.99$, weight-decay $= 0$ and initial learning rate $= 1e^{-4}$. For three loss terms, we set $\lambda_c = 0.1$, $\lambda_p = 5.0$ and $\lambda_g = 1.0$. For the GLH-Transformer encoder, we set the window size of GL-MSA and SW-MSA to 7. The feature dimension after four transformer stages is $96, 192, 384$, and 768, respectively. For color memory module, we set $m = 512$, $k = 64$, $n = 4$, $d_1 = 512$ and $d_2 = 256$. The network is trained for 200,000 iterations with batch size of 16 and the learning rate is decayed by 0.5 at 80,000, 120,000 and 160,000 iterations. The training and evaluation images are resized to $256 \times 256$. We conduct all experiments with 4 T V100 GPUs.

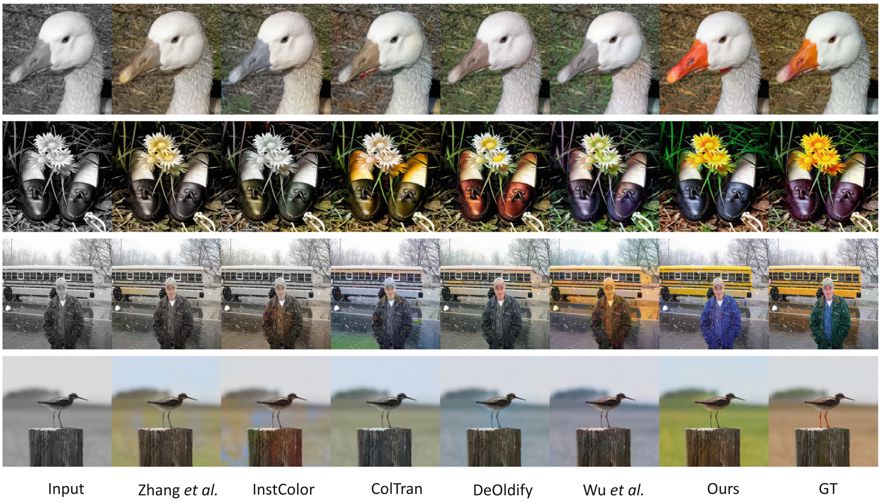

**Fig. 5. Visual comparisons with previous automatic colorization methods.** Our method is able to generate semantic consistent and color vivid images.

## 4.2 Comparison with State-of-the-Art Methods

**Quantitative Comparison.** We compare our method with previous methods on three datasets and list the quantitative results in Table 1. All competing methods use codes and model parameters provided by authors. Our method achieves the lowest FID on the ImageNet, indicating that our method could generate realistic and natural color images. On COCO-Stuff and CelebA-HQ datasets, our method also gains the lowest FID, demonstrating the generalization of our method. For the colorfulness score, some methods are higher than ours. However, as mentioned in [30,33], the higher CF may be because they encourage rare colors, leading to unreal colorization results, which are also reflected in their high FID. Therefore, we provide the absolute CF difference between the colorization images and the ground truth images. We exclude all grayscale images

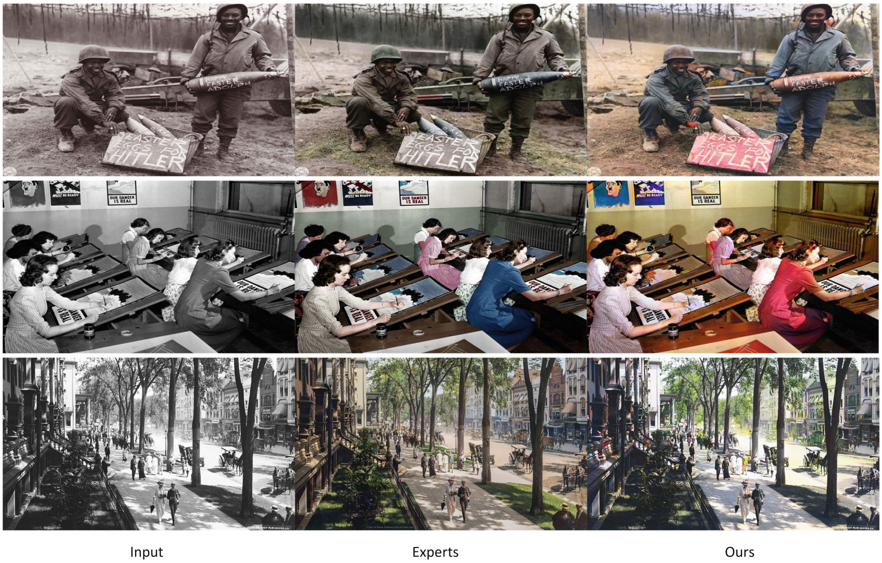

**Fig. 6.** Visual results on real-world black-and-white photos.

in the ground truth images to calculate ground truth CF, which is different from [30]. The lower $\Delta$CF indicates more precise colorization results, and we achieve the lowest $\Delta$CF on all three datasets.

**Qualitative Comparison.** We then visualize the grayscale image colorization results in Fig. 5. Here we display comparisons of images in different scenes from ImageNet validation dataset. Note that the GT images are provided for reference only but the evaluation criterion should not be color similarity. Overall, our results are more reasonable and vivid compared to other competitors. We can see that InstColor produces wrong colors due to miscalculating the semantics, as shown by the duckbill in the first row and the dog ear in the penultimate row. Zhang *et al.* and DeOldify tend to obtain results with uneven and unstable colors in surface like bus affected by luminance change in the grayscale input, while our results look more natural. ColTran and Wu *et al.* produce colorful results, but with unpleasant chromaticity appearing in blue or yellow. Instead, our method can generate semantic-consistent and vivid colorization in complex scenes such as shoes and flowers displayed together, and the bokeh grass.

**User Study.** We conduct a subjective user study to evaluate which colorization approach is preferred by human observers. We choose InstColor [26], ColTran [21], DeOldify [1] and Wu *et al.* [30] as competing methods for their low FID. We randomly select 50 images from the ImageNet validation set. For each participant, we show him/her five shuffled colorization images at one time and ask for the participant to choose the preferred one. We totally invite 20 volunteers to participate in the user study. The result is shown in Fig. 7 through boxplots. Our method is preferred by 35.16% of users, outperforming all other

**Table 2. Quantitative comparisons for ablation studies on the ImageNet dataset.** $CM_1$ means one group of color prior and $CM_4$ means four. $*$ indicates that the CM is initialized randomly without Memory Build.

| Encoder | Decoder | FID↓ | CF↑ | $\Delta$CF↓ | #Param. |
|---|---|---|---|---|---|
| ResNet50 | w/o CM | 4.10 | 34.27 | 4.82 | 44.5M |
| Swin-Trans. | w/o CM | 2.68 | 36.03 | 3.06 | 41.1M |
| Twins | w/o CM | 2.48 | 36.46 | 2.63 | 36.1M |
| GLH-Trans. | w/o CM | 1.85 | 36.70 | 2.39 | 43.4M |
| GLH-Trans. | w/$CM_1$ | **1.71** | 36.96 | 2.13 | 44.4M |
| GLH-Trans. | w/$CM_4^*$ | 1.80 | 38.25 | 0.84 | 44.8M |
| GLH-Trans. | w/$CM_4$ | **1.71** | **39.76** | **0.67** | 44.8M |

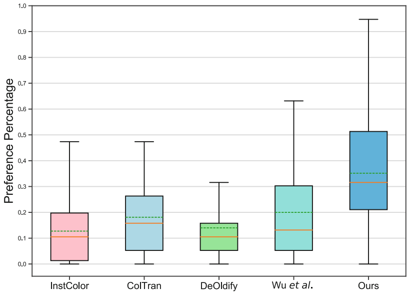

**Fig. 7. Boxplots of user study for five methods.** Green dash lines represent the mean preference percentage by users. Our method outperforms all other competing methods by a large margin. (Color figure online)

competing methods (InstColor 12.74%, ColTran 18.11%, DeOldify 14%, Wu *et al.* 20%), which is consistent with the FID score.

**Runtime and Model Parameters.** We illustrate speed, parameters and FID comparison among SOTA colorization methods in Fig. 1(b). Our method colorizes 256 × 256 gray images at 40 FPS with 44.8M model parameters, which is ×4.4 speed faster and ×4.7 parameters fewer than the previous SOTA method [30].

**Visual Results on Real-World Black-and-White Photos.** We collect some historical black-and-white pictures from a website[1] and compare our results with manually colorized ones by human experts, as shown in Fig. 6. The results demonstrate the practicality of our method.

### 4.3 Ablation Studies

To inspect the effectiveness of the proposed GLH-Transformer encoder and Color Memory decoder in the image colorization task, we conduct a series of ablation studies and list results in Table 2. We select ResNet50 [11] encoder as baseline model for its comparable parameters to our full model. We also adopt Twins [6] as our backbone, which also combines global and local attention.

**GLH-Transformer Encoder.** Semantic consistency of color is one key point to image colorization. Traditional CNNs are weak in building long-range dependencies. As shown in the left part of Fig. 8, with ResNet50 as encoder, some areas are not reasonable and semantic consistent, which is also reflected by its high FID. Transformer is notable for its use of attention to model long-range dependencies

[1] https://www.boredpanda.com/colorized-history-black-and-white-pictures-restored-in-color/.

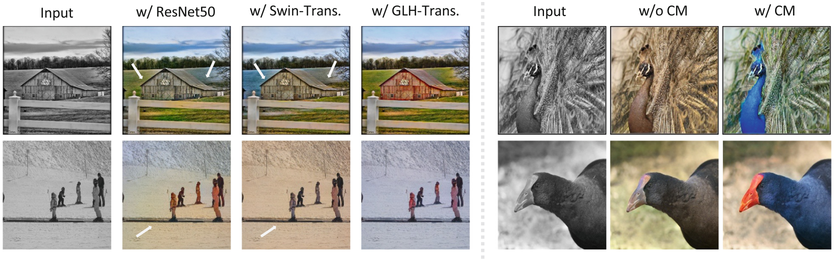

**Fig. 8. Illustrations of ablation studies.** GLH-Transformer helps produce semantic reasonable and consistent results, and CM leads to vivid results.

in the data. When replacing ResNet50 with Swin-Transformer in the baseline model, FID is reduced to 2.68 from 4.1. However, to reduce the computation complexity, Swin-Transformer calculates self-attention within each non-overlap local window, leading to the receptive field being relatively small in low-level features. Therefore the colorization results are still unreasonable in some areas. By introducing GLH-Transformer, with the help of building global and local hybrid dependencies for each window, FID is further reduced to 1.85, resulting in reasonable and natural visual results. Compared to the baseline model, our GLH-Transformer reduces FID by 55% with fewer parameters. We also compare our GLH with other backbone, Twins [6], and GLH achieves lower FID score. The keys and values of Global sub-sampled attention (GSA) in Twins only come from global features, while the keys and values of GL-MSA have both global and local patch features, which provide more effective feature fusion.

**Memory Decoder.** The color richness of colorization images is another key point. As in Table 2 and right part of Fig. 8, with CM, our method tends to generate color diversity and vivid images, and the colorfulness score improves from 36.7 to 39.76. We also explore the effectiveness of multiple groups of *values*. Compared to one group, multi groups of *values* can improve colorfulness by 7% with only 0.4M extra parameters. In addition, the CM performs better with pre-stored memory, although it can be trained from random scratch.

**Hyperparameters.** We conduct more ablation studies on $m$, $k$, $d_1$, $d_2$, and list quantitative results in Table 3 and Table 4. Increasing these hyperparameters can slightly improve performance, therefore we set them according to the principle of complexity balance.

**Table 3.** Quantitative results of CM under different $m$ **and** $k$**.**

| $m$ | 512 | 512 | 512 | 256 | 1024 |
|---|---|---|---|---|---|
| $k$ | 64 | 32 | 128 | 64 | 64 |
| FID↓ | 1.71 | 1.87 | 1.78 | 1.93 | **1.68** |
| CF↑ | 39.76 | 39.12 | 39.57 | 38.56 | **39.89** |

**Table 4.** Quantitative results of CM under different $d_1$ **and** $d_2$**.**

| $d_1$ | 512 | 256 | 768 | 512 | 512 |
|---|---|---|---|---|---|
| $d_2$ | 256 | 256 | 256 | 128 | 512 |
| FID↓ | 1.71 | 1.85 | 1.74 | 1.91 | **1.65** |
| CF↑ | 39.76 | 38.94 | 39.81 | 38.88 | **39.86** |

## 5 Conclusion

In this work, we design a colorization network based on hybrid attention and color memory to improve semantic consistency and color richness. On one hand, we propose the GL-MSA operation, suitable to capture long-range dependencies along with efficient computation. On the other hand, proposed color memory module introduces image-adaptive color priors for feature queries. The experimental results show that the model's accurate understanding of semantics and the introduction of more color priors help obtain more vivid results. What's more, instead of splitting the colorization into multiple steps, we verify that the end-to-end architecture can achieve better results while ensuring efficiency.

**Limitations.** When dealing with extreme low-quality old images with difficult scenes, our method may produce unreasonable artifacts or incoherent colors, which are also hard cases for recent works. Fortunately, this might be alleviated to some extent by performing image restoration beforehand.

## References

1. Antic, J.: A deep learning based project for colorizing and restoring old images (2018)
2. Anwar, S., Tahir, M., Li, C., Mian, A., Khan, F.S., Muzaffar, A.W.: Image colorization: a survey and dataset. arXiv preprint arXiv:2008.10774 (2020)
3. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
4. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff: thing and stuff classes in context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1209–1218 (2018)
5. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 415–423 (2015)
6. Chu, X., et al.: Twins: revisiting the design of spatial attention in vision transformers. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
7. Deng, J.: A large-scale hierarchical image database. In: Proceedings of IEEE Computer Vision and Pattern Recognition 2009 (2009)
8. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
9. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 2672–2680 (2014)

10. Hasler, D., Suesstrunk, S.E.: Measuring colorfulness in natural images. In: Human Vision and Electronic Imaging VIII, vol. 5007, pp. 87–95. International Society for Optics and Photonics (2003)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
12. He, M., Chen, D., Liao, J., Sander, P.V., Yuan, L.: Deep exemplar-based colorization. ACM Trans. Graph. (TOG) **37**(4), 1–16 (2018)
13. Hendrycks, D., Gimpel, K.: Gaussian error linear units (GELUs). arXiv preprint arXiv:1606.08415 (2016)
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
15. Ho, J., Kalchbrenner, N., Weissenborn, D., Salimans, T.: Axial attention in multidimensional transformers. arXiv preprint arXiv:1912.12180 (2019)
16. Huang, Y.C., Tung, Y.S., Chen, J.C., Wang, S.W., Wu, J.L.: An adaptive edge detection based colorization algorithm and its applications. In: Proceedings of the 13th Annual ACM International Conference on Multimedia, pp. 351–354 (2005)
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
18. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
19. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: International Conference on Learning Representations (2018)
20. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (Poster) (2015)
21. Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer. In: International Conference on Learning Representations (2021)
22. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 577–593. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_35
23. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: ACM SIGGRAPH 2004 Papers, pp. 689–694 (2004)
24. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Su, J.W., Chu, H.K., Huang, J.B.: Instance-aware image colorization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7968–7977 (2020)
27. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
28. Vitoria, P., Raad, L., Ballester, C.: ChromaGAN: adversarial picture colorization with semantic class distribution. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2445–2454 (2020)

29. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, pp. 277–280 (2002)
30. Wu, Y., Wang, X., Li, Y., Zhang, H., Zhao, X., Shan, Y.: Towards vivid and diverse image colorization with generative color prior. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14377–14386 (2021)
31. Xu, Z., Wang, T., Fang, F., Sheng, Y., Zhang, G.: Stylization-based architecture for fast deep exemplar colorization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9363–9372 (2020)
32. Yoo, S., Bahng, H., Chung, S., Lee, J., Chang, J., Choo, J.: Coloring with limited data: few-shot colorization via memory-augmented networks. IEEE (2019)
33. Zhang, B., et al.: Deep exemplar-based video colorization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8052–8061 (2019)
34. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_40
35. Zhang, R., et al.: Real-time user-guided image colorization with learned deep priors. ACM Trans. Graph. (TOG) **36**(4), 1–11 (2017)
36. Zhao, J., Liu, L., Snoek, C.G., Han, J., Shao, L.: Pixel-level semantics guided image colorization. arXiv preprint arXiv:1808.01597 (2018)