ColorFormer: A Novel Colorization Method Based on a Transformer

Hamza Shafiq[1], Truong Nguyen[2], and Bumshik Lee[1,*]

[1]: Hamza Shafiq and Bumshik Lee are affiliated with Chosun University, Department of ICE, 309 Pilmundaero, Gwangju, South Korea, 61452.

[2]: Truong Nguyen is affiliated with University of California, San Diego, Department of ECE, 9500 Gilman Dr, La Jolla, CA 9209.

[*]: Corresponding author (bslee@chosun.ac.kr)

Abstract

This paper introduces a novel grayscale image colorization method named ColorFormer, which leverages a transformer-based architecture enhanced with adversarial learning. Grayscale image colorization often suffers by problems such as inaccurate color placement, loss of fine details, poor color consistency across different regions, and the complexity of effectively capturing both local and global features. Conventional methods frequently produce images with artifacts, muted colors, and poor color consistency. ColorFormer addresses these challenges by introducing novel architecture with a lightweight multi-head self-attention mechanism within the ColorFormer Block. The proposed design not only enhances the colorization process but also reduces the complexity of the self-attention mechanism. In addition, we employ a conditional Wasserstein generative adversarial network (CWGAN) framework to ensure improved color accuracy, stable training, and superior visual quality of the generated colorized images. To further enhance the visual quality of the colorized images, we incorporate perceptual loss and adversarial loss during the training phase. Experimental results demonstrate that ColorFormer significantly outperforms other state-of-the-art colorization techniques, producing more realistic and vibrant colorized images.

Keywords

Colorization; Generative adversarial networks; Transformer

1. Introduction

Image colorization is used to restore colored images from black-and-white images. Colorization has many applications, including colorizing various legacy grayscale images from the past and cartoons [1], [2], restoring old photos [3], detecting fake color [4], and even assisting in classification and segmentation [5]. However, the color of an object can vary significantly; for example, the colors of leaves can be green or red. Therefore, colorization presents significant challenges due to the difficulty in accurately assigning colors based solely on intensity values. Effectively assigning the appropriate color to each object in an image remains an active area of research. The limitations of automatic image colorization may result in challenges such as color bleeding, misplacement of objects, and semantic ambiguity.

Multiple methods have been recently proposed to overcome the limitations of image colorization. These methods can be classified into two categories: user-guided colorization and automatic colorization. User-guided colorization requires user intervention to assign colors to the objects in an image, which is a labor-intensive task. In user-guided

colorization, the assigned colors depend on user selection, making this method less prone to errors. There are two types of user interaction: scribble-based and reference images. Scribble-based methods use user hints in the form of color scripts to colorize objects based on their actual colors. Example-based methods use reference-colored images similar to the input grayscale image and transfer the color to the input image. Both methods require human intervention to colorize the input images. In contrast, automatic colorization methods do not require user intervention and can generate a color image using a learning-based model. These methods learn the end-to-end mapping of grayscale to color images. Deep learning methods for colorization have recently become popular owing to their efficacy and the large number of publicly available datasets, such as ImageNet [6] and Places [7], which have 1.3 million and 1.8 million images, respectively.

Although automatic colorization methods achieve better results, the problems of unnatural colors, color bleeding, loss of fine details, and the inability to maintain color consistency across different regions persist. Semantic clues and segmentation information were added to the colorization network to address these problems [8]. Although these methods solve the problems to a certain extent, applying them to every situation was difficult. Moreover, the network complexity remained high.

To address these limitations, we propose a transformer-based image colorization network named ColorFormer. The proposed network is designed to overcome specific challenges inherent in automatic colorization methods, including the generation of unnatural colors, color bleeding, loss of fine details, poor color consistency, and the high complexity associated with capturing both local and global features.

The proposed ColorFormer is designed based on a combination of convolutional and transformer layers within ColorFormer Block to learn local and global information effectively. It is a conditional Wasserstein generative adversarial network (CWGAN)-based approach that combines the features of the Wasserstein GAN (WGAN) [9] and conditional GAN (CGAN) [10] to achieve better image colorization. The main advantage of the proposed ColorFormer method is that it captures local and global information. A lightweight multi-head self-attention (LW-MHSA) mechanism was implemented in the ColorFormer Block to enhance the colorization process and significantly reduce the complexity of self-attention. Moreover, a color feedforward network (CFFN) was used instead of the conventional feedforward network (FFN) to capture local information. The benefit of using a CFFN in the ColorFormer Block was its ability to effectively capture local features, such as edges, textures, or colors, using convolutional layers. CFFN preserves the spatial information in input images, which is crucial for image colorization. A comprehensive evaluation of the proposed method through extensive experiments demonstrates its superior performance compared with the state-of-the-art colorization techniques and highlights the potential of ColorFormer in image colorization.

The main contributions of our paper are summarized as follows:

- We propose ColorFormer, a novel colorization method based on the proposed transformer architecture, which leverages the advantages of LW-MHSA, CFFN, and CWGAN to address the drawbacks of traditional methods.
- The proposed ColorFormer Block integrates the convolutional layer with the transformer architecture, which enables the effective extraction of local and global information, resulting in high-quality and visually appealing

colorized images.

- LW-MHSA is proposed to enhance the colorization process while significantly reducing the complexity of self-attention in the ColorFormer Block.

2. Related Works

Early studies on colorization were mainly user-guided methods. The user provides hints to colorize pixels, which can be in the form of scribble-based or example-based approaches. In scribble-based methods, the user provides a high-level scribble. The colors are then propagated based on low-level similarity matrices. For example, an early method assigned colors similar to pixels with the same luminance [11]. This is a labor-intensive task that requires accurate scribbles for good colorization. In [12], colorization using an edge detection technique was proposed. Luminance-weighted chrominance bleeding was proposed in [13] for fast colorization. Zhang et al. [14] proposed a method that uses an additional deep prior from a convolutional neural network (CNN) to ensure colorization without scribbles. Moreover, AdaColViT [15] is proposed to reduce the complexity of the transformer and employ user-guided pruning of redundant patches and layers. This approach ensures real-time interaction for colorization. The iColoriT [16], is another colorization Vision Transformer network, that optimally utilizes Transformers' global receptive field to propagate user hints, achieving real-time colorization with minimal input. However, these methods may result in color bleeding because pixels with similar intensities produce similar colors.

In example-based methods, a color image is provided as a reference for colorizing a grayscale image. In [17], luminance values were extracted and matched with grayscale images to transfer colors. In [18], a reference source image was segmented, and color information from the segmented image was used as a scribble in [11] to transfer colors. Moreover, an automatic reference image retrieval method was proposed to reduce the effort required to select a reference image [19]. However, these methods are highly dependent on the reference image and provide unnatural results if the semantics of the reference image does not match those of the input grayscale image. In [20] and [21], colorization methods were proposed to match the semantics of the reference image to the target image. In addition, the authors used different types of references, such as words [22], [23] and sentences [24]. Although these methods have improved over the years, they still require user interference, and their results depend on the provided information.

Fully automatic methods generally use deep learning-based structures to learn semantic information for colorizing grayscale images. In [25], the use of a CNN for color images in a fully automatic manner was first attempted, wherein patches were used to colorize images with a simple model architecture. A class-rebalancing scheme to resolve the inherent ambiguity and multimodal nature of colorizing grayscale images was proposed in [14], in which a visual geometry group (VGG) network was adopted to colorize images. In [26], a network was jointly trained for classification and colorization using a labeled dataset. The VGG network was used to augment the input grayscale images in [27] and pass through CNN networks. However, these methods still exhibit color bleeding and semantic confusion. In [8], [28], additional semantic information was used to resolve these problems, and promising results were achieved by generating more contextually accurate colorizations by leveraging semantic information. However, despite their achievements, these approaches still have limitations, such as color bleeding and unsaturated results,

which lead to less visually appealing colorized images. Another disadvantage is the reliance on semantic information, which may not always be available or accurately estimated, thereby limiting the applicability of these methods.

Recently, GANs [29] have become popular. These generative models facilitate multimodal colorization. An image-to-image translation model was proposed using a CGAN in [10], wherein a U-Net-based generator was used; this resulted in more vivid colorized images owing to adversarial training. The model was generalized to high-resolution images in [30]. The input noise was sampled at various times to obtain diverse colorization [31]. The grayscale images were mapped to a Gaussian mixture model using a mixture density network [32]. In [33], class distribution was additionally used in the WGAN model. In [34], more focus was placed on colorization using generative priors, which provided the spatial structures of the generated image. Zhao et al. [35] proposed a saliency map-guided GAN that leverages saliency information to prioritize accurate colorization of salient regions. Du et al. [36] refined the GAN architecture with a double-channel guided approach, employing distinct channels for color and structure information, enhancing colorization fidelity. Wu et al. [37] introduced a generative color prior, trained on a vast collection of colorized images to achieve vivid and diverse colorization outcomes. Furthermore, [38], based on the CGAN architecture, featuring a novel loss function, a multi-scale discriminator, and a channel and spatial attention mechanism. Additionally, Liu et al. [39] proposed a PatchGAN-based image colorization model incorporating CBAM. Liu et al. show that CBAM can help the model focus on important regions of the image, leading to improved performance on benchmark datasets. Moreover, image colorization using color-features is proposed, which uses GAN based architecture and learnable color features to colorize images [40].

Transformers [41] have received considerable attention in the field of computer vision. The transformer architecture was first introduced by Vaswani et al. [41]. Later, a new architecture for image classification was proposed using transformers called vision transformers (ViTs) [42]. The ViT architecture divides the input image into a grid of patches, which are subsequently processed by the transformer network to perform classification tasks. In addition to image classification, transformers have been applied to other image-processing tasks, such as object detection, segmentation, image super-resolution, denoising, and colorization.

In [43], Swin Transformer, a hierarchical vision transformer that uses shifted windows to achieve long-range dependencies without sacrificing efficiency, was proposed. Furthermore, ColTran [44] is an early example of using transformers in image colorization, which coarsely colorizes grayscale images using a conditional autoregressive transformer, followed by two parallel networks for upsampling, resulting in finely colored high-resolution images. Transformers have persistently been employed in image colorization. CT2 [45], [46] represents another approach that uses transformers, encoding colors as tokens and guiding interactions between grayscale image patches and color tokens through color attention and query modules. DDColor [46] introduces a dual decoder GAN architecture for image colorization, where the initial decoder generates a coarse colorization while the secondary decoder refines this output by incorporating semantic details.

Recently, a diffusion model [47] have emerged as a powerful tool for image generation and colorization. The diffusion models use a probabilistic approach to generate images by iteratively denoising a sample from a Gaussian distribution.

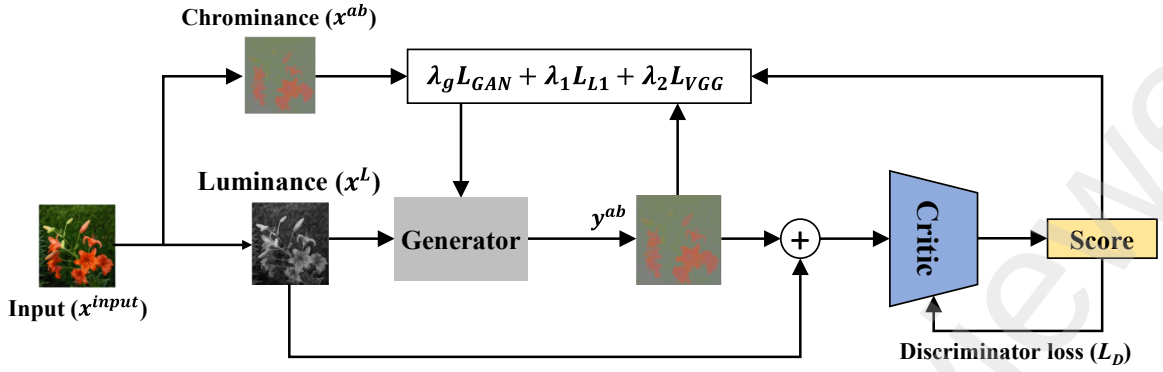


Fig. 1. Overall GAN architecture of the proposed ColorFormer network.

In [48], [49], diffusion models for image colorization were proposed, where the model learns to add and then remove noise to generate realistic colorizations. These models have shown promise in producing high-quality colorized images with less color bleeding and more accurate color representations. However, diffusion models often require a large amount of computational resources and extensive training times, making them less practical for real-time applications. Additionally, their performance can be susceptible to the choice of noise schedule and the quality of the training data, leading to potential limitations in generalizability and robustness.

The current CNN-based and transformer-based colorization networks have several limitations, including color bleeding, unsaturated results, and difficulties in capturing local and global features. To address these problems, we propose ColorFormer, a novel method that combines a CWGAN framework with transformers and convolutional layers to improve color consistency and stable training. Unlike conventional techniques, LW-MHSA is further introduced to enhance colorization while reducing the complexity of the self-attention mechanism. The notable feature of reduced model complexity, achieved through the window mechanism, sets our approach apart in efficiently handling long-range dependencies. The ColorFormer Block is designed to integrate local and global information within the network while preserving its overall complexity. Our approach achieves superior and visually pleasing colorized images and outperforms state-of-the-art methods.

3. Proposed Method

In the proposed method, the CIELAB [50] (also called Lab) color space was used to represent all colors visible to the human eye, wherein L^* and a^*, b^* represent the luminance (brightness) and chrominance (a*: red to green; b*: blue to yellow) channels, respectively. Given a grayscale image $x^g \in \mathbb{R}^{H\times W\times 1}$, the goal is to predict the two color channels a and b, where $x^{Lab} \in \mathbb{R}^{H\times W\times 3}$ is the original color image. The use of the CIELAB color space provides more precise control over the colorization process and a more accurate representation of the color. Fig. 1 shows the overall architecture of the proposed ColorFormer colorization system. As shown in Fig. 1, given an input image $x^{input} \in \mathbb{R}^{H\times W\times 3}$, the image is first converted to the CIELAB color space and split into L and ab channels. L channel image $x^L \in \mathbb{R}^{H\times W\times 1}$ is passed through the generator, and the generator outputs ab channel image $y^{ab} \in \mathbb{R}^{H\times W\times 2}$ that has the color information. This image y^{ab} is passed through the discriminator along with a real image x^{ab}, and the discriminator evaluates the quality of the generated image. The model is trained in an adversarial manner.

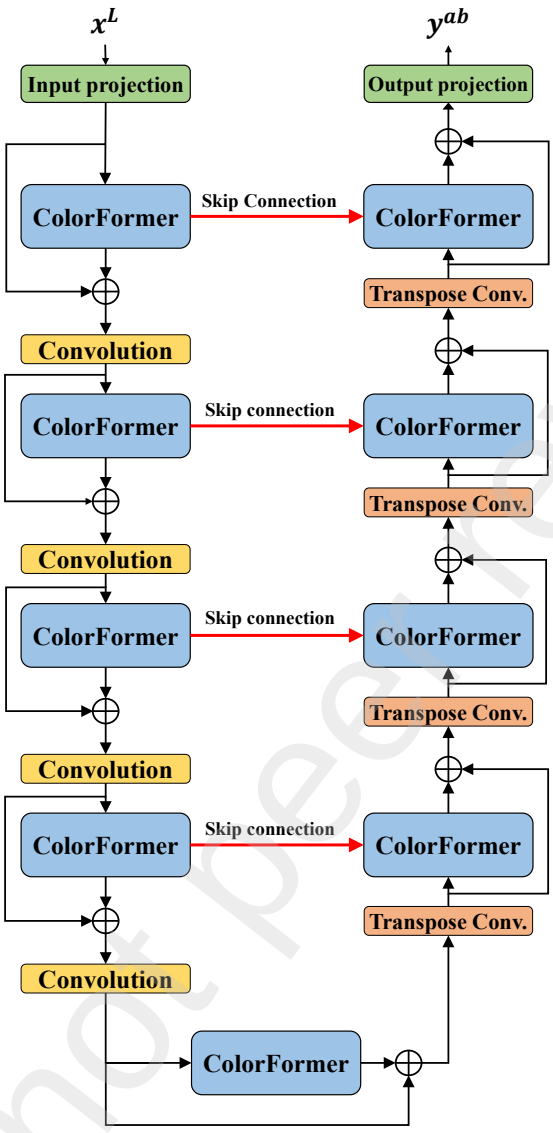


Fig. 2. Generator architecture in the proposed colorization method.

The ColorFormer structure is based on the GAN architecture, which uses transformer layers in the generator for natural and diverse colorization. The network consists of a generator and a discriminator. The generator is based on ColorFormer Blocks, which consist of LW-MHSA, a CFFN, a normalization layer, and convolutional layers. The discriminator is based on the Markovian discriminator architecture (PatchGAN) [10], which focuses on capturing high-frequency components using local patches.

The GAN architecture with the ColorFormer network was designed based on two key concepts: the CGAN architecture and the WGAN, also called the CWGAN. The CWGAN was used to improve the colorization accuracy and stability. The CGAN incorporates additional information such as grayscale images, resulting in more accurate and visually appealing colorization results. However, the WGAN addresses some common limitations of traditional

GANs, such as instability and mode collapse. Overall, the CWGAN produces more realistic and high-quality colorizations by leveraging the strengths of the conditional GAN and WGAN. Instead of the conventional adversarial loss, we used the Earth mover distance-based objective function (Wasserstein distance) [9] for the GAN, which improved the overall stability and convergence of the model. The overall architecture of our proposed method is based on the CWGAN. GAN architecture is directly trained as CWGAN. In this section, we introduce the proposed generator and critic architectures and present the objective function for training the proposed GAN.

3.1. Proposed Generator Architecture

The overall generator architecture is divided into three parts: the encoder, decoder, and skip connections. The encoder extracts important features from the input image, understanding its unique properties. The decoder then uses these features to reconstruct the image with color details. Skip connections help smoothly pass important information from the encoder to the decoder, making sure the model creates colorful and visually appealing images effectively. The generator architecture is based on transformer layers that use a window mechanism to reduce the complexity. The generator consists of ColorFormer Blocks and convolutional layers. Convolutional layers are used to down-sample the image features. First, the input image x^L is passed through the input projection layer, which comprises convolution and activation. A rectified linear unit [51] is used as the activation function for the input projection layer. The image is then flattened and passed through the ColorFormer layer, which consists of depth-wise convolution (DWC), layer normalization (LN), LW-MHSA, and a CFFN. DWC is used to extract local information while reducing complexity, which is very helpful for colorization because the colors of neighboring pixels are dependent on each other.

A window mechanism is used to reduce the complexity of the model. In LW-MHSA, heads are split between shifted windows. In an LW-MHSA layer with a cyclic shift size of 1, the input can be divided into two window configurations: one using the original window and the other using a cyclically shifted window. This allows N/2 attention heads to use the original window and the remaining N/2 heads to use the shifted window. Since the shifted window returns to its original position after one shift, this approach enables the model to capture both local and global information. The final output of the self-attention layer is obtained by adding the results, as shown in Fig. 4. This process can reduce the network complexity by improving long-range dependencies. Finally, the input of the ColorFormer Block is added to the output using the residual connection to compensate for missing information.

Convolutional layers are employed between the ColorFormer Blocks for image downsampling and upsampling. Additionally, the decoder is responsible for the retrieval of spatial information, while utilizing skip connections from the encoder to enhance spatial information recovery. The architectural configuration of the overall generator closely resembles that of U-Net. [52]. The output of the generator is a two-channel image $Y_G \in \mathbb{R}^{H \times W \times 2}$ with the color information alone. This image is then passed on to the discriminator along with x^L as a condition to predict the score.

3.1.1. Proposed ColorFormer

The ColorFormer Block is proposed to address the challenges of transformer regarding computational complexity while capturing global and local contextual information. Self-attention in ViTs captures long-range dependencies.

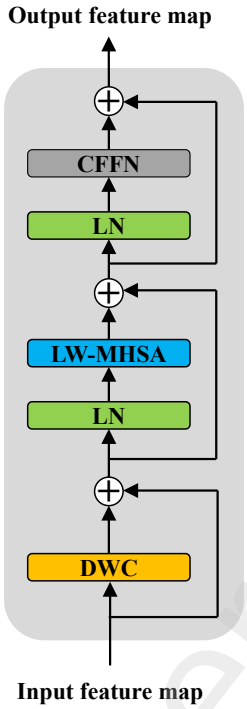


Fig. 3. Proposed ColorFormer Block.

The standard transformer calculates the relationship between each token and all other tokens in the feature map. Since this operation has quadratic computational complexity, it is unsuitable for applying global self-attention to high-resolution feature maps. Moreover, although colorization requires local and global information, transformers are more biased toward long-range dependencies because the color of a pixel in an image can depend on its local context (such as the colors of adjacent pixels) and its global context (such as the overall color distribution of the image).

To address these challenges, we proposed a ColorFormer Block designed to capture local and global information while minimizing computational complexity. The proposed ColorFormer Block includes self-attention to capture global information and convolutions for local contextual information. Fig. 3 shows the architecture of the proposed ColorFormer Block. The ColorFormer Block consists of three components: (1) DWC, (2) LW-MHSA, and (3) a CFFN.

The input is passed through the DWC, which is used to capture local information while reducing the complexity. The DWC output is added elementwise to the input features and then passed to the LN before self-attention. The output from the LN is passed through the LW-MHSA, and then element-wise addition is performed with the input features for the LN. Finally, the features are passed through an LN step before being processed by CFFN. The output of the CFFN is then added element-wise to the input features. Overall, this ColorFormer Block is designed to process the input features hierarchically, where each layer processes an intermediate output from the previous layer to capture

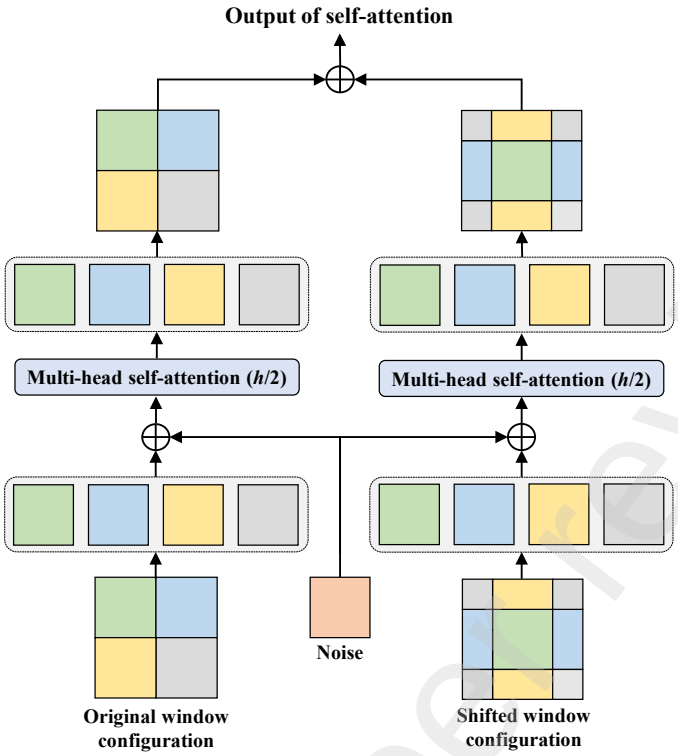


Fig. 4. Proposed window mechanism in LW-MHSA, where h is the number of head.

increasingly complex representations of the input data. The DWC, LW-MHSA, and CFFN allow the network to capture local and global information in the input features. The overall process can be expressed by (1), (2), and (3).

$$x^{l-1} = DWC(x^{l-1}) + x^{l-1}, \tag{1}$$

$$x^{l} = LWMHSA(LN(x^{l-1})) + x^{l-1}, \tag{2}$$

$$x^{l} = CFFN(LN(x^{l})) + x^{l}, \tag{3}$$

where x is the input feature map to the ColorFormer Block, DWC denotes the depth-wise convolution layer, LWMHSA denotes the lightweight multi-head self-attention, LN denotes the layer normalization, and CFFN denotes the color feedforward network network block. A more detailed explanation of LW-MHSA and CFFN is given below.

a) **LW-MHSA**

Given a feature map X, we split it into non-overlapping patches of size $M \times M$,

$$X = \{X^1, X^2, ..., X^n\} \tag{4}$$

where n is the number of patches. We then split each patch into windows of size W × W and flattened the windows to pass through the self-attention block. To capture dependencies outside the window, we used a shifted-window mechanism. The windows are shifted in a cyclic manner and split according to the number of heads of self-attention.

If the shift size is 1, that is, the window returns to the same position after the 2nd shift, then there are two window configurations. The N/2 heads of self-attention were given one window configuration and another with a different window configuration. Fig. 4 shows the window mechanism of the proposed LW-MHSA. Half of the heads are given one window configuration and the other half a shifted window, and both are added to obtain the final output of self-attention. The overall process of the proposed self-attention algorithm is defined in (5) and (6).

$$\{X_s^n, X_{s+1}^n, \ldots, X_{s+w}^n\} \tag{5}$$

where X_s^n is the window configuration for the n-th patch, and w is the number of shifted window configurations for the *n*-th patch.

$$Y_j^i = ATN_k\left(X_s^i W_j^Q, X_s^i W_j^K, X_s^i W_j^V\right) + ATN_k\left(X_{s+1}^i W_j^Q, X_{s+1}^i W_j^K, X_{s+1}^i W_j^V\right) \tag{6}$$

$$i, j \in \{1, 2, \ldots, N\}$$

where Y_j^i is the output of LW-MHSA for the *i*-th patch; $W^Q, W^K, and\ W^V$ are the projection matrices of the query, key and value for the single head, respectively. The k (= $floor(N/(w+1))$) is the number of attention heads that can use a single window configuration, which is equal to the total number of attention heads (N) divided by the number of windows (w+1) that can be created from the input sequence. That is, if the number of windows (w+1) is 2 and the number of heads (N) is 16, then k will be 8 (8 heads for each window configuration). s+1 is the shifted window and ATN_k is the function of the attention module for k number of heads.

$$\hat{X}_j = \{Y_j^1, Y_j^2, \ldots, Y_j^n\} \tag{7}$$

where $\hat{X}_j$is the output of the LW-MHSA after combining all the patches. The total number of heads of self-attention was split between the window configurations. We added relative position encoding to the attention module inspired by [53], [54]

$$Atn\ (Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}} + B\right)V \tag{8}$$

where B is the relative position bias term; Q, K, and V are the query, key, and values, respectively; and Softmax is a softmax function [55]. The proposed LW-MHSA mechanism enhances the conventional Multi-Head Self-Attention (MHSA) used in the Swin Transformer by introducing a dynamic window mechanism. Unlike MHSA which utilizes fixed and non-overlapping windows, the proposed LW-MHSA splits the feature map into patches and divides each patch into overlapping windows using a cyclic shifted-window mechanism. This approach allows LW-MHSA to capture local dependencies more effectively and reduces computational complexity by focusing attention on smaller, relevant regions. Additionally, LW-MHSA incorporates noise during the attention process, acting as a regularization technique to enhance model robustness and prevent overfitting. The inclusion of noise improves the resilience of the model by preventing it from memorizing specific details in the training data. Furthermore, the integration of relative position encoding further enhances spatial awareness.

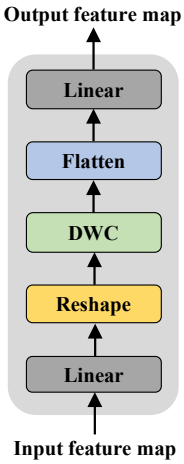


Fig. 5. Proposed CFFN

b) **CFFN**

The CFFN block consists of linear and convolutional layers. According to [56], [57], the FFN of transformers is unable to capture local contextual information. The neighboring pixel information is crucial in colorization. To address this issue, we added convolutional layers to the FFN, as in [57], [58], [59]. In CFFN, the first linear layer is applied, and the features obtained are reshaped to apply a convolution. The DWC enables the CFFN to capture local features while reducing the complexity of the model. After the convolutional layers, the features are flattened and passed through the linear layer. The GeLU [60] activation function is used for each layer.

3.2. Discriminator Architecture

The discriminator also referred to as a Critic in the proposed method, is based on the Markovian discriminator architecture (PatchGAN) [10], where the PatchGAN discriminator focuses on the high-frequency structure of the image generated by the generator and uses local patches in an image to determine whether a generated image is real or fake. The Critic produces high scores for the input and ground-truth (GT) pairs and low scores for the input and generated pairs. The discriminator was conditioned on a grayscale image (luminance channel image) by receiving both luminance and chrominance channels as inputs to predict a score indicating whether the image is real or fake.

3.3. Objective Function

The objective function of the proposed architecture is defined as

$$L_{total} = \lambda_g L_{GAN} + \lambda_1 L_{L1} + \lambda_2 L_{VGG} \tag{9}$$

where L_{GAN}, L_{L1}, and L_{VGG} represent the GAN, L1 loss, and VGG losses, respectively, and λ_g, λ_1, and λ_2 are the hyperparameters. The first term $\lambda_g L_{GAN}$ in (9) denotes the adversarial loss for GAN training. The WGAN loss functions are defined as (10) and (11).

$$L_D = E_y[\, D(y,x)] - E_{\tilde{y}}[\, D(\tilde{y}, x)] + \lambda \cdot GP, \tag{10}$$

$$L_G = -E_{\tilde{y}}[\,D(\tilde{y},x)] \tag{11}$$

where L_D and L_G represent the loss for the discriminator and generator of the WGAN [9], respectively; y and $\tilde{y}$ are the GT and generated chrominance image, respectively; x is given a grayscale image and passed to the discriminator as a condition similar to the CGAN and *GP* is the gradient penalty used for stable training of the GAN. The WGAN loss offers better properties than other GAN losses, resolves the problem of a vanishing gradient, and achieves stable training of the GAN [9]. L_{L1} is the pixel-wise L1 loss function defined as (12).

$$L_{L1} = ||y - \tilde{y}||_1 \tag{12}$$

We used the VGG loss function to improve the perceptual quality of the generated images. The VGG loss function in (9) is defined by the rectified linear unit activation layer of the pretrained VGG network.

$$L_{VGG} = ||\varphi_k(y) - \varphi_k(\tilde{y})||_2^2 \tag{13}$$

where φ_k refers to the features of the *k*-th layer of the pretrained VGG network. The VGG loss was used to measure the semantic similarity between the generated and GT images. Hence, the total losses for the generator and discriminator can be expressed as (14) and (15), respectively, by rewriting (10) and (11), respectively.

$$L_G = -E_{\tilde{y}}[\,D(\tilde{y})] + \lambda_1 L_{L1} + \lambda_2 L_{VGG} \tag{14}$$

$$L_D = E_y[\,D(y)] - E_{\tilde{y}}[\,D(\tilde{y})] + \lambda \cdot GP \tag{15}$$

4. Experimental Results

4.1. Implementation Details

For the experiments, we utilized the two publicly available datasets, PascalVOC [61] and ImageNET [6]. The PascalVOC dataset consists of 17,125 images, while ImageNET comprises nearly 1.3 million images for training. ImageNET was used as the training set for the training of the colorization network, and the first 5000 images from the ImageNET validation set were used for testing. The input images were resized to 256×256 pixels using bilinear interpolation. We observed that resizing was better than random cropping because cropping can negatively affect color learning. The input images were normalized to the range [−1, 1]. The training images were divided into l and ab channels and used as the input and GT, respectively, during training. We used an ADAM optimizer [62] with learning rates of 1×10^{-4} and 2×10^{-4} for the generator and discriminator, respectively. The size of the embedding dimension was set to 16, and exponential decay rates β_1 and β_2 values in the ADAM optimizer were set to 0.5 and 0.999, respectively. The generator and discriminator of ColorFormer are trained until the network converges. The hyperparameters λ_g, λ_1, and λ_2 in (9) were empirically set to 0.5, 100, and 1000, respectively. The network was implemented in the PyTorch framework with a GeForce RTX3090 GPU.

4.2. Quantitative Metrics and Comparisons

The peak signal-to-noise ratio (PSNR) [63] and structural similarity index measure (SSIM) [64] were used to measure the quality of colorized images with respect to the GT Images. PSNR and SSIM are popular metrics for evaluating the colorization performance. The PSNR measures the difference between original and processed images in terms of the ratio of the maximum possible power of a signal to the power of the corrupting noise. Meanwhile, the SSIM measures the structural similarity between two images by considering the luminance, contrast, and structure of the images. Higher PSNR and SSIM values generally indicate better image quality and a closer similarity to the original image. The colorfulness metric [65], which evaluates the amount of color variation in an image, was used to evaluate the quality of colorized images. The colorfulness metric [65] is based on the standard deviation of the chrominance channels (a* and b*) of the image and is used to quantify the overall color of the image. The standard deviation represents the degree to which the chrominance values in an image vary from the average values. A higher standard deviation indicates a greater range of chrominance values and, thus, a more colorful image. △Colorfulness measures the difference in colorfulness values between colored and GT images. Additionally, the Learned Perceptual Image Patch Similarity (LPIPS) [66] metric was used to evaluate the perceptual quality of the colorized images. LPIPS measures the perceptual similarity between images by comparing deep feature representations extracted from neural networks, with lower LPIPS values indicating higher perceptual similarity to the GT images.

(a) Convolution-based methods

(b) Transformer-based methods

Fig. 6. Visual comparisons for the colorized output images (a) Convolution-based methods (b) Transformer-based methods

TABLE I
QUANTITATIVE COMPARISONS

Models \ Datasets	PascalVOC					ImageNET				
	PSNR (dB)	**SSIM**	**Colorfulness**	**△Colorfulness**	**LPIPS**	**PSNR (dB)**	**SSIM**	**Colorfulness**	**△Colorfulness**	**LPIPS**
CIC [14]	21.00	0.92	30.43	2.55	0.15	22.64	0.91	31.60	4.72	0.22
Deoldify [67]	22.97	0.91	16.60	16.38	0.15	21.12	0.83	22.70	13.62	0.24
BigColor [34]	21.47	0.88	35.71	2.73	0.17	21.26	0.89	38.65	2.00	0.17
InstCol [68]	22.91	0.91	22.21	10.77	0.17	23.28	0.91	24.87	11.44	0.21
ChromaGAN [33]	23.63	0.88	21.89	11.09	0.18	23.35	0.90	27.88	8.43	0.21
ColTran [44]	23.83	0.86	35.74	2.76	0.19	20.95	0.80	20.60	15.72	0.29
CT2 [45]	19.30	0.91	**36.04**	3.06	0.15	23.50	0.92	38.48	2.17	0.19
ColorDiffusion [48]	18.12	0.59	16.09	16.89	0.39	17.54	0.54	16.36	24.29	0.41
ColorFormer	**24.51**	**0.94**	31.38	**1.60**	**0.15**	**24.37**	**0.92**	**38.95**	**1.70**	**0.16**

TABLE II
COMPLEXITY ANALYSIS

Models	GFlops	Number of parameters
Swin Transformer Block [43]	3.48	54.98k
Proposed ColorFormer Block	**2.26**	**53.44k**

We compared the proposed method with fully automatic state-of-the-art colorization methods, including CIC [14], Deoldify [67], BigColor [34], InstColor [68], ChromaGAN [33], ColTran [44], CT2 [45] and ColorDiffusion [48]. Table I presents the quantitative results and comparisons. Table I shows that our proposed method achieves significantly higher PSNR and SSIM values than other state-of-the-art methods. The LPIPS values for images colorized using ColorFormer are consistently lower compared to the state-of-the-art methods. Fig. 6 (a) and (b) shows the visual results of colorization for the proposed and other methods. As shown in Fig. 6, the proposed method shows more natural colorization than the other methods. For instance, the horse color (column 2) of the CIC [14] appears unnatural and reddish compared with our results. In addition, our method does not produce rare colors, and the output images are close to the GT. Although BigColor [34] demonstrated more saturated results in image colorization, it exhibited a notable drawback in generating unnatural outputs that deviated from the true colors of GT images. Coltran [44], a transformer-based colorization network, reported desaturated results with bleeding artifacts. However, the resulting images of our proposed method may not achieve higher colorfulness values because it is likely to encourage rare colors. However, our proposed method shows colored images that are more similar to the GT with higher PSNR and SSIM values, indicating that our method successfully reproduced the original colors accurately. Although Coltran [44], BigColor [34], and CT2 [45] produce rare colors, they exhibit bleeding artifacts and unnatural colorization. The △colorfulness values obtained in our experiments indicate that our proposed method produces color variations closely aligned with the GT images. Despite the potential of the diffusion models, the visual results of ColorDiffusion suffer from artifacts due to the iterative denoising process, which may introduce blurriness and lack of fine details, compromising the overall quality and sharpness of the colorized images. These findings demonstrate the effectiveness of our approach in generating more accurate and natural colorizations, thereby enhancing the overall quality and visual appeal of colorized images. Overall, the proposed network achieved more natural and consistent results.

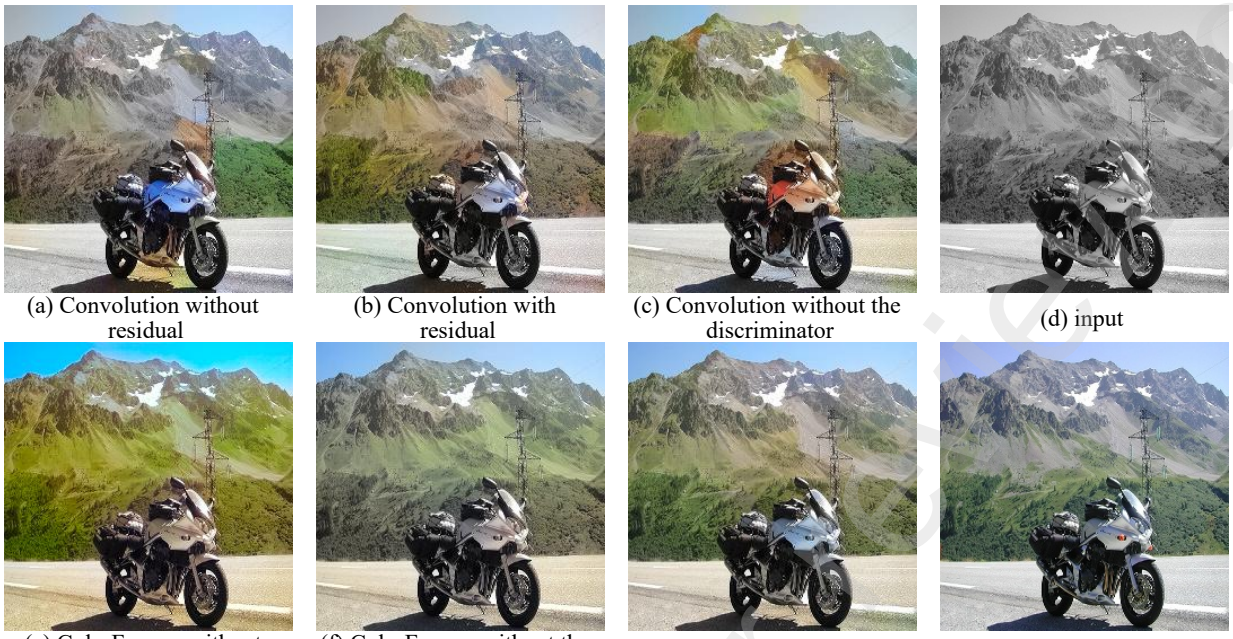
Table II shows a detailed complexity analysis of the Swin Transformer Block and our proposed ColorFormer Block. The analysis in the table focuses on two critical aspects: the computational complexity measured in GFlops and the model size quantified by the number of parameters. The Swin Transformer Block has a computational complexity of 3.48 GFlops, comprising 54.98k parameters. In contrast, the ColorFormer Block has a lower computational complexity of 2.26 GFlops with 3.44k parameters of less number of parameters. These comparative values highlight the efficiency of the ColorFormer Block, which has less computational complexity than the Swin Transformer Block.

Fig. 7. Failure cases

TABLE III
SETUP FOR ABLATION STUDY

Test items	Transformer block	GAN architecture	CFFN	Residual connection
Convolution w/ discriminator	✗	✓	✗	✗
Convolution w/ residual	✗	✓	✗	✓
Convolution w/o discriminator	✗	✗	✗	✓
ColorFormer w/o residual	✓	✓	✓	✗
ColorFormer w/o discriminator	✓	✗	✓	✓
Feedforward (MLP)	✓	✓	✗	✓
ColorFormer	✓	✓	✓	✓

Despite the success of our proposed method, it is essential to acknowledge its limitations and potential failure cases. The proposed model struggles in particularly challenging scenarios, including images with low-light conditions and high complexity, as shown in Fig. 7. In low-light situations, the model may struggle to discern intricate features, leading to suboptimal colorization results. Similarly, complex images with intricate patterns or multiple objects can pose challenges, as the model may find it challenging to assign colors to intricate regions accurately. Two common issues observed in failure cases are desaturation and color bleeding. The other state-of-the-art methods also have these issues, as shown in Fig. 6. Desaturation may occur when the model fails to capture essential features accurately, resulting in dull or faded color representations. This issue is particularly evident in low-light images. Furthermore, color bleeding, where colors extend beyond their intended boundaries, can significantly degrade overall image quality. This problem is especially pronounced in complex scenes, such as those featuring small objects against larger backgrounds, leading to color bleeding, as illustrated in Fig. 7.



(a) Convolution without residual (b) Convolution with residual (c) Convolution without the discriminator (d) input

(e) ColorFormer without residual (f) ColorFormer without the discriminator (g) ColorFormer (h) GT

Fig. 8. Visual comparison of ablation study for the proposed method

4.3. Ablation Studies

We conducted extensive ablation studies to demonstrate the effectiveness of the ColorFormer Block, residual connections, and adversarial learning. Table III shows the setup of the ablation studies. Fig. 8, Tables IV, V, and VI show the results.

Convolution: To test the contribution of the transformer block to the overall performance, we replaced the ColorFormer Block with a convolutional layer. Transformers play an important role in capturing long-range dependencies. Experimental results show that the convolution block produced inconsistent and unnatural colors in the image. Finally, ColorFormer achieved a gain of approximately 1.6 dB in the PSNR, as shown in Table IV. Moreover, the LPIPS value for ColorFormer was significantly lower (0.15) than the convolution block (0.19), which indicates that ColorFormer achieves better perceptual quality.

Residual connection: We added residual connections to every ColorFormer Block to compensate for the missing information in our proposed network. The colors were dull and unsaturated if the residual connections were disabled, as shown in Fig. 8(a) and (b). The results indicated that the residual connections alleviated the problem of vanishing gradients and improved the flow of information. The residual connections showed promising results for both the PSNR and SSIM values, as listed in Table IV. Additionally, the LPIPS value improved from 0.21 without residual connections to 0.15 with residual connections, highlighting the enhancement in perceptual quality.

Adversarial learning: We investigated the effects of the discriminator and adversarial loss on colorization. The discriminator was removed, and the generator was trained using perceptual loss. Adversarial learning can substantially affect colorization by enabling the generative model to generate colorized images that are more realistic and consistent with real color images. Adversarial learning helps to overcome the limitations of conventional colorization algorithms

TABLE IV
ABLATION STUDY ON THE EFFECT OF CONVOLUTION AND RESIDUAL CONNECTIONS

Variations	PSNR (dB)	SSIM	Colorfulness	△Colorfulness	LPIPS
Convolution without residual	22.806	0.936	26.51	6.47	0.19
Convolution with residual	22.965	0.937	25.32	7.66	0.18
ColorFormer without residual	24.414	0.930	29.45	3.53	0.21
ColorFormer	**24.517**	**0.943**	**31.38**	**1.60**	**0.15**

TABLE V
ABLATION STUDY ON THE EFFECT OF ADVERSARIAL LEARNING

Variations	PSNR (dB)	SSIM	Colorfulness	△Colorfulness	LPIPS
Convolution without discriminator	22.793	0.935	26.73	6.26	0.18
ColorFormer without discriminator	24.362	**0.952**	17.48	15.50	0.17
ColorFormer	**24.517**	0.943	**31.38**	**1.60**	**0.15**

TABLE VI
ABLATION STUDY ON THE EFFECT OF CFFN

Variations	PSNR (dB)	SSIM	Colorfulness	△Colorfulness	LPIPS
Feedforward (MLP)	24.291	**0.944**	19.44	13.54	0.16
ColorFormer	**24.517**	0.943	**31.38**	**1.60**	**0.15**

that rely on heuristics and color distribution assumptions. Table V presents the contributions of adversarial learning to the overall performance of the network. Non-adversarial learning approaches generate less realistic colorization results, as shown in Figs. 8(c) and (f), where the absence of adversarial learning leads to suboptimal colorization quality. The LPIPS value further supports this observation, with a lower LPIPS value of 0.15 for ColorFormer compared to 0.17 for the non-adversarial approach, confirming the perceptual improvement brought by adversarial learning.

FFN: The FFN within the transformer layer plays a crucial role in introducing nonlinear transformations, enabling the model to learn more complex features and relationships, thereby enhancing its expressive power and representation capability. Incorporating convolution within the FFN offers additional benefits, such as efficient local feature extraction and capturing spatial relationships in images, which helps the model better understand the structure and context of the input data. A conventional FFN was used in this ablation study to test the effects of convolution on the FFN. As a result, the PSNR values increase, and the output images appear more natural when the convolutional layer is used in the FFN, as shown in Fig. 8(g). The colorfulness and △colorfulness values in Table VI show that convolution in the FFN significantly enhances the colorization results. The LPIPS value also decreased to 0.15 for ColorFormer, demonstrating improved perceptual similarity compared to the conventional FFN with a higher LPIPS value of 0.16.

5. Conclusion

In this study, we developed a novel and robust image colorization technique that leverages the strengths of the CWGAN and a ColorFormer Block, which employs a window-based MHSA mechanism for lightweight and efficient processing. By integrating convolutional layers into the ColorFormer Block, our approach can effectively capture local and global features, leading to superior colorization results than existing state-of-the-art methods. Extensive ablation studies demonstrated the effectiveness of our proposed method and its ability to generate visually appealing

and realistic colorized images. The proposed adversarial training methodology ensures that the generated colorizations are plausible and visually compelling. The proposed method has significant potential for various applications, including art restoration, video colorization, and the enhancement of low-quality or historical images. In conclusion, the proposed ColorFormer architecture demonstrates the effectiveness of leveraging GANs and transformer blocks in the image colorization domain, paving the way for further advancements and research in this area.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean Government under Grant 2022R1I1A3065473

6. REFERENCES

[1] L. Zhang, C. Li, T.-T. Wong, Y. Ji, and C. Liu, "Two-stage sketch colorization," *ACM Trans Graph*, vol. 37, no. 6, pp. 1–14, Dec. 2018, doi: 10.1145/3272127.3275090.

[2] Y. Qu, T.-T. Wong, and P.-A. Heng, "Manga colorization," in *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*, New York, New York, USA: ACM Press, 2006, p. 1214. doi: 10.1145/1179352.1142017.

[3] Y. Chen, Y. Luo, Y. Ding, and B. Yu, "Automatic Colorization of Images from Chinese Black and White Films Based on CNN," in *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, IEEE, Jul. 2018, pp. 97–102. doi: 10.1109/ICALIP.2018.8455654.

[4] Y. Guo, X. Cao, W. Zhang, and R. Wang, "Fake Colorized Image Detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 1932–1944, Aug. 2018, doi: 10.1109/TIFS.2018.2806926.

[5] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a Proxy Task for Visual Understanding," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jul. 2017, pp. 840–849. doi: 10.1109/CVPR.2017.96.

[6] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int J Comput Vis*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[7] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 Million Image Database for Scene Recognition," *IEEE Trans Pattern Anal Mach Intell*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018, doi: 10.1109/TPAMI.2017.2723009.

[8] J. Zhao, L. Liu, C. G. M. Snoek, J. Han, and L. Shao, "Pixel-level Semantics Guided Image Colorization," in *British Machine Vision Conference*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:51926798

[9] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., in Proceedings of Machine Learning Research, vol. 70. PMLR, Nov. 2017, pp. 214–223. [Online]. Available: https://proceedings.mlr.press/v70/arjovsky17a.html

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jul. 2017, pp. 5967–5976. doi: 10.1109/CVPR.2017.632.

[11] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans Graph*, vol. 23, no. 3, pp. 689–694, Aug. 2004, doi: 10.1145/1015706.1015780.

[12] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proceedings of the 13th annual ACM international conference on Multimedia*, New York, NY, USA: ACM, Nov. 2005, pp. 351–354. doi: 10.1145/1101149.1101223.

[13] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1120–1129, May 2006, doi: 10.1109/TIP.2005.864231.

[14] P. and E. A. A. Zhang Richard and Isola, "Colorful Image Colorization," in *Computer Vision – ECCV 2016*, J. and S. N. and W. M. Leibe Bastian and Matas, Ed., Cham: Springer International Publishing, 2016, pp. 649–666.

[15] G. Lee, S. Shin, T. Na, and S. S. Woo, "Real-Time User-Guided Adaptive Colorization With Vision Transformer," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 484–493.

[16] J. Yun, S. Lee, M. Park, and J. Choo, "iColoriT: Towards Propagating Local Hints to the Right Region in Interactive Colorization by Leveraging Vision Transformer," *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1787–1796, 2023.

[17] T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to greyscale images,” *ACM Trans Graph*, vol. 21, no. 3, pp. 277–280, Jul. 2002, doi: 10.1145/566654.566576.
[18] R. Irony, D. Cohen-Or, and D. Lischinski, “Colorization by Example,” in *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, in EGSR ’05. Goslar, DEU: Eurographics Association, 2005, pp. 201–210.
[19] A. Y.-S. Chia *et al.*, “Semantic colorization with internet images,” *ACM Trans Graph*, vol. 30, no. 6, pp. 1–8, Dec. 2011, doi: 10.1145/2070781.2024190.
[20] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, “Deep exemplar-based colorization,” *ACM Trans Graph*, vol. 37, no. 4, pp. 1–16, Aug. 2018, doi: 10.1145/3197517.3201365.
[21] B. Zhang *et al.*, “Deep Exemplar-Based Video Colorization,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2019, pp. 8044–8053. doi: 10.1109/CVPR.2019.00824.
[22] S. and C. W. and P. D. K. and W. Z. and M. X. and C. J. Bahng Hyojin and Yoo, “Coloring with Words: Guiding Image Colorization Through Text-Based Palette Generation,” in *Computer Vision – ECCV 2018*, M. and S. C. and W. Y. Ferrari Vittorio and Hebert, Ed., Cham: Springer International Publishing, 2018, pp. 443–459.
[23] H. Kim, H. Y. Jhoo, E. Park, and S. Yoo, “Tag2Pix: Line Art Colorization Using Text Tag With SECat and Changing Loss,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2019, pp. 9055–9064. doi: 10.1109/ICCV.2019.00915.
[24] C. Zou, H. Mo, C. Gao, R. Du, and H. Fu, “Language-based colorization of scene sketches,” *ACM Trans Graph*, vol. 38, no. 6, pp. 1–16, Dec. 2019, doi: 10.1145/3355089.3356561.
[25] Z. Cheng, Q. Yang, and B. Sheng, “Deep Colorization,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Dec. 2015, pp. 415–423. doi: 10.1109/ICCV.2015.55.
[26] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Let there be color!,” *ACM Trans Graph*, vol. 35, no. 4, pp. 1–11, Jul. 2016, doi: 10.1145/2897824.2925974.
[27] M. and S. G. Larsson Gustav and Maire, “Learning Representations for Automatic Colorization,” in *Computer Vision – ECCV 2016*, J. and S. N. and W. M. Leibe Bastian and Matas, Ed., Cham: Springer International Publishing, 2016, pp. 577–593.
[28] J. Zhao, J. Han, L. Shao, and C. G. M. Snoek, “Pixelated Semantic Colorization,” *Int J Comput Vis*, vol. 128, no. 4, pp. 818–834, 2020, doi: 10.1007/s11263-019-01271-4.
[29] I. Goodfellow *et al.*, “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf
[30] K. Nazeri and E. Ng, “Image Colorization with Generative Adversarial Networks,” *arXiv preprint arXiv:1803. 05400*, 2018.
[31] Z. and Z. W. and Y. Y. Cao Yun and Zhou, “Unsupervised Diverse Colorization via Generative Adversarial Networks,” in *Machine Learning and Knowledge Discovery in Databases*, J. and T. L. and V. C. and D. S. Ceci Michelangelo and Hollmén, Ed., Cham: Springer International Publishing, 2017, pp. 151–166.
[32] A. Deshpande, J. Lu, M.-C. Yeh, M. J. Chong, and D. Forsyth, “Learning Diverse Image Colorization,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jul. 2017, pp. 2877–2885. doi: 10.1109/CVPR.2017.307.
[33] P. Vitoria, L. Raad, and C. Ballester, “ChromaGAN: Adversarial Picture Colorization with Semantic Class Distribution,” in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Mar. 2020, pp. 2434–2443. doi: 10.1109/WACV45572.2020.9093389.
[34] K. and K. S. and L. H. and K. S. and K. J. and B. S.-H. and C. S. Kim Geonung and Kang, “BigColor: Colorization Using a Generative Color Prior for Natural Images,” in *Computer Vision – ECCV 2022*, G. and C. M. and F. G. M. and H. T. Avidan Shai and Brostow, Ed., Cham: Springer Nature Switzerland, 2022, pp. 350–366.
[35] Y. Zhao, L.-M. Po, K.-W. Cheung, W.-Y. Yu, and Y. A. U. Rehman, “SCGAN: Saliency Map-Guided Colorization With Generative Adversarial Network,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3062–3077, Aug. 2021, doi: 10.1109/TCSVT.2020.3037688.
[36] K. Du, C. Liu, L. Cao, Y. Guo, F. Zhang, and T. Wang, “Double-Channel Guided Generative Adversarial Network for Image Colorization,” *IEEE Access*, vol. 9, pp. 21604–21617, 2021, doi: 10.1109/ACCESS.2021.3055575.
[37] Y. Wu, X. Wang, Y. Li, H. Zhang, X. Zhao, and Y. Shan, “Towards Vivid and Diverse Image Colorization with Generative Color Prior,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2021, pp. 14357–14366. doi: 10.1109/ICCV48922.2021.01411.
[38] S. Treneska, E. Zdravevski, I. M. Pires, P. Lameski, and S. Gievska, “GAN-Based Image Colorization for Self-Supervised Visual Feature Learning,” *Sensors*, vol. 22, no. 4, p. 1599, Feb. 2022, doi: 10.3390/s22041599.
[39] C. Liu and Y. Tu, “Image Colorization with Convolution Block Attention Modules,” https://github.com/kliu513/Image-Colorization.
[40] H. Shafiq and B. Lee, “Image Colorization Using Color-Features and Adversarial Learning,” *IEEE Access*, vol. 11, pp. 132811–132821, 2023, doi: 10.1109/ACCESS.2023.3335225.
[41] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[42] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *ArXiv*, vol. abs/2010.11929, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:225039882
[43] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2021, pp. 9992–10002. doi: 10.1109/ICCV48922.2021.00986.
[44] M. Kumar, D. Weissenborn, and N. Kalchbrenner, "Colorization Transformer," in *International Conference on Learning Representations*, 2021.
[45] S. Weng, J. Sun, Y. Li, S. Li, and B. Shi, "$$\mathrm {CT^2}$$: Colorization Transformer via Color Tokens," in *Computer Vision – ECCV*, 2022, pp. 1–16. doi: 10.1007/978-3-031-20071-7_1.
[46] X. Kang, T. Yang, W. Ouyang, P. Ren, L. Li, and X. Xie, "DDColor: Towards Photo-Realistic Image Colorization via Dual Decoders," *IEEE/CVF International Conference on Computer Vision*, Dec. 2023.
[47] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," *arXiv preprint arXiv:2006.11239*, 2020.
[48] E. Millon, "Color Diffusion." 2023. Accessed: Jun. 01, 2024. [Online]. Available: https://github.com/ErwannMillon/Color-diffusion.git
[49] H. Liu, J. Xing, M. Xie, C. Li, and T.-T. Wong, "Improved Diffusion-based Image Colorization via Piggybacked Models," *arXiv preprint arXiv:2304.11105*, 2023.
[50] International Commission on Illumination (CIE), *Colorimetry*, 3rd ed. Vienna, Austria: CIE, 2004.
[51] V. Nair and G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair," in *Proceedings of ICML*, Nov. 2010, pp. 807–814.
[52] P. and B. T. Ronneberger Olaf and Fischer, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, J. and W. W. M. and F. A. F. Navab Nassir and Hornegger, Ed., Cham: Springer International Publishing, 2015, pp. 234–241.
[53] T. Plötz and S. Roth, "Benchmarking Denoising Algorithms with Real Photographs," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2750–2759, 2017, [Online]. Available: https://api.semanticscholar.org/CorpusID:9715523
[54] H. Son, J. Lee, S. Cho, and S. Lee, "Single Image Defocus Deblurring Using Kernel-Sharing Parallel Atrous Convolutions," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2622–2630, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:237259856
[55] Bishop and M. Christopher, *Pattern Recognition and Machine Learning*. Springer, 2006.
[56] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning Texture Transformer Network for Image Super-Resolution," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2020, pp. 5790–5799. doi: 10.1109/CVPR42600.2020.00583.
[57] S. W. Zamir *et al.*, "CycleISP: Real Image Restoration via Improved Data Synthesis," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2020, pp. 2693–2702. doi: 10.1109/CVPR42600.2020.00277.
[58] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image Restoration Using Swin Transformer," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, IEEE, Oct. 2021, pp. 1833–1844. doi: 10.1109/ICCVW54120.2021.00210.
[59] J. Shi, L. Xu, and J. Jia, "Just noticeable defocus blur detection and estimation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015, pp. 657–665. doi: 10.1109/CVPR.2015.7298665.
[60] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv preprint arXiv:1606.08415*, Jun. 2016.
[61] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int J Comput Vis*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.
[62] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6980, 2014, [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106
[63] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron Lett*, vol. 44, no. 13, p. 800, 2008, doi: 10.1049/el:20080522.
[64] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.
[65] D. Hasler and S. E. Suesstrunk, "Measuring colorfulness in natural images," in *Human vision and electronic imaging*, B. E. Rogowitz and T. N. Pappas, Eds., Jun. 2003, p. 87. doi: 10.1117/12.477378.
[66] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
[67] "DeOldify." Accessed: Nov. 04, 2023. [Online]. Available: https://github.com/jantic/DeOldify
[68] J.-W. Su, H.-K. Chu, and J.-B. Huang, "Instance-Aware Image Colorization," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7965–7974, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:218763285