

ELF-CandyGAN: A Candy Color Coloring Method for Image Local Feature Enhancement

Mei Xu^{*}, Huan Xu

School of Information Engineering, Technology & Media University of Henan Kaifeng, Henan, Kaifeng, 475000, China

**Corresponding author: xm@home.hpu.edu.cn*

Keywords: Local Feature Enhancement; Candy Color; Generative Adversarial Networks; Bi-discriminator Networks; Feature Structure Loss Function

Abstract: Candy color is a new phenomenon in the field of photography, and its high brightness, low saturation, and low contrast bring a unique color experience to the world. The CandyCycleGAN network is good at realizing the transformation from ordinary color to candy color. Still, there will be problems as some of the image details are not dealt with properly, so to solve the above problems, this paper designs a candy based on the local feature enhancement of the image color coloring method (ELF-CandyGAN). Based on the generator of the U-Net network, a color learning module is designed in the downsampling process to learn the distribution, relationship, and features of candy color, which helps the network to better learn and understand the color information and keep the naturalness and authenticity of the color, and at the same time, a unique jump connection is designed to add the results in the upper layer convolution as part of the results in the lower layer convolution; secondly, a global context module is introduced in the color learning module. To ensure that the chromaticity values learned during the entire network training process remain within the candy color range, a global context module is introduced. This module also reduces computational complexity and accelerates network training. Subsequently, a feature enhancement module is designed, which introduces additional enhancement operations to enable deeper mining and processing of network features, thereby improving the performance and effectiveness of the network in the coloring task. This module also helps to enhance and restore the detail information lost during the downsampling process. Furthermore, a dual discriminator network based on PatchGAN is constructed. The first discriminator, D1, adopts a multi-scale discriminative structure to guide the generator toward producing richer image details. The second discriminator, D2, is designed to compute the structural similarity between the generated image and the original input image, encouraging the generator to produce structurally consistent outputs. Finally, a feature structure loss function is proposed to impose constraints on the structural similarity between the generated and input images, ensuring that the generated images retain more original detail features and exhibit higher realism.

1. Introduction

In 2018, a participant named Ben Thomas won the Hasselblad Masters photography competition with a fresh and unique post-processing style called candy color, which subsequently became widely recognized and popular as a new tonal mode. Characterized by low contrast and high saturation, candy color makes the colors in images more vivid, conveying richer color information. With its distinctive tonality, candy color has introduced a novel photographic perspective to the photography industry and provided the world with a fresh color experience. Currently, the realization of the candy color tone is primarily achieved through image recoloring technology, which alters visual image effects by modifying an input image's color, brightness, and contrast. Researchers have designed various methods to implement image recoloring.

Zongnan Chen et al [1] improved the activation function of the CycleGAN model, using the Parametric Rectified Linear Unit activation function in the generator to facilitate model training. The discriminator employs the PatchGAN network architecture to enhance color detail at high resolutions. Shisong Zhu [2] proposed a Candy CycleGAN network based on chrominance verification to achieve candy color recoloring. Based on the CycleGAN network, multi-scale fusion was implemented to enhance detailed features in output images and improve overall image quality; additionally, a chrominance verification process was designed to constrain the generated chrominance value range, ensuring the final results meet expectations. Wenhua Ding et al.[3] adaptively learn style features for images of different styles through the layer-consistent dynamic convolution method and achieve multi-image-domain translation by fusing content and style features of input images. Wenhui Qin et al.[4] designed a model named MCGAN, capable of transforming multiple artistic styles in a single operation. Based on the CycleGAN network combined with CGAN, this model addresses the issue of style homogenization among multiple styles by introducing channel-wise and spatial attention mechanisms as well as a style discretization loss function. Jiawei Zhou [5] proposed a novel image recoloring algorithm based on moving least squares, significantly improving the locality and accuracy of color editing. Sihong Meng[6] introduced an image colorization algorithm based on semantic similarity propagation. This approach utilizes a deep neural network to extract semantic features from an input grayscale image. The image colorization problem is addressed by solving for chrominance values of the grayscale image through energy function optimization, thereby propagating user-specified stroke colors to other regions of the image. Hong'an Li [7] proposed a grayscale image coloring method combining Pix2Pix Generative Adversarial Network. It first improves the U-Net architecture, then employs both L1 loss and smooth L1 loss to measure the difference between generated and real images, and finally introduces a gradient penalty to construct new data distributions between the generated and real image distributions. LIANG et al.[8] combined the Control Net with a coloring model, leveraging the diffusion model to achieve finer control over image generation while also realizing multimodal coloring effects.

2. Related Work

The CandyCycleGAN network achieves good results for candy color coloring. However, since the original image is fed into the generator for image generation, part of the image feature information is lost during the learning process, which leads to insufficient utilization of the input image information. To address some issues existing in the coloring results of the CandyCycleGAN network, this article continues to research candy color coloring methods based on Generative Adversarial Network, aiming to solve problems such as loss of local feature information after coloring.

Therefore, this article proposes a candy color coloring method based on image local feature enhancement (Candy color coloring method for image local feature enhancement) (ELF-CandyGAN), capable of addressing issues like partial detail feature loss in generated images while also improving

computational efficiency. In the generator network, this chapter adopts a structure based on the U-Net network as the foundation. An RGB image is used as input to the generator, and a color learning module is designed to capture chrominance information in the image. Additionally, a feature enhancement module is designed to recombine features from the original input image and the output image, forming a new output image. In the discriminator network, this chapter uses a structure based on PatchGAN as the foundation and implements a dual-discriminator architecture. One discriminator, referred to as the first discriminator D1, is used to determine the authenticity of the generated image. The D1 discriminator network employs a multi-scale discrimination structure to guide the generator network in producing images with richer details. The other one, referred to as the second discriminator D2, is used to calculate the structural similarity between images. The D2 discriminator computes the structural similarity between the generated image produced by the generator and the original input image, helping the generator produce images with higher structural consistency and thereby improving the quality of the generated images. Using the original input RGB image and the output image as inputs to the discriminator network, they first pass through the D1 discriminator network. If the D1 discriminator identifies the image as real, the image proceeds to the D2 discriminator network. Otherwise, the loss of the D1 discriminator is calculated as the total discriminator loss. If the image passes through the D1 discriminator and is identified as real by the D2 discriminator, the generator loss is calculated. Otherwise, the loss of the D2 discriminator is computed. Finally, the losses from both the D1 and D2 discriminators are summed to obtain the final discriminator loss.

3. Network Model

The CandyCycleGAN network employs an image coloring algorithm based on chrominance verification and candy color, which demonstrates certain advantages in transforming images from base colors to candy color. However, it also has several limitations. First, due to the unique architecture of the network-featuring two generators and one discriminator-it consumes significant memory and computational resources during subsequent network training, demanding a high-performance experimental environment and resulting in relatively long overall training times. Second, the complexity of the CandyCycleGAN network requires tuning numerous hyperparameters to achieve optimal results; this parameter adjustment process is excessively time-consuming and affects the overall experimental progress. Lastly, some image details are lost during the convolution process in the coloring output. To address the aforementioned issues, this article designs the ELF-CandyGAN network.

3.1 The Candy Color Coloring Network Generator Based on Enhanced Local Image Features

The generator structure in this article based on U-Net is shown in Figure 1, named the ELF-CandyGAN network generator. It consists of three main parts: the encoding module (downsampling section), the feature enhancement module, and the decoding module (upsampling section).

In the encoding module, the input image first passes through a 3×3 convolutional layer, then proceeds through four cascaded downsampling modules and color learning modules alternately before reaching the feature learning module. The color learning modules learn and understand the characteristics and patterns of candy color, enabling them to assign appropriate candy color tonal values to the colored regions within the image. Feature maps processed by the downsampling modules have their dimensions halved and channel count reduced by half. Conversely, feature maps passing through the color learning modules maintain the same output dimensions and number of channels.

The feature enhancement module of the generator comprises nine local feature enhancement blocks. This part of the network transforms and strengthens edge features and color features into generated image features, reinforcing the relationship between inputs and outputs to achieve more

effective feature learning. The detailed structure is shown as follows.

Finally, the generator's upsampling operation performs the inverse of the downsampling operation, gradually restoring the image size and number of channels. This network fuses feature maps from skip connections and outputs from the previous layer via convolutional layers, enhances feature learning through residual modules, and finally applies the Pixel Shuffle [9] method for upsampling to restore the original resolution.

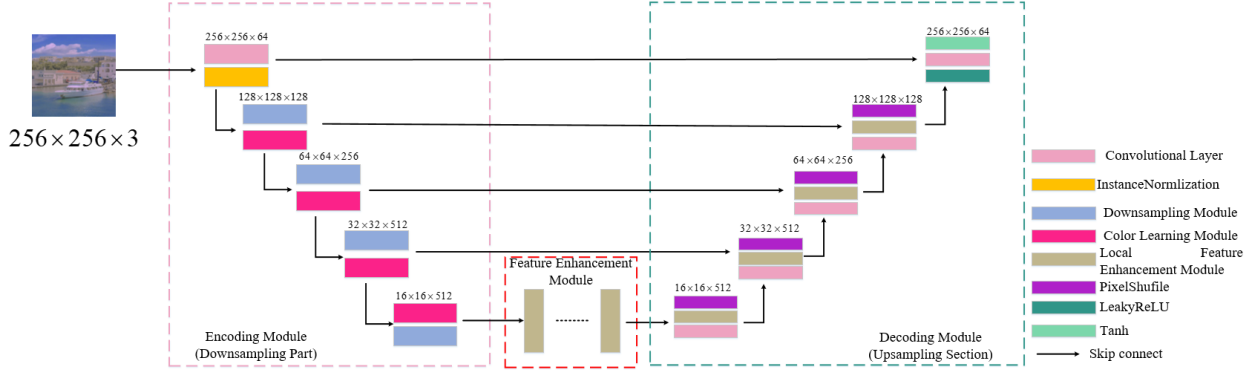


Figure 1: ELF-CandyGAN Network Generator

3.2 Color Learning Module

This article opts for a more effective Global Context Network (GC Net) module [10], which models relationships among all image pixels, helping the network better learn inter-pixel correlations and improve the final image colorization quality.

The GC Net module combines the advantages of both Non-Local Networks (Non-Local Net, NL Net) and SE Net. It can establish long-range dependencies and capture global information like NL Net [11], as well as reduce computational complexity similar to SE Net. The detailed architecture is shown in Figure 2: Comparison of GC Block network structure and related modules.

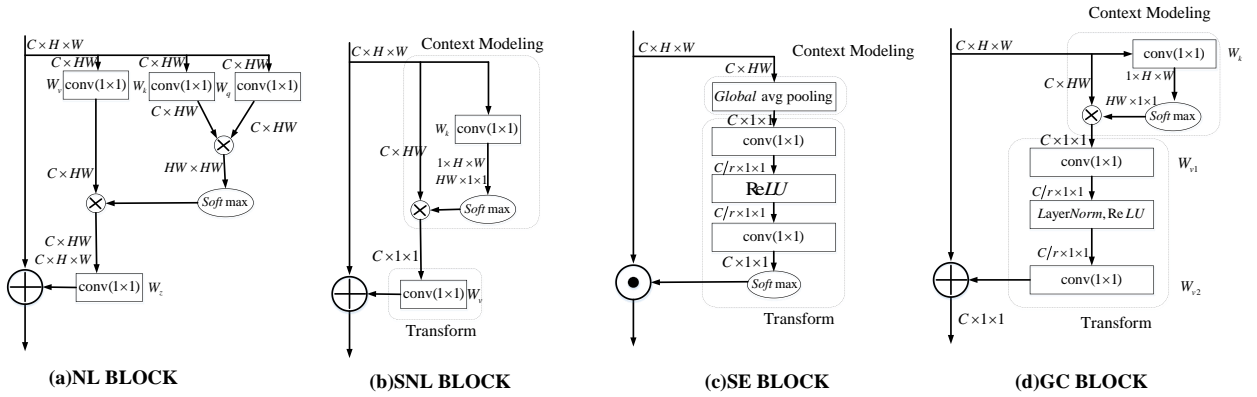


Figure 2: GC Block network structure and its related module structure comparison

Specifically, (a) represents the traditional NL Block. In NL Net, the attention feature maps calculated for different pixel positions are essentially identical. By computing pairwise relationships between each search position and all other positions to form an attention map (a). (b) denotes the simplified NL Block (SNL Block). SNL Block simplifies the original block by employing a global attention map computed from all pixels, thereby reducing the computational cost to a fraction of the original. It first performs Global Attention Pooling (GAP), where the attention weights are obtained through a 1×1 convolution W_k , followed by pooling to acquire global contextual features. The second

step involves feature transformation via another 1×1 convolution W_v . Finally, the global contextual features from each position are aggregated through an addition operation. (c) is the SE Block, which first applies global average pooling, utilizes a Bottleneck Transform (BoT) module to compute the importance of each channel, and then employs matrix multiplication to recalibrate the channel features. (d) is the GC Block, which can be viewed as a combination of (c) and (d). It first performs context modeling, then replaces the original convolution process with the BoT module to learn inter-channel dependencies, computes the number of parameters, and adds the feature weights to the mapping of input features for feature fusion. Additionally, a normalization layer is added before the activation function in the BoT module to reduce the optimization complexity caused by stacking two BoT modules and improve generalization performance.

The color learning module is shown in Figure 3 Color Learning Module. The convolution kernel size is 3×3 , the stride is 2, the padding value is 1, and the activation layer is LeakyReLU. Additionally, a GC block is added to learn and maintain the candy color style. InstanceNormalization is applied to all layers except the sixth and the final convolutional layers. The first three steps are used for learning the candy color style, while the following five steps utilize convolution and deconvolution operations to restore the input image size, ensuring that the proposed color learning process does not affect the subsequent generation network's convolutional operations.

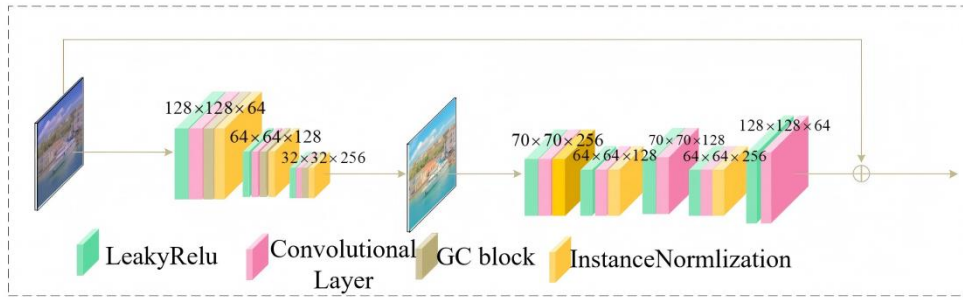


Figure 3: GC Block network structure and comparison of its related module structures

3.3 Local Feature Enhancement Module

This article designs a local feature enhancement module applied to adjacent local regions to strengthen the expression of local features, as shown in Figure 4, the structural diagram of the local feature enhancement module. Starting from the image center as the adjacent local region, this approach aggregates local features with high similarity to supplement missing information in features extracted by the backbone network [12]. ResNet is employed in this article as the base network for local feature enhancement, aiming to improve the model's perception capability for subtle image features and color variations, thereby enhancing overall performance and effectiveness. The depth of the ResNet architecture facilitates learning complex image features, enabling a better understanding of image content and improving the accuracy of image coloring.

The output features $T1$ and $T2$ from adjacent layers of the U-Net network are first normalized. Taking $T1$ as an example, two linear mapping functions (ω_v and ω_s) are used to generate dimension-reduced features $Tv1$ and $Ts1$. Additionally, $T1$ and $T2$ are concatenated to form an integrated feature Tq , which combines features from $T1$ and $T2$. Then, interactions between these features and their respective inputs are leveraged to enhance the features. Specifically, another linear mapping function ω_q is applied on Tq to reduce its dimensionality to $c/2$, followed by a softmax function to generate the weight map $T\omega_q$. Subsequently, the mapping result is used to perform a weighting operation on $Tv1$ via element-wise multiplication, followed by an adaptive average pooling operation ($P(\cdot)$) to reduce computational cost. This group of operations can be represented as $F(\cdot)$, as formulated in

Equation 1:

$$\hat{T}_q = F(T_s, T_q) = P(T_s \odot \text{soft max}(\omega_q(T_q))) \quad (1)$$

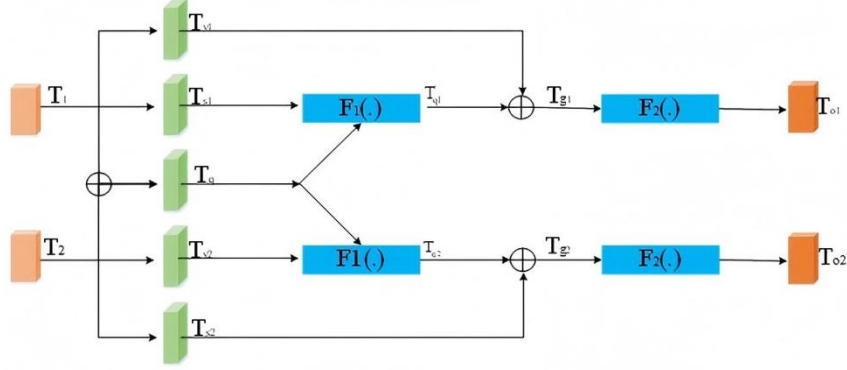


Figure 4: Structure of the Local Feature Enhancement Module

Finally, the features T_{v1} and T_{q1} are combined to form T_g . Then, a deserialization operation is applied to obtain the locally enhanced feature map T_{o1} . Similarly, T_i can undergo the above operations to obtain the locally enhanced feature T_{oi} .

3.4 Colorization Network Discriminator Based on Local Image Feature Enhancement candy color

This article designs a dual-discriminator network, one being the first discriminator D1 used to distinguish real from generated images, and the other being the second discriminator D2 used to calculate structural similarity between images.

The ELF-CandyGAN network employs the PatchGAN structure for its D1 discriminator, incorporating full-size, 1/2-size, and 1/4-size convolution processes. The architecture of the ELF-CandyGAN discriminator D1 is illustrated in Figure 5 Network Structure of Discriminator D1. Finally, three outputs of different sizes are obtained, forming a feature map matrix where each element represents the discrimination result of the corresponding region. The average of these three outputs serves as the final result.

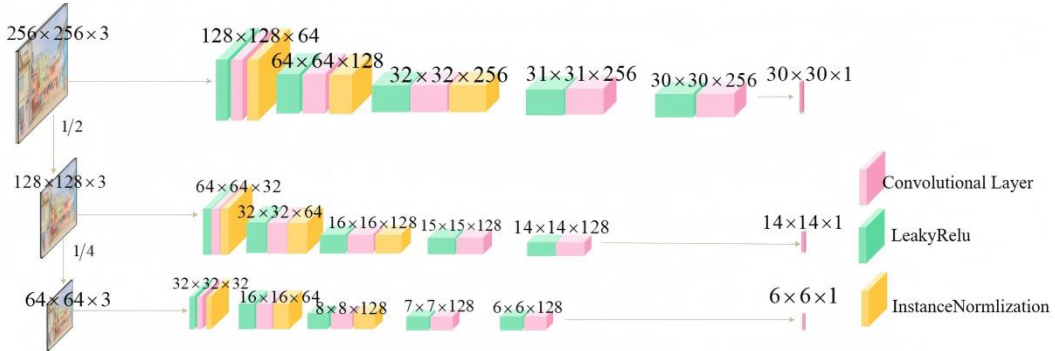


Figure 5 Discriminator Network D1 Architecture

The ELF-CandyGAN network discriminator D2 adopts the PatchGAN architecture. As shown in Figure 6, the network structure of Discriminator D2 consists of six convolutional layers. The kernel size of all convolutional layers is 3×3 . The first three layers have a stride of 2 and padding value of 1. The last three layers use kernels of 3×3 with stride 1 and padding 1. LeakyReLU is used as the activation function. Additionally, InstanceNormalization is applied in the first four layers. The final

layer changes the number of channels from 512 to 3, followed by an SSIM-Loss function layer that outputs the final discriminative matrix. The top input is the image generated by the generator, while the bottom input is the original image.

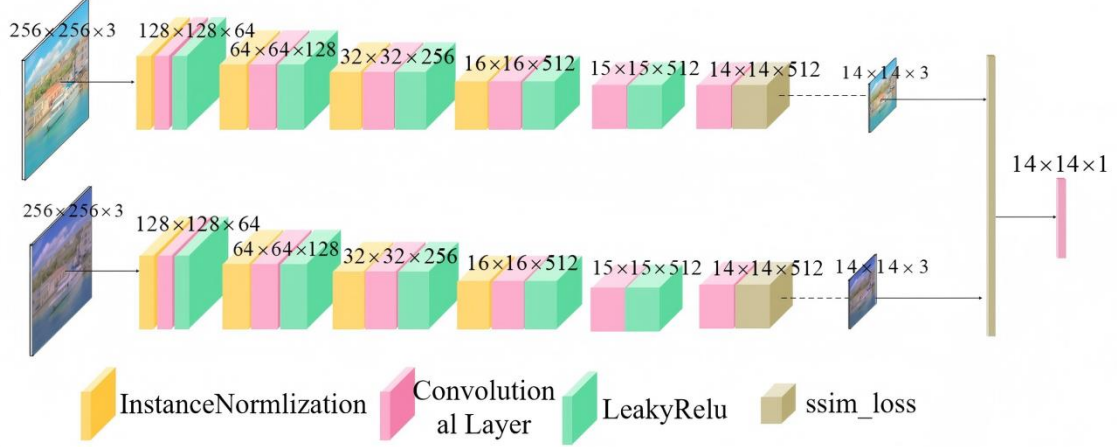


Figure 6: Discriminator D2 network structure

3.5 Loss function

The loss function designed in this article mainly consists of two parts: adversarial loss and feature structure loss (Feature and Structure loss, FS loss). The adversarial loss is used to learn the differences between generated samples and real images. Additionally, considering the particularity of using dual discriminators, we introduce the feature structure loss to stabilize the training of discriminator D2. The overall loss function expression is given in Eq. (2):

$$L(G, D) = L_{GAN}(G, D) + \lambda L_{FS}(G, D_2) \quad (2)$$

Here, L_{GAN} represents the GAN loss, L_{FS} represents the feature structure loss, and λ is a weight hyperparameter. In this article's experiments, $\lambda=10$. D2 denotes the D2 discriminator.

The adversarial loss consists of two components: generator network loss and discriminator network loss. Their expressions are given in (3)~(5):

$$L_{GAN}(G, D) = L_G + L_D \quad (3)$$

$$L_G = E_{x \sim P_G} [D(G(x, y))] \quad (4)$$

$$L_D = E_{x \sim P_G} [D_1(G(x, y))] + E_{y \sim P_{D_1}} [D_2(D_1(G(x, y)))] \quad (5)$$

Here, L_G is the loss of the generator network, L_D is the loss of the discriminator network, x represents the input image, y denotes the generated image, D1 refers to the D1 discriminator, and D2 refers to the D2 discriminator.

The feature structure loss L_{FS} calculates the structural dissimilarity between the generated image and the input image. To preserve low-frequency information in the images, an L1 Loss is added to the loss function. L_{FS} is shown in Equation (6):

$$L_{FS}(x, y) = 1 - \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} + c_3 E(\|G(y) - x\|_1) \quad (6)$$

Here, x is the input image, y is the generated image, μ_x is the mean of x , μ_y is the mean of y , σ_x^2 is the variance of x , σ_y^2 is the variance of y , and σ_{xy} is the covariance between x and y ; $c1$, $c2$, and $c3$ are all constants.

To verify that the L_{FS} constructed in this article imposes effective constraints on the ELF-CandyGAN network, $L_{GAN}(G, D)$ and $L(G, D)$ functions are compared. The comparison result is shown in Figure 7, which presents a graph comparing the loss functions $L_{GAN}(G, D)$ and $L(G, D)$.

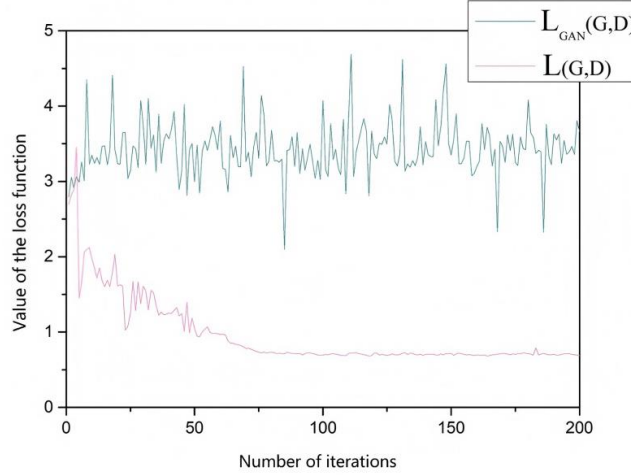


Figure 7: Comparison of the loss functions $L_{GAN}(G, D)$ and $L(G, D)$

As can be seen from the Figure 7, compared to using only $L_{GAN}(G, D)$, during the entire network training process the loss function does not show a converging trend and exhibits significant overall fluctuations. This results in the model failing to converge throughout training, requiring a longer training time to achieve ideal performance. In this article, we introduce the L_{FS} loss function to help balance the training process between the generator and discriminator, as shown in the Figure 7, compared to $L_{GAN}(G, D)$, the value of the $L(G, D)$ loss function is significantly lower overall, starting to converge at around 50 iterations and gradually stabilizing during later stages of training. This indicates that the L_{FS} loss function helps accelerate the convergence speed of the model, making the training process more efficient.

4. Experimental Results and Analysis

4.1 Dataset

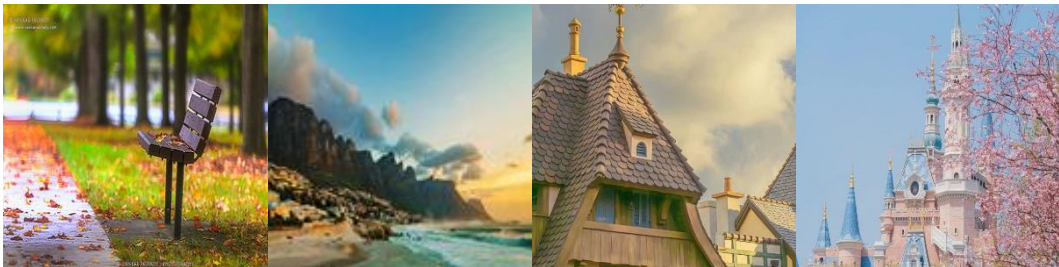


Figure 8: Scenic Spot and Landscape Image Dataset

The dataset in this article was collected through mobile phone photography and online sources, as shown in Figure 8: the landmark and scenic image dataset. This dataset contains a total of 6287 images across 8 categories, with each category containing approximately 600 to 800 images, evenly

distributed in quantity. The image sizes are respectively 1024×1024, and the total file size of the dataset is 2.1GB.

4.2 Experimental Setup

The experimental environment is based on the Linux operating system and the PyTorch deep learning framework, with an NVIDIA GeForce RTX 4090 (GPU).

Regarding the parameters during the training process, after comparing experimental results and considering the actual GPU memory capacity, the final parameter settings were chosen as follows: The Batchsize was set to 10, the number of epochs was set to 200 (number of iterations), the ADAM optimization algorithm was used with decay rates β_1 set to 0.5 and β_2 set to 0.999. Additionally, the network learning rate was set to 0.00003. All experiments were conducted on a single computer system.

4.3 Experimental Results and Analysis

4.3.1 Experimental evaluation metrics

Inception Score (IS) is a standard often used to measure the quality of generated samples from generative models such as GANs. It is defined as shown in Equation (7):

$$IS(P_g) = e^{E_{x \sim P_g} [D_{KL}(p(y|x} \| p(y))]} \quad (7)$$

Here, $x \sim p_g$ denotes the images generated by the generator, $p(y|x)$ represents the probability distribution of the generated x across all categories, $p(y)$ indicates the probability obtained from N generated images, and KL is used to measure the distance between two probability distributions.

User subjective evaluation (User Study)[13] subjectively assesses tasks that cannot be measured by fixed metrics through questionnaires. User subjective evaluations not only help accurately assess the strengths and weaknesses of each coloring algorithm but also provide suggestions for improving future algorithms.

PSNR is a full-reference image quality assessment method that uses mean squared error to measure the difference between the generated image and the input image, evaluating the similarity between the generated image and the ground truth image. A higher PSNR value indicates lower distortion in the generated image, meaning better image colorization results. The formula is shown in (8):

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (8)$$

To better evaluate the final image colorization results, the color richness metric (COLORFUL)[2] is used to measure the diversity of colors. This indicates the richness of colors in the colorized image and thus assesses the image quality, with larger values representing better performance. The calculation formula is shown in Equation (9):

$$COLORFUL = \sigma_{rgyb} + 0.3 \times \mu_{rgyb} \quad (9)$$

Here, σ_{rgyb} and μ_{rgyb} denote the standard deviation and mean value, respectively, along the matrix $rgyb$ resulting from the element-wise multiplication operation. As shown in Equations (10)~(11):

$$\sigma_{rgyb} := \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} \quad (10)$$

$$\mu_{rgyb} := \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \quad (11)$$

Here, σ_{rg} and μ_{rg} denote the standard deviation and mean along the matrix rg dot product operation, respectively, as shown in Equations (12)~(13):

$$rg = R - G \quad (12)$$

$$yb = \frac{1}{2}(R + G) - B \quad (13)$$

FLOPs are used to measure the training time, resource consumption, and inference efficiency of a model during training. The smaller the FLOPs, the better. As shown in Equation (14):

$$FLOPs = 2 \cdot K_H \cdot K_W \cdot C_{in} \cdot C_{out} \cdot H \cdot W \quad (14)$$

Here, K_H and K_W are the height and width of the convolution kernel, C_{in} is the number of input channels, C_{out} is the number of output channels, and H and W are the height and width of the output feature map, respectively.

4.3.2 Qualitative Analysis

To intuitively demonstrate the coloring performance of the proposed network model, we conduct comparative experiments between the improved algorithm ELF-CandyGAN and four existing algorithms: CandyCycleGAN, Bo Li, Hong'an Li, and Zongnan Chen. All comparative experiments are conducted under the same experimental environment. The results of the comparative experiments are shown in Table 1, which presents the evaluation metrics of the comparison experiments.

Table 1: Results of comparison experiments on evaluation metrics

Method	PSNR	IS	COLORFUL	FLOPs/B
Bo Li	20.8667	0.6928	36.9423	1018.3846
Hong'an Li	21.9942	0.7162	27.2753	932.9320
Zongnan Chen	24.0943	0.7936	45.2850	993.8429
CandyCycleGAN	30.9647	0.9318	53.8438	810.1649
ELF-CandyGAN	33.6791	0.9560	58.1097	786.5327


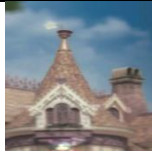


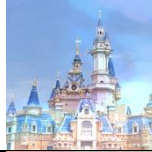










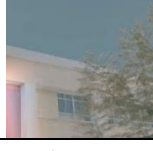
Note: Bold indicates the optimal value

The table records the average values of various metrics on the validation set. As shown in the table, ELF-CandyGAN achieves the best results across all metrics. Compared with the worst-performing method, Bo Li, ELF-CandyGAN reduces model computational cost by 22.77%, improves PSNR by 61.40%, IS by 37.99%, and COLORFUL by 57.30%. Compared with the second-best model, CandyCycleGAN, computational cost decreases by 2.92%, PSNR increases by 8.77%, IS improves by 2.6%, and COLORFUL rises by 7.92%. The comparative experimental results demonstrate that the ELF-CandyGAN model not only reduces computational cost and enhances model performance but also effectively mitigates artifacts present in images generated by the CandyCycleGAN model, thereby improving overall image generation quality. Based on objective metrics, the proposed method in this chapter demonstrates superiority over the other four algorithms, indicating its effectiveness.

To more intuitively illustrate the final coloring performance of the ELF-CandyGAN network,

comparison results between our model and the four other models are presented in Table 2: Comparison of Results Between This Work and Four Other Algorithms. From the experimental results, it can be observed that the ELF-CandyGAN network exhibits certain advantages in the final output. Compared to the final output images of the CandyCycleGAN network, the final output of the ELF-CandyGAN network displays richer detail features. Artifacts observed in some images generated by CandyCycleGAN are effectively addressed in the ELF-CandyGAN network, resulting in better preservation of input details, reduced information loss during training, and effective retention of low-frequency components, leading to higher expressive power in the final images. Among them, the image coloring result of Hong'an's Li algorithm is more consistent with the original color of the image. The overall color brightness value of the image after Zongnan's Chen algorithm coloring is too low, the color saturation is too high, and the contrast between colors is excessive, which does not meet the candy color feature requirements. Moreover, these two algorithms-Hong'an Li and Zongnan Chen suffer from inaccurate coloring regions, insignificant image detail features, and blurred edges during the coloring process. Bo Li's output results are based on reference example images and appear relatively worst in coloring performance compared to the other four algorithms, showing less image detail and more image noise artifacts.


Table 2: Comparison of results between this article and four other algorithms

Method Real image	ELF- CandyGAN	Candy- CycleGAN	Bo Li	Hong'an Li	Zongnan Chen
					
					
					
					
					

To ensure fairness in the experimental data, a total of 48 participants were involved in the user evaluation method described in this article. Among them, there were 24 participants aged between 18 and 35 years, 12 participants aged between 35 and 55 years, and another 12 participants aged between 55 and 65 years. The gender ratio was controlled at 1:1. Each participant was shown four comparison experiments along with effect images generated by the ELF-CandyGAN network. Without being informed which coloring model corresponds to each colored image, participants were asked to select

within five seconds the image they personally considered to have the best coloring effect. The results are shown in Table 3 Comparison of User Evaluations for Coloring Experiments Across Different Models. The value R on the right side of each image represents the percentage of participants who believed that the corresponding model achieved the best coloring effect. It can be seen from the table that the percentages obtained by the ELF-CandyGAN network are the highest across all metrics. Experimental results indicate that the coloring effects produced by the ELF-CandyGAN network are more widely accepted by users.

Table 3: User Evaluation Comparison of Coloring Experiments on Different Models

Images	ELF-CandyGAN	R%	Candy CycleGAN	R%	Bo Li	R%	Hong'an Li	R%	Zongnan Chen	R%
		48.5		39.6		0		0		11.9
		53.6		40.7		0		0		5.7
		45.6		43.5		0		2.2		8.7
		50.1		42.3		0		0		7.6
		58.7		39.3		0		1.2		0.8

Note: Bold indicates the optimal value.

4.3.3 Quantitative Analysis

To investigate the performance improvement of the ELF-CandyGAN network model compared to the baseline U-Net Generative Adversarial Network model, ablation experiments were conducted by comparing the baseline U-Net Generative Adversarial Network with the proposed ELF-CandyGAN network model in this article. The results are shown in Table 4 Ablation Experiment Results.

Table 4: Ablation Experiment Results

Network model	PSNR	IS	COLORFUL	FLOPs/B
U-Net	21.6293	0.5589	35.2910	1168.3790
U-Net + Color Learning	26.3672	0.5896	50.1793	991.3712
U-Net + Local Feature Enhancement	30.4790	0.7892	42.2730	961.8023
U-Net + Color Learning + Local Feature Enhancement	31.5802	0.7918	52.5843	920.4769
U-Net + D2	28.1949	0.6901	37.1939	876.3696
U-Net + D2 + FS Loss	30.2617	0.7768	45.2948	870.2793
U-Net + Color Learning + Local Feature Enhancement + D2 + FS Loss	35.2368	0.9583	60.2696	781.3649

Note: Bold indicates the optimal value.

From the results shown in the evaluation metrics, it can be observed that compared to the original U-Net, adding the color learning module and local feature enhancement module individually leads to slight improvements across all metrics. When both modules are added simultaneously, PSNR improves by 46.00%, IS improves by 41.67%, COLORFUL improves by 49.00%, and computational cost decreases by 21.22%. After incorporating the D2 discriminator, PSNR increases by 30.36% and computational cost reduces by 24.99%. Furthermore, when the FS loss is added upon the D2 discriminator, all metrics achieve additional small improvements, among which PSNR improves by 39.91%, IS improves by 38.99%, COLORFUL improves by 28.35%, and computational cost

decreases by 25.51%. The final ablation experiment involves the ELF-CandyGAN network, where all metrics show significant improvement compared to the U-Net network: PSNR improves by 62.91%, IS improves by 71.46%, COLORFUL improves by 70.78%, and computational cost decreases by 34.66%. These experimental results validate that the coloring capability of the ELF-CandyGAN network is credible.

4.4 Experimental Results

This article performs coloring based on RGB images, resulting in candy color-styled images that are rich in color and detailed features, aligning with the characteristics of candy color-high brightness, low contrast, and high saturation. From the experimental results, it is evident that this article significantly improves both image color representation and detail feature preservation compared to CandyCycleGAN. Additionally, it effectively reduces artifacts present in some generated images and emphasizes the preservation of low-frequency image information. The result visualization of the ELF-CandyGAN network is shown in Figure 9: ELF-CandyGAN Result Demonstration.

In this article, ELF-CandyGAN is designed to better achieve candy color-style coloring. A color learning module is designed to capture the unique color characteristics of candy color, while a global context module is introduced to better constrain the chrominance value range of the generated images, ensuring the output conforms to the candy color traits of high brightness, low contrast, and high saturation. Furthermore, a local feature enhancement module is designed to enrich the detail features of the output image and improve overall image quality. A D2 discriminator is also designed to enforce structural similarity between the generated and input images, reconstructing structural features from the input and enhancing training stability of the Generative Adversarial Network. Finally, a feature structure loss is designed to guide the ELF-CandyGAN network in preserving image structural information, avoiding distortions and blurring, thereby improving both the accuracy and stability of the coloring process.



Figure 9: ELF-CandyGAN Results Display

5. Conclusion

To better achieve candy color coloring, reduce network training time, minimize artifacts in generated images, and improve the final image generation quality, this article constructs the generator of this chapter using a U-Net network. We design a color learning module and a local feature enhancement module, and simultaneously develop a dual discriminator to construct the candy color coloring model. This article first introduces in detail the basic architecture and related concepts of the ELF-CandyGAN network. Secondly, it describes the dataset used for the candy color coloring task in the experiments and presents the hyperparameter settings of the model. Subsequently, it briefly explains the evaluation metrics used to assess the model. Finally, through qualitative analysis comparing with four algorithms-CandyCycleGAN, Bo Li, Hong'an Li, and Zongnan Chen-and quantitative analysis of the ELF-CandyGAN network itself, we verify the effectiveness of the ELF-CandyGAN network in the candy color coloring task.

References

- [1] Zongnan Chen, Yaoguang Ye, Jiahui Pan, *Grayscale Image Colorization Method Based on CycleGAN [J]*, *Computer Systems & Applications*, 2023, 32(08): 126-132, DOI: 10.15888/j.cnki.csa.009195.
- [2] Shisong Zhu, Mei Xu, Bibo Lu, et al. *CandyCycleGAN: Candy Color Coloring Algorithm Based on Chromaticity Verification[J]*. *Academic Journal of Computing & Information Science*, 2023, 6(13).
- [3] Wenhua Ding, Junwei Du, Lei Hou et al., *Fashion Content and Style Transfer Based on Generative Adversarial Network [J/OL]*, *Computer Engineering and Applications*: 1-11 [2023-07-05].
- [4] Wenhui Qin, Yilai Zhang, *Design and Implementation of Multi-school Style Transfer Algorithm[J]*, *Fujian Computer*, 2023, 39(04):42-48[DOI:10.16707/j.cnki.fjpc.2023.04.008].
- [5] Jiawei Zhou, Zikang Huang, Junhao Peng, et al. *image recoloring Based on Moving Least Squares[J]*. *Journal of Zhejiang University (Science Edition)*, 2025, 52(01): 10-21.
- [6] Sihong Meng, Hao Liu, Haotian Fang, et al. *image colorization Based on semantic similarity propagation[J]*. *Journal of Graphics*, 2025, 46(01): 126-138.
- [7] Hongan Li, Qiaoxue Zheng, Jing Zhang, et al. *Grayscale Image Colorization Method Combined with Pix2PixGenerative Adversarial Network [J]*. *Journal of Computer-Aided Design & Computer Graphics*, 2021, 33(06): 929-938.
- [8] LIANG Z X,LI Z C,ZHOU S C,et al. *Control color:multimodal diffusion-based interactive image colorization[J]*. 2024,05,07.<https://arxiv.org/abs/2402.10855>.
- [9] Shi W,Caballero J,F Huszar,et al,*Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network[C]*,*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016:1874-1883.
- [10] Cao Y,XuJ,Lin S,et al,*Gcnet: Non-local networks meet squeeze-excitation networks and beyond[C]*,*Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*,2019.
- [11] Ruan Yiting, *Research on Interactive Line Art Coloring Method Based on Generative Adversarial Network [D]*, *Zhejiang University*, 2023, DOI:10.27461/d.cnki.gzjdx.2022.000937.
- [12] He Wenhao, Ge Haibo, *Camouflaged Object Detection Method Based on Local-Global Feature Mutual Compensation [J/OL]*, *Journal of Computer Science and Technology*: 1-15 [2024-03-11], <http://kns.cnki.net/kcms/detail/11.5602.TP.20240226.1044.006.html>.
- [13] Zhou Wang, A, C, Bovik, H, R, Sheikh, et al. *Image quality assessment: from error visibility to structural similarity[J]*. *IEEE Transactions on Image Processing*, 2004, 13(4): 600-612.