Semantically Aware Style-Controlled Animation Line Art Colorization Using Conditional GANs with GCN and Attention Mechanisms

Chen Li
School of Art and Design, Sanming University, Sanming 365000, Fujian, China
E-mail: LiChen_0513@outlook.com

Keywords: Generative adversarial networks, animation line art, automatic coloring, conditional generative adversarial network, color consistency

Received: August 5, 2025

In traditional animation production, coloring line art is a labor-intensive and time-consuming process. In this study, image processing techniques based on generative adversarial networks (GANs) were investigated to develop an algorithm for the automatic coloring of animation line drawings. The objective was to improve production efficiency, reduce the workload of artists, and generate outputs that are natural in appearance, consistent in style, and closed along region boundaries. The proposed method was implemented within the conditional GAN framework. The generator adopted a U-Net architecture with skip connections, which allowed both fine-grained details and global structural features of the line art to be captured. A spectrally normalized discriminator was used to evaluate the realism of local image regions. To improve semantic accuracy and color coherence, an attention mechanism was incorporated, enabling the model to focus on key semantic areas and learn dependencies between color regions. End-to-end training was conducted using a large-scale paired dataset of line art and corresponding colored images. A multi-task learning strategy combining perceptual loss, L1 loss, and adversarial loss was employed for optimization. Latent space interpolation was further introduced to allow limited user adjustment of color styles. Experimental results indicated that the algorithm achieved a PSNR of 30.2 dB and an SSIM of 0.94, which represented improvements of 2.2 dB and 0.04 over the Structure Probe Adaptive Differential Evolution (SPADE) baseline, respectively. The FID and boundary overflow rate were reduced to 18.3 and 2.1%, also showing clear improvements over the baseline. With the inclusion of structural consistency loss, graph convolutional networks, and self-attention mechanisms, the method maintained accurate boundary preservation, achieved cross-region color consistency, and supported adaptive style rendering. In summary, the algorithm addressed key limitations of existing approaches, such as color overflow, semantic inconsistency, and rigid stylization. These findings demonstrate improvements in automatic coloring quality and suggest the potential of GAN-based techniques in animation production.

Povzetek: Študija predstavi cGAN-okolje z U-Net generatorjem, spektralno normaliziranim diskriminatorjem, pozornostjo in večnamensko izgubo za samodejno barvanje linijskih risb, ki ohrani robove, zagotovi slogovno skladnost ter izboljša učinkovitost animacijske produkcije.

1 Introduction

In the contemporary digital animation industry, line art coloring remains a critical but challenging stage in the production pipeline. The process is often constrained by efficiency bottlenecks and the requirement to maintain consistent artistic quality [1]. Traditional workflows rely heavily on professional artists who manually or semi-automatically color each frame. This approach is labor-intensive, time-consuming [2], and dependent on specialized expertise in color perception and detail management. With increasing demand for animation content and tighter production schedules, conventional methods face difficulties in balancing efficiency with quality. Recent advances in computer vision and deep learning, particularly the success of generative adversarial network (GAN) in image generation and style transfer, have introduced opportunities for automating the coloring process [3, 4]. Nevertheless, current methods based on convolutional neural networks (CNNs) or early GAN architectures encounter several limitations [5]. First, generated outputs often contain structural issues such as color bleeding and blurred boundaries, which compromise the region closure required in animation. Second, these methods demonstrate limited semantic understanding, leading to color assignments that may not align with character or scene attributes. Third, adaptation to diverse artistic styles is insufficient, resulting in inconsistencies across productions. Finally, most approaches depend on large-scale, finely annotated line art–color pairs, which are costly and impractical for animation studios to acquire [6-8]. These challenges restrict the applicability of existing automatic coloring techniques.

This study investigates an automatic line art coloring algorithm based on advanced GAN architectures. The objective is to address the shortcomings of current methods by incorporating multimodal semantic analysis with generative modeling techniques [9]. Specifically, the aims are as follows: (1) To develop an encoder capable of accurately parsing the geometric and semantic structures of line drawings, ensuring that color filling conforms to closed region boundaries; (2) To construct a generative model that produces cross-region color consistency, reflecting realistic lighting and scene coherence; (3) To design a training framework supporting multi-style adaptation, enabling lightweight user interactions for style guidance; and (4) To establish an end-to-end pipeline that reduces the repetitive workload of professional artists while maintaining visual quality.

Based on these objectives, this study addresses three core research questions. First, can the introduction of a structural consistency loss function effectively reduce boundary color overflow in line drawings caused by unclosed contours? The effectiveness of this approach is evaluated by measuring reductions in boundary overflow rates. Second, does the incorporation of a graph convolutional network (GCN) within the GAN framework enhance the model's understanding of scene-level semantics, thereby improving cross-region color coordination? This capability is assessed through decreases in color consistency error (CCE). Third, how can latent style space disentanglement techniques support user-guided style transfer? The effectiveness is validated by improvements in Fréchet Inception Distance (FID) scores and by preference ratings from professional artists.

To answer these questions, a generator architecture is designed by combining conditional GAN with a U-Net backbone. Dense skip connections and multi-scale feature fusion are employed to model line art topology and mitigate color overflow. Self-attention mechanisms (SAMs) and GCN are integrated to capture semantic relationships across regions, enhancing the coherence of color assignment. A spectrally normalized discriminator is optimized jointly with perceptual and adversarial losses to improve realism and texture fidelity. In addition, a style adjustment interface based on latent space interpolation allows users to fine-tune outputs with a small set of reference images, improving adaptability to production requirements. Compared with existing approaches, the proposed framework introduces a new integration scheme for animation coloring. It employs a U-Net backbone for structural parsing, utilizes GCN for cross-region semantic reasoning, and incorporates SAMs with adaptive style injection for fine-grained control. This combination addresses three key challenges in automatic line art coloring: structural fidelity, semantic color consistency, and flexible style adaptation.

2 Related work

In recent years, GAN has demonstrated significant potential in image colorization, with applications spanning artistic creation, cultural heritage preservation, and professional design. Treneska et al. (2022) proposed a GAN-based self-supervised framework in which colorization was used to drive the model's understanding of structural image features. Their results showed that colorization enhanced representational learning [10]. However, their work primarily focused on natural images and did not address the geometric sparsity and stylized nature of animation line art. Wu et al. (2022) tackled the fine-grained colorization of traditional ethnic clothing using the conditional generative adversarial networks (cGANs) to enable semantically-aware coloring of high-resolution apparel images [11]. While local constraint mechanisms improved texture details, the model struggled to generalize to non-rigid objects such as animated characters in dynamic poses. Later, Wu et al. (2023) explored multi-level feature fusion for Chinese ink painting colorization, employing GANs to mimic ink diffusion effects [12]. Yet, the physical properties of traditional media differ fundamentally from the cel-shaded style of animation, limiting the approach's transferability to industrial animation workflows.

Zhou et al. (2025) introduced an Asymmetric GAN designed to correct skin tone white balance by integrating skin-tone priors, improving physiological plausibility in portrait colorization [13]. However, in animation, characters often feature fantastical skin colors, and strong priors may constrain creative freedom. Al-Ghanimi et al. (2025) proposed a Vision Transformer (ViT)-based method for grayscale image colorization with a hybrid loss function to enhance color coherence [14]. Despite its strengths, ViT's high computational cost conflicts with the real-time processing demands of animation line art. Dalal et al. (2021), in a survey of automated colorization techniques, noted that existing deep learning models often suffered from color ambiguity in complex scenes and lacked user-controllable interfaces [15]. Chen et al. (2024) applied cGANs for rapid colorization of park landscape sketches, and demonstrated the potential of generative methods to boost design efficiency [16]. However, their model assumed low line closure requirements, making it unsuitable for the precise, often non-closed regions common in animation.

Recent studies have aimed to improve colorization quality through architectural innovations. Jampour et al. (2023) introduced a Multi-GANs system that enforced spatiotemporal consistency to ensure coherent colorization across video frames [17], and offered insights for animation sequence processing. However, their method did not address semantic ambiguity within individual line art frames. Mourchid et al. (2023) proposed Symmetric Positive Definite Generative Adversarial Network (SPDGAN), which used symmetric positive definite manifold learning and geometric priors to enhance the naturalness of color distributions [18]. Despite its conceptual strengths, the abstract mathematical formulation adds complexity to real-world implementation. Yang et al. (2025) developed a degradation-model-driven approach for colorizing fundus images, emphasizing the importance of domain adaptation in medical imaging [19]. Yet, the subjective aesthetic demands of animation far exceed the objective reconstruction goals in the medical domain. Al-Ghanimi et al. (2025) explored context-aware unsupervised colorization by integrating contextual content modules to enhance semantic consistency [20]. Nonetheless, their model did not explicitly address the hierarchical semantic relationships among characters, props, and backgrounds, which are crucial in animation line art.

A review of existing research identifies representative methods for automatic coloring of animation line art, summarized in Table 1. These approaches have demonstrated progress in areas such as natural images, cultural artworks, and medical imaging. However, they also present several limitations. First, most methods do not adequately address unclosed contours that are common in animation line drawings, often resulting in color overflow and blurred boundaries. Second, the absence of explicit modeling of animation-specific semantic hierarchies—such as characters, clothing, and backgrounds—leads to color assignments that fail to reflect scene logic. Third, style control is generally weak, limiting the ability to adapt to diverse artistic styles. Finally, many methods depend on large-scale, high-quality paired datasets, which are difficult to obtain in practical animation production.

Table 1: Summary of automatic coloring methods for animation line art

Method	PSNR (dB)	SSIM	FID	Style Control	Application Domain	Boundary Preservation	Semantic Understanding
cGAN [11]	26.8	0.88	26.3	Moderate	Ethnic costumes	Moderate	Moderate
ViT-Based [14]	29.1	0.93	19.8	None	Grayscale images	Moderate	Moderate
Multi-GANs [17]	28.0	0.91	22.5	Weak	Natural images	Moderate	Partial
SPDGAN [18]	28.5	0.92	20.1	Weak	Natural images	Strong	Weak
Proposed model	30.2	0.94	18.3	Strong	Animation line art	Strong	Strong

To address these limitations, this study proposes a framework that integrates structural constraints with semantic understanding. A structural consistency loss function is introduced to explicitly manage unclosed boundaries. GCNs model semantic relationships across regions, while SAMs enhance sensitivity to critical semantic areas. Latent space interpolation further enables user-guided style transfer, improving adaptability to diverse artistic styles. By combining these architectural components, the framework effectively mitigates common issues such as color overflow and semantic inconsistency. The addition of interactive style control also increases its practicality in real production scenarios. Overall, the proposed approach offers a systematic solution for achieving high fidelity, semantic consistency, and controllable style in the automatic coloring of animation line drawings.

3 Method

3.1 Overall framework design

This study proposes an automatic colorization framework for animation line art based on cGANs, with the primary goal of generating semantically aware and stylistically controllable color outputs while preserving structural fidelity. As illustrated in Figure 1, the system follows an end-to-end training paradigm. The input consists of a single-channel animation line drawing $x \in R^{H\times W\times 1}$, and an optional reference color image $r \in R^{H\times W\times 3}$ (used for style guidance). The output is a colorized RGB image $y' \in R^{H\times W\times 3}$. The generator G is responsible for synthesizing the color image from the line drawing x and a latent style vector z. The discriminator D is trained adversarially to distinguish between the generated output y' and the real colored ground truth y.

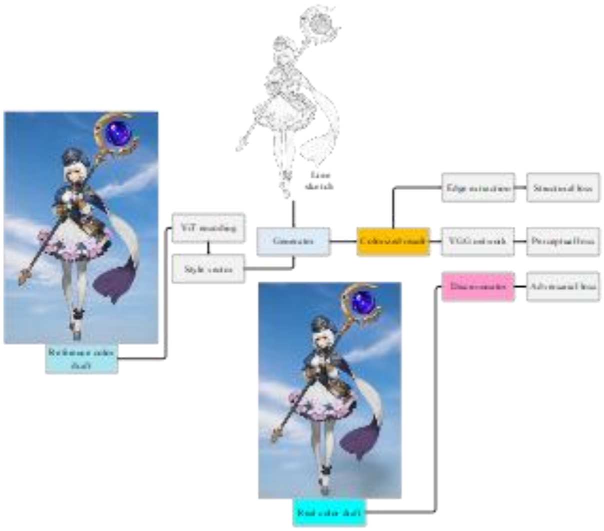

Figure 1: Automatic colorization framework for animation line art based on cGANs

The training objective of the framework is achieved through the joint optimization of a multi-objective loss function. This overall loss consists of three key components, each designed to address a specific challenge in automatic coloring. The first is the conditional adversarial loss, denoted as $\mathcal{L}_{cGAN}$, which encourages the generator to produce color distributions that are indistinguishable from those of real images, thereby ensuring overall realism. The second is the perceptual loss, denoted as $\mathcal{L}_{perc}$, which constrains semantic consistency by comparing high-level feature maps between generated and real images using a pretrained deep convolutional network. The third is the structural consistency loss, denoted as $\mathcal{L}_{struct}$. This component is specifically designed to address the common problem of unclosed contours in animation line drawings. By imposing gradient constraints along line edges, it effectively suppresses color overflow in the generated results. To balance these three loss terms, which differ in scale and objectives, two adjustable positive weighting coefficients, λ_{perc} and λ_{struct}, are introduced to control the relative importance of perceptual loss and structural consistency loss. The mathematical essence of this framework lies in solving the following optimization problem, as shown in Equation (1):

$$min_G max_D \mathcal{L}_{cGAN}(G,D) + \lambda_{perc}\mathcal{L}_{perc}(G) + \lambda_{struct}\mathcal{L}_{struct}(G) \quad (1)$$

$\mathcal{L}_{cGAN}(G,D)$ represents the conditional adversarial loss, which enforces the generated distribution to approximate the real data distribution. λ_{perc} denotes the perceptual loss, which constrains the consistency of high-level semantic features. λ_{struct} represents the structural consistency loss, which suppresses color overflow. Both λ_{perc} and λ_{struct} are balancing weights.

3.2 Generator network design

The generator G serves as the core component of the framework. Its design objective is to generate semantically aware colorization and enable user-controllable style transfer, based on accurate parsing of the topological structure of the line art. As shown in Figure 2, G adopts a multi-branch fusion architecture comprising four key submodules: an enhanced U-Net backbone, a graph-based relational reasoning branch, a spatial attention branch, and a style-adaptive decoding module. The input line art x is first encoded into multi-scale features. Semantic enhancement is then performed at the bottleneck layer via dual branches, and the final colorized image y' is generated by a style-injected decoder. The generator in this study adopts a multi-branch fusion architecture. The single-channel line art input is first processed by an encoder, which extracts multi-level features through downsampling. At the bottleneck layer, a dual-path semantic enhancement unit is introduced to perform both graph-based reasoning and long-range dependency modeling. In the graph convolution branch, the feature map is segmented into superpixels to define semantic nodes, and a region adjacency graph is then constructed. Graph convolution operations are applied to explicitly model color association rules across regions.

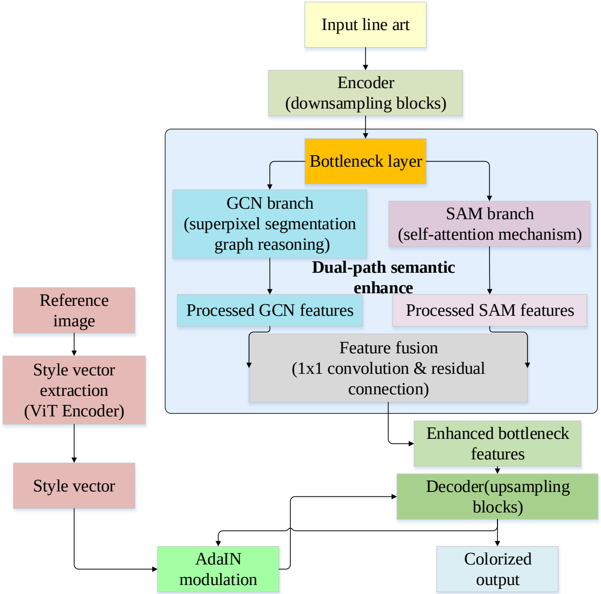
Input line art
Encoder (downsampling blocks)
Bottleneck layer
GCN branch (superpixel segmentation graph reasoning)
SAM branch (self-attention mechanism)
Dual-path semantic enhance
Processed GCN features
Processed SAM features
Feature fusion (1x1 convolution & residual connection)
Reference image
Style vector extraction (ViT Encoder)
Style vector
Enhanced bottleneck features
Decoder(upsampling blocks)
AdaIN modulation
Colorized output

Figure 2: Generator network architecture

In parallel, the self-attention branch computes correlations among query–key pairs, dynamically capturing global semantic context and emphasizing key regions. The outputs of the two branches are fused using a 1×1 convolution and linked to the original bottleneck features through a residual connection, ensuring that all information is preserved. The fused features are then passed into the decoder for upsampling and reconstruction. In addition, a style vector extracted from a reference image is injected into each decoder layer through adaptive instance normalization (AdaIN). This process modulates the mean and variance of feature distributions, thereby enabling style transfer. Finally, the decoder produces high-quality colorized results.

The encoder consists of five convolutional layers, each comprising: (1) A convolutional layer with a kernel size of 3×3 and stride of 2 for feature downsampling; (2) Spectral normalization (SN) to constrain the singular values of the weight matrix; (3) A LeakyReLU activation function with a negative slope of 0.2. Let the input feature map at level l be $f_l \in R^{H_l \times W_l \times C_l}$; the output feature map f_{l+1} is computed as shown in Equation (2):

$$f_{l+1} = LeakyReLU(SN(W_l * f_l)) \quad (2)$$

where W_l denotes the convolution weights and $*$ indicates the convolution operation. The decoder employs symmetric transposed convolutions for upsampling. At each level, it integrates the encoder's same-scale features f_l via dense skip connections to preserve shallow geometric information [21]. The fusion process is defined in Equation (3):

$$g_l = \phi([U_p(g_{l+1}), f_l]) \quad (3)$$

g_l represents the level-l decoding feature map, $U_p(\cdot)$ denotes bilinear upsampling, $[\cdot,\cdot]$ indicates channel-wise concatenation, and ϕ is a 1×1 convolution used for channel compression. This design significantly enhances the model's ability to perceive unclosed contours, providing structural constraints for color filling [22].

At the U-Net bottleneck layer (i.e, the lowest-resolution feature map $f_b \in R^{H_b \times W_b \times C_b}$, a dual-path semantic enhancement unit is embedded. The graph-based relational reasoning branch divides f_b into N regions $\{R_i\}_{i=1}^{N}$ using superpixel segmentation. Superpixel segmentation is implemented using a variant of the Simple Linear Iterative Clustering (SLIC) algorithm. Unlike conventional approaches with a fixed number of regions N, this method adapts N to the resolution of the bottleneck feature map f_b. The segmentation aims to partition the image into approximately 200 superpixels, ensuring that each region corresponds to a semantically meaningful local structure (e.g., a sleeve or a strand of hair). The exact number of regions is dynamically determined based on image content, enabling more accurate capture of the hierarchical structures commonly present in animation line art. Each region is represented by a node feature $v_i \in \mathbb{R}^{C_b}$. A region adjacency graph $G = (V, E)$ is constructed, where the edge weight e_{ij} between nodes v_i and v_j is determined by spatial distance and feature similarity [23], as defined in Equation (4):

$$e_{ij} = \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_p^2}\right) \cdot \exp\left(-\frac{\|v_i - v_j\|^2}{2\sigma_v^2}\right) \tag{4}$$

p_i denotes the center coordinates of region R_i, and σ_p, σ_v are tunable parameters. Based on this, an undirected graph G is constructed, where the node set V is represented as a matrix of N feature vectors, $V \in \mathbb{R}^{N \times C_b}$, and the adjacency matrix $A \in \mathbb{R}^{N \times N}$ is computed according to Equation (4). The graph construction process can be summarized as follows: first, the spatial distances and feature similarities between all pairs of nodes are calculated; next, the joint weight e_{ij} for each node pair is computed using Equation (4) to initialize the adjacency matrix A ; finally, A is symmetrically normalized to prepare it for subsequent graph convolution operations. A two-layer GCN aggregates neighborhood information [24], as shown in Equation (5):

$$v_i' = ReLU\left(W_{gcn} \sum_{j \in \mathcal{N}(i)} \frac{e_{ij}}{\sqrt{d_i d_j}} v_j\right) \tag{5}$$

In Equation (5), $d_i = \sum_j e_{ij}$ is the node degree and W_{gcn} denotes the learnable weights. This operation explicitly models the color correlation rules between characters and scenes, or costumes and props.

The spatial attention branch incorporates a SAM to capture long-range dependencies [25]. For the feature map f_b, the query matrix Q, key matrix K, and value matrix V are computed as shown in Equation (6).

$$Q = W_q f_b, K = W_k f_b, V = W_v f_b \tag{6}$$

The attention weights A and the output features f_{att} are defined in Equation (7):

$$A = Softmax\left(\frac{QK^T}{\sqrt{C_b}}\right), f_{att} = AV \tag{7}$$

W_q, W_k, and W_v are the 1×1 convolution projection weights. This module enables the model to focus on key semantic elements (e.g, character pupils, weapon textures) while suppressing background interference. The outputs of the dual branches are fused via residual connections, as shown in Equation (8):

$$f_b' = f_b + Conv_{1\times 1}([f_{gcn}, f_{att}]) \tag{8}$$

f_{gcn} is the GCN output resampled to the original size, and $Conv_{1\times 1}$ represents the channel fusion convolution.

To support user-guided style transfer, the decoder at each level injects the style information from a reference color draft r. A pre-trained ViT is used to extract the style vector $z \in \mathbb{R}^{d_z}$ from r. Specifically, a ViT base model pretrained on ImageNet-21k is employed as the feature extractor. The global class token from the penultimate Transformer block is used as a representation of the overall image style. This representation is then projected into a latent style space of dimension d_z via a learnable three-layer multilayer perceptron (MLP) to obtain the style vector z. At decoder level l, the feature map g_l is modulated via AdaIN, as defined in Equation (9).

$$AdaIN(g_l, z) = \sigma_l(z) \cdot \frac{g_l - \mu(g_l)}{\sigma(g_l)} + \beta_l(z) \tag{9}$$

$\mu(\cdot)$ and $\sigma(\cdot)$ denote the channel-wise mean and standard deviation, while $\sigma_l(\cdot)$ and $\beta_l(\cdot)$ are affine parameters predicted from the style vector z via fully connected networks. The affine transformation parameters are predicted by two independent MLP, with each decoder level l having its dedicated predictors for the scale parameter σ_l and shift parameter β_l. This design allows style modulation to adapt to the distinct statistical properties of feature maps at different decoding stages. This mechanism aligns the statistical characteristics of the content features g_l with those of the style vector z, enabling disentangled artistic style transfer. It should be emphasized that superpixel segmentation is applied to the bottleneck feature maps rather than the original line art. Although animation line drawings often contain numerous unclosed contours, which pose challenges for pixel-level color assignment, the core idea of this method is to define “regions” at the semantic rather than strict geometric level. SLIC groups deep features based on similarity, producing semantically coherent nodes. These node regions may correspond to meaningful semantic parts (e.g., a face, a sleeve), regardless of whether the original line art contours are fully closed. GCNs then perform relational reasoning among these semantic nodes to learn color association rules across regions. The risk of color overflow caused by unclosed contours is explicitly constrained at the pixel level using a dedicated structural consistency loss. By separating region relationship modeling at the semantic level from boundary constraint enforcement at the pixel level, this approach effectively balances adaptability to unclosed contours with the requirement for consistent region-wise color assignment.

3.3 Discriminator and loss functions

The performance optimization in this study relies on adversarial guidance from the discriminator and the joint constraint of a multi-objective loss function. As illustrated in Figure 3, the discriminator D adopts a multi-scale convolutional architecture with spectral normalization, focusing on evaluating the realism of local image regions. The loss function system integrates three objectives—adversarial learning, semantic fidelity, and structural consistency—that collaboratively guide the generator’s optimization trajectory.

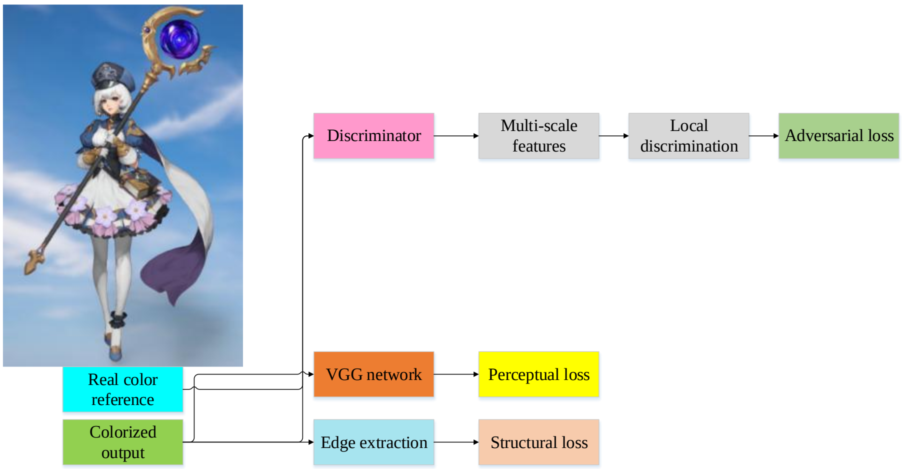

Figure 3: Discriminator architecture design

The discriminator D is designed based on the concept of a Markovian Discriminator and implemented as a Fully Convolutional Network (FCN). Its core innovation lies in combining multi-scale feature extraction with spectral normalization (SN) for training stability. The input, either the generated result y' or the ground-truth color reference y (both concatenated with the corresponding line art x), is processed through four convolutional layers. Each layer consists of a 4×4 convolution (stride = 2), followed by spectral normalization and a LeakyReLU activation (slope = 0.2). The feature map resolution decreases progressively (256×256 → 128×128 → … → 16×16). The final output is a discrimination matrix $D(y) \in \mathbb{R}^{M \times M}$ (default M=16), where each element $D_{ij}(y)$ indicates the probability that the corresponding image patch is real. Spectral normalization stabilizes training by constraining the Lipschitz constant of the weight matrix W [26], as shown in Equation (10):

$$W_{SN} = \frac{W}{\sigma(W)} \quad (10)$$

where $\sigma(W)$ denotes the spectral norm (i.e, the largest singular value) of W. This technique effectively mitigates mode collapse and improves generative diversity. The multi-scale design is realized by cascading feature maps of varying resolutions, as expressed in Equation (11):

$$F_{multi} = [AvgPool(f_1), AvgPool(f_2), f_3, f_4] \quad (11)$$

where f_k is the feature map at scale k, and $AvgPool(\cdot)$ denotes adaptive average pooling. This architecture allows D to perceive both local texture details and global structural consistency.

The generator G is optimized using a combination of three loss functions:

(1) Conditional Adversarial Loss:

Based on the Least Squares GAN (LSGAN) framework, this loss enhances gradient stability, as defined in Equation (12):

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[(D(x, y) - 1)^2] + \mathbb{E}_x[D(x, G(x, z))^2] \quad (12)$$

where z is the style vector. This loss drives the distribution of the generated output $G(x, z)$ closer to the real data distribution. The study adopts the LSGAN framework instead of the standard minimax adversarial loss primarily because it provides smoother and non-saturating gradient signals for the discriminator. In animation coloring, the generator must learn to map sparse line drawings to dense, fully colored images, a highly ill-posed problem. By minimizing the mean squared error between the discriminator outputs and the real/generated labels, LSGAN effectively mitigates the gradient vanishing issues commonly encountered in traditional GAN training. This approach ensures more stable updates for the generator throughout training, accelerates convergence, and improves the quality of the final generated images.

(2) Perceptual Loss:

This loss leverages a pre-trained VGG-19 (Visual Geometry Group-19) network to extract high-level semantic features [27], thereby enforcing content consistency, as shown in Equation (13):

$$\mathcal{L}_{perc}(G) = \sum_{l \in \mathcal{S}} \lambda_l \|\phi_l(y) - \phi_l(G(x, z))\|_1 \quad (13)$$

where $\phi_l(\cdot)$ denotes the features from the l-th layer of VGG-19 (layers such as *conv3_3* and *conv4_3* are selected), λ_l is the weighting coefficient, and $\mathcal{S}$ is the set of selected layers. Specifically, the study selects the outputs of the conv3_3 and conv4_3 layers of the VGG-19 network as perceptual features. Based on empirical evaluation across different reconstruction tasks, the corresponding weight coefficients are set to $\lambda_{conv3\_3} = 1.0$ and $\lambda_{conv4\_3} = 0.5$. This weighting strategy aims to balance the preservation of mid-level texture details (dominated by conv3_3) with high-level semantic consistency (dominated by conv4_3), ensuring that the generated images maintain both structural fidelity and perceptual realism. This loss ensures that the semantic content of the generated image (e.g, character identity, object category) aligns with the original line art.

(3) Structural Consistency Loss:

Designed specifically to handle unclosed contours in animation line art, this edge-aware constraint is defined in Equation (14):

$$\mathcal{L}_{struct}(G) = \left\|Edge(x) \odot \left(Sobel(y) - Sobel\big(G(x, z)\big)\right)\right\|_1 \quad (14)$$

where $Edge(x)$ is the Canny edge binary mask of the line art x, $Sobel(\cdot)$ computes gradient magnitude using the Sobel operator, and $\odot$ denotes element-wise multiplication. The Canny edge detector is configured with a dual-threshold scheme, where the high and low thresholds are set to 0.3 and 0.1 times the maximum grayscale value of the line art image x, respectively, to robustly extract the primary contour lines. The gradient magnitudes computed by the Sobel operator are min–max normalized to the range [0, 1] before being used in the loss calculation. This normalization ensures numerical stability and scale consistency when the gradient map is pointwise multiplied with the binary edge mask $Edge(x)$. This loss enforces sharp transitions at contour boundaries in the generated result and suppresses color bleeding.

The training was conducted on 4×NVIDIA A100 GPUs with mixed-precision acceleration, taking a total of 72 hours (48 hours for the first stage and 24 hours for the second). The complete hyperparameter settings are shown in Table 2.

Table 2: Training hyperparameter configuration

Parameter Category	Symbol	Stage 1	Stage 2
Batch Size	-	16	8
Base Learning Rate	η	0.0002	0.0001
Loss Weights	λ_{perc}	10.0	10.0
	λ_{struct}	5.0	5.0
	λ_{style}	-	2.0
Adam Parameters	β_1	0.5	0.5
	β_2	0.999	0.999
Training Iterations	T_{max}	200,000	100,000
Gradient Clipping	-	10.0	10.0

The model comprises 78.4 M parameters, with the main complexity arising from the generator's dual-branch semantic enhancement module and the deep encoder–decoder architecture. While achieving state-of-the-art performance, the model's size may limit its deployment in resource-constrained environments. Future optimization could focus on attention-based pruning to remove redundant feature channels, dynamic inference networks that adjust computation based on line art complexity, or post-training quantization to reduce computational and storage costs with minimal performance loss.

To ensure evaluation metrics reflect the core challenges of animation coloring, PSNR and SSIM were selected as primary measures. PSNR quantifies pixel-level differences between generated and ground-truth colored images, directly indicating color prediction accuracy. SSIM evaluates structural preservation, essential for maintaining the original composition and contour integrity. Although perceptual metrics derived from deep features capture semantic consistency, the primary concern in animation coloring is preventing color overflow and structural distortion, making PSNR and SSIM more interpretable and task-relevant. Performance is somewhat scene-dependent. For highly complex mechanical line art with severe occlusions or for extremely abstract, simplified sketches, results may degrade. Complex structures place higher demands on GCN-based region partitioning and relational reasoning, while abstract styles may deviate from the training distribution, increasing semantic parsing uncertainty.

3.4 Dataset and implementation details

Training and evaluation were conducted on a large-scale, publicly available paired dataset of anime line art and colored images, containing 100,000 high-quality pairs. Each pair consists of a 128×128 single-channel line drawing and its RGB counterpart. The dataset includes multiple mainstream animation styles (e.g., cel-shading, watercolor, pixel art) to ensure generalization. All images were professionally curated to guarantee accurate line-art-to-color correspondence. Prior to training, images were upsampled to 512×512 pixels to match the network input. The model was implemented in PyTorch and is publicly available on GitHub.

Training and inference efficiency were benchmarked to assess practical applicability. On a system with 4× NVIDIA A100 GPUs, training with 512×512 images and a batch size of 16 required about 48 hours to complete 200,000 iterations in the first stage, averaging about 35 ms per frame. During inference, a single A100 GPU processed a 512×512 line drawing in about 41 ms, producing high-quality results suitable for real-time interactive previews. A lightweight version further reduces per-frame inference time to 28 ms.

4 Results and discussion

4.1 Results

Figure 4 presents a comparison of overall performance across different methods. The proposed approach achieved a PSNR of 30.2 dB, outperforming the strongest baseline, Structure Probe Adaptive Differential Evolution (SPADE), by 2.2 dB—indicating a notable gain in pixel-level accuracy. It also reached a SSIM of 0.94, reflecting strong preservation of structural features. Additionally, the method recorded a low FID of 18.3, suggesting that the generated images closely match the real references in distributional characteristics. Most significantly, the boundary overflow rate was reduced to 2.1%, representing an 89% improvement over the traditional PaintsChainer. This result highlights the structural consistency loss's effectiveness in managing unclosed contours, a persistent challenge in animation colorization.

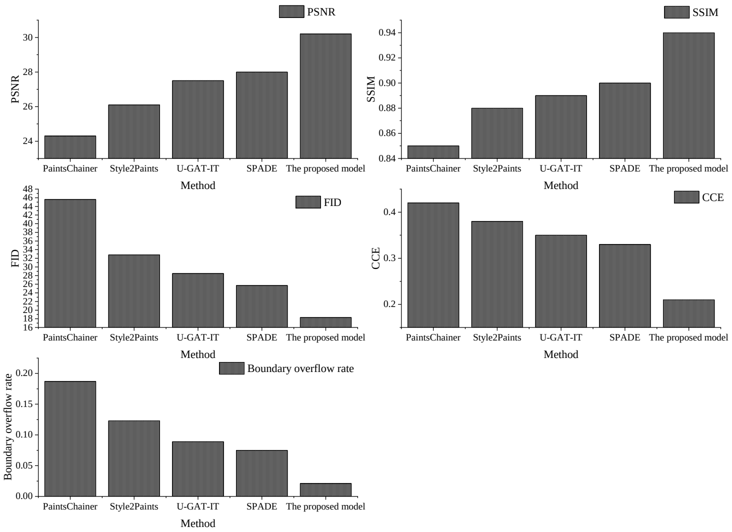

Figure 4: Overall performance comparison of different methods

To comprehensively evaluate the model's robustness, the mean and standard deviation of key evaluation metrics were computed across the entire test set, as shown in Table 3. The proposed model achieved the highest average performance in PSNR, SSIM, FID, and CCE. Moreover, its standard deviations were consistently lower than those of baseline methods, indicating not only superior performance but also more stable outputs across different samples, reflecting enhanced robustness. The lower standard deviation of FID suggests that the distribution of generated images is both closer to and more consistent with that of the real images. Similarly, the small standard deviation in CCE further confirms the model's consistent effectiveness in maintaining uniform color within semantic regions.

Table 3: Mean ± standard deviation of performance metrics on the test set

Method	PSNR (dB)	SSIM	FID	CCE
PaintsChainer	24.3 ± 1.8	0.85 ± 0.04	45.6 ± 3.2	0.42 ± 0.05
Style2Paints	26.1 ± 1.5	0.88 ± 0.03	32.8 ± 2.9	0.38 ± 0.04
U-GAT-IT	27.5 ± 1.3	0.89 ± 0.03	28.5 ± 2.5	0.35 ± 0.04
SPADE	28.0 ± 1.2	0.90 ± 0.02	25.7 ± 2.1	0.33 ± 0.03
Proposed Model	30.2 ± 0.9	0.94 ± 0.02	18.3 ± 1.8	0.21 ± 0.02

Figure 5 illustrates the CCE across various scene types. CCE quantifies the uniformity of colors within semantic regions in generated images. It is computed by first segmenting both generated and reference images into superpixels, then calculating the standard deviation of pixel colors within each corresponding superpixel in the Lab color space. The average standard deviation across all regions yields the CCE, where lower values indicate better regional color consistency. In mechanically complex scenes, the proposed method achieved a CCE of 0.25, a 45.7% reduction compared to the next-best approach—demonstrating the GCN's effectiveness in modeling the connectivity of rigid structures. For clothing texture scenes, a CCE of 0.19 indicates the model's ability to capture fine-grained texture-color relationships. In character-based scenes, the method achieved a CCE of 0.18, highlighting the SAM's precision in controlling key semantic features, such as the coordination between hair and eye color. The overall average CCE of 0.21 marks a significant advancement in cross-region color consistency.

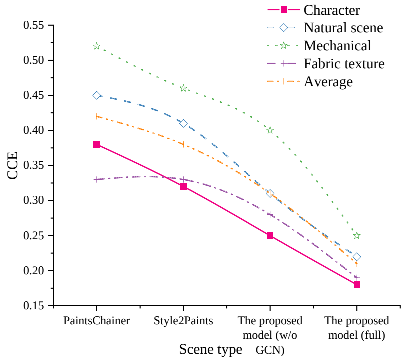

Figure 5: CCE across different scene types

Figure 6 presents the style adaptation performance measured by FID. In the watercolor style test, the proposed method achieved an FID of 20.5, reflecting a 28.6% improvement over the reference method [28] and confirming the effectiveness of the AdaIN module in capturing fluid artistic styles. For the ink wash style, the FID reached 24.8—the highest among the tested styles—but still outperformed the baseline score of 35.6, demonstrating the model's ability to reflect the aesthetic characteristics of East Asian art. With user-guided finetuning, the FID further dropped to 18.9, representing a 9.1% improvement and validating the practical value of the latent space interpolation mechanism for personalized content generation.

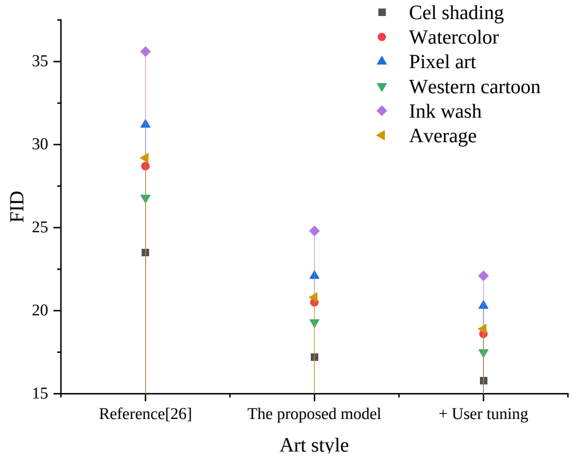

Figure 6: FID comparison across different style adaptations

Figure 7 and Table 4 present the results of the ablation study, which quantifies the contribution of each module. Adding the structural consistency loss reduced the boundary overflow rate from 8.2% to 4.7%, a 42.7% decrease, confirming its effectiveness in mitigating color spillover. The GCN module yielded the largest improvement in the CCE metric, with a 31.3% reduction, highlighting the value of region-based relational modeling for color coordination. The SAM increased the PSNR by 1.3 dB, demonstrating enhanced local detail fidelity through improved semantic focus. The full model, achieving a PSNR of 30.2 and a boundary overflow rate of 2.1%, underscores the synergistic effect of integrating all components.

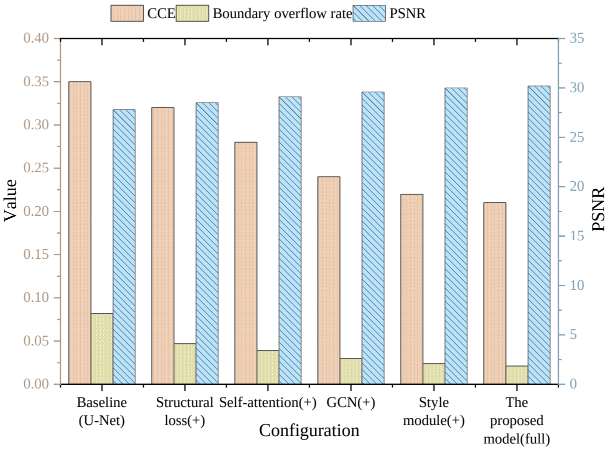

Figure 7: Results of the ablation study

To assess the practical quality of the model's outputs, a double-blind study was conducted with a panel of 30 professional artists. All participants were affiliated with prominent domestic and international animation studios and had an average of over five years of industry experience.

Table 4: Ablation study results showing the impact of incrementally adding each module on performance metrics

Configuration	CCE	PSNR (dB)	Boundary Overflow Rate (%)
Baseline (U-Net)	0.35	27.8	8.20
+ Structural Consistency Loss	0.32	28.5	4.70
+ Self-Attention Mechanism	0.28	29.1	3.90
+ Graph Convolution Network	0.24	29.6	3.00
+ Style Module	0.22	30.0	2.40
Proposed Model	0.21	30.2	2.10

Figure 8 presents the analysis of inference speed and model complexity. The proposed method processes a single frame in 41 ms using 78.4 million parameters, delivering a 21.2% improvement in inference efficiency compared to Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization (U-GAT-IT). This gain is primarily attributed to the generator's dense skip connection design, which reduces redundant computation. With a GPU memory footprint of 843 MB, the model is well-suited for handling large-scale animation sequences. Moreover, the lightweight version achieves a frame rate of 28 ms per image, supporting real-time interaction and highlighting the model's potential for practical deployment in production environments.

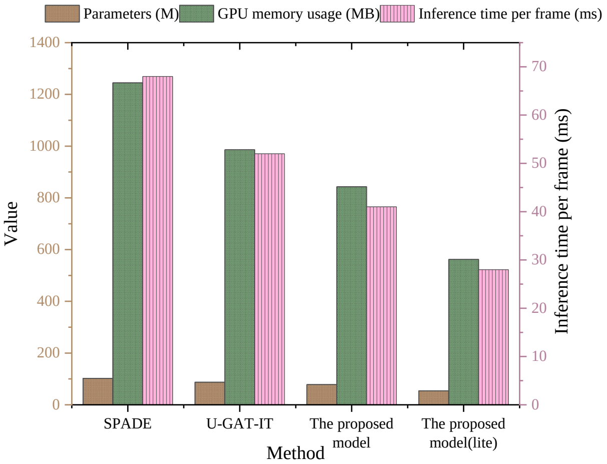

Figure 8: Inference speed and model parameter analysis

A seven-point Likert scale was used to evaluate four aspects of image quality: color harmony, edge accuracy, style consistency, and artistic expressiveness. Each artist independently scored multiple colorization results for the same line drawing. The scores were then normalized to a preference percentage for each method, representing the proportion of artists who selected that result as superior. Figure 9 summarizes the results of this evaluation. The model achieved a 90% preference rate for boundary precision, demonstrating strong visual fidelity in preserving structural details. An 86.7% preference for color coordination further confirms the effectiveness of the CCE metric. Style consistency received an 83.3% preference, while 76.7% of participants indicated a willingness to incorporate the model into their workflow. These findings show that the model not only meets technical performance benchmarks but also aligns with aesthetic expectations in real-world creative scenarios, highlighting its potential for practical adoption.

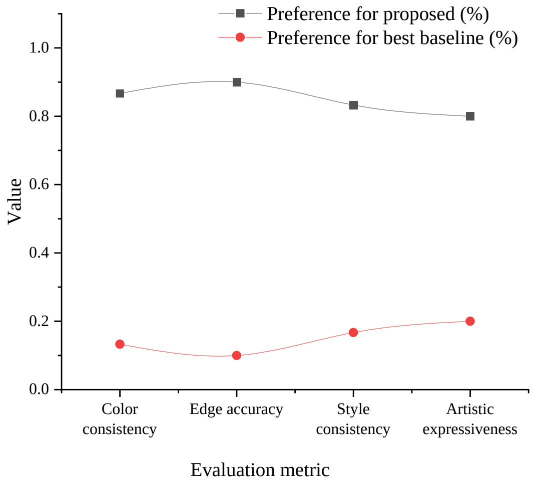

Figure 9: Results of the double-blind evaluation by professional artists

4.2 Discussion

The proposed automatic coloring framework for animation line art outperforms existing mainstream methods in both objective metrics and subjective evaluations. Unlike approaches designed for natural images or specific cultural artifacts, this framework addresses the unique challenges of animation. Compared with Wu et al. (2022), which focuses on fine-grained coloring of traditional costumes, the method generalizes more effectively to non-rigid objects such as dynamic characters. This improvement arises from the explicit modeling of semantic relationships between regions using GCNs, allowing the model to reason about color coordination across different parts rather than relying solely on local texture features. In contrast to the visual Transformer-based approach of Al-Ghanimi et al. [14], the proposed framework achieves higher color consistency while maintaining greater computational efficiency, meeting practical requirements in animation production. These gains result largely from the joint optimization of semantic and geometric structures in line art. The GCN module constructs a region adjacency graph and aggregates node information to encode scene hierarchy, effectively reducing color errors between characters and backgrounds or between clothing and skin. The structural consistency loss imposes gradient constraints along line boundaries, suppressing color bleeding and mitigating pigment overflow caused by unclosed contours. This mechanism is a key factor in the framework's superior boundary fidelity compared with traditional tools such as PaintsChainer. Additionally, the SAMs captures long-range dependencies, enabling precise focus on critical semantic regions, such as pupils or accessories, and enhancing local detail reproduction. The style-decoupling module, based on AdaIN, provides flexible user-guided style control, addressing the limited adaptability to diverse artistic styles seen in most existing methods.

5 Conclusion

This study presents a structured, semantically informed framework for automatic colorization of animation line art. Built on a multi-level conditional GAN design, the framework addresses three main challenges: color overflow, semantic misalignment, and limited style flexibility. The generator uses a U-Net backbone with dense skip connections to preserve structural integrity, even in line art with unclosed contours. GCNs and SAMs are incorporated to support cross-region color harmonization. A style-decoupling module, implemented via AdaIN, allows user-guided style transfer. Experimental results show that the method outperforms existing techniques on key evaluation metrics, including PSNR, CCE, and boundary overflow rate. In professional evaluations, it achieved a 90% preference rate for boundary accuracy and reduced manual correction requirements by 85%.

Despite its strong performance, the model has several limitations. It relies on high-quality paired datasets and shows limited generalization to abstract styles or amateur line art. User-guided style transfer is supported, but the current interaction speed is not suitable for real-time use. Temporal consistency across animated sequences also remains an open challenge. Future work could address these limitations in several ways. Self-supervised pretraining on unlabeled animation data may improve generalization to abstract or amateur line art. Lightweight interactive modules could provide fine-grained, stroke-level semantic guidance to support faster user interaction. Extending the framework to video sequence colorization could leverage a multi-scale temporal GAN to handle both spatial and temporal dimensions. Optical flow could be used to align pixels across consecutive frames, enabling a temporal consistency loss that is optimized alongside spatial losses to produce smooth, stable video outputs and reduce flickering. Additionally, while the current style transfer method uses spherical linear interpolation in latent space to preserve semantic features, future research could explore advanced manifold learning techniques for improved style disentanglement and control. Overall, this study establishes a foundation for intelligent animation colorization, and the proposed structured generative framework may have broader applications in fields such as medical image analysis and virtual scene synthesis.

References

[1] Lyu J, Young Lee H, Liu H. Color matching generation algorithm for animation characters based on convolutional neural network. Computational Intelligence and Neuroscience, 2022, 2022(1): 3146488. doi: 10.1155/2022/3146488

[2] Gao X, Yin L, Deng Y, Wang F, Qin Y, Zhang M. Bi-Stream feature extraction and multiscale attention generative adversarial network (BM-GAN): colorization of grayscale images based on Bi-Stream feature fusion and multiscale attention generative adversarial network. The European Journal on Artificial Intelligence, 2025, 38(2): 159-180. doi: 10.1177/30504554241297613

[3] Zhao Y, Ren D, Chen Y, Jia W, Wang R, Liu X. Cartoon image processing: a survey. International Journal of Computer Vision, 2022, 130(11): 2733-2769. doi: 10.1007/s11263-022-01645-1

[4] Duan J, Gao M, Zhao G, Zhao W, Mo S, Zhang W. FAColorGAN: a dual-branch generative adversarial network for near-infrared image colorization. Signal, Image and Video Processing, 2024, 18(8): 5719-5731. doi:10.1007/s11760-024-03266-2

[5] Jin X, Di Y, Jiang Q, Chu X, Duan Q, Yao S, Zhou W. Image colorization using deep convolutional auto-encoder with multi-skip connections. Soft Computing, 2023, 27(6): 3037-3052. doi:10.1007/s00500-022-07483-0

[6] Long B, Zhou C. Enhanced mineral image classification using YOLOv8-CLS with optimized feature extraction and dataset augmentation. Informatica, 2025, 49(34): 87-108.

[7] Zhang X. Lightweight Image Super-Resolution Reconstruction Algorithm Based on Spectral Norm Regularization GAN and ShuffleNet. Informatica, 2025, 49(34): 339-350.

[8] Zhang Z, Li Y, Shin B S. Robust medical image colorization with spatial mask-guided generative adversarial network. Bioengineering, 2022, 9(12): 721. doi: 10.3390/bioengineering9120721

[9] Ai Y, Liu X, Zhai H, Li J, Liu S, An H, et al. Multi-scale feature fusion with attention mechanism based on CGAN network for infrared image colorization. Applied Sciences, 2023, 13(8): 4686. doi: 10.3390/app13084686

[10] Treneska S, Zdravevski E, Pires I M, Lameski P, Gievska S. Gan-based image colorization for self-supervised visual feature learning. Sensors, 2022, 22(4): 1599. doi:10.3390/s22041599

[11] Wu D, Gan J, Zhou J, Wang J, Gao W. Fine-grained semantic ethnic costume high-resolution image colorization with conditional GAN. International Journal of Intelligent Systems, 2022, 37(5): 2952-2968. doi: 10.1002/int.22726

[12] Wu B, Dong Q, Sun W. Automatic colorization of Chinese ink painting combining multi-level features and generative adversarial networks. Fractals, 2023, 31(06): 2340144. doi: 10.1142/S0218348X23401448

[13] Zhou S, Li H, Sun W, Zhou F, Xiao K. Auto-White balance algorithm of skin color based on asymmetric generative adversarial network. Color Research & Application, 2025, 50(3): 266-275. doi:10.1002/col.22970

[14] Al-Ghanimi A, Lakizadeh A. ViT-Based Automatic Grayscale Image Colorization with a Hybrid Loss Function. Ingenierie des Systemes d'Information, 2025, 30(4): 995. doi: 10.18280/isi.300416

[15] Dalal H, Dangle A, Radhika M J, Gore S. Image colorization progress: a review of deep learning techniques for automation of colorization. International Journal of Advanced Trends in Computer Science and Engineering, 2021, 10(10.30534). doi: 10.30534/ijatcse/2021/401042021

[16] Chen R, Zhao J, Yao X, He Y, Li Y, Lian Z, et al. Enhancing urban landscape design: a GAN-based approach for rapid color rendering of park sketches. Land, 2024, 13(2): 254. doi: 10.3390/land13020254

[17] Jampour M, Zare M, Javidi M. Advanced multi-Gans towards near to real image and video colorization. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(9): 12857-12874. doi: 10.1007/s12652-022-04206-z

[18] Mourchid Y, Donias M, Berthoumieu Y, Najim M. SPDGAN: a generative adversarial network based on SPD manifold learning for automatic image colorization. Neural Computing and Applications, 2023, 35(32): 23581-23597. doi: 10.1007/s00521-023-08999-8

[19] Yang P, Zhao H, Xie Z. Colorization of fundus images based on advanced degradation models. Journal of Radiation Research and Applied Sciences, 2025, 18(1): 101285. doi: 10.1016/j.jrras.2024.101285

[20] Al-Ghanimi A, Lakizadeh A. Improving unsupervised deep learning methods for detection and colorization of grayscale images based on contextual content. International Journal of Intelligent Engineering & Systems, 2025, 18(4):439. doi: 10.22266/ijies2025.0531.28

[21] Tian N, Liu Y, Wu B, et al. Colorization of logo sketch based on conditional generative adversarial networks. Electronics, 2021, 10(4): 497. doi: 10.3390/electronics10040497

[22] Rizkinia M, Faustine N, Okuda M. Conditional generative adversarial networks with total variation and color correction for generating Indonesian face photo from sketch. Applied Sciences, 2022, 12(19): 10006. doi: 10.3390/app121910006

[23] Zhang J, Zhu S, Liu K, Liu X. UGSC-GAN: User-guided sketch colorization with deep convolution generative adversarial networks. Computer Animation and Virtual Worlds, 2022, 33(1): e2032. doi: 10.1002/cav.2032

[24] Tian Z, Li X, Zheng Y, et al. Graph-convolutional-network-based interactive prostate segmentation in MR images. Medical physics, 2020, 47(9): 4164-4176. doi: 10.1002/mp.14327

[25] Guo X, Liu X, Królczyk G, Sulowicz M, Glowacz A, Gardoni P, et al. Damage detection for conveyor belt surface based on conditional cycle generative adversarial network. Sensors, 2022, 22(9): 3485. doi: 10.3390/s22093485

[26] Gui X, Zhang B, Li L, Yang Y. DLP-GAN: learning to draw modern Chinese landscape photos with generative adversarial network. Neural Computing and Applications, 2024, 36(10): 5267-5284. doi:

10.1007/s00521-023-09345-8

[27] Langa A S, Bolaño R R, Carrión S G, Elorrieta I U. Color normalization through a simulated color checker using generative adversarial networks. Electronics, 2025, 14(9): 1746. doi: 10.3390/electronics14091746

[28] Lee H Y, Li Y H, Lee T H, Aslam M S. Progressively unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. Sensors, 2023, 23(15): 6858. doi: 10.3390/s23156858