Electronic Research Archive

ERA, 33(9): 5377–5400.
DOI: 10.3934/era.2025241
Received: 28 July 2025
Revised: 29 August 2025
Accepted: 02 September 2025
Published: 12 September 2025

https://www.aimspress.com/journal/ERA

Research article

Analysis and comparison for image colorization with machine learning based on PyTorch and ChromaGAN

Jinyi Luo and Xi Li*

School of Mathematical Sciences, Chengdu University of Technology, Sichuan 610059, China

* **Correspondence:** Email: lixi@cdut.edu.cn.

Abstract: Colorization of grayscale images has found widespread applications across artistic, historical, scientific, medical, and industrial domains. Traditional manual colorization methods, however, are labor-intensive, time-consuming, and prone to subjective interpretation. In recent years, various deep learning (DL) approaches have been explored to automate the colorization process. Despite this progress, the efficacy and accuracy of these DL methods remain largely unexamined. One of the primary challenges is the accurate handling of color boundaries, which significantly affects the realism and visual clarity of the colorized output. In this study, we investigated the impact of different DL algorithms and training epochs on the quality of colorized images, comparing them to their corresponding ground truths. To explore these effects, a semantic-guided adversarial generative framework was employed, and five commonly used DL algorithms were compared.

Keywords: image colorization; deep learning; ChromaGAN

1. Introduction

Image colorization is a technique that involves generating realistic and semantically consistent color information from grayscale images. As a classic example of an ill-posed inverse problem in computer vision, colorization has attracted significant interest due to its broad applicability domains, including historical photo restoration, remote sensing, digital heritage preservation, and medical image analysis [1]. A single grayscale intensity can correspond to multiple valid color values, making the problem inherently ambiguous without strong contextual or semantic guidance. Over the years, image colorization techniques have undergone significant evolution, transitioning from manual-intensive, heuristic-driven methods to automated, learning-based systems. Broadly, existing approaches can be categorized into three major paradigms: Scribble-based methods, example-based methods, and machine learning (ML)-based methods [2].

Scribble-based methods rely on humans to provide color cues, typically in the form of sparse strokes or doodles, that are propagated throughout the grayscale image using spatial or edge-aware algorithms. Levin et al. [3] introduced a pioneering framework in which neighboring pixels with similar luminance are encouraged to share similar chrominance values, solved through a quadratic cost function. Subsequent enhancements addressed propagation artifacts such as color bleeding and edge blurring. For instance, Huang et al. [4] incorporated edge information to preserve structure; Yatziv et al. [5] developed a luminance-weighted blending scheme to reduce sensitivity to scribble location; and Luan et al. [6] leveraged image segmentation to improve region-level consistency. While conceptually intuitive, scribble-based methods are labor-intensive, demand expert intervention for complex scenes, and yield non-reproducible results.

Example-based or reference-based methods reduce manual intervention by transferring color information from one or more reference images to the grayscale target. Welsh et al. [7] proposed an early technique based on local luminance and texture matching, but the lack of spatial and semantic coherence often led to unstable color assignments. Subsequent improvements incorporated region-based segmentation [8], probabilistic transfer models [9], and semantic priors [10,11] to better align reference and target content. For instance, Chia et al. [11] used scene classification to guide transfer, while Gupta et al. [12] employed a superpixel feature matching for enhanced structural consistency. Despite the reduced user burden, the efficacy of these methods remains highly dependent on the similarity between reference and target images, and the performance can degrade substantially under poor alignment or when appropriate references are unavailable.

The advent of deep learning (DL) has led to transformative advances in image colorization by enabling fully automated, end-to-end models capable of learning rich mappings between grayscale and color domains. Early models such as Cheng et al. [13] utilized convolutional neural networks (CNNs) to integrate semantic features with pixel-wise color prediction. Iizuka et al. [14] jointly trained classification and colorization branches to incorporate multi-scale semantic understanding, while Zhang et al. [15] introduced a quantized color prediction approach with class rebalancing to better model the rare chromatic variations. Noaman et al. [16] noted in their review that early coloring methods relied on manual features and heuristic rules. Huang et al. [1] emphasized that deep learning overcomes the limitations of traditional methods through end-to-end training. Anwar et al. [17] systematically reviewed commonly used datasets (such as ImageNet and COCO) and emerging large-scale multi-domain datasets (such as Landscape-Color), and pointed out the impact of data bias (such as dominant color distribution) on model performance.

Later, deep learning models have incorporated user guidance [18], hybrid prediction-transfer frameworks [19], and generative adversarial networks (GANs) to enhance perceptual realism. Notably, Isola et al. [20] combined a U-Net generator with adversarial training to significantly improve realism and texture consistency. Further developments include noise sampling for diversity [21] and memory networks to recover rare or subtle color features [22]. These deep learning methods leverage hierarchical representations and probabilistic modeling, enabling them to bridge low-level visual features and high-level semantic understanding, yielding substantial improvements in automation and output quality.

Here, we employ the ChromaGAN framework due to its targeted approach to the core challenge of semantic ambiguity. Unlike standard GANs, ChromaGAN integrates semantic perception and adversarial learning by jointly predicting a semantic map to guide color generation. This provides a crucial prior for plausible color choices.

Despite the tremendous progresses made in the past decade, automated image colorization continues to face key challenges. Foremost among these is the inherent ambiguity in grayscale-to-color mapping, particularly in domains such as historical imagery where ground-truth colors are often unknown. Achieving semantic consistency while avoiding implausible or surreal results requires models to reason about scene content, object identity, and contextual relationships. Moreover, maintaining sharp and accurate color boundaries remains technically difficult. Inadequate boundary preservation leads to artifacts such as color bleeding, edge blurring, and unnatural transitions. These problems are exacerbated in GAN-based models due to adversarial optimization instabilities.

In this study, we investigate how different deep learning algorithms and network configurations affect the quality of color filling and boundaries in image colorization. We employ a semantic-guided adversarial generative framework capable of learning plausible color distributions from grayscale inputs. The end-to-end model leverages the semantic context to guide the colorization process, while adversarial training improves visual realism and semantic coherence.

The outline of this paper is as follows. In Section 2, we introduce the related mathematical foundations. In Section 3, we describe the experiment setup and the generator's architecture in detail. In Section 4, we present experimental results and discussions. Finally, in Section 5, we list the major conclusions of this study.

2. Relevant mathematical foundations

2.1. ChromaGAN

ChromaGAN is an innovative grayscale image coloring framework that achieves high-quality automatic coloring of grayscale images by combining adversarial learning and semantic category distribution. The core innovation of this framework lies in integrating semantic information into the coloring process, enabling the learning of class-aware color distributions without the need for manual annotation. This section will introduce ChromaGAN's objective function, training method, and key technologies from a mathematical principle perspective.

2.1.1. Generator architecture

For a given grayscale input image L, ChromaGAN attempts to learn the following mapping [2]:

$$G : L \to (a, b). \tag{1}$$

This process generates $I = (L, a, b)$ a plausible color image, where a and b are chrominance channels in the CIE *Lab* color space. The coloring process of this learning procedure is accomplished by a generator $\mathcal{G}_\theta$, which consists of two sub-modules:

$$\mathcal{G}_\theta = (\mathcal{G}^1_{\theta_1}, \mathcal{G}^2_{\theta_2}), \tag{2}$$

where $\theta = (\theta_1, \theta_2)$ represents all generator parameters, and

- $\mathcal{G}^1_{\theta_1}: L \to (a,b)$ is responsible for predicting the chrominance channel (a,b) from the grayscale input image L;
- $\mathcal{G}^2_{\theta_2}:\ L \to y$ is responsible for predicting the semantic branch from the grayscale input image L, and this semantic branch outputs a probability distribution vector:

$$y \epsilon \mathbb{R}^m\ ,\ \sum_{i=1}^{m} y_i = 1. \tag{3}$$

Here, m represents the predefined number of categories, and y_i indicates the probability that an image belongs to category i. This design enables the model to learn semantic information without manual annotation, laying the foundation for subsequent semantic-guided coloring.

2.1.2. Adversarial learning framework

The objective loss function of ChromaGAN is defined as:

$$\mathcal{L}_e(\mathcal{G}_\theta, D_\omega) = \mathcal{L}_e\left(\mathcal{G}^1_{\theta_1}\right) + \lambda_g \mathcal{L}_g\left(\mathcal{G}^1_{\theta_1}, D_\omega\right) + \lambda_s \mathcal{L}_s\left(\mathcal{G}^2_{\theta_2}\right). \tag{4}$$

- The first item $\mathcal{L}_e\left(\mathcal{G}^1_{\theta_1}\right)$ represents the color error loss. By calculating the mean-square error (MSE) between the predicted chrominance map $\mathcal{G}^1_{\theta_1}(L)$ generated by the generator $\mathcal{G}^1_{\theta_1}$ and the real chrominance maps (a,b), we evaluate the expected performance on the real data distribution $\mathbb{P}$. This achieves the objective of supervising the generator $\mathcal{G}^1_{\theta_1}$ to learn how to predict the chrominance channel from the luminance channel. The specific definition of this error loss is:

$$\mathcal{L}_e\left(\mathcal{G}^1_{\theta_1}\right) = \mathbb{E}_{(L,a,b)\sim\mathbb{P}}\left[\left\|\mathcal{G}^1_{\theta_1}(L) - (a,b)\right\|_2^2\right]. \tag{5}$$

Here, $\mathbb{P}$ represents the distribution of real color images, and $\|\cdot\|_2$ represents the Euclidean norm.

- The second term $\lambda_g \mathcal{L}_g\left(\mathcal{G}^1_{\theta_1}, D_\omega\right)$ represents the Wasserstein GAN loss (WGAN loss) [23]:

$$\mathcal{L}_g\left(\mathcal{G}^1_{\theta_1}, D_\omega\right) = \mathbb{E}_{\tilde{I}\sim\mathbb{P}}[D_\omega(\tilde{I})] - \mathbb{E}_{(a,b)\sim\mathbb{P}_{\mathcal{G}^1_{\theta_1}}}[D_\omega(L,a,b)] - \mathbb{E}_{\hat{I}\sim\mathbb{P}_{\hat{I}}}\left[\left(\left\|\nabla_{\hat{I}} D_\omega(\hat{I})\right\|_2 - 1\right)^2\right]. \tag{6}$$

In this formula, parameter λ_g is used to enhance the naturalness of color image $\tilde{I}$, the system samples real color images from the real color image distribution $\mathbb{P}$. $(a,b)\sim\mathbb{P}_{\mathcal{G}^1_{\theta_1}}$ represents the generated chrominance channels sampled from the distribution of $\mathcal{G}^1_{\theta_1}$. $\mathbb{P}_{\mathcal{G}^1_{\theta_1}}$ denotes the model distribution (model distribution) of $\mathcal{G}^1_{\theta_1}(L)$, where L follows the distribution $\mathbb{P}_{rg}$ of grayscale input images. The inherent antagonism in the WGAN loss lies in its dual objective: While the discriminator $D_\omega(\cdot)$ aims to maximize this loss through optimization, the generator $\mathcal{G}^1_{\theta_1}$ strives to minimize this loss. The WGAN loss defined above consists of three components: The first component represents the expectation of the real color image $\tilde{I}$ on the discriminator network $D_\omega(\cdot)$; the second component denotes the expectation of the generated image composed of the true grayscale input L and the chromaticity (a,b) predicted by the generator on the discriminator network $D_\omega(\cdot)$; and the third component is the WGAN-GP (WGAN with gradient penalty) with a gradient penalty term [24], where $\hat{I}$ is a sample collected from the line connecting the real distribution and the generated distribution, i.e., $\hat{I} = \epsilon\tilde{I} + (1-\epsilon)(L,a,b)$, where $\epsilon \sim U[0,1]$, and $\nabla_{\hat{I}} D_\omega(\hat{I})$ represents the gradient of the discriminator on input $\hat{I}$. This penalty term forces the gradient norm of the discriminator to approach 1, ensuring that the discriminator satisfies the 1-Lipschitz constraint required in Wasserstein distance calculations.

- The third term $\lambda_s \mathcal{L}_s(\mathcal{G}_{\theta_2}^2)$ represents the class distribution loss, which aims to maintain the semantic rationality of the color in the generated image:

$$\mathcal{L}_s(\mathcal{G}_{\theta_2}^2) = \mathbb{E}_{L \sim \mathbb{P}_{rg}}\left[KL\left(y_V \| \mathcal{G}_{\theta_2}^2(L)\right)\right]. \tag{7}$$

Among them, parameter λ_s is used to enhance semantic consistency; $L \sim \mathbb{P}_{rg}$ indicates that L follows the distribution $\mathbb{P}_{rg}$ of grayscale input images; $y_V \in \mathbb{R}^m$ represents the distribution vector [25] output by the pre-trained VGG-16 model when applied to grayscale images; and $KL(\cdot \| \cdot)$ denotes the Kullback-Leibler divergence.

Therefore, based on the loss function (4), we calculate the weights of $\mathcal{G}_\theta$ and D_ω by solving the following min-max problem:

$$\min_{\mathcal{G}_\theta} \max_{D_\omega \in \mathcal{D}} \mathcal{L}(\mathcal{G}_\theta, D_\omega). \tag{8}$$

The hyperparameters λ_g and λ_s are fixed at 0.1 and 0.003, respectively.

2.2. PyTorch

Mathematically, PyTorch is a system that operates on multidimensional real vector spaces, where a tensor is a multidimensional array, formally written as:

$$T \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}. \tag{9}$$

Moreover, PyTorch supports function composition and differential. By defining each operation $\emptyset_i$ as a node in the graph:

$$V_{i+1} = \emptyset_i(V_i). \tag{10}$$

For a composite function $y = f\left(g\left(h(x)\right)\right)$, its derivative is computed via the chain rule:

$$\frac{dy}{dx} = \frac{dy}{dg} \cdot \frac{dg}{dh} \cdot \frac{dh}{dx}, \tag{11}$$

then, the gradients are backpropagated through the graph using this chain rule. Therefore, the loss function $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ with parameter $\theta \in \mathbb{R}^n$ can be computed via reverse-mode automatic differentiation:

$$\nabla_\theta \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \theta_1}, \ldots, \frac{\partial \mathcal{L}}{\partial \theta_n}\right). \tag{12}$$

Finally, the objective functions can be optimized via gradient-based algorithms:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \mathcal{L}\left(\theta^{(t)}\right), \tag{13}$$

where η is the learning rate.

3. Experiment setup

In the ChromaGAN model, the generator is one of the core components responsible for mapping the input grayscale image to the color space (predicting the chroma channels a and b). Its design is based on the generative adversarial network (GAN) framework, combining semantic information with adversarial learning to achieve realistic and diverse image coloring. To systematically explore the impact of generator design on performance (specifically, the quality of color boundaries in image coloring), we examine the following five distinct generator architectures: original, compact, transposed convolutions, depthwise convolutions, and pixel shuffle. The five generators (principles and parameters) are explained below.

Original

The original generator uses two Upsample-Conv2D-Conv2D blocks (with nearest-neighbor upsampling) to progressively decode fused mid-level and global features into ab color channels (the two chroma components in the LAB color space, predicting color information for grayscale inputs), ending with a final Upsample for full 224 × 224 (a standard resolution in computer vision tasks, e.g., ImageNet) output.

Principle: The original generator serves as the baseline architecture, utilizing a series of upsampling and convolutional layers to decode features into color space. It aims to balance complexity and performance with a straightforward design.

- Decoder convolutional layers: 4.
- Decoder parameters: 129,762.

Compact

This version adopts an approach similar to the original generator but uses only single convolutional layer after each upsampling operation. The Upsample is set to bilinear rather than nearest neighbor, and the output is produced through a Tanh activation layer.

Principle: This version aims to reduce model complexity and computational cost compared to the original generator. It employs fewer convolutional layers and uses bilinear upsampling, which is computationally efficient but may lead to smoother and potentially less detailed outputs.

- Decoder convolutional layers: 3.
- Decoder parameters: 92,834.

Transposed convolutions

This version replaces Upsample with ConvTranspose2d (“deconvolution”)-based upsampling blocks, interleaved with a 3×3 convolution operation, to learn its own upsampling kernels.

Principle: This generator replaces upsampling layers with transposed convolutional layers (also known as deconvolutions). Transposed convolutions can learn its own upsampling kernels during training, potentially leading to more precise and detailed reconstructions compared to fixed upsampling methods like bilinear or nearest-neighbor. This often comes at the cost of increased parameters and computational requirements.

- Decoder convolutional layers: 3 (Conv) +3 (Transposed).
- Decoder parameters: 208,058.

Depthwise convolutions

This architecture replaces the standard Conv2d with depthwise separable Conv blocks (one depthwise Conv + one pointwise Conv) throughout the decoder to reduce parameter count.

Principle: This architecture extensively uses depthwise separable convolutional blocks. These blocks factorize standard convolutions into a depthwise convolution (applying a single filter per input channel) and a pointwise convolution (1 × 1 convolution to combine channel outputs). This design significantly reduces the number of parameters and computational cost compared to standard convolutions, making the model more lightweight and efficient, albeit potentially at the expense of representational capacity.

- Decoder convolutional layers: 3 (Depthwise) +3 (Pointwise) +1 (Output).
- Decoder parameters: 14,066.

Pixel shuffle

This version uses progressive sub-pixel convolution (1 × 1 conv → pixel shuffle) blocks for learned upsampling, interleaved with a standard convolution for final refinements.

Principle: This generator utilizes sub-pixel convolution (via the pixel shuffle operation) for learned upsampling. The pixel shuffle layer rearranges elements in a low-resolution multi-channel feature map to form a higher-resolution output with fewer channels. This approach can efficiently increase resolution while potentially preserving more information from the feature maps compared to other upsampling techniques, but it can also be parameter-intensive.

- Decoder convolutional layers: 3 (Pre-pixel shuffle) +1 (Output).
- Decoder parameters: 378,638.

4. Experimental results and discussion

In this section, we systematically analyze the impact of generator parameter changes and training epochs on the quality of generated images. We focus on examining the clarity and realism of boundaries in color transition areas between objects and evaluate whether the generated results are consistent with actual scenes by comparing changes in interface details. Furthermore, we conduct a comparative analysis of colors generated at different epoch stages to summarize the characteristics of the chromaGAN generator architecture in terms of color filling.

4.1. Datasets

Imagenette is selected for training in this study. Imagenette is a subset of ImageNet that contains only 10 representative categories, with a data volume approximately 1/100*th* that of ImageNet. It significantly reduces training time and computing resource requirements, making it suitable for rapid verification of model prototypes or algorithm effectiveness while retaining the multimodal characteristics of ImageNet.

4.2. Image evaluation based on generator parameters

We select nine sets of images with an epoch of 900 to analyze the effects and characteristics of different generators. The selected images are characterized by their vibrant and saturated color palette, where each hue is distinctly separated from the others with sharp, well-defined boundaries. This high color contrast and clarity make it easy to differentiate between elements within the images, as shown in Figures 1(a)–(i).

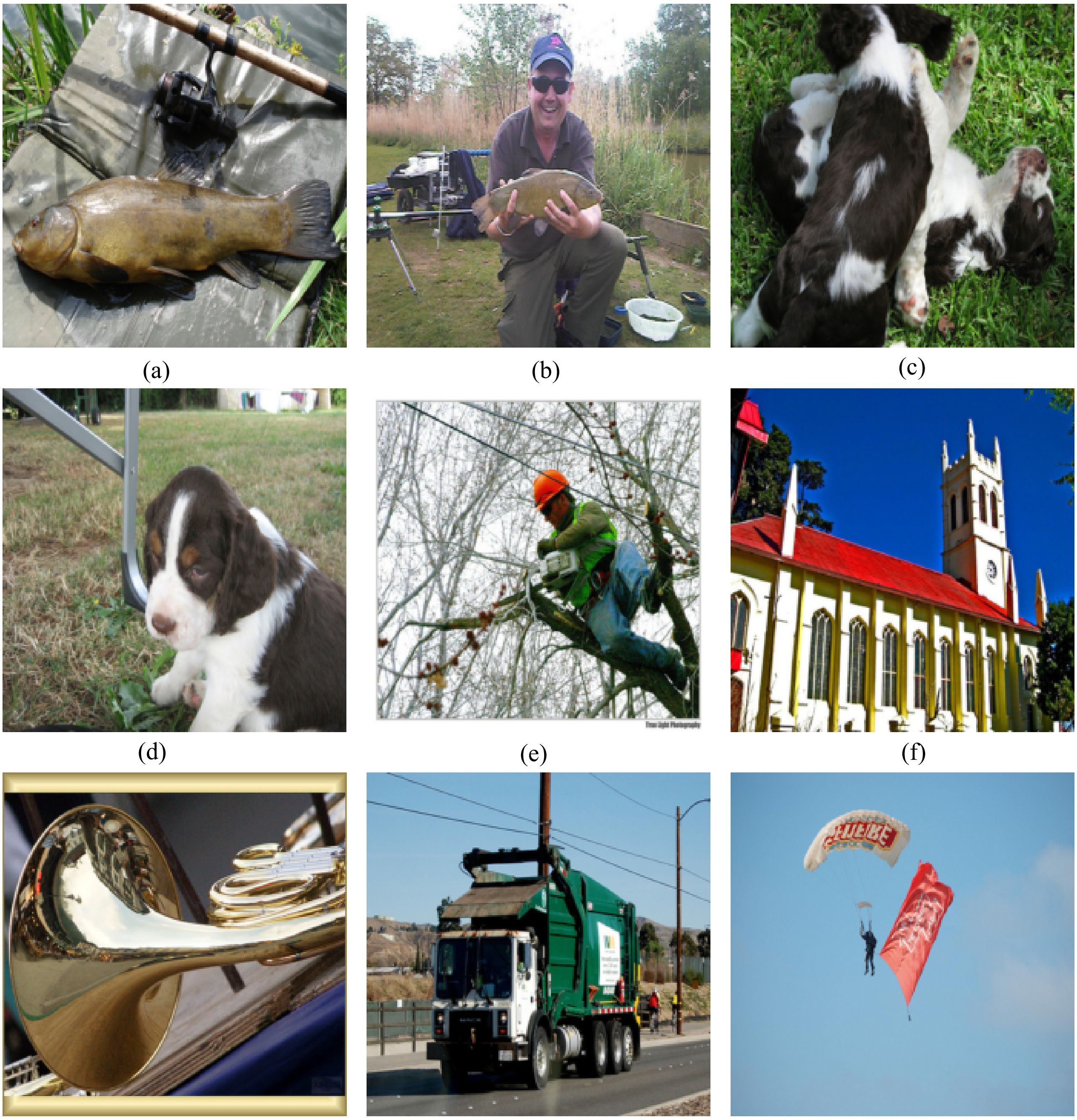

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 1. Ground truth images used to display the original colors.

The sample images shown in Figures 2–6 are representative of the overall trends observed across the evaluation dataset. The qualitative characteristics and performance issues demonstrated in these examples, such as color inaccuracies, boundary artifacts, and semantic inconsistencies, are consistently observed in the broader set of results. While minor variations occur depending on specific image content and complexity, the selected figures provide a characteristic illustration of each generator's typical behavior and limitations.

4.2.1. Original

Figure 2 shows the colorized images using the original generator, in correspondence to the ground truth images shown in Figure 1. The differences between the colorized images and the ground truth images are analyzed below to identify the coloring characteristics of the original generator.

Figure 2(a) shows the grass in gray and lacks the lime green tones, while the fishing rod, tarp, fish body, and lake edges appear overly orange, particularly in the shadowed areas where they contact the fish. In Figure 2(b), the lawn correctly displays lime green, yet issues arise elsewhere: The dirt patches, stone piers, and background dead grass incorrectly show green instead of yellow. Additionally, the blue tool cart appears gray where it meets the lawn, and the hat suffers the same discoloration. Red overfills the fish, and the fishman's shoulders, hands, pants, and head.

In Figure 2(c), the lawn skews yellowish overall, with the light green zones turning gray. Moreover, the dog's paws and nose exhibit an unnatural light red tint.

Figure 2(d) continues the lawn's color inaccuracies, presenting a yellowish-green hue with uniformly green dirt and metal brackets. The plants beneath the dog's paws appear grayish, and the puppy's fur loses its white purity, adopting a reddish gradient.

In Figure 2(e), the sky's grayish-white sections are wrongly tinged with blue, and the tree branches appear unnaturally black and yellowish. The workers' fluorescent helmets and vibrant clothing darken into a dull brownish-red, while their red masks and the tree's red ornaments render as black.

Figure 2(f) follows with subdued tones: the sky, greenery, and church roof wall appear overly dark, while reddish-brown gradients dominate the church's wall and roof edges.

In Figure 2(g), the round horn carries a yellowish-reddish cast, and the blue tabletop incorrectly fills with dark gray. The wood and tabletop also display an orange-reddish gradient where they intersect the horn.

Figure 2(h) introduces further anomalies: The sky fades to gray near the pole, dead grass and guardrails are uniformly green instead of yellow, the road shows an abrupt blue patch, and the green truck morphs into greyish-black and dark red.

Finally, Figure 2(i) features an unnaturally dark sky. The parachute, intended as a white umbrella with white lettering on red, erroneously appears yellow with yellow text. Similarly, the red flag loses its vibrancy, washed out in a pale red.

The overall characteristics of the original generator can be summarized as follows: Bright colors are easily filled with greyish or darker shades of the same color; light colors are easily filled with reddish shades and gradient colors, blurring the interface; and multiple color images show that it is difficult to distinguish between the lawn and the soil, with the soil appearing only in small areas of the lawn and filled with green that matches the surrounding lawn. This indicates that the original generator lacks the ability to recognize the overall color of objects and the color boundaries between local details.

This may be due to the fixed-nucleus sampling (nearest-neighbor algorithm) having semantic blind spots, leading to color bleeding and blurred boundaries. Subsequent convolution operations cannot fully correct this defect, resulting in significantly reduced distinguishability between areas of similar brightness (such as soil and grass).

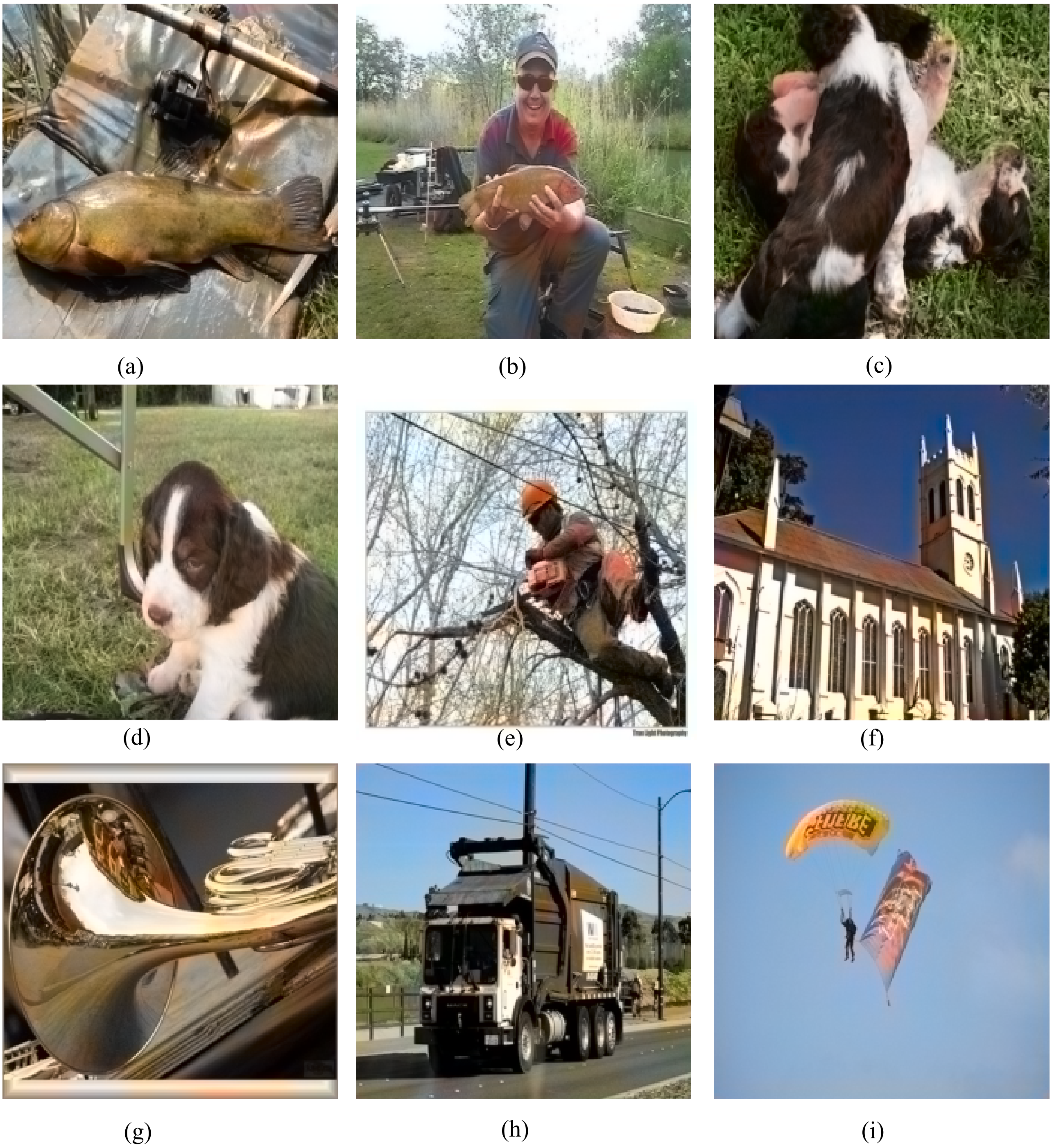


Figure 2. Images generated by the original generator.

4.2.2. Compact

Figure 3 shows the colorized images generated by the compact generator corresponding to the images shown in Figure 1.

In Figure 3(a), the grassland appears in a low-saturation light green, while the fishing rod, waterproof cloth, and certain fish body areas display an orange-red to copper-yellow gradient, with the fish's overall tone appearing noticeably yellowish.

In Figure 3(b), we see the lawn, soil, stone piers, and withered grass uniformly rendered in green, while the dark blue utility vehicle and adjacent lawn edges appear gray along with the character's hat. The character's clothing shows a bluish-gray tint with green gradients around knee areas contacting the lawn.

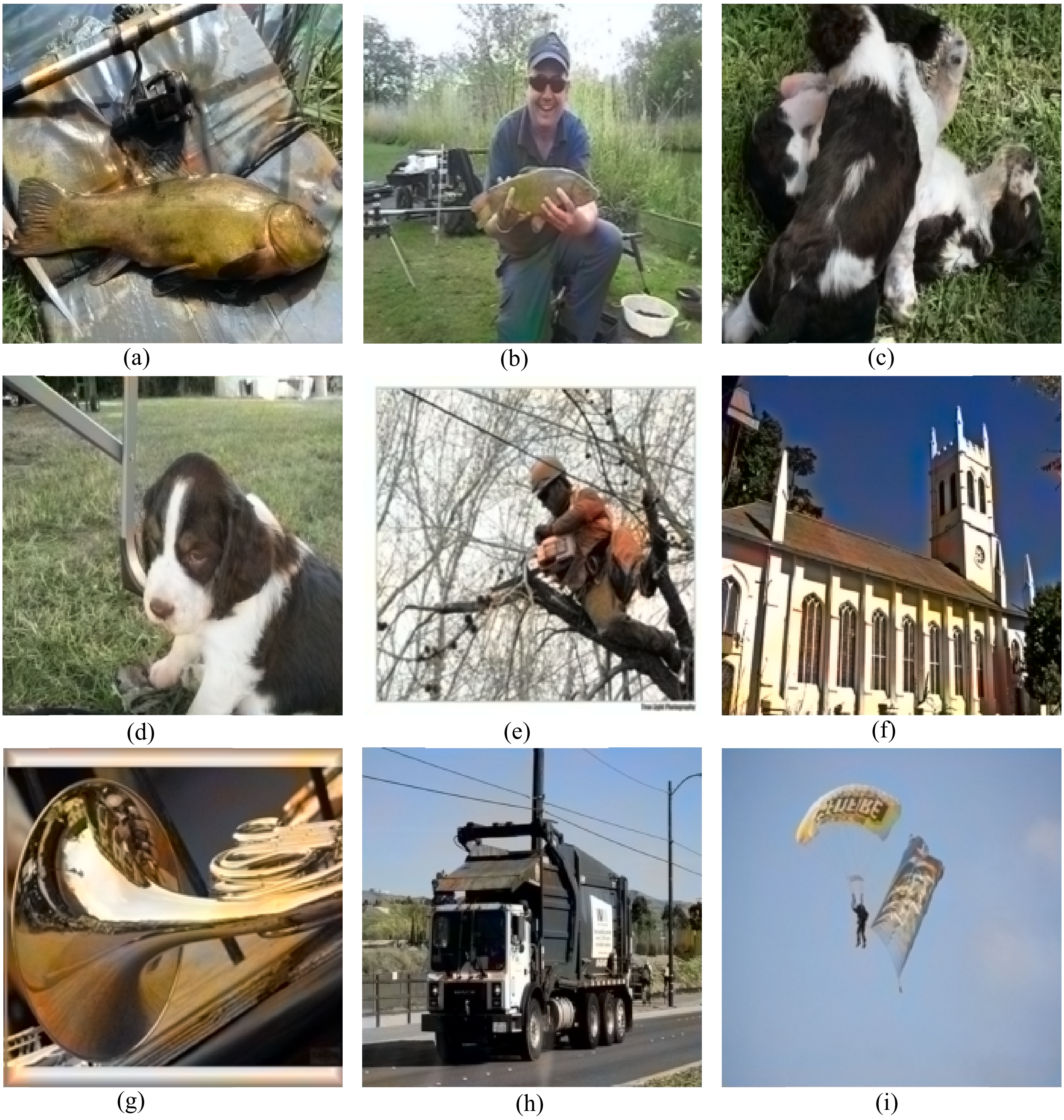


Figure 3. Images generated by the compact generator.

Figure 3(c) presents a grayish-green background lawn, with the dog's white fur showing faint reddish gradients and its light pink paws uniformly gray.

Similarly, in Figure 3(d), nearby soil and withered grass appear uniformly green, with grayish bleeding near steel pipes. The plants beneath the dog's paws render as gray, while the puppy's white fur near withered grass shows grayish-yellow tints, and the house's colorful decorations appear dull gray.

For Figure 3(e), the sky shows a grayish-white yellowish tint, with red tree decorations and masks rendered dark gray. Workers' red hats and green clothing uniformly appear grayish-brown.

Figure 3(f) continues this pattern with a dark gray sky, gray vegetation, dull grayish-brown church roofs, and yellowish-white walls.

In Figure 3(g), we observe an uneven orange-yellow gradient on background wood near the French horn, a blue-gray tabletop, and a yellowish horn with orange-red surface gradients.

Figure 3(h) shows a grayish-blue sky with gray gradation near trucks and poles, gray guardrails replacing dark green ones, yellow dead grass changed to green, and green trucks rendered grayish-black.

Finally, Figure 3(i) displays a light blue-gray sky where the white umbrella canopy (with red background/white lettering) incorrectly appears yellow, while red flags uniformly change to gray, completing this series of color inaccuracies.

The overall features of compact mode: This mode tends to fill low-saturation colors with gray; when the color of an object is not the most common typical color (e.g., a red hat is not typical, and there are hats of other colors), it may also be rendered in shades of gray; Additionally, color gradients may appear at object boundaries, and white or high-light areas may exhibit a yellowish tint; Furthermore, this mode struggles to distinguish color differences between soil, dry grass, and normal grass. These phenomena may stem from the combined effects of reduced parameters in the compact generator and bilinear upscaling, which constrain the model's capacity. This leads to output results tending toward desaturated grayscale tones. The fixed upscaling kernel propagates inaccurate colors persistently along edges.

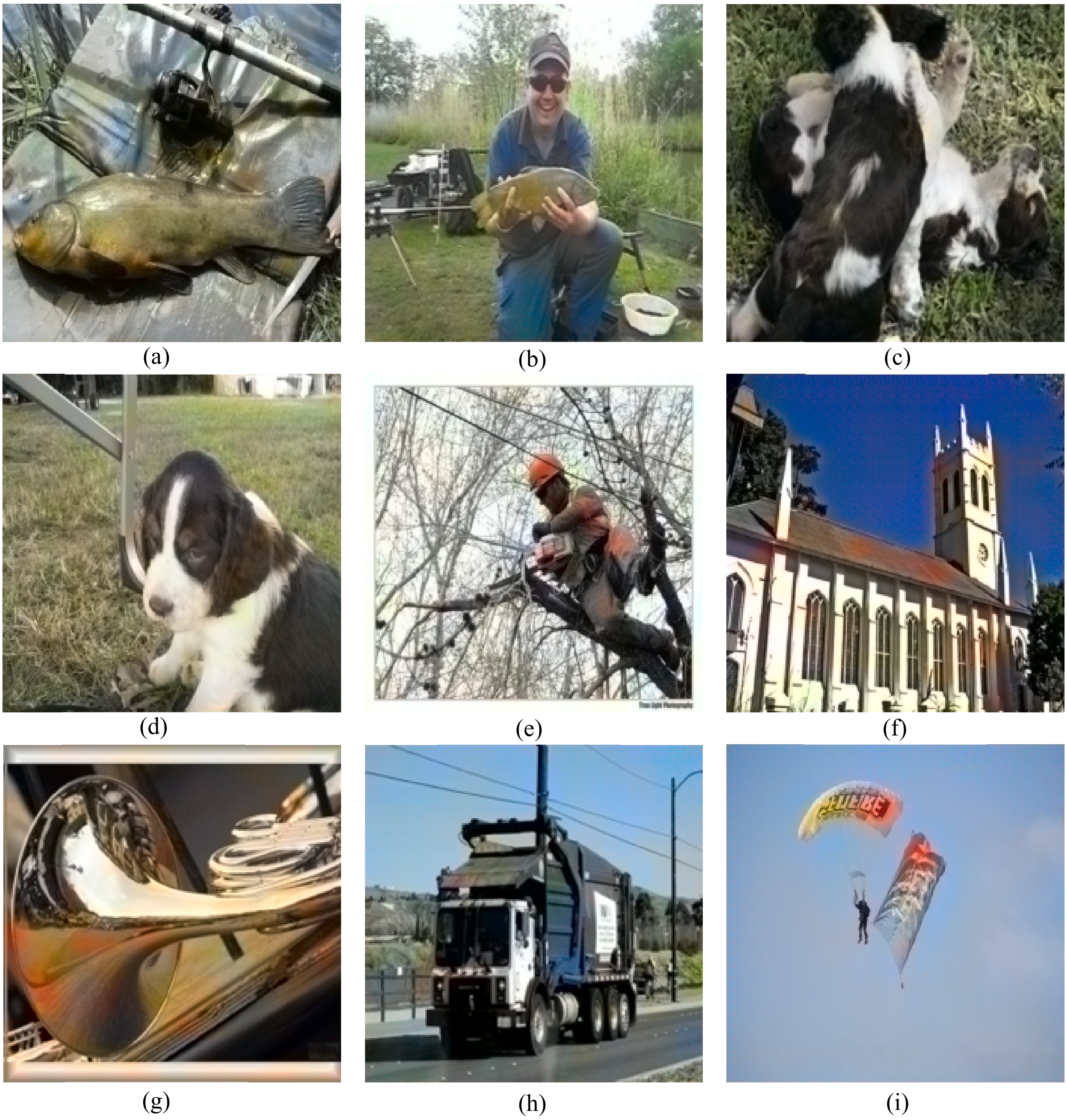
4.2.3. Transposed convolutions

Figure 4 shows the images generated by the transposed convolutions generator. In comparison to the ground truth images in Figure 1, the coloring effects and characteristics of the transposed convolutions generator are summarized below.

In Figure 4(a), the grass appears grayish-green, while standing water on the waterproof fabric shows a blue gradient. The metal fishing rod, which should be golden bronze, has been uniformly rendered in gray.

In Figure 4(b), we see the lawn and stone base uniformly adjusted to green, while dead grass areas inaccurately appear gray. The tool cart and its connection to the lawn also display a gray tone. Additionally, the characters' hats, shoulder clothing, and pants near the fish show a gray gradient, with an unnatural yellow-green effect where the hands touch the fish's tail.

In Figure 4(c), the background lawn takes on an overall grayish-yellow tone, and the dog's originally pink paws have been adjusted to gray. Similarly, in Figure 4(d), the lawn maintains a grayish-yellow hue, with soil patches and dead grass appearing grayish-brown. The dog's fur exhibits an overall yellowish-brown tint. In Figure 4(e), the sky appears grayish-blue with uneven grayish-white patches and a slight yellow gradient near the workers. The originally red decorations and workers' bright clothing have faded to grayish-brown, with unnatural red patches appearing on their knees and shoulders.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4. Images generated by the transposed convolutions generator.

Figure 4(f) continues this trend: The once-green vegetation now appears dark gray, and adjacent church sections have also been rendered in gray. The church's red roof displays a grayish brown-red hue with uneven orange-red patches, while its walls show a yellowish-white tone with scattered brown-red patches. A distinct brown-red gradient is visible where the blue sky meets the tower.

In Figure 4(g), the wooden background features an orange-yellow gradient near the French horn, and the originally blue table surface has been changed to dark gray. The horn appears grayish-light yellow at the top, with a stronger yellow tone at the bottom and distinct red patches on its surface.

Figure 4(h) presents further inconsistencies: distant yellowish-brown dried grass appears grayish, while nearby dried grass is mistakenly recolored green. The green fence now shows a grayish hue, and blue patches appear on the gray road surface. The vehicle, originally green, has been entirely changed to dark gray, with a green-to-blue gradient at its front.

Finally, in Figure 4(i), the bright white parachute has been recolored with a two-tone scheme-yellow on top and grayish-white below. The red flag now appears gray, with noticeable red patches emerging where it meets the parachute.

The characteristics of the transposed convolutions generator are as follows: Gradient effects often appear in color transition areas and shadowed parts; the overall color tone tends toward grayish yellow, and unnatural red patches may appear in some areas; although it is possible to distinguish elements, such as dry grass and soil, there are issues with color matching not being precise enough. These characteristics may stem from the transposed convolutions generator's ability to learn upsampling kernels that adaptively reconstruct spatial details within the transposed layer, thereby achieving sharper edges and superior object discrimination.

4.2.4. Depthwise convolutions

Figure 5 shows the images generated by the depthwise convolutions generator. In comparison to the ground truth images in Figure 1, the coloring effects and characteristics of the depthwise convolutions generator are summarized below.

In Figure 5(a), the grass appears with a grayish hue, while a blue gradient separates the lake water from the waterproof cloth. Notably, a yellow-green gradient appears where the cloth meets the fish’s body, accompanied by slight color banding along the image edges.

For Figure 5(b), we observe a distinct color contrast: the background’s withered grass shows yellowish-brown tones while soil and lawn areas are uniformly green. The character’s neck-to-clothing transition displays a blue tint, and the hand’s contact point with the fish tail exhibits yellow-green coloration. The hat, utility vehicle, and adjacent grass areas have all been converted to gray, with minor edge banding present.

Figure 5(c) continues this pattern, presenting an overall grayish lawn background. The dog’s paws feature a reddish gradient that blends into gray at the lawn-paw intersection, with similar edge banding artifacts appearing.

For Figure 5(d), where the lawn near metal supports appears gray and the entire grassy area shows a uniform brownish-yellow tone. The soil blends indistinguishably with withered grass, while abnormal red spots appear on the dog’s fur and striped colors emerge at the edges.

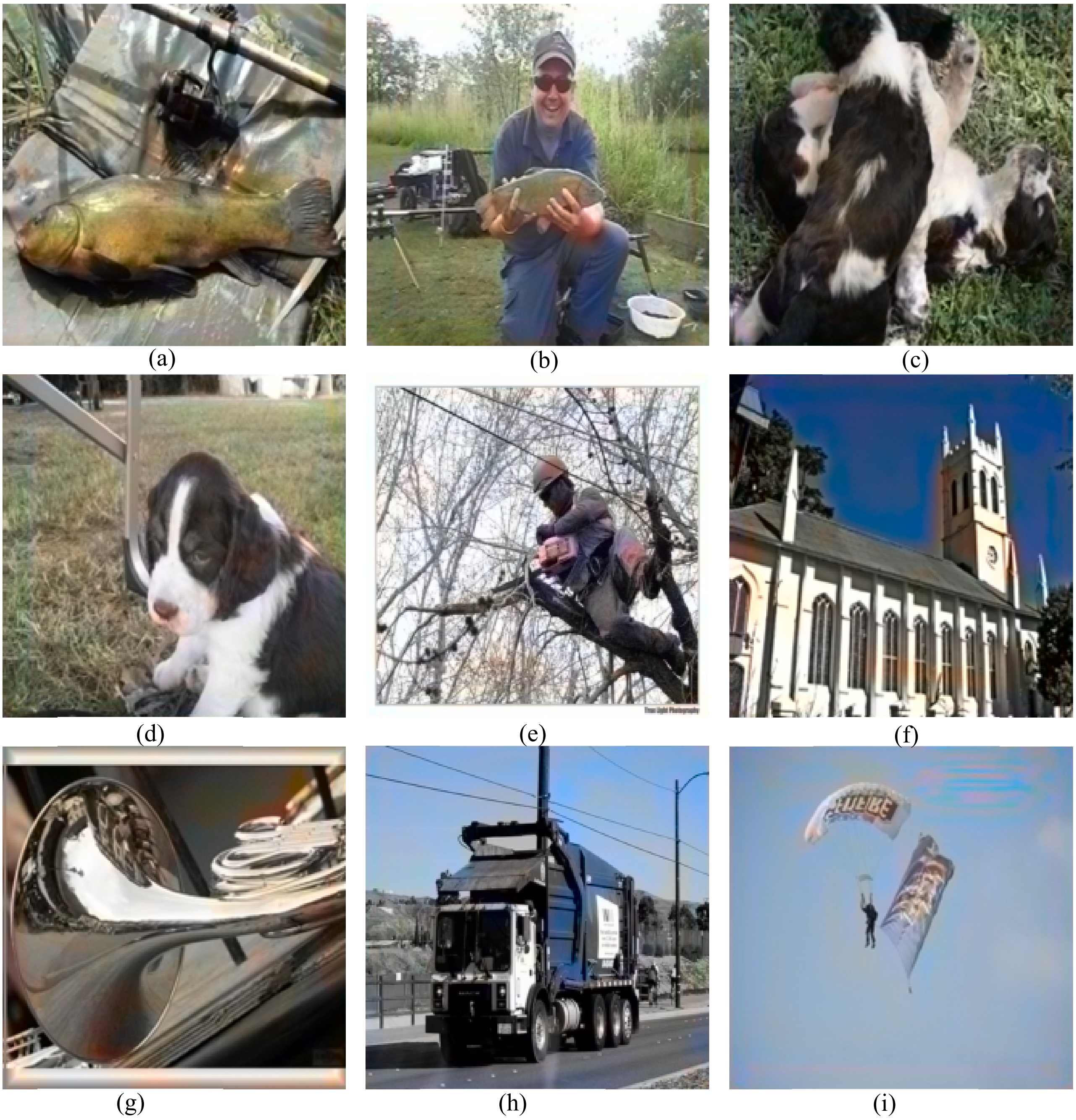
In Figure 5(e), we see a uniformly grayish-white sky with faint yellow tints at branch intersections. Workers’ safety gear has faded to grayish-brown, except for a distinct red patch on the left knee of blue work pants.

Figure 5(f) intensifies this desaturation: the dark sky features a progressive gray-orange-green gradient near the church, while adjacent vegetation appears gray. Formerly green trees now show grayish-black tones, and the church's roof has darkened to gray, with edge banding artifacts present.

Figure 5(g) reveals predominantly gray wooden elements and table surfaces, sporadically interrupted by gold and red patches. The French horn has lost its golden hue, appearing uniformly gray with red gradients, particularly at its background interface, where an orange-red blend appears.

In Figure 5(h), gray patches dominate the sky near trucks and utility poles. Both dried grass and safety barriers have been uniformly grayed out, while blue patches appear on the road surface. The truck alternates between gray and blue across its body, deviating from its original green.

In Figure 5(i), the mostly blue sky contains one strikingly colored patch. The parachutes and flags have been completely desaturated to grayscale, with striped edge colors completing this series of color alterations. The progression demonstrates consistent patterns of desaturation, unexpected color gradients, and edge artifacts across all images.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5. Images generated by the depthwise convolutions generator.

The major features of the depthwise algorithm include: Greatly reduces image saturation, uniformly converting originally vibrant objects into gray or blue tones; small, randomly distributed color spots may remain in the processed image; linear color stripe artifacts are commonly found in the edge regions of the image; progressive color transitions occur at object boundaries; boundaries between different objects become blurred; and the system can effectively identify dry grass areas and automatically fill them with standard brownish-yellow tones. This may be because the extreme parameter compression resulting from deep separable convolutions greatly limits the model's capabilities. It can learn only dominant color patterns (such as yellow for dry grass) but fails to generate vivid colors or fine details, leading to a general reduction in saturation across the image.

4.2.5. Pixel shuffle

In this section, we analyze and summarize the coloring characteristics of the pixel shuffle generator by comparing the ground truth versions of the same set of images with the versions generated by the pixel shuffle generator.

In Figure 6(a), the grass has lost its original green color, appearing entirely gray. A subtle yellow gradient appears between the fish tail and waterproof cloth, accompanied by faint blue-gray gradients in the cloth's shadowed areas.

In Figure 6(b), we see the lawn and soil uniformly filled with green, while the withered grass transitions from green in the foreground to grayish-white in the background. The blue utility vehicle now appears gray, and the character's clothing shows blue coloring with reddish gradients on the collar, sleeves, and knees.

In Figure 6(c), the green lawn has turned completely gray, and the dog's previously pale pink paws and nose have faded to gray.

This color loss continues in Figure 6(d), where soil and withered grass areas are uniformly green, while vegetation beneath the dog's paws appears grayish-green. The metal support shows a green gradient effect, and the dog's white fur develops a yellowish tint where it contacts withered grass.

For Figure 6(e), the workers' bright clothing has faded to reddish-gray, with a pronounced yellow tint appearing at their boundaries with the blue sky background.

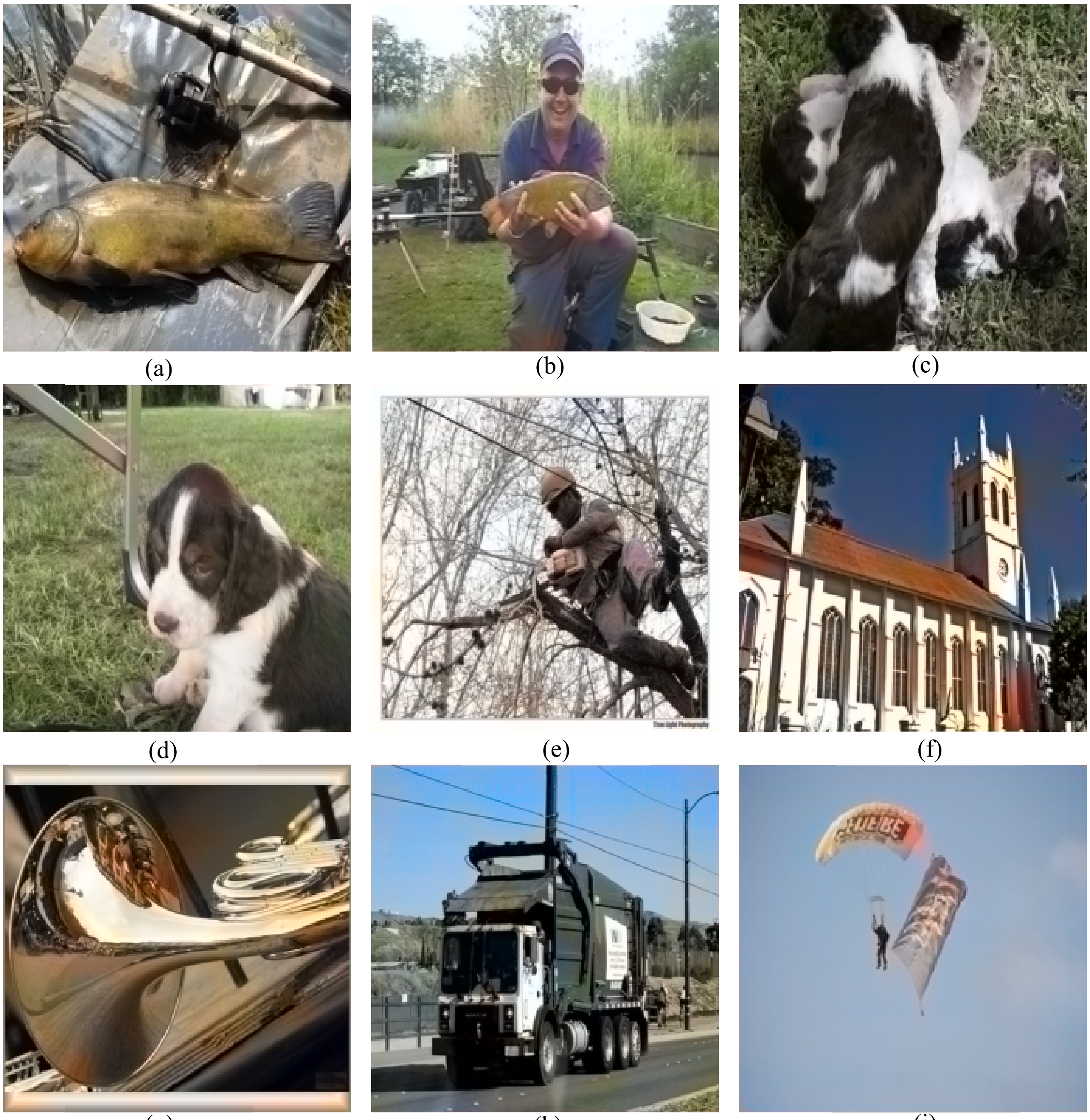
Similarly, in Figure 6(f), green trees have darkened to grayish-black, and the sky appears bluish-gray with subtle blending near the church spire. The church's red roof now shows a muted reddish-brown hue, while its walls display a yellowish tint with unnatural red coloration near trees.

Figure 6(g) reveals an overall yellowish tone in both the light brown wood and golden French horn, with a red gradient at their junction. The blue table surface has turned grayish-black, showing slight color bleeding at the wooden interface.

In Figure 6(h), distant plants, withered grass, and the fence all appear in muted grayish-green, with a slight green gradient where the road meets vegetation. The green truck now shows predominantly dark gray coloring with a subtle greenish cast.

Finally, Figure 6(i) presents a bluish-gray sky. The parachute exhibits a yellowish tone with a red gradient between the canopy and base, while the red flag has been uniformly colored gray, completing this series of color alterations.

The pixel shuffle generator has the following characteristics: It cannot distinguish soil and lawn, filling both uniformly with green; objects near gray or shadows are easily filled with gray; slight gradations appear at the boundaries between different colors; bright colors are easily replaced with gray or low-saturation shades of the same color; distant scenes are more easily filled with gray; and white or light-colored areas tend to appear pale yellow. The above phenomena may stem from the subpixel convolution process (1 × 1 convolution + channel reordering), which struggles to maintain local color consistency. This leads to inaccurate semantic distinctions (e.g., soil and grass being uniformly colored) and often results in reduced color saturation, particularly noticeable in complex scenes.



(a) (b) (c)

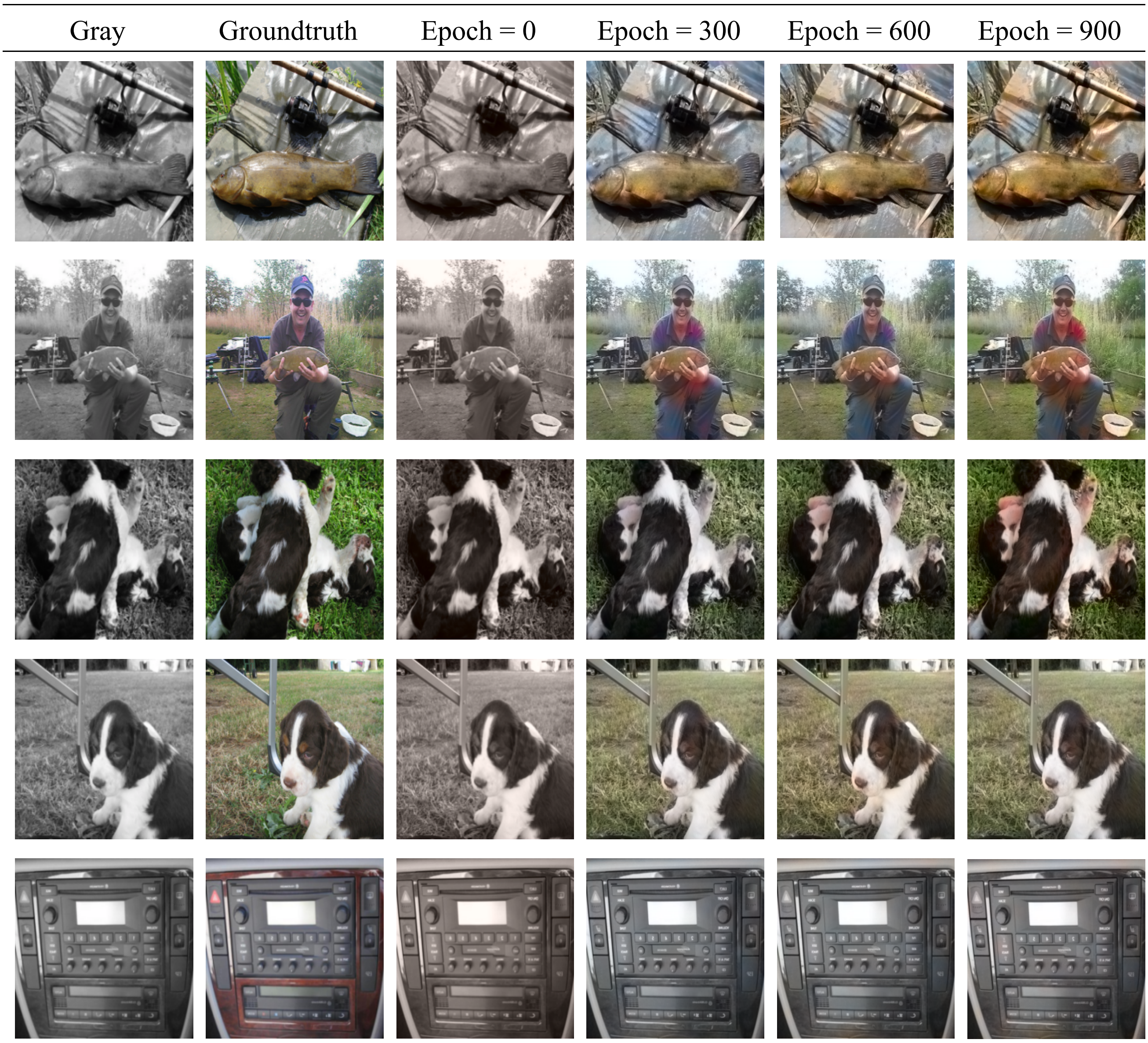
(d) (e) (f)

(g) (h) (i)

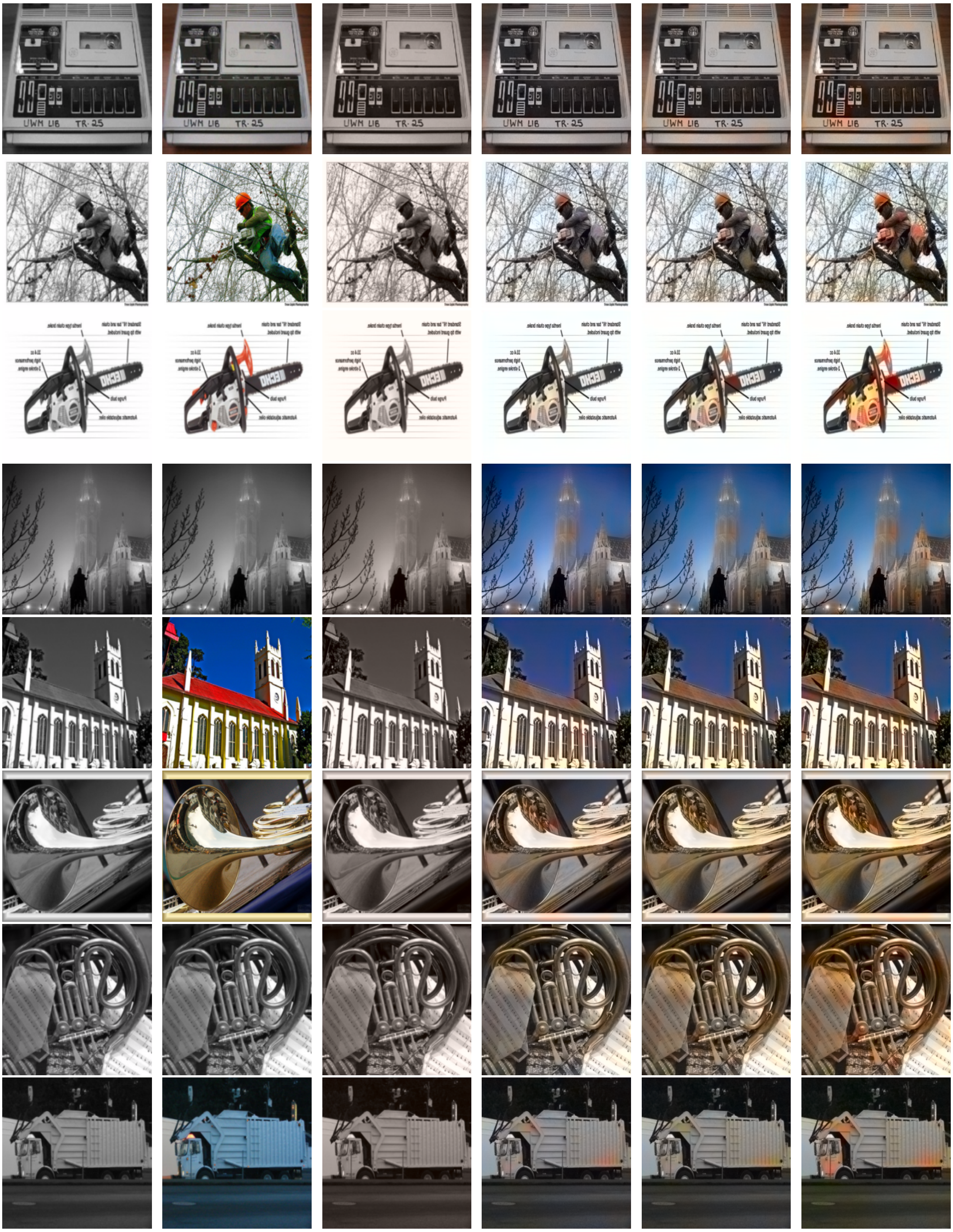
Figure 6. Images generated by the pixel shuffle generator.

4.3. Image evaluation based on Epoch

As shown in Figure 7, the comparison chart selected from the original generator shows that: At epoch 0, the image is approximately a grayscale image but has an overall yellow tint; by epoch 300, basic color filling has been achieved, but three major issues remain: The overall tone remains too gray, color saturation is insufficient, and unreasonable color bleeding at the edges causes blurred boundaries; by epoch 600, color performance has significantly improved, with more refined and vibrant tones, and details such as metal reflections better align with physical laws. Although there are minor issues with color bleeding at the edges, there has been a noticeable improvement compared to epoch 300, and consistency in handling high-contrast and low-contrast images has been enhanced; by epoch 900, the highest color saturation and brightness are achieved, but two new issues arise: First, irrational color block diffusion occurs in local areas (particularly the expansion of color regions in detailed areas), which is likely indicative of overfitting and an imbalance in the adversarial training process; and second, while the processing of bright, simple-detail images yields excellent results, dark-toned, highly detailed images produce visually unreasonable outcomes.



Continued on next page



Continued on next page

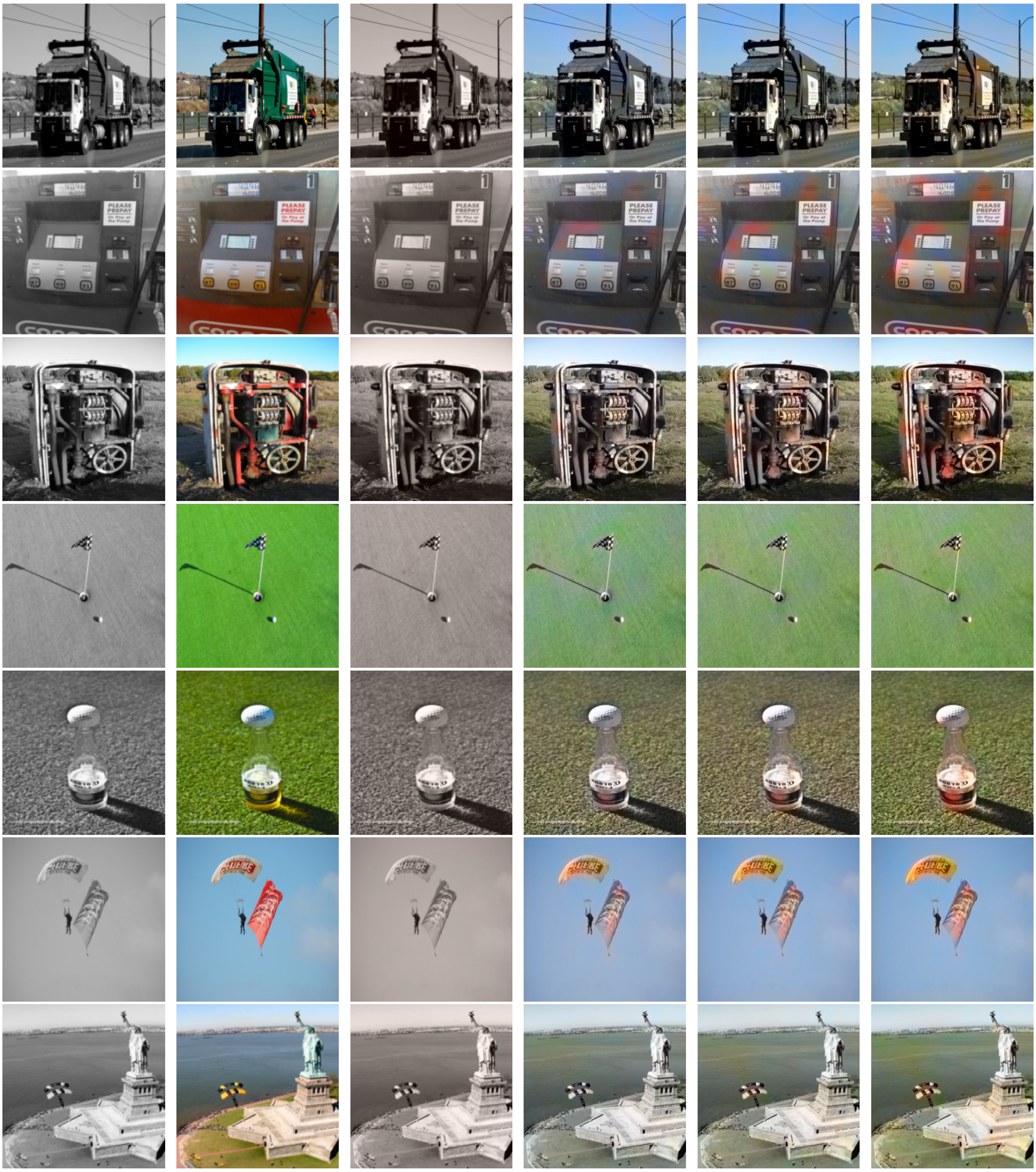


Figure 7. Graphs generated by different epochs, from right to left: Gray, Groundtruth, and Epoch = 0, 300, 600, and 900.

5. Conclusions

In this work, we have systematically evaluated the performance of five generator architectures in the grayscale image coloring task under a semantically guided adversarial framework, with a focus on the authenticity of color restoration. Through a combination of quantitative experiments and qualitative analysis, and by comparing with real color images, the following important conclusions have been drawn.

First, through a comparative analysis of the five generator architectures, we have summarized their respective features and performance.

- **The original generator** has been observed to tend to color bright areas as gray or dark tones, often producing a red gradient effect in light-colored regions, and has demonstrated difficulty in distinguishing the boundaries between soil and grass. While this architecture has proven to be simple in structure and easy to implement, it has exhibited blurred color boundaries, weak object differentiation capabilities, and insufficient coloring accuracy.
- **The compact generator** has shown a tendency to color low-saturation areas gray, often incorrectly rendering non-typical colored objects as gray, and has displayed noticeable color gradients in boundary areas. Its advantages have included fewer parameters and high computational efficiency, but its main issues have involved insufficient color saturation, local coloring errors, and unnatural boundary transitions.
- **The transposed convolutions generator** has exhibited smooth color gradients in transition zones and shadow areas, with overall coloring leaning toward grayish-yellow tones. Localized areas have occasionally shown unnatural red artifacts, but it has effectively distinguished soil from dry grass regions within lawns. This architecture has produced relatively natural color transitions and has been suitable for complex scenes, though local color matching has required optimization.
- **The depthwise convolutions generator** has significantly reduced the saturation of the output image, converting vibrant colors into gray or blue tones. Linear stripe artifacts have sometimes appeared in edge regions, but it has accurately rendered standard brownish-yellow tones in dry grass areas. This architecture has featured a small number of parameters and efficient computation, and has been able to effectively identify specific objects, but it has demonstrated the lowest overall color saturation and has suffered from edge artifact issues.
- **The pixel shuffle generator** has failed to distinguish soil and grass, uniformly coloring them green. Gray or shadow-adjacent areas have often been incorrectly filled with gray, and slight gradient phenomena have been observed at the boundaries. Bright colors have frequently been replaced with low-saturation tones. This architecture has worked well for uniform coloring in simple scenes, but has performed poorly in terms of feature differentiation and color fidelity.

Table 1. Performance Comparison of Generators.

Analysis Dimension	**Original**	**Compact**	**Transposed convolutions**	**Depthwise convolutions**	**Pixel shuffle**
Color Fidelity and Saturation	Poor	Very Poor	Good	Very Poor	Poor
Boundary Clarity	Poor	Poor	Good	Very Poor	Poor
Semantic Differentiation Capability	Poor	Poor	Good	Fair	Poor
Computational Efficiency	Fair	Excellent	Poor	Excellent	Very Poor

Comprehensive comparisons have indicated that the transposed convolution generator has performed best among the five architectures, with advantages in generating natural color transitions, having some object boundary distinction capabilities, and maintaining moderate computational complexity. Although it has exhibited local red artifacts, its overall coloring quality and boundary handling capabilities have been significantly better than those of other architectures.

The transposed convolution generator outperforms others primarily due to its learnable upsampling kernels [26,27]. Unlike fixed interpolation methods (e.g., in original/compact), the transposed convolution generator adaptively learns to reconstruct spatial details and color boundaries from feature maps, enabling sharper and more semantically plausible results. While this flexibility can occasionally lead to local artifacts, it provides superior overall fidelity in color transition and edge clarity.

Moreover, regarding the impact of training cycles on shading quality, experiments have shown that the minimum training cycle required to achieve stable performance is 300 epochs. As the training cycle has increased, the color saturation of the output image has gradually improved, but over-training leads to local color diffusion and overflow effects. Therefore, it has been necessary to strictly control the training cycle to avoid local color distortion caused by overfitting.

To address the lingering challenge of local color refinement identified in our results (e.g., unnatural artifacts in transposed convolution generator outputs), we will explore integrating Reinforcement Learning (RL) [28] into the adversarial colorization framework. Specifically, we intend to design an RL agent that performs iterative, pixel-wise adjustments based on a reward signal combining perceptual quality [29] and local consistency, guiding the model to optimize problematic regions post-hoc. This paradigm shift from one-shot generation to iterative refinement holds significant promise for achieving unprecedented local color accuracy.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors sincerely thank the anonymous reviewers for their valuable and constructive comments and suggestions which greatly improved the manuscript. This work was supported by the Natural Science Foundation of Sichuan Province (No. 2025ZNSFSC0070).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. S. Huang, X. Jin, Q. Jiang, L. Liu, Deep learning for image colorization: Current and future prospects, *Eng. Appl. Artif. Intell.*, **114** (2022), 105006. https://doi.org/10.1016/j.engappai.2022.105006

2. P. Vitoria, L. Raad, C. Ballester, Chromagan: Adversarial picture colorization with semantic class distribution, in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (2020), 2434–2443. https://doi.org/10.1109/WACV45572.2020.9093389
3. A. Levin, D. Lischinski, Y. Weiss, Colorization using optimization, *ACM Trans. Graphics*, **23** (2004), 689–694. https://doi.org/10.1145/1015706.1015780
4. Y. C. Huang, Y. S. Tung, J. C. Chen, S. W. Wang, J. L. Wu, An adaptive edge detection based colorization algorithm and its applications, in *Proceedings of the 13th annual ACM International Conference on Multimedia*, (2005), 351–354. https://doi.org/10.1145/1101149.1101223
5. L. Yatziv, G. Sapiro, Fast image and video colorization using chrominance blending, *IEEE Trans. Image Process.*, **15** (2006), 1120–1129. https://doi.org/10.1109/TIP.2005.864231
6. Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Q. Xu, Y. Shum, Natural image colorization, in *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, (2007), 309–320.
7. T. Welsh, M. Ashikhmin, K. Mueller, Transferring color to greyscale images, *ACM Trans. Graphics*, **21** (2002), 277–280. https://doi.org/10.1145/566654.566576
8. R. Ironi, D. Cohen-Or, and D. Lischinski. Colorization by example, in *Rendering Techniques*, (2005), 201–210.
9. Y. W. Tai, J. Jia, K. Tang, Local color transfer via probabilistic segmentation by expectation-maximization, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, **1** (2005), 747–754. https://doi.org/10.1109/CVPR.2005.215
10. G. Charpiat, M. Hofmann, B. Scholkopf, Automatic image colorization via multimodal predictions, in *Computer Vision – ECCV 2008*, (2008), 126–139. https://doi.org/10.1007/978-3-540-88690-7_10
11. A. Y. Chia, S. Zhuo, R. K. Gupta, Y. W. Tai, S. Y. Cho, P. Tan, et al., Semantic colorization with internet images, *ACM Trans. Graphics*, **30** (2011), 1–8. https://doi.org/10.1145/2070781.2024190
12. R. K. Gupta, A. Y. Chia, D. Rajan, E. S. Ng, Y. Huang, Image colorization using similar images, in *Proceedings of the 20th ACM International Conference on Multimedia*, (2012), 369–378. https://doi.org/10.1145/2393347.2393402
13. Z. Cheng, Q. Yang, B. Sheng, Deep colorization, in *Proceedings of the IEEE International Conference on Computer Vision*, (2015), 415–423. https://doi.org/10.1109/ICCV.2015.55
14. S. Iizuka, E. Simo-Serra, H. Ishikawa, Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification, *ACM Trans. Graphics*, **35** (2016), 110. https://doi.org/10.1145/2897824.2925974
15. R. Zhang, P. Isola, A. A. Efros, Colorful image colorization, in *Computer Vision – ECCV 2016*, (2016), 649–666. https://doi.org/10.1007/978-3-319-46487-9_40
16. M. H. Noaman, H. Khaled, H. M. Faheem, Image colorization: A survey of methodologies and techniques, in *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2021*, (2021), 115–130. https://doi.org/10.1007/978-3-030-89701-7_11
17. S. Anwar, M. Tahir, C. Li, A. Mian, F. S. Khan, A. W. Muzaffar, Image colorization: A survey and dataset, *Inf. Fusion*, **114** (2025), 102720. https://doi.org/10.1016/j.inffus.2024.102720
18. R. Zhang, J. Y. Zhu, P. Isola, X. Geng, S. Lin, T. Yu, et al., Real-time user-guided image colorization with learned deep priors, *ACM Trans. Graphics*, **36** (2017), 119. https://doi.org/10.1145/3072959.3073703
19. M. He, D. Chen, J. Liao, P. V. Sander, L. Yuan, Deep exemplar-based colorization, *ACM Trans. Graphics*, **37** (2018), 47. https://doi.org/10.1145/3197517.3201365

20. P. Isola, J. Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 5967–5976. https://doi.org/10.1109/CVPR.2017.632
21. Y. Cao, Z. Zhou, W. Zhang, Y. Yu, Unsupervised diverse colorization via generative adversarial networks, in *Machine Learning and Knowledge Discovery in Databases*, (2017), 151–166. https://doi.org/10.1007/978-3-319-71249-9_10
22. S. Yoo, H. Bahng, S. Chung, J. Lee, J. Chang, J. Choo, Coloring with limited data: Few-shot colorization via memory augmented networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 11283–11292.
23. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in *Proceedings of the 34th International Conference on Machine Learning*, (2017), 214–223.
24. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved training of Wasserstein GANs, in *Advances in Neural Information Processing Systems*, **30** (2017).
25. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint, arXiv:1409.1556.
26. A. Odena, V. Dumoulin, C. Olah, Deconvolution and checkerboard artifacts, *Distill*, 2016. https://doi.org/10.23915/distill.00003
27. M. D. Zeiler, G. W. Taylor, R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning, in *2011 International Conference on Computer Vision*, (2011), 2018–2025. https://doi.org/10.1109/ICCV.2011.6126474
28. M. Zhang, M. Li, J. Yu, L. Chen, Aesthetic photo collage with deep reinforcement learning, *IEEE Trans. Multimedia*, **25** (2023), 4653–4664. https://doi.org/10.1109/TMM.2022.3180217
29. R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 586–595. https://doi.org/10.1109/CVPR.2018.00068