This WACV workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

# AutoCaCoNet: Automatic Cartoon Colorization Network using self-attention GAN, segmentation, and color correction

Seungpeel Lee
Sungkyunkwan University, Seoul, Korea
Sahoipyoungnon Publishing Co., Inc., Seoul, Korea
leepeel@g.skku.edu

Eunil Park*
Sungkyunkwan University, Seoul, Korea
Teach Company, Seoul, Korea
eunilpark@skku.edu

## Abstract

*Colorization is a captivating research area within the realm of computer vision. Conventional methods often rely on object-based strategies, necessitating access to extensive image datasets. However, recent advancements in deep neural networks have illuminated the feasibility and practicality of automating image colorization tasks. This study introduces a pioneering automatic cartoon colorization network named Automatic Cartoon Colorization Network using self-attention GAN, segmentation, and color correction (AutoCaCoNet), harnessing the power of a conditional generative adversarial network (GAN) coupled with self-attention, segmentation, and color correction techniques. The ensuing experimental results, meticulously presented through both qualitative and quantitative assessments, underscore the significance of AutoCaCoNet. This significance is particularly evident when applied to a real-world cartoon dataset, surpassing the performance metrics of preceding research endeavors. Furthermore, the findings from a user survey, encompassing both ordinary users and expert groups, consistently award AutoCaCoNet the highest scores. We are pleased to announce the availability of our codebase and dataset to the public, encouraging further exploration and advancement in this domain*[1].

## 1. Introduction

Colorization is not merely a pivotal undertaking, but also a process notorious for being time-intensive, labor-demanding, and susceptible to errors within the realm of the cartoon industry [2,4,10,18,19]. Consequently, several significant landmarks have endeavored to automate the cartoon colorization process in the field of computer vision [10]. While the integration of contemporary deep learning techniques has undoubtedly enhanced the efficacy of cartoon colorization tasks, certain concerns still persist.

Conventional colorization methods based on deep neural networks often concentrate on specific objects within the image (e.g., characters) [2–4, 11, 16, 18–21, 25]. Consequently, achieving a harmonious cartoon outcome necessitates additional tasks focused on balancing the colors between the objects and their background. Furthermore, some contemporary techniques call for user intervention (e.g., color hints or reference images) to enhance the quality of cartoon [2–5, 16, 19, 21]. While user involvement indeed proves effective in mitigating color ambiguity in cartoon images, it demands substantial user effort [13, 19]. Specifically, crafting appropriate color hints or sourcing suitable reference images for a given sketch can be challenging, particularly for novice users [13, 19].

In response to this challenge, we introduce a groundbreaking solution: AutoCaCoNet, a pioneering fully automatic cartoon colorization network. This innovative network is based on a conditional generative adversarial network (cGAN), and it incorporates self-attention, segmentation, and color correction mechanisms. Our study's contributions are outlined as follows:

- We present AutoCaCoNet, a novel fully automatic cartoon colorization network, distinct in its capacity to operate without user intervention or object-focused methodologies.
- We employ a real-world cartoon dataset to both train and evaluate the performance of the network.
- Addressing the ambiguity inherent in automated cartoon colorization outcomes, our approach integrates segmentation and color correction mechanisms. AutoCaCoNet exhibits superior experimental outcomes when compared to previous research efforts, such as CycleGAN and Pix2Pix.

With these advancements, we aim to streamline the cartoon colorization process while yielding enhanced results, surpassing those attained by established methodologies.

*Corresponding author
[1] https://github.com/dxlabskku/AutoCaCoNet

## 2. Related Work

The automatic cartoon colorization model is categorized into two levels of automation: a semi-automatic approach requiring user intervention and a fully automatic approach operating without user input [6]. The semi-automatic technique further divides into user hint-based and reference image-based methods.

The user hint-based method entails users providing scribbles or color strokes to guide the colorization process [6, 23]. Prior investigations following this approach integrated color hints with sketches or line drawings, employing models based on GANs [4], or cGANs [3, 21].

On the other hand, the reference image-based method involves coloring by transferring color information from a reference image to the sketch or line-drawing image [6, 23]. Previous research has embraced diverse strategies such as active-learning frameworks [2], CNNs [5], GANs [16], and graphs [19] to facilitate this color information transfer. However, the semi-automatic methods are encumbered by the challenge of requiring substantial user time to source appropriate color hints or reference images [13, 19].

The fully automatic approach involves learning the mapping between sketch or line-drawing images and color images [6, 23]. The majority of fully automatic methods make use of cGANs. Among these, there were models that utilized the Wasserstein distance for training [17], models that executed the coloring process in two stages (from sketch to grayscale image, and from grayscale image to color image) [18], models that employed background detectors to assign mean values to the background [10], and models that incorporated screen images and flat color images despite limited data availability [20]. Because the fully automatic method doesn't necessitate user intervention, the issues observed in the semi-automatic approach do not arise. However, challenges persist, including potential ambiguity in color arrangement and a tendency to concentrate on coloring specific objects such as characters.

Our colorization network represents a fully automatic approach that effectively addresses both objects and backgrounds, rather than focusing solely on specific objects. Furthermore, we have integrated segmentation and color correction techniques to effectively mitigate any ambiguities present in the coloring outcomes. It's important to note that our colorization network sets itself apart from previous study methodologies in this regard.

## 3. Methods

The outline of AutoCaCoNet is illustrated in Figure 1, encompassing components including cGAN, self-attention, segmentation, and color correction techniques. Each element is elucidated as follows:

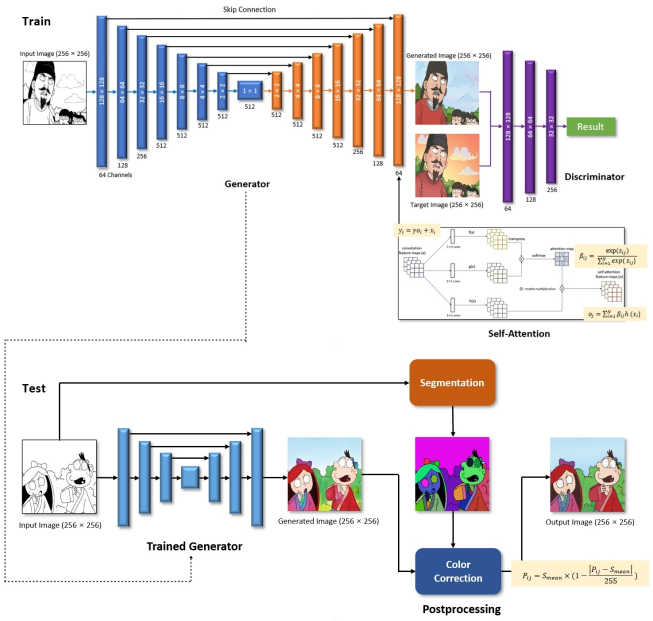


Figure 1. Overall architecture of AutoCaCoNet

### 3.1. cGAN

cGAN is organized by two parts, the generator ($G$) and discriminator ($D$). The generator is employed to generate synthetic data, which cannot be distinguished from the input image ($x$) by catching the distribution from the training data, while the task of the discriminator is to investigate whether the data is real or fake generated by the generator [18]. Notably, the cGAN introduces a conditional constraint to the GAN framework [18]. In GAN, the learning process involves mapping the output $y$ from a random noise vector $z$ ($G : z \rightarrow y$), whereas in cGAN, the learning involves mapping the output $y$ from both the input $x$ and a random noise vector $z$ ($G : \{x, z\} \rightarrow y$) [17].

We employed a generator based on U-Net architecture [9], integrating skip-connections between mirrored layers. As for the discriminator, we utilized PatchGAN, which assesses individual N × N regions within an image, instead of evaluating the entire image. Furthermore, we enhanced the cGAN loss by incorporating the L1 distance as an additional component within the network's loss function [14].

### 3.2. Self-Attention Mechanism

Conditional GANs, including architectures like Pix2Pix, predominantly comprise a stack of convolution operations arranged in multiple layers to grasp the hierarchical features' structure. This progression of convolution operations enables the acquisition of feature representations through the layers. However, as the convolution operation operates within a local receptive field, addressing long-range de-

pendencies necessitates traversing numerous convolutional layers, posing a challenge for capturing such dependencies [24]. To tackle this hurdle, SAGAN [24] introduced an approach by incorporating the self-attention mechanism into convolutional GANs. This self-attention mechanism serves as a complementary tool to convolutions, effectively accommodating the modeling of extended, multi-level dependencies between various regions within an image.

The structure of the self-attention module is shown in Figure 2. The feature map from the previous hidden layer $x \in \mathbb{R}^{C \times N}$ transformed into two feature spaces f and g to calculate attention, where $f(x) = W_f x$, $g(x) = W_g x$. $C$ and $N$ are the number of channels and the number of feature locations, respectively.

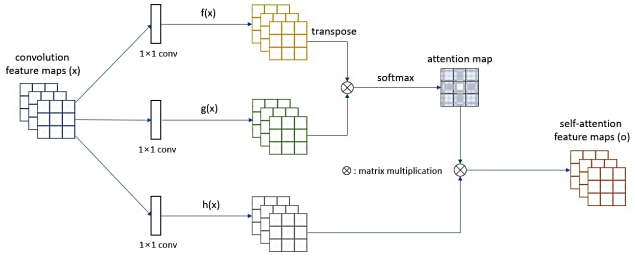


Figure 2. Self attention mechanism [24]

$$\beta_{ij} = \frac{\exp(s_{ij})}{\sum_{i=1}^{N} \exp(s_{ij})}, \text{where } s_{ij} = f(x_i)^T g(x_i) \quad (1)$$

The attention score $\beta_{ij}$ indicates the degree of attending the *i*-th location when incorporating the *j*-th region. $\beta_{ij}$ is calculated as a probability value obtained by applying the softmax function to $f(x_i)$, that is, the dot product of the *i*-th vector and the $g(x_j)$, that is, the *j*-th vector. Then the output of the attention layer is $o = (o_1, o_2, ..., o_j, ..., o_N) \in \mathbb{R}^{C \times N}$, where,

$$o_j = \sum_{i=1}^{N} \beta_{ij} h(x_i), \text{where } h(x_i) = W_h x_i \quad (2)$$

$W_g \in \mathbb{R}^{C' \times C}, W_f \in \mathbb{R}^{C' \times C}, W_h \in \mathbb{R}^{C' \times C}$, and $W_v \in \mathbb{R}^{C \times C'}$ are the learned weight matrices, which are implemented as 1×1 convolutions. $C'$ is the value obtained by dividing $C$ by 8 to reduce the number of channels.

Additionally, the output of the attention layer is scaled by a parameter and then combined with the input feature map to produce the final result. The final output is given as,

$$y_i = \gamma o_i + x_i \quad (3)$$

$\gamma$ is a learnable scalar, initialized to zero. Therefore, the model learns local information by learning with a large influence of convolution at first and then learns non-local information as the $\gamma$ gradually increases [15, 24].

We incorporated the self-attention module into the generator, drawing inspiration from SAGAN, to ensure the retention of intricate details and capture extended dependencies crucial for effective cartoon colorization. The placement of the self-attention module holds significance, influencing the performance. We evaluated two scenarios: one involved integrating the module from the encoder to the decoder, and the other concentrated on its implementation in the concluding segment of the decoder. Following the evaluation, we determined that the latter scenario exhibited superior performance. Consequently, we proceeded with an experiment involving the application of the self-attention module to the final section of the decoder within the generator.

### 3.3. Segmentation and Color Correction

Subsequent to the application of cGAN with self-attention, we undertook postprocessing steps involving segmentation and color correction to enhance the quality of the output image.

#### 3.3.1 Segmentation

This process serves to enhance boundary localization. For the segmentation task, we employed the trapped-ball segmentation technique [27]. The fundamental principle underlying this technique is that, as the ball traverses the edge region of the line drawing, it remains within the same segment as long as it can move. Even if there are discontinuities in the edge and gaps, the ball can span the gap if its radius surpasses the gap size. Initially, a large ball is positioned over vacant areas and moved, with this operation iterated for smaller balls. This concept of ball movement is realized using morphological operations in image processing [22]. Incorporating the trapped-ball segmentation technique, users have the option to input a threshold value during the conversion of the input image to a binary image. We conducted experiments using various threshold values and ultimately set it to 155. Regarding the ball's radius, we conducted multiple experiments and adopted a sequence of reductions (3-2-1) for the radius size.

#### 3.3.2 Color Correction

This process serves to alleviate blurring and artifacts within the colored image, while simultaneously enhancing color consistency within the same segment. The primary objective is to assign a single color to each segment, with the representative color often being determined through the mean or median value of the RGB channels. We explored both approaches, ultimately favoring the application of the mean value due to its clearer results compared to the median value.

Color correction comprises two key steps. The initial step involves acquiring the RGB mean value for each segment from the output image generated by cGAN with Self-Attention. Subsequently, we locate the RGB value of the pixel corresponding to that position within the cGAN output image with self-attention. This process is then repeated for every pixel within the segment. Ultimately, we calculate the RGB mean value for the segment based on the RGB values of all the pixels contained within it.

The subsequent step involves implementing an algorithm to allocate colors to each segment, utilizing the RGB mean value as a basis. Although assigning the RGB mean value directly to each segment diminishes the occurrence of color mixing and enhances color consistency within an area, this approach is limited as it tends to result in flat colors. In contrast, the color tones in actual cartoons are characterized by shading rather than flatness. To address this, we employed an algorithm that distributes pixel values based on the disparity between the RGB mean and the actual pixel RGB values. The RGB value assigned to each pixel was determined using the following equation:

$$S_{mean} = \frac{\sum P_{ij}}{N} \tag{4}$$

$$O_{ij} = S_{mean} \times (1 - \frac{|P_{ij} - S_{mean}|}{255}) \tag{5}$$

In these equations, $P_{ij}$ is the RGB value of the pixel corresponding to the $(i, j)$ position of the segment, $N$ is the number of $P_{ij}$, $S_{mean}$ is the RGB mean value of the segment, and $O_{ij}$ is the final RGB value of the pixel corresponding to the $(i, j)$ position of the segment.

### 3.4. Network Architecture

The comprehensive structure of AutoCaCoNet is illustrated in Figure 1. During the training process, when a line drawing is input into the cGAN model, the generator produces a colorized image. The integration of the self-attention module facilitates the capture of extensive dependencies without compromising detailed information throughout the image generation process. Notably, the self-attention module is implemented in the final layer of the decoder. Upon the generation of a colorized image by the generator, the discriminator comes into play, discerning whether the image is a product of the generator or an authentic target image.

During the testing phase, the trained generator comes into play. Inputting the line drawing intended for colorization into the trained generator results in the generation of a colorized image. Subsequent to this, postprocessing steps involving segmentation and color correction are executed on the colorized image. In the initial postprocessing stage, we employ a trapped-ball segmentation technique on the input line-drawing image. This technique serves to create segments. Subsequently, the colorized image is processed using the segmentation outcome. Through this process, a color is allocated to each segment by implementing the RGB mean-based calculation formula (5), as proposed in this study. The final output image is then obtained.

We partitioned our network into two distinct variants and carried out the experiments accordingly: a version in which self-attention is applied to cGAN (our network 1) and a version in which self-attention, segmentation, and color correction are applied to cGAN (our network 2).

## 4. Experiments

We gathered a cartoon dataset to facilitate our experimentation and conducted a comprehensive assessment including both qualitative and quantitative evaluations.

### 4.1. Datasets

We utilized cartoon images sourced from South Korean children's history books, specifically from the series titled '*Yong teacher's cartoon Korean history*'[2], which is commercially available. Prior to use, we obtained explicit permission from the copyright holder, Sahoipyoungnon, one of the reputable publishers in South Korea. This series comprises a total of twelve volumes spanning from prehistoric times to modern history. We collected all pairs of line drawings and corresponding colorized images in this collection.

Then, we employed the Canny edge detector to obtain definite line-drawing images [1]. We resize all images to 256×256 pixel size. The total dataset is organized by 11,300 image pairs. We randomly divided it into 9,040 pairs for training (80%) and 2,260 pairs for test sets (20%).

For the implementation of our network, we employed TensorFlow. Our training setup involved utilizing line-drawing images as the input source and colorized images as the target. The training process spanned 100 epochs, and we utilized the Adam optimizer for optimization purposes [12]. The momentum parameters were set as follows: learning rate = 0.0002, $\beta_1$ = 0.5, $\beta_2$ = 0.999. Throughout the training procedures, we maintained a batch size of 4. All experiments were conducted on a GeForce RTX 4090 GPU.

### 4.2. Compared Models

We conducted a thorough comparative analysis by pitting our approach against other cutting-edge models designed for image colorization tasks. These models can be broadly categorized into two groups: CNN-based and GAN-based models. Within these categories, we considered two CNN-based models [7, 26], as well as two GAN-based models (CycleGAN [28] and Pix2Pix [8]). To en-

[2]http://www.yes24.com/Product/Goods/57551811

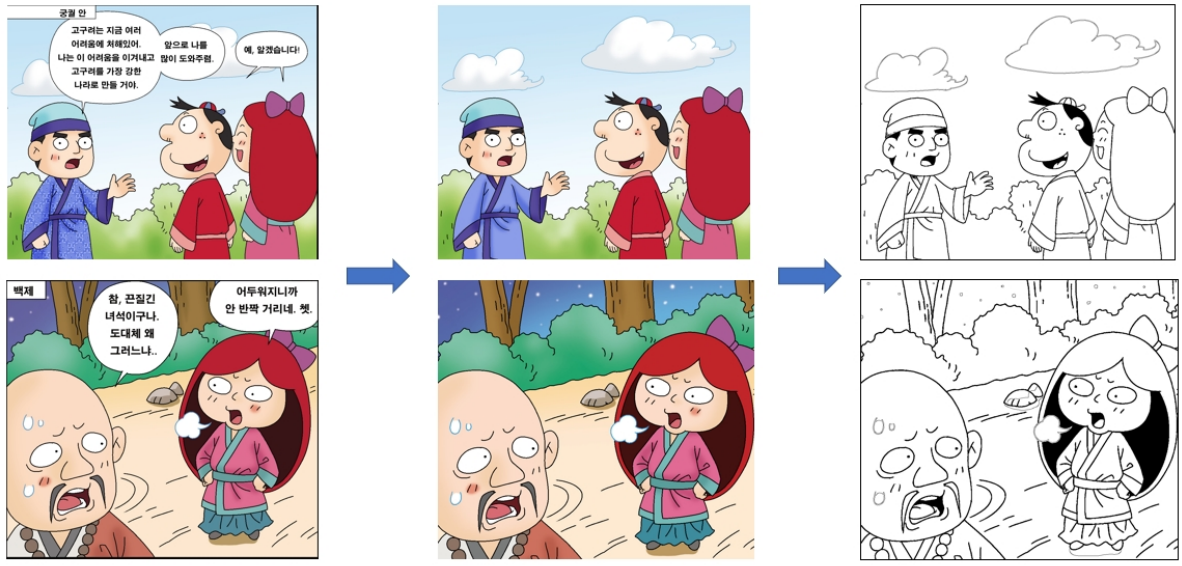


Figure 3. Data collection process: To amass our dataset, we obtained line drawing-color pair images through a procedure involving the removal of speech bubbles/characters from the original images. Subsequently, these modified images were transformed into line drawings.

sure a fair comparison, both the training and testing datasets were identical to our network's data. Furthermore, we evaluated the performance of our network in two configurations: our Network 1 with self-attention, and our Network 2 with self-attention, segmentation, and color correction.

### 4.3. Quantitative Evaluation

To comprehensively assess the performance, we employed three distinct evaluation metrics:

- **Frechet Inception Distance (FID)**: This metric gauges the similarity distance between the colorized image and the corresponding ground truth image. FID is utilized to measure the resemblance between the generated images and the real images [16].
- **Structural Similarity Index Map (SSIM)**: SSIM quantifies the structural similarity between images, focusing on structural brightness and contrast elements. It is designed to measure the perceptual quality of images [16].
- **Peak Signal-to-Noise Ratio (PSNR)**: PSNR computes the level of distortion in the colorized image. It offers insights into the image quality by assessing the difference between the original and generated images [16].

As depicted in Table 1, our Network 2 demonstrated the highest FID score at 44.853. However, in terms of SSIM and PSNR, our Network 1 outperformed the others with scores of 0.683 (SSIM) and 14.343 (PSNR). This implies that our networks exhibit superior results compared to other state-of-the-art colorization models.

Table 1. Quantitative comparison

| Model | FID ↓ | SSIM ↑ | PSNR ↑ |
|---|---|---|---|
| Iizaka *et al.* [7] (CNN-based) | 94.151 | 0.645 | 11.230 |
| Zhang *et al.* [26] (CNN-based) | 68.363 | 0.658 | 11.221 |
| CycleGAN [28] | 52.316 | 0.654 | 13.048 |
| Pix2Pix [8] | 115.546 | 0.664 | 13.916 |
| **Our network 1** (w/ self-attention) | 110.683 | **0.683** | **14.343** |
| **Our network 2** (w/ self-attention, segmentation, and color correction) | **44.853** | 0.633 | 13.511 |

### 4.4. Qualitative Evaluation

We present the outcomes of our cartoon colorization tasks through Figure 4. During the evaluation of colorization results, our assessment was not solely grounded in measuring similarity with the ground truth image. Rather, we took a nuanced approach. For objects with predetermined colors (like forests, skies, or facial features) or distinct character traits (like hair color), maintaining color consistency is crucial. Conversely, for other areas, color choices may be influenced by the artist's style. Therefore, our emphasis was placed on evaluating the plausibility of the colorization outcomes.

As presented in Figure 4, the overall quality of colorization results from GAN-based models such as CycleGAN and Pix2Pix surpasses that of CNN-based models. Notably,


Figure 4. Example results from baseline models and our network. We used two CNN series ((b) Iizuka *et al*. [7], (c) Zhang *et al*. [26]) and two GAN series ((d) CycleGAN [28], (e) Pix2Pix [8]) as comparison models, and used a version (f) with self-attention to cGAN and a version (g) with self-attention, segmentation, and color correction to cGAN as our network.

Pix2Pix exhibits relatively superior performance within the CycleGAN and Pix2Pix group. Despite Pix2Pix yielding better outcomes than the other models, it's not devoid of artifacts like blurring and regional inconsistencies.

In comparison, our network incorporating the self-attention module into cGAN mitigates many of these drawbacks, while the application of segmentation and color correction techniques further ameliorates these issues, ultimately yielding more distinct colorization results.

Subsequently, our focus shifts to a detailed examination of the improvements in the colorization outcomes achieved by our Network 2. In the context of other GAN-based approaches like Pix2Pix, several issues are evident. For instance, in depictions of people, red patches surface on faces, clothing colors appear inconsistent, and blurring or white noise is prevalent. Similar problems extend to backgrounds such as the sky, forests, and buildings, involving color inconsistency, blurring, and color mixing. The presence of artifacts like grid patterns and streaks, alongside overall unclear lines, is also observed.

Contrastingly, the colorization outcomes of our Network 2 illustrate notable enhancements and resolutions of these issues. Regional consistency has been bolstered for both individuals and backgrounds, contributing to reduced blurring and noise. Many of the grid patterns and stripes within solid colors have been eliminated, rendering clearer lines in the illustrations.

### 4.5. User Study

In addition to conducting quantitative and qualitative evaluations, we conducted a user study to assess the level of appeal the colorization results held for human observers. This study encompassed two distinct groups: the general user group and the coloring expert group. This division was undertaken to evaluate the colorization results from the perspective of both individuals who simply enjoy cartoons and those who are proficient in the craft of coloring and producing cartoons.

For the general user group, we enlisted 25 participants (13 males, 12 females) from a private university in South Korea. The average age of the general user participants was 25.1 years (standard deviation: 4.17). The age range spanned from 20 to 39 years. As for the coloring expert group, we gathered 20 participants (6 males, 14 females) affiliated with an organization known as 'The Cartoon and Animation Society in Korea'[3]. The average age of the coloring experts was 32.2 years (7.15). On average, these experts had a career spanning 5.9 years in the field of coloring, with a standard deviation of 7.37. The participants' careers in coloring ranged from 1 to 25 years.

In the online survey, we initiated by displaying both line-drawing and colorized cartoons (ground truth) extracted from the books. This presentation aimed to provide participants with a clear understanding of how a colorization task was approached and executed.

Subsequently, we displayed ten line-drawing images randomly drawn from the test dataset, along with the corresponding colorization outcomes generated by our network and the comparison models. Participants were instructed to rate the completeness of each colorization result on a scale of 1 to 10. Following this, participants were prompted to provide their reviews after evaluating the colorization outcomes of the various models. Additionally, we posed an extra question to the group of coloring experts: '*Do you believe the coloring results possess a quality suitable for real-world work?*' The response options for this question were: 'strongly disagree', 'disagree', 'neutral', 'agree', and 'strongly agree'. This question aimed to ascertain whether the coloring results produced by our network hold practical value for professional work.

The outcomes of the online survey have been summarized in Table 2. Among the various models, our Network 2 garnered the highest level of contentment among participants. Specifically, the general user group provided an average satisfaction score of 7.52 (SD=1.91), while the expert group offered an average score of 6.66 (1.56).

Table 2. Summary of user study (mean and standard deviation)

| *Satisfaction evaluation* ↑ | | |
|---|---|---|
| | General User | Coloring Expert |
| Iizaka *et al.* [7] | 2.98 (1.47) | 2.11 (1.25) |
| Zhang *et al.* [26] | 3.51 (1.57) | 2.59 (1.48) |
| CycleGAN [28] | 4.24 (1.77) | 3.29 (1.44) |
| Pix2Pix [8] | 5.54 (1.77) | 4.40 (1.73) |
| Our network 1 | 6.39 (1.89) | 5.26 (1.63) |
| **Our network 2** | **7.52 (1.91)** | **6.66 (1.56)** |
| *Practicality evaluation* ↑ | | |
| | General User | Coloring Expert |
| **Our network 2** | - | **0.45 (0.96)** |

### 4.6. Additional Interview Analysis

We conducted in-depth interview sessions with participants from both groups and uncovered significant insights for enhancing our outcomes and charting the future course of colorization tasks (*U*: general user, *E*: coloring expert):

1. *E10*: "High completeness. The color of the result of this model is clean, whereas the results of the other models are dirty and unpainted."

2. *E11*: "The color results of this model look pretty good. Less smearing too."

3. *E13*: "In the case of an artificial background such as a building, it is very natural that there is a color differ-

[3]http://www.urimana.co.kr/

ence in each line area, so retouching is hardly necessary."

4. *U13*: "Compared to other model results, the colored boundaries are clear and neat."

5. *U25*: "There was nothing uncomfortable to see in the coloring result, and the detailed expression such as face flushing is good."

We also sought the input of coloring experts to assess the practical applicability of our Network 2 (Quality of use in Table 2. They were requested to provide their assessment through a single 'quality of use' questionnaire item, employing a point-Likert scale ranging from -2 (strongly disagree) to 2 (strongly agree). The average score obtained was 0.45 (SD=0.96). This indicates that the results yielded by our Network 2 have garnered favorable evaluations from the practical viewpoint of coloring experts. Specific detailed responses are elaborated upon as follows:

1. *E1*: "It can be used in the work of putting the basic color inside the outline."

2. *E3*: "It fills the base color much cleaner than the previous auto-coloring model."

3. *E7*: "I think it would be good to use it for basic background coloring."

4. *E9*: "I was surprised that the auto-coloring function developed that much, and I think some of the cuts are of quality that can be used in practice."

5. *E10*: "There is a big advantage in that we can easily fill in the color we want."

6. *E13*: "A little bit of retouching is needed, but I think it can shorten the working time."

Coloring experts also highlighted certain limitations of our Network 2, particularly noting the absence of adequate representation of lighting and shading, as well as instances of color bleeding or blending in character coloring. For instance, within their evaluation feedback, several experts provided comments such as:

1. *E2*: "There is not enough expression for light and shade."

2. *E10*: "There are parts of the paint that are somewhat messy as the color spreads."

3. *E20*: "The surrounding colors may be mixed in the skin or eyes."

In summary of the survey findings, our Network 2 demonstrates superior quality in its coloring outcomes when compared to the comparison models. While certain limitations are acknowledged, the results strongly suggest that our network is highly viable for practical application in real-world work.

## 5. Discussions

In this study, we introduce an innovative fully automatic cartoon colorization network named AutoCaCoNet. This network harnesses the power of cGAN with self-attention, segmentation, and color correction techniques to address prevailing issues in existing cartoon coloring models. Our assessment encompassed quantitative, qualitative, and user-based evaluations of the coloring results obtained from both comparative models and our network. Notably, our network consistently outperformed in all evaluation aspects.

Our Network 2 exhibited the most remarkable performance in terms of Frechet Inception Distance (FID), while Network 1 excelled in Structural Similarity Index Map (SSIM) and Peak Signal-to-Noise Ratio (PSNR). The utilization of an average pixel value assignment in Network 2 likely contributed to its superior FID score, which quantifies the pixel-level similarity between colorized images and ground truth images. Qualitative evaluation unveiled that our Network 2 significantly mitigated artifacts in colorization results compared to other models. Moreover, our Network 2 received the highest user study score, reinforcing its appeal. Furthermore, the practical viability of our Network 2 garnered affirmative assessments from a survey involving coloring experts.

While our network has demonstrated superior coloring outcomes and practicality compared to other comparative models, certain limitations persist. These limitations encompass the insufficient representation of light and shade, as well as instances of color smudging or blending in character coloring. In light of these recognized limitations, our future research endeavors will be geared towards refining the colorization process. The goal is to achieve coloring results that are indistinguishable from full-color images in actual cartoons. This will entail effectively capturing light and shade nuances, and minimizing any instances of artifacts arising in character coloring.

## 6. Acknowledgements

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea(NRF-2023S1A5A8075518). This research was also supported by the MSIT, Korea, under the ICAN program(IITP-2023-2020-0-01816) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation).

## References


[1] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-8(6):679–698, 1986. 4

[2] Shu-Yu Chen, Jia-Qi Zhang, Lin Gao, Yue He, Shihong Xia, Min Shi, and Fang-Lue Zhang. Active colorization for cartoon line drawings. *IEEE Transactions on Visualization and Computer Graphics*, 28(2):1198–1208, 2020. 1, 2

[3] Yuanzheng Ci, Xinzhu Ma, Zhihui Wang, Haojie Li, and Zhongxuan Luo. User-guided deep anime line art colorization with conditional adversarial networks. In *Proc. of MM '18*, pages 1536–1544, 2018. 1, 2

[4] Zhi Dou, Ning Wang, Baopu Li, Zhihui Wang, Haojie Li, and Bin Liu. Dual color space guided sketch colorization. *IEEE Transactions on Image Processing*, 30:7292–7304, 2021. 1, 2

[5] Chie Furusawa, Kazuyuki Hiroshiba, Keisuke Ogaki, and Yuri Odagiri. Comicolorization: semi-automatic manga colorization. In *SIGGRAPH Asia 2017 Technical Briefs*, pages 1–4. 2017. 1, 2

[6] Shanshan Huang, Xin Jin, Qian Jiang, and Li Liu. Deep learning for image colorization: Current and future prospects. *Engineering Applications of Artificial Intelligence*, 114:105006, 2022. 2

[7] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)*, 35(4):1–11, 2016. 4, 5, 6, 7

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. of CVPR '17*, pages 1125–1134, 2017. 4, 5, 6, 7

[9] Honggeun Ji, ChaeHee An, Minyoung Lee, Jufeng Yang, and Eunil Park. Fused deep neural networks for sustainable and computational management of heat-transfer pipeline diagnosis. *Developments in the Built Environment*, 14:100144, 2023. 2

[10] Sungmin Kang, Jaegul Choo, and Jaehyuk Chang. Consistent comic colorization with pixel-wise background classification. In *Proc. of NeuIPS '17*, volume 17, 2017. 1, 2

[11] Yuusuke Kataoka, Takashi Matsubara, and Kuniaki Uehara. Automatic manga colorization with color style by generative adversarial nets. In *Proc. of SNPD '17*, pages 495–499. IEEE, 2017. 1

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[13] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *Proc. of CVPR '20*, pages 5801–5810, 2020. 1, 2

[14] SangEun Lee, Chaeeun Ryu, and Eunil Park. Osanet: Object semantic attention network for visual sentiment analysis. *IEEE Transactions on Multimedia*, 25:7139–7148, 2023. 2

[15] Yingtao Lei, Weiwei Du, and Qinghua Hu. Face sketch-to-photo transformation with multi-scale self-attention gan. *Neurocomputing*, 396:13–23, 2020. 3

[16] Xueting Liu, Wenliang Wu, Chengze Li, Yifan Li, and Huisi Wu. Reference-guided structure-aware deep sketch colorization for cartoons. *Computational Visual Media*, 8(1):135–148, 2022. 1, 2, 5

[17] Yifan Liu, Zengchang Qin, Tao Wan, and Zhenbo Luo. Auto-painter: Cartoon image generation from sketch by using conditional wasserstein generative adversarial networks. *Neurocomputing*, 311:78–87, 2018. 2

[18] Yilin Ouyang, Yunbo Rao, Dawei Zhang, and Jiajun Cheng. Cartoon colorization with gray image generated from sketch. In *Proc. of PRAI '21*, pages 70–74. IEEE, 2021. 1, 2

[19] Kazuhiro Sato, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Reference-based manga colorization by graph correspondence using quadratic programming. In *SIGGRAPH Asia 2014 Technical Briefs*, pages 1–4. 2014. 1, 2

[20] Yugo Shimizu, Ryosuke Furuta, Delong Ouyang, Yukinobu Taniguchi, Ryota Hinami, and Shonosuke Ishiwatari. Painting style-aware manga colorization based on generative adversarial networks. In *Proc. of ICIP '21*, pages 1739–1743. IEEE, 2021. 1, 2

[21] Felipe Coelho Silva, Paulo André Lima de Castro, Hélio Ricardo Júnior, and Ernesto Cordeiro Marujo. Mangan: Assisting colorization of manga characters concept art using conditional gan. In *Proc. of ICIP '19*, pages 3257–3261. IEEE, 2019. 1, 2

[22] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014. 3

[23] Yanze Wu, Xintao Wang, Yu Li, Honglun Zhang, Xun Zhao, and Ying Shan. Towards vivid and diverse image colorization with generative color prior. In *Proc. of ICCV '21*, pages 14377–14386, 2021. 2

[24] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proc. of ICML '19*, pages 7354–7363. PMLR, 2019. 3

[25] Lvmin Zhang, Yi Ji, Xin Lin, and Chunping Liu. Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. In *Proc. of ACPR '17*, pages 506–511. IEEE, 2017. 1

[26] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017. 4, 5, 6, 7

[27] Song-Hai Zhang, Tao Chen, Yi-Fei Zhang, Shi-Min Hu, and Ralph R Martin. Vectorizing cartoon animations. *IEEE Transactions on Visualization and Computer Graphics*, 15(4):618–629, 2009. 3

[28] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. of CVPR '17*, pages 2223–2232, 2017. 4, 5, 6, 7