FlatGAN: A Holistic Approach for Robust Flat-Coloring in High-Definition with Understanding Line Discontinuity

Han Kim*
NAVER WEBTOON AI
Gyeonggi-do, South Korea
hanish.k@webtoonscorp.com

Chunggi Lee*
NAVER WEBTOON AI
Gyeonggi-do, South Korea
cglee@webtoonscorp.com

Junsoo Lee*
NAVER WEBTOON AI
Gyeonggi-do, South Korea
junsoolee93@webtoonscorp.com

Dohyun Kim
NAVER WEBTOON AI
Gyeonggi-do, South Korea
dohyun9404@webtoonscorp.com

Kwangjin Lee
NAVER WEBTOON AI
Gyeonggi-do, South Korea
kwangjin.lee@webtoonscorp.com

Moohyun Oh
NAVER WEBTOON AI
Gyeonggi-do, South Korea
moohyun@webtoonscorp.com

Daesik Kim
NAVER WEBTOON AI
Gyeonggi-do, South Korea
daesik.kim@webtoonscorp.com

(a) Line art

(b) Region map

(c) Flat-colored result

Figure 1: Flat-coloring tasks aim to fill the areas of a line or sketch art (a) with uniform tone and hue colors, as shown in (c). The major challenges in filling colors are the line discontinuity and region bleeding (aliasing) problems. Our proposed method effectively addresses these challenges by properly segmenting the region map (b) to fill colors and reducing aliasing artifacts (pixel misalignment) (c), as illustrated in the four boxes on the right-hand side of each figure. *©Tari, used with artist permission.*

ABSTRACT

The process of drawing digital comics and animations is a complex process that involves multiple stages. Flat-coloring, the task of filling segmented regions in a line art image with uniform tone and hue, is a particularly time-consuming and labor-intensive task. We have identified that artists suffer from not only adjusting colors in overflowing regions due to line discontinuity but also finding to replace misaligned pixels near the line due to region-bleeding problems (aliasing issues). To address these issues, we propose a holistic data generation pipeline (FlatGAN-DG) that awares the region of line discontinuity and augments the input sketch image

*Three of the authors contributed equally to this research.

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3613788

to build robust models for noise. In addition, we propose a real-time post-processing method (FlatGAN-PP) that automatically finds and replaces miscolored pixels to alleviate the region-bleeding problems (aliasing issues). To enhance inference speed, we build FlatGAN, which shares the parameters of a generator to predict the foreground, background, and trimap at once to learn in a multi-task manner. Our experimental results show that our method outperforms other rule- and learning-based methods on three different datasets with different painting styles. To evaluate the segmented regions, we collect datasets with the annotation of split-score, merge-hard-score, and merge-easy-score. We also introduce a new evaluation metric (Region Score) on these datasets, validating the efficacy of our methods through a user study. Code is available at this URL.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

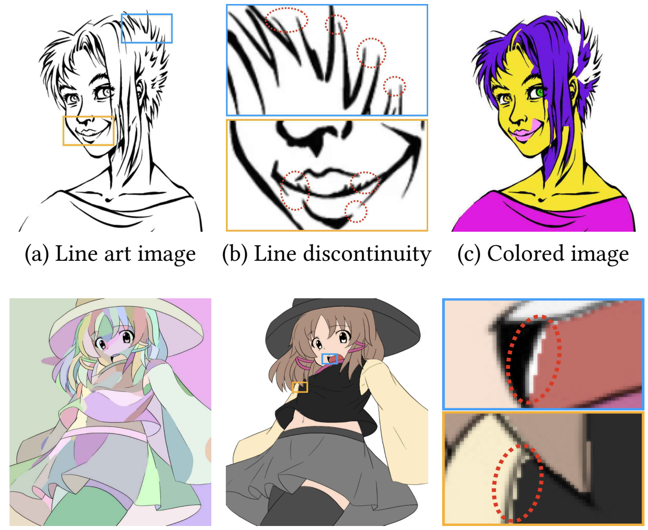
KEYWORDS

synthetic dataset generation, neural networks, line art flat-coloring

ACM Reference Format:
Han Kim, Chunggi Lee, Junsoo Lee, Dohyun Kim, Kwangjin Lee, Moohyun Oh, and Daesik Kim. 2023. FlatGAN: A Holistic Approach for Robust Flat-Coloring in High-Definition with Understanding Line Discontinuity. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3613788

1 INTRODUCTION

The popularity of digital comics and animations has soared, indicating the potential for the overall market to grow gradually [1–3]. However, drawing these art pieces is an extremely time-consuming and labor-intensive task, especially in terms of completing in-between cuts of comics or frames of animations within a short period. The process of colorization is particularly complex and involves multiple stages, such as line drawing, flat colorization, shading, lighting, and special effects [22]. As shown in Figure 1c, flat-coloring refers to the process of recognizing the segmented regions (Figure 1b) of a line-art image (Figure 1a) and then filling the regions with uniform tones and hues of colors before incorporating shading, lighting, and special effects. Our interviews have revealed that flat colorization is the most time-consuming stage and demands a tremendous amount of labor. Painters have to deal with not only tens or hundreds of ***line discontinuity regions***, but also the ***region-bleeding phenomenon*** in high resolution in order to enhance the precision of colorizing, which the current tools do not fully support. Line discontinuity regions (Fig.2a-b), where the line segments are not properly connected due to the intentions or styles of artists, require the drawing of virtual lines to block overflowing color to the other objects or manual brushing of mis-colored regions. On the other hand, the region-bleeding phenomenon (Fig.2f), which causes pixel misalignment errors around the line art, requires finding and replacing colors since the inference results of the model need to be recovered to a high-resolution image. These are serious issues that need to be addressed to improve the quality of images.



(a) Line art image (b) Line discontinuity (c) Colored image

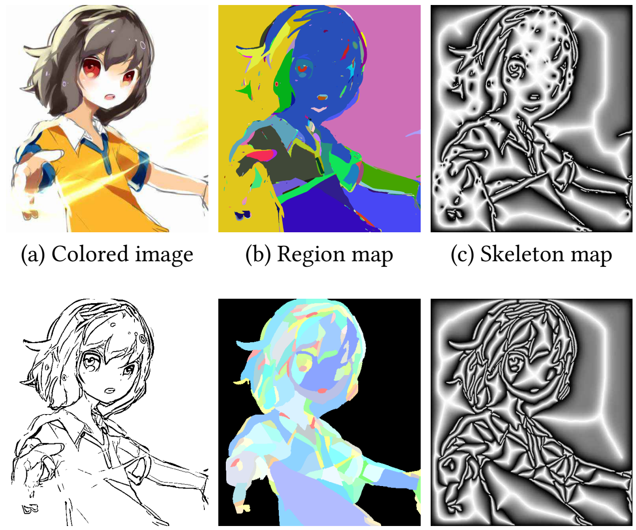
(d) Region map (e) Colored Image (f) Region-bleeding

Figure 2: The flat-coloring task faces two significant challenges. The line discontinuity problem is where line segments such as those highlighted in the blue and yellow boxes of Figure (b) are not correctly connected due to the intentions or style of the artists. The artists may fill colors like the example shown in Figure (c). The region bleeding problem (aliasing issue) that arises in recovering high resolution region map (d) after model inference. The blue and yellow boxes in Figure (e) highlight the region bleeding (f) caused by pixel misalignment. *©Top: Sara Teramo, used with artist permission.*

To address these issues, rule-based flat-filling methods [9, 14, 21, 26] and learning-based approaches [5, 22–25] have exhibited the ability to fill flat colors. Some rule-based techniques [9, 21, 26] have proposed approaches to achieve high-quality vectorized results that prevent leakages at unfinished curves and lines. Although these methods show promising performance, they still face challenges, such as incorrect or over-segmented flat color region boundaries for largely-opened or noisy sketches. On the other hand, there have been learning methods that overcome the limitations of rule-based methods by addressing the color-bleeding problem [22] and proposing an interactive flat-coloring framework [25]. However, based on our professional artists' interviews, they have shown reluctance to use auto-coloring and interaction-based methods due to their concerns about the method not maintaining their own painting styles and the additional time required for interaction. Additionally, they have expressed concern that they spend a lot of time struggling to replace overflowed colors that result from line discontinuities in the artwork.

In this paper, we propose a new holistic pipeline, called FlatGAN-DG, which not only generates ground-truth images with line discontinuity awareness but also augments sketch images from colored ones. Our pipeline improves the building of robust models by segmenting regions through an understanding of line discontinuity, greatly helping in obtaining high-quality flat region maps. This means that our models can properly segment regions that should be divided and do not overflow into the background in those regions. Consequently, artists no longer need to spend time drawing virtual lines to segment line discontinuity regions correctly. Furthermore, we present a real-time post-processing algorithm, called FlatGAN-PP, to solve the region bleeding problem and pixel misalignment errors around line art. Our algorithm extracts contours from near lines, which mainly cause the problem, and replaces the target pixel one-by-one by comparing unmarked neighbor regions. In addition, we share the parameters of a generator, predicting foreground and background as well as a trimap through multi-task learning to increase inference speed and output quality. All of our methods enable painters to improve the quality of their artwork by reducing the time spent on redundant and repetitive tasks, and allowing them to focus on more creative aspects, such as storytelling. Our extensive ablation studies demonstrate that our newly proposed components meaningfully contribute to the final records. Finally, we propose a new metric, called ***Regions score***, to evaluate the region connectivity for flat-coloring and conduct a user study to validate our methods in real-world environments. We also release an evaluation dataset called *RegionConnectivity* with annotations to evaluate the Region Score metric, contributing to this research field. In short, our contributions can be summarized in threefold:

- Our new holistic pipeline generates sketch and colored image pairs to address the issue of overflowing regions due to line discontinuity, achieving state-of-the-art results.
- We introduce a real-time post-processing algorithm to incorporate with our model to alleviate the region-bleeding problem (aliasing issues) by increasing the quality of paints and reducing working time.
- We propose a novel evaluation method, ***Regions score***, that validates the connectivity of regions based on line art. We release the RegionConnectivity dataset, which can utilize this metric, thereby contributing to the research field. We conduct a user study and show outperformed results in real-world environment to validate our methods.



(a) Colored image (b) Region map (c) Skeleton map

(d) Line art image (e) Region map (f) Skeleton map

Figure 3: DanbooRegion (DB) [24] and FlatFilling [25] take a colored image (a) and a line art image (d) as inputs to generate a skeleton map (c and f), respectively. These methods generate intermediate outputs such as a region map (b and e) and then convert the skeleton map.

2 RELATED WORK

2.1 Rule-based flat-filling

Since the advent of *bucket* tools, which utilizing the flood-filling algorithm, the major interest in the flat-coloring has been the line discontinuity regions, i.e., areas that are not completely closed. For example, so called Trappedball segmentation algorithm [26], which utilized on morphological operators, effectively divided areas for small gaps missed by the user. Fourey *et al.* [9] re-adopted the presegmentation-based approach. Some work [8, 13] propose methods which rasterize sketch image to simplify line and reduce the number of disconnected lines. However, they still limited to grasp high-level representation, which lead to incorrect or over-segmented flat color region boundaries for largely-opened or noisy sketches.

2.2 Learning-based flat-coloring

Recently, learning-based flat coloring methods [22, 25] have been proposed to overcome the limitations of rule-based methods. This work [5] addresses color-bleeding issues by introducing segmentation fusion mechanism. In addition, FlatFilling [23, 25] proposed an interactive flat-coloring framework. In these method, the final or intermediate output such as color, region and influence maps are generated based on the condition of user scribbles. However, for each input scribble, all previously entered scribbles are processed by the neural network. This confuses the user since the re-processed output does not preserve the previous flat color image. Also, every scribble inference is time consuming and computationally intensive. FlatMagic [22] has raised an issue of color-bleeding problem on opening sketches. FlatMagic [22] uses a paired dataset consisting of opened sketches and corresponding flat color maps. However, large dataset is required to handle the raised issue, but FlatMagic [22] fails to handle the issue due to lack of paired data. Based on [25], we refine the structure to fully reflect user interaction immediately. More importantly, we propose a data generation method based on GIMP [9] to overcome color-bleeding problem in opening sketch. On the other hand, several methods have been proposed to enhance the performance of editing by cleaning up lines from sketch images to raster vectored images, including [15–18].

3 FORMATIVE STUDY

In order to investigate the importance and difficulties of flat-coloring, we conducted a formative study with 12 professional artists (3 men and 9 women) from nine digital comics painters and three illustrators. The participants are contacted directly and have professional experience ranging from 1 year to 16 years (less than 5 years: 6, 5-10 years: 4, more than 10 years: 2). Our observations are consistent with similar results [22] that indicate they spend between 20% to 50% of their time filling colors during their entire creating process.

The most cumbersome task for artists during flat-coloring is to fill regions with line discontinuities since existing tools usually support segment closed lines. To fill these regions, they manually draw virtual lines to divide areas that should be different colors and repeatedly fill the colors until they are satisfied. This process constitutes a significant portion of their work. For instance, digital comics artists draw an average of 60 cuts for each episode every week, while illustrators fill colors for entire large images. Three digital comics artists mentioned that they want to focus on more critical and quality work such as improving their story and paint details, rather than filling colors. In addition, we ask the participants about AI-driven auto-coloring methods or interacting with AI models to understand their professional perspectives. Eleven artists believe that the auto-coloring method would be beneficial for amateur artists, but not for them. The professionals have their own painting style, and they need to maintain a consistent style. Thus, it is challenging to adopt the method that automatically fills in entire colors. Interacting with AI models [25] could enhance their workflow, but they would hesitate to use it if it cannot be processed in real-time since doing coloring manually is faster. Consequently, they require a method that 1) properly segments line discontinuity regions and 2) process in real-time as they do not want to wait for time-consuming methods or models for interactions.

4 METHOD

This section explains our work in three aspects: i) FlatGAN-DG (4.2), an algorithm to generate synthetic datasets, ii) FlatGAN (4.3), a main neural architecture, and iii) FlatGAN-PP (4.4), a new post-processing algorithm to alleviate the region-bleeding problem in the high-definition generation.

4.1 Preliminary

Flat-coloring aims to fill a segmented region map, which can have varying numbers of regions, based on the complexity of objects in a given line art image. The segmentation of regions plays an important role in the flat-coloring task since the well-segmented region maps make painters easy to fill colors. Taking inspiration from existing literature [22, 25], we build a model by utilizing an intermediate representation that can be inter-converted between the desired region maps and a skeleton map or an edge map, as shown in Figure 3. Predicting the edge map [22] or skeleton map [25] is reasonable option to convert the region map. However, we choose the skeleton map (Figure 3f), since the skeleton map contains richer information and is more robust during the early training stage compared to the edge map representation, which only contains sparse information with values of 0 and 1, and is unstable due to the lack of gradient information. We explain two methods that convert a color image ($I_c \in \mathcal{R}^{3xhxw}$, Figure 3a) or a line art image ($I_s \in \mathcal{R}^{1xhxw}$, Figure 3d) into a region map (Figure 3b and e) or a skeleton map (Figure 3b and e). First, DanbooRegion (DB) [24] takes the color image (Figure 3a) and estimates the skeleton map (Figure 3c). However, the results of the DR model are not only unstable, such as the combined area between the head and face in Figure 3b, but also insufficient to be used as ground truth. On the other hand, rule-based method takes the line art (Figure 3d) and generates region map (Figure 3e), showing robust results in line discontinuity where the line segments are not connected properly and overflow to the background or other objects, as shown in Figure 3e. Therefore, we have adopted the GIMP [9] method to enhance our dataset.

4.2 FlatGAN-DG: Data Generation Pipeline

To obtain a line discontinuity-aware and sketch augmentation dataset, we propose a new holistic pipeline that enables the model to be more robust to painters' styles and noises. First, we need dataset that contains a diverse range of styles, since real-world line arts vary greatly in content, style, and complexity. Next, we require ground truth label for line discontinuity regions, where the line segments are not connected properly due to the intention or styles of artists, to build robust models, because manually drawing virtual lines to fill the colors repeatedly in line discontinuity regions is time-consuming, as we mentioned in Section 3. The pipeline consists of two main components, generating target for ground-truth for line discontinuity regions and input images for augmenting line art images, represented by A and B in Figure 4, respectively. Given a color image (Figure 4A-a), our pipeline utilizes synthetic sketch extracted by SketchKeras [12] and then simplify the line art image (Figure 4A-b) using SketchSimplifier [19], which reduces the noise of the lines. The line art image is then converted to a segmented regions map (Figure 4A-c) using the GIMP [9] method. We use the color images (Figure 4A-a), that we collected white background images, and the segmented regions (Figure 4A-c) to generate a foreground segmented regions map (Figure 4A-d) using the watershed method. The skeleton maps (Figure 4A-e and -f) are then generated from the foreground regions map using this method [24, 25]. Finally, we obtain the tri-map (Figure 4A-g) by utilizing both foreground and background information. Therefore, we can generate ground-truth images for line discontinuity regions to prevent from overflowing coloring.

We also describe the generating pipeline for input images to build a model that is robust to different painting styles and noise. The second pipeline extracts a sketch image (Figure 4B-a) from the color image (Figure 4A-a) and then finds contours (Figure 4B-b) to remove some parts of the lines, such as the red lines in Figure 4B-c, in order to augment the sketch images (Figure 4B-d). Lastly, we obtain the final sketch image (Figure 4B-e) by adding random noise. This enhances the effect of generating color and skeleton image pairs as it creates more diverse input images. In other words, we can consider the different style of painters' sketches.

4.3 FlatGAN: Network Architecture

We draw inspiration from AlacGAN [6], a widely used model in the auto-colorization task, which consists of a common encoder and decoder. The AlacGAN reduces the domain gap between real and synthetic sketches through the local feature network. We adopt this architecture since it generates a skeleton map from a synthetic sketch. However, unlike AlacGAN, which automatically fills all areas in the line art image, FlatGAN aims to predict an exact region map that can be converted from a skeleton map. The automatic flat-coloring not only causes discomfort but also results in additional interaction costs when the output does not align with the user's intention and needs to be revised, as mentioned in section 3.

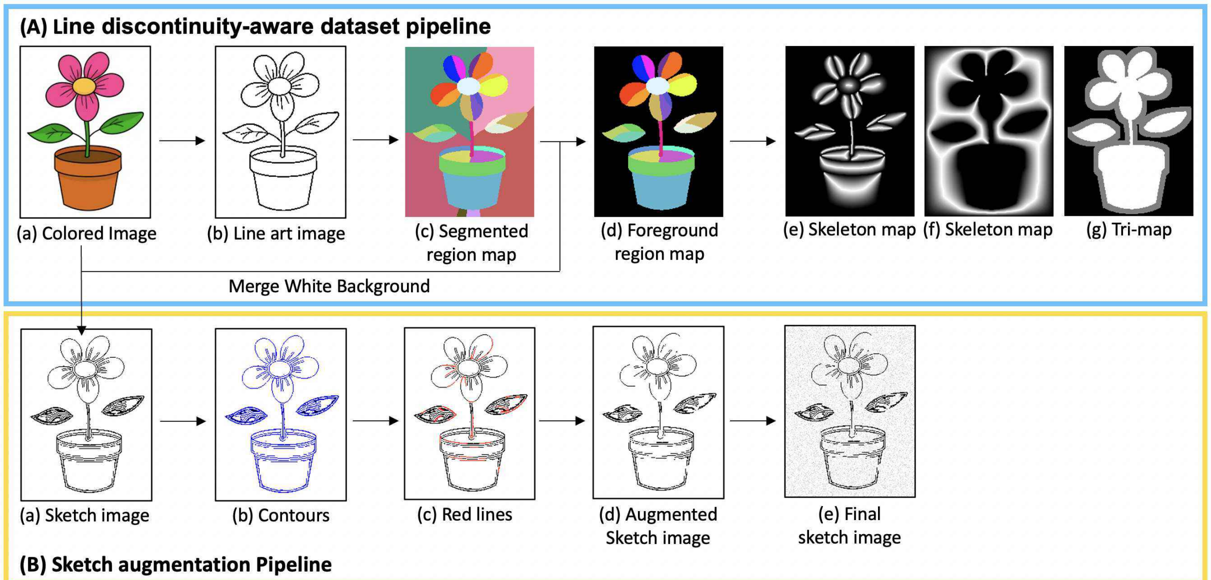


Figure 4: Overview of our line discontinuity aware dataset generation (A) and sketch augmentation pipeline (B). The (A) pipeline takes a colored image and generates two skeleton maps and a tri-map to generate labels. The (B) pipeline shows the sketch augmentation process aimed at building a robust model that is capable of handling different painting styles and noise levels.

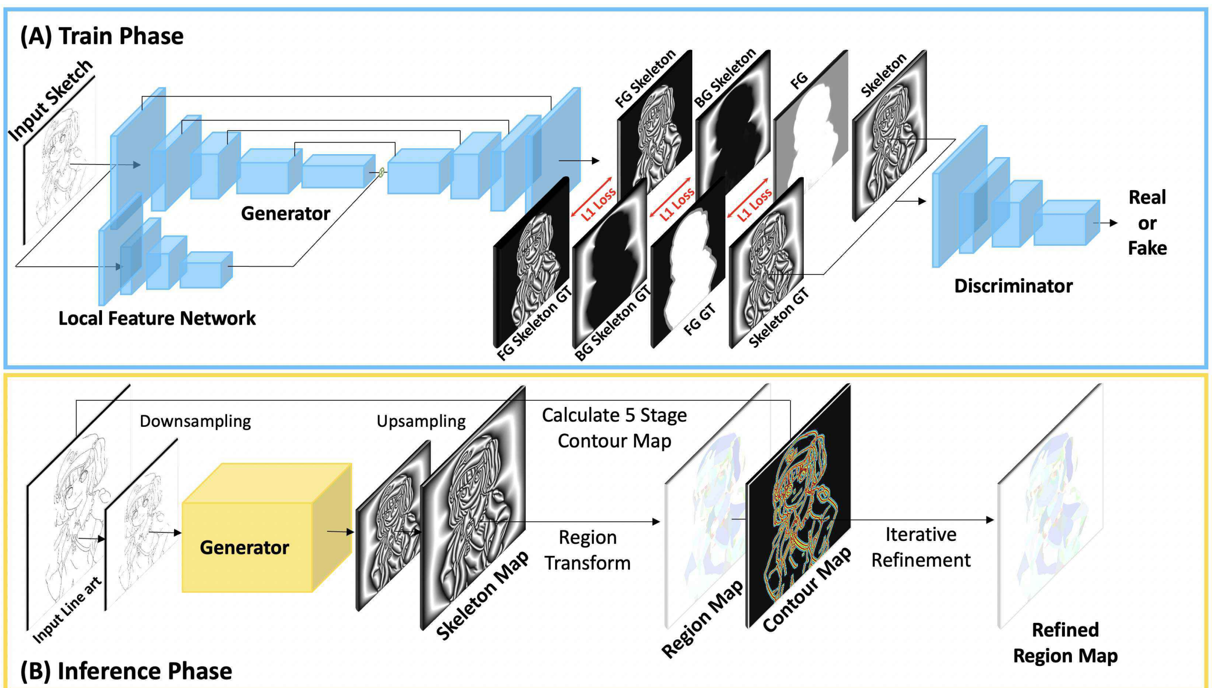


Figure 5: Overview of our network architecture. Our training phase (A) shares parameters of a generator and estimates 3 channel outputs (Skeleton FG, Skeleton BG, Trimap FG) to perform multi-task learning. Our inference phase (B) generates sketch map and transform region map. Lastly, our algorithm iteratively refine the region map to alleviate aliasing issue.

Furthermore, AlacGAN is highly time-consuming since it must process images per each individual user interaction. Therefore, we share the generator's parameters to enhance multi-task learning and predict foreground and background skeleton maps and a trimap. This improves the quality of results and increases the inference speed.

Figure 5(A) presents the overall architecture of FlatGAN, which consists of a feature encoder E, a skeleton generator G, and a discriminator D. We have added an additional network at the last layer of generator G to infer the foreground skeleton map ($\hat{S_f}$), background skeleton map ($\hat{S_b}$), and Trimap ($\hat{T}$). In summary, our model aims to predict these three outputs given a line art image Is:

$$(\hat{S_f}, \hat{S_b}, \hat{T}) = G(E(I_s)). \tag{1}$$

Our loss function is formulated similar to [22, 25] as a combination of a reconstruction loss, a perceptual loss, and an adversarial loss:

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{prec}\mathcal{L}_{perc} + \lambda_{adv}\mathcal{L}_{adv} \tag{2}$$

, where the λs are coefficients of the above-mentioned losses. The pixel-wise reconstruction loss aims to minimize the gap between fore- (S_f) and background skeleton map (S_b), and tri-map (T).

$$\mathcal{L}_{rec} = \left\|\hat{S_f} - S_f\right\|_1 + \left\|\hat{S_b} - S_b\right\|_1 + \left\|\hat{T} - T\right\|_1 \tag{3}$$

We utilize the perceptual loss [10], which involves a L1-loss based on the generated and target skeleton map, to enhance the color and structural features, as it is crucial in comparison to the edge map. This approach aims to improve the overall quality of the skeleton map in our flat-coloring method.

$$\mathcal{L}_{perc} = \sum_j \left\|\phi_j(\hat{S}) - \phi_j(S)\right\|_1 \tag{4}$$

, where $S = S_f * T + S_b * (1 - T)$ and ϕ_j indicates the j-th intermediate feature maps of pre-trained convolutional based model such as VGG19 [20] networks. We takes an adversarial loss as our additional loss, founding that the loss improves the flat-coloring performance. It is defined as following:

$$\mathcal{L}_{adv} = \mathbb{E}_{S\sim\mathcal{D}}\left[\log D(S)\right] + \mathbb{E}_{\hat{S}\sim G}\left[\log\left(1 - D(\hat{S})\right)\right] \tag{5}$$

4.4 FlatGAN-PP: Alleviating region-bleeding

In the professional art world, the typical resolutions of input paintings range from 2k to 3k. To address a high-resolution input, a common approach is to downsample the original image to 512px or 1024px, infer the line art image, and then upsample it back to its original size due to GPU memory or speed limitations. However, this method often results in the region-bleeding issues, which causes minor pixel misalignment errors around the line art, as shown in Figure 6b. Our feedback from professional artists indicates that region bleeding not only reduces the overall quality of the paintings but also requires additional manual work to identify and fill in these areas. In particular, this phenomenon is more evident when the surrounding colors differ significantly, such as in the case of eyes (white) and hair (black, yellow, etc.). Also, FlatMagic [22] has proposed a method to address the region-bleeding problem, but its speed is highly unlikely to be suitable for real-world environments. We evaluate these speed comparison in our experiment section.

We propose a real-time post-processing algorithm, called FlatGAN-PP, that consists of two parts to address the issue of region bleeding. Since the bleeding artifact mainly occurs near border lines (Figure 2f), we first detect the lines of given sketch (Figure 6d) and extract n number of contours near the lines (Figure 6e). In our experiment, we use five contours that represent red (the innermost contour), yellow, green, blue, and sky blue (the outermost contour) as shown in Figure 6e. Next, each color in the counter from sky blue

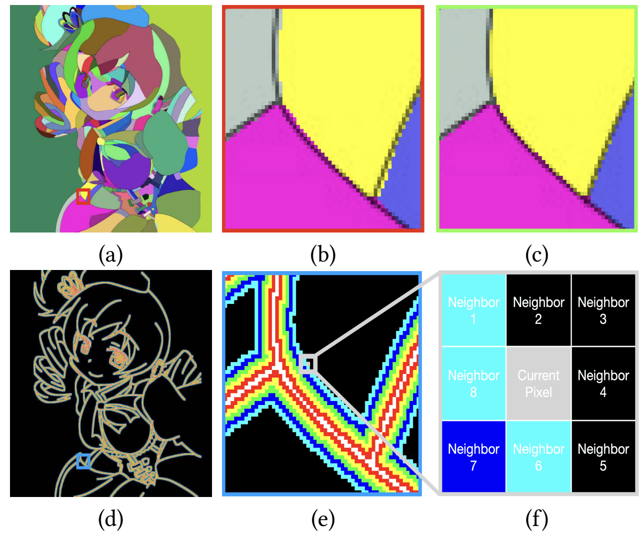


Figure 6: Our algorithm alleviates the *aliasing problem.* This algorithm takes the region map (a) and automatically finds aliasing areas (b) to refine them, as shown in the results (c). It extracts the contour map (d) from the region map (a) near the line art. From the outermost to innermost contour, the algorithm replaces the misaligned pixel based on the unmarked majority of neighbor pixels (f).

to red replaces one-by-one based on unmarked regions, i.e., near neighbor areas that are not contour lines. For example, the current pixel in Figure 6f is selected to change color based on dominant neighbor pixels as a black color, which is not included in contours. Our method iteratively repeats this process to the innermost contour lines. We believe that our approach is effective and practical since the bleeding areas locally occur around the lines—our algorithm is designed to focus on them. We also provide detailed information, such as a pseudo-code and parallel processing code, in the supplementary material.

5 EXPERIMENTS

5.1 Datasets

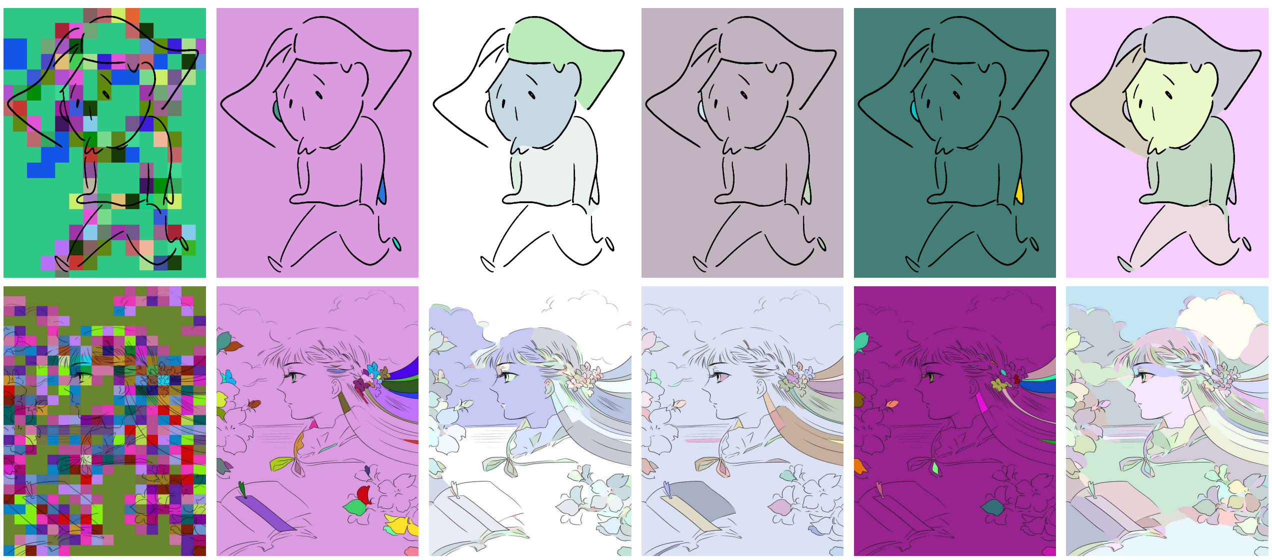
In our training, we collect one million illustrations from Danbooru [4]. Subsequently, we select about 128,000 illustrations with white backgrounds, named Safebooru-WB, to utilize the FlatGAN-DGpipeline, as mentioned in Section 4.2. To compare performances between baselines and ours in the quantitative manner, we build a new evaluation dataset, named RegionConnectivity, which has a total of 300 line arts collected from Safebooru [11], Webtoons [7], and the Internet. For the RegionConnectivity, we manually label annotations which is used to calculate merge-hard-score, split-score, and merge-easy-score—described in Section. 5.2.

5.2 Evaluation metric (Region Score)

Before presenting our experiments, we introduce a new evaluation metric, ***Regions score***, and dataset for region segmentation in flat-coloring. Most colorization papers use FID [8] to evaluate the quality of the reconstructed generated image. However, this

	t	Method		Safebooru-WB [11]				Webtoons [7]				Internet			
		DG	backbone	$s\uparrow$	$m_e\uparrow$	$m_h\uparrow$	$RS\uparrow$	$s\uparrow$	$m_e\uparrow$	$m_h\uparrow$	$RS\uparrow$	$s\uparrow$	$m_e\uparrow$	$m_h\uparrow$	$RS\uparrow$
(a)	R	super-pixel	N/A	84.13	13.81	8.15	14.49	74.85	15.50	12.20	18.76	82.17	18.61	6.32	13.38
(b)		trapped-ball	N/A	4.13	99.61	92.74	11.40	1.71	99.70	99.40	4.95	6.97	100.00	98.26	18.33
(c)		GIMP	N/A	70.34	76.65	18.42	36.78	31.52	92.28	68.01	52.38	56.92	69.41	20.34	36.97
(d)	L	FlatFilling	FlatFilling	16.55	96.30	74.62	35.62	4.57	99.70	95.23	12.53	31.00	98.02	58.78	50.44
(e)		FlatMagic	FlatMagic	1.37	99.80	96.07	3.99	0.10	99.70	99.70	0.29	0.77	100.00	99.56	2.27
(f)		FlatFilling	FlatGAN	18.62	96.19	71.29	38.39	9.23	99.57	94.81	23.26	34.61	98.43	58.78	53.50
(g)		GIMP	FlatGAN	78.62	60.89	12.38	27.29	61.41	71.71	29.39	46.68	76.15	56.47	13.85	29.15
(h)		FlatGAN-DG	FlatFilling	66.89	80.93	25.98	45.59	44.57	92.17	54.76	58.19	67.44	77.47	36.60	54.48
(i)		FlatGAN-DG	FlatMagic	66.20	73.15	15.40	32.01	61.71	82.27	34.82	52.55	69.76	67.58	11.68	26.14
(j)		FlatGAN-DG	FlatGAN	80.00	77.82	41.08	**60.37**	71.19	87.14	43.22	**61.65**	86.82	68.77	38.96	**57.99**

Table 1: Comparisons using the *Regions score* over the RegionConnectivity dataset. (a-c), *i.e.*, R, are grouped from the rule-based methods and (d-j),*i.e.* L, are the results of learning-based methods. We study the performance of baselines and ours in terms of two criteria: data generation (DG) strategy and backbone network architecture. s, m_e, m_h, RS indicate split, merge-easy, merge-hard, and *Regions score* respectively.



(a) Super-pixel (b) Trapped-ball [26] (c) GIMP [9] (d) FlatFilling [25] (e) FlatMagic [22] (f) FlatGAN (Ours)

Figure 7: Comparison with existing approaches: from left to right, rule-based approaches including Trapped-ball [26] and GIMP [9], learning-based approaches including FlatFilling [25] and FlatMagic [22], and our proposed approach. The samples in the first row are from Webtoon [7], while those in the second row are from the internet. *©Top: Jenny Seo, used with artist permission.*

distance-based metric is not feasible to evaluate our purpose that how well areas are segmented for artists in the real-world environment. Therefore, we propose a new evaluation metric and release a new dataset for this purpose, which will contribute to future work. To evaluate whether the regions are well-segmented from the user's perspective, various factors need to be considered: 1) whether the areas that need to be divided are properly segmented, and 2) whether the areas that should not be divided are not segmented. According to our analysis, most methods are mainly in trouble in line discontinuity, where the lines are not connected properly like Fig. 2(b). Thus, we annotate the dataset based on the following criteria: 1) ***split-score***: whether regions are segmented in line discontinuity between different objects (e.g., between eyes and hair). 2) ***merge-hard-score***: whether the regions are not segmented in line discontinuity and the same object (e.g., hair). 3) ***merge-easy-score***: whether the regions are not segmented in the same object (e.g., clothes). Based on these criteria, we propose ***Regions score***, the harmonic mean of ***split-score***, ***merge-hard-score***, and ***merge-easy-score*** (like F1 metric) to evaluate the proper splitting of regions and their semantics.

5.3 Comparison for FlatGAN over baselines

To validate our proposed FlatGAN method, we compare baselines from both the rule-based algorithm [9, 26] and learning-based methods [22, 25], which are widely used in flat-coloring tasks. We train all the baselines and our proposed method on Safebooru-WB [11] dataset, and then evaluated their performance on the RegionConnectivity dataset, as mentioned in Section 5.1. Overall, we find that our FlatGAN method (Table 1j) outperforms all baseline methods with respect to ***Regions score*** (*RS*). The rule-based baselines (Table 1 a-c) exhibit different characteristics. Super-pixel segmentation (Table 1a) has high split-score$s$, but low merge-hard-scorem_h since it divides a region into small segments, as shown in Figure 7a. On the other hand, the trapped-ball algorithm (Table 1b) shows overflowing drawbacks, as shown in Figure 7b, in accordance with the low split-score. GIMP (Table 1c) exhibits relatively better performance than the above two, but the merge-hard score (m_h) is lower than ours.

Next, to confirm the effectiveness of data pipeline and model, we compare several architecture of backbone networks such as FlatFilling [25], FlatMagic [22], and FlatGAN with varying data generation pipelines. We train our FlatGAN with other data pipelines (f and g), showing overall improvement of *RS* and the advantages of multi-task learning. In addition, we investigate the effectiveness of FlatGAN-DG by training other baselines (h and i) which increase the performance of ***Regions score*** compared to (d and e). Consequently, we observe that FlatGAN-DG enhances all models to be the robustness about line discontinuity by augmenting the dataset. We observe that our decoder, which generates three results and merges them, enhances overall performance compared to other learning-based models. Furthermore, we demonstrate the robustness of FlatGAN in terms of line discontinuity problems, as shown in the green circle of Figure. 3c in the supplementary material. Inferring foreground and background skeleton maps and trimaps not only enhances multi-task learning but also improves boundary awareness, allowing the generators to share their entire parameters and learn the quality of segmenting region maps.

These results are further supported by Figure 7. The first row of Figure 7 depicts a simple line art image with several line discontinuities. The Trapped-ball (b), FlatFilling (d), and FlatMagic (e) methods segment the entire character as few regions, resulting in a burdensome task to refill these areas. In contrast, our methods segment proper regions to fill each component, such as hairs, face, pants, and shirt. The second row of Figure 7 shows a more complicated example. Other methods segment small and complex areas into several regions, resulting in artists needing to manually re-segment them. However, FlatGAN segments more appropriate regions compared to the baselines, enabling artists easily to fill color object-by-object.

5.4 Comparison for FlatGAN-PP

To validate our FlatGAN-PP, we conduct a experiment comparing it to FlatMagic [22], a method designed to tackle the region-bleeding problem. Figure 8 presents a comparison of different image resolutions and devices (CPU and GPU). As image resolutions increase, FlatMagic [22] shows a steep increase in inference speed, while our FlatGAN-PP shows a significantly smaller increase even with CPU, and FlatGAN-PP with GPU can infer high resolutions (over 3k) in less than 1s, as shown in Figure 8. Furthermore, the region-bleeding phenomenon usually occurs more noticeably in larger resolutions (over 3k), typically found in paintings by artists in real-world environment. Thus, we firmly believe that our FlatGAN-PP method not only improves the workflow for artists but also enhances the overall

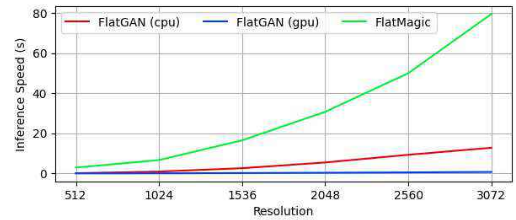


Figure 8: Inference speed comparison by resolution

quality of their paintings, by alleviating the burden of manually revising colors and releasing them to focus on more creative tasks.

5.5 User Study

To evaluate the effectiveness of our overall methods, we conduct a user study in which participants who intend to create digital comics are instructed to color randomly selected 5 line art images from the Safebooru-WB, Webtoons, and Internet datasets. To assess only the time taken for coloring, we provide ground truth colored images to the participants, as they are not professional artists, so do not need to spend time considering which colors to select. The study employs a between-subject design, with each of the 5 participants assigned to color 5 line art images (a total of 25 images per method) using GIMP (rule-based method), FlatFilling (learning-based method), and FlatGAN (our method). We record the time taken to fill colors and conduct a statistical test to demonstrate that our approach improves the flat-filling task regarding time efficiency. The Shapiro-Wilk test reveal that our time data is not a normal distribution (p-value=0.00002). Thus, we conduct Kruskal-wallis test and observe the a significant difference among three methods (p-value = 0.0027). Results from Dunn's test indicate that our method (mean: 214s) has a statistically significant difference compared to FlatFilling (mean: 373s) and GIMP (mean: 409s). In summary, we believe that all methods contribute to reducing the workload of flat-filling compared to other methods.

6 CONCLUSION

Through this work, we investigate a useful method and its components (FlatGAN, FlatGAN-DG, and FlatGAN-PP) to support artists who suffer from flat-coloring—the most burden task during the entire digital comics, animations, and illustrations. To support these artists, we propose a holistic data generation pipeline (FlatGAN-PP) and a real-time post-processing algorithm to reduce inference time. Our methods not only prevent from overflowing of line discontinuity regions to background but also automatically find to replace aliased areas due to the resolution size of model inference. To evaluate the segmented regions well, we collect and release datasets including the annotation of split-score, merge-hard-score, and merge-easy-score. We also introduce a new evaluation metric (***Regions score***) for these datasets. Lastly, our experimental results show our methods outperform over other rule- and learning-based methods, showing the efficacy through a user study.

REFERENCES

[1] 2020. *Comic Book Market Size, Share Industry Analysis, By Product Type, By Distribution Channel, and Regional Forecast 2021-2028. Accessed: 2023-03-27.* https://www.fortunebusinessinsights.com/comic-book-market-103903

[2] 2022. *3D Animation Market Size, Share Trends Analysis Report By Technique, By Components, By Deployment, By End-use, And Segment Forecasts, 2022 - 2030. Accessed: 2023-03-27.* https://www.grandviewresearch.com/industry-analysis/3d-animation-market

[3] Korea Creative Content Agency. 2020. *CCartoon Industry White Paper. Ministry of Culture, Sports, and Tourism.*

[4] Anonymous, Danbooru community, and Gwern Branwen. 2021. Danbooru2020: A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset. https://www.gwern.net/Danbooru2020. https://www.gwern.net/Danbooru2020 Accessed: DATE.

[5] Ruizhi Cao, Haoran Mo, and Chengying Gao. 2021. Line art colorization based on explicit region segmentation. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 1–10.

[6] Yuanzheng Ci, Xinzhu Ma, Zhihui Wang, Haojie Li, and Zhongxuan Luo. 2018. User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks. In *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*. ACM, 1536–1544.

[7] Naver Webtoons Corp. 2023. . https://comic.naver.com/webtoon/weekday

[8] Jean-Dominique Favreau, Florent Lafarge, and Adrien Bousseau. 2016. Fidelity vs. simplicity: a global approach to line drawing vectorization. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–10.

[9] Sébastien Fourey, David Tschumperlé, and David Revoy. 2018. A fast and efficient semi-guided algorithm for flat coloring line-arts. In *International Symposium on Vision, Modeling and Visualization*.

[10] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. Springer, 694–711.

[11] Alexander Lamson. 2021. *Safebooru - Anime Image Metadata.* https://www.kaggle.com/datasets/alamson/safebooru

[12] lllyasviel. 2018. . https://github.com/lllyasviel/sketchKeras

[13] Amal Dev Parakkat, Marie-Paule R Cani, and Karan Singh. 2021. Color by numbers: Interactive structuring and vectorization of sketch imagery. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11.

[14] Amal Dev Parakkat, Pooran Memari, and Marie-Paule Cani. 2022. Delaunay Painting: Perceptual Image Colouring from Raster Contours with Gaps. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 166–181.

[15] Kazuma Sasaki, Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Joint gap detection and inpainting of line drawings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5725–5733.

[16] Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. 2018. Mastering sketching: adversarial augmentation for structured prediction. *ACM Transactions on Graphics (TOG)* 37, 1 (2018), 1–13.

[17] Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. 2018. Real-time data-driven interactive rough sketch inking. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

[18] Edgar Simo-Serra, Satoshi Iizuka, Kazuma Sasaki, and Hiroshi Ishikawa. 2016. Learning to simplify: fully convolutional networks for rough sketch cleanup. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

[19] Edgar Simo-Serra, Satoshi Iizuka, Kazuma Sasaki, and Hiroshi Ishikawa. 2016. Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup. *ACM Transactions on Graphics (SIGGRAPH)* 35, 4 (2016).

[20] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

[21] Daniel Sỳkora, John Dingliana, and Steven Collins. 2009. Lazybrush: Flexible painting tool for hand-drawn cartoons. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 599–608.

[22] Chuan Yan, John Joon Young Chung, Yoon Kiheon, Yotam Gingold, Eytan Adar, and Sungsoo Ray Hong. 2022. FlatMagic: Improving Flat Colorization through AI-driven Design for Digital Comic Professionals. In *CHI Conference on Human Factors in Computing Systems*. 1–17.

[23] Mingcheng Yuan and Edgar Simo-Serra. 2021. Line art colorization with concatenated spatial attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3946–3950.

[24] Lvmin Zhang, Yi JI, and Chunping Liu. 2020. DanbooRegion: An Illustration Region Dataset. In *European Conference on Computer Vision (ECCV)*.

[25] Lvmin Zhang, Chengze Li, Edgar Simo-Serra, Yi Ji, Tien-Tsin Wong, and Chunping Liu. 2021. User-Guided Line Art Flat Filling with Split Filling Mechanism. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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