User-Guided Line Art Flat Filling with Split Filling Mechanism

LVMIN ZHANG, CHENGZE LI, EDGAR SIMO-SERRA, YI JI, TIEN-TSIN WONG, CHUNPING LIU

User-Guided Line Art Flat Filling with Split Filling Mechanism

Lvmin Zhang
Soochow University / Style2Paints Research
China
lvminzhang@acm.org

Chengze Li
The Chinese University of Hong Kong / Style2Paints Research
China
ljsabc@gmail.com

Edgar Simo-Serra
Waseda University / JST PRESTO
Japan
ess@waseda.jp

Yi Ji
Soochow University
China
jiyi@suda.edu.cn

Tien-Tsin Wong
The Chinese University of Hong Kong
China
ttwong@cse.cuhk.edu.hk

Chunping Liu
Soochow University
China
cpliu@suda.edu.cn

Abstract

Flat filling is a critical step in digital artistic content creation with the objective of filling line arts with flat colors. We present a deep learning framework for user-guided line art flat filling that can compute the "influence areas" of the user color scribbles, i.e., the areas where the user scribbles should propagate and influence. This framework explicitly controls such scribble influence areas for artists

1. Introduction

Flat filling is a process to color line arts with fairly flat colors according to the artists' specifications. This technique originates from the on-paper cartoon animation of the 1930s and remains critical in the digital painting era, as these flat-colored results exhibit great versatility in a wide variety of art workflows. Not only these flat colors can be directly blended with the line arts to create cartoon-like illustrations, they can also be used as independent foundations without

Introduction

- Flat filling is a process to color line arts with flat colors according to the artists' specifications.
- In computer vision and graphics, two broad paradigms of user-guided line art colorization exist: traditional user scribble propagation and learning-based interactive colorization.
- We propose the Split Filling Mechanism (SFM) to achieve controlling the influence areas of use scribbles accurately to meet professional use cases, while at the same time incorporating data-driven color generation capability to inspire and facilitate content creation

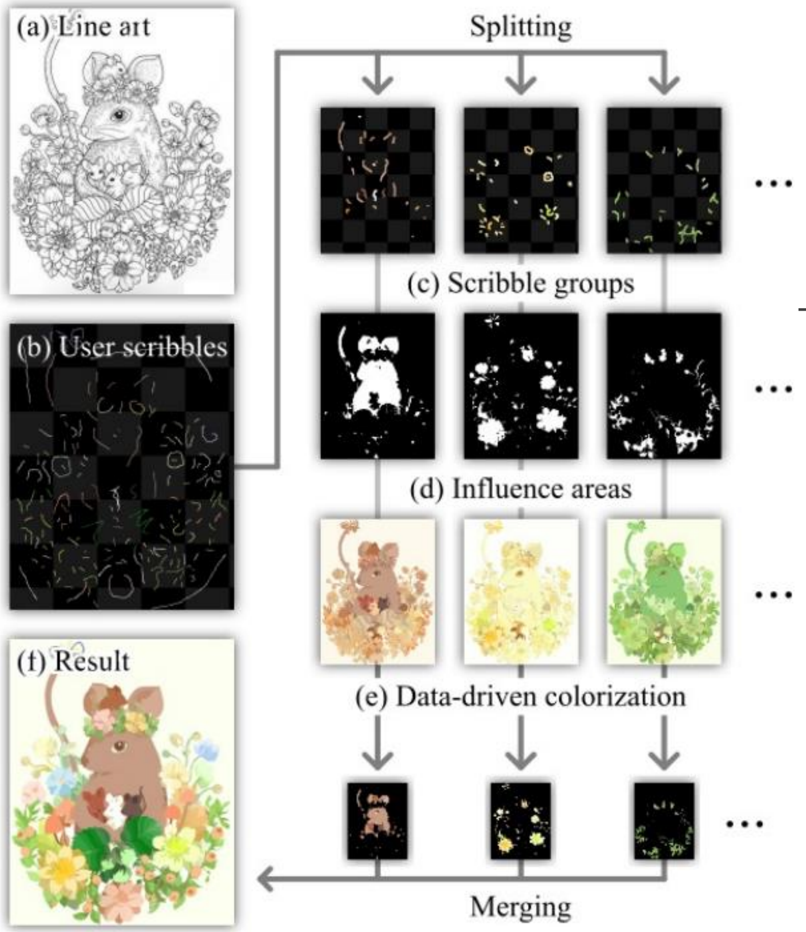


Figure 1. **Split Filling Mechanism (SFM).** Given (a) a line art and (b) some user scribbles, the SFM separate the scribbles into (c) several groups and estimate (d) the influence area of each group to accurately control the color segmentation. Afterwards, the SFM performs (e) data-driven colorization in each group to generate visually satisfying color combinations to assist artists. The outputs are merged together to achieve (f) the result. *Flower Mouse.*

Introduction

- Our contributions are as follows:
  - We analyze the merits and goals of traditional propagation-based and interactive learning-based colorization methods, and then motivate the problem to get the best of both worlds to simultaneously control the influence areas of user scribbles and generate plausible color combinations.
  - We propose the Split Filling Mechanism (SFM) framework consisting of the split scribble processing and data-driven colorization steps.
  - We show that the proposed approach can handle a diversity of complex line drawings, enabling both artists and amateurs to easily achieve high-quality flat filling results

Related Work

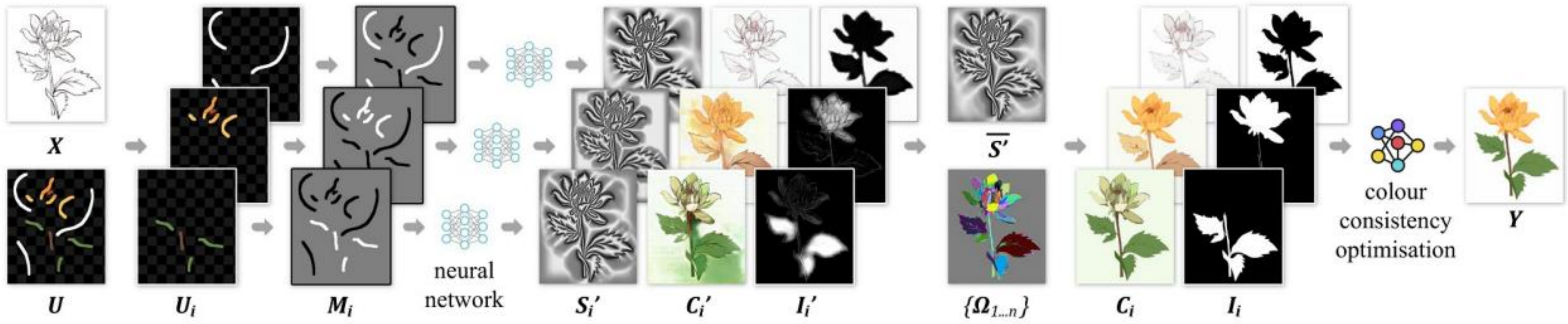
Propagation-based line art colorization.

- Shaw et al. proposes to speed up the flat colorization process using an efficient tree search algorithm.
- Optimization-based approaches have also been used to propagate the color from user scribbles in black-and white photograph and are further extended to the case of filling colors in patterned manga screen-tones, and many more.
- The popular open-source software GIMP uses an auto-closing algorithm to compute user intended flat colorization regions in line drawings.

Related Work

Learning-based line art colorization

- The prosperity of large-scale learning techniques has popularized the initial researches of monochrome photography colorization
- Multiple solutions for direct sketch colorization have been proposed and widely used, including the commercial product PaintsChainer and a two-stage solution.
- The reference-based coloring can also work well online video sequences, as proposed by using an attention-based framework. Image stylization translate cartoon images or artistic drawings from/to photographs or human portraits.
- Those approaches tend to produce results with pixel-level texture learned from their training data, whereas in many standard line-drawing-based artistic creation workflows (e.g., cel-coloring, cel-shading, etc.), artists have the requirement to fill color in regions, and the colors in each region must be flat and consistent.



(a) Line drawing $\boldsymbol{X}$; scribble map $\boldsymbol{U}$; split scribble map $\boldsymbol{U}_i$; split scribble mask $\boldsymbol{M}_i$

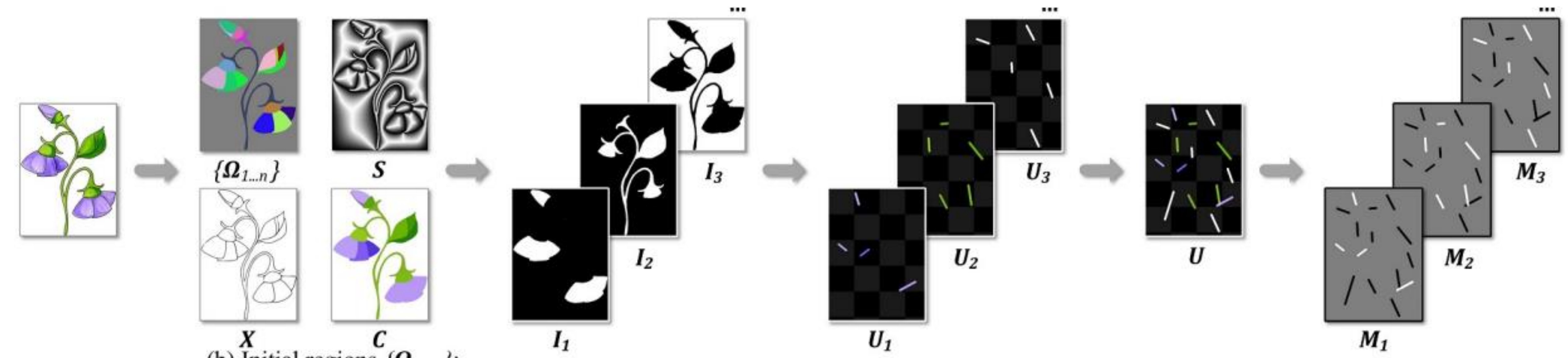
(b) Predicted region skeleton map $\boldsymbol{S}_i'$, colour map $\boldsymbol{C}_i'$ and influence map $\boldsymbol{I}_i'$

(c) Average skeleton map $\overline{\boldsymbol{S}'}$; initial regions $\{\boldsymbol{\Omega}_{1 \ldots n}\}$; flattened colour map $\boldsymbol{C}_i$ and influence map $\boldsymbol{I}_i$

(d) Output merged result $\boldsymbol{Y}$

Figure 2. **Inference pipeline.** We visualize the components involved in the inference pipeline of our framework with splitting and merging.

Method



(a) Illustration (b) Initial regions $\{\Omega_{1...n}\}$; region skeleton map S; line drawing X; flat colour map C. (c) Influence map I_i (d) Split scribble map U_i (e) Merged scribble map U (f) Split scribble mask M_i

Figure 3. **Training data synthesis pipeline.** We illustrate the involved components within our dataset synthesis for splitting and merging.

Method

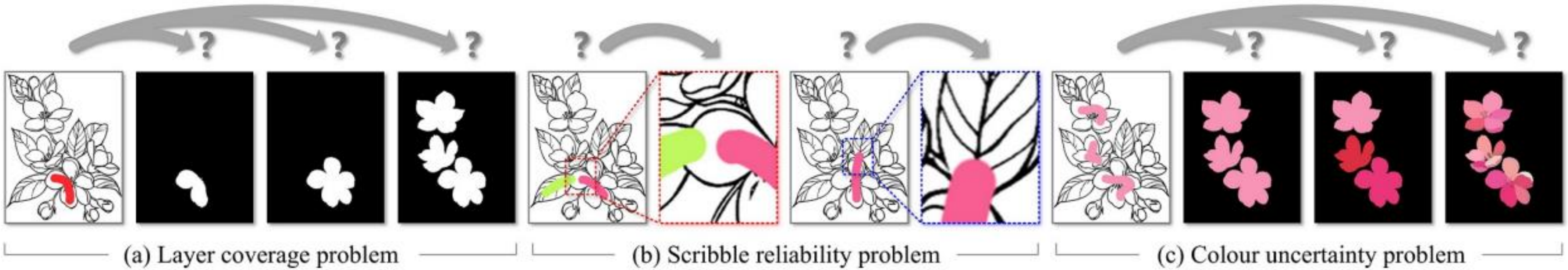


Figure 4. **Scribble problems.** We illustrate common problems associated with the user scribbles synthesizing steps that need to be resolved.

Method

Method

- Although the SFM framework naturally mitigates these problems, care must be taken when synthesizing the training scribbles. In order to deal with the layer coverage issue, we randomly manipulate the region coverage of each scribble, so that the model can learn to estimate appropriate regions that are affected by each scribble. To be specific, instead of sampling from the same region Ω_j for p_1 and p_2, we allow p_2 to be taken from a region that is reachable from p_1 within the region set Φ_i .
- We implement this by performing a random walk from Ω_j to find a random k-step-neighbor region Ω_k to sample p_2 from. We do a $k = 3$ step random walk to not obtain regions that are too far away. Next, to tackle the scribble reliability problem, we not only sample scribble endpoint positions within fixed region area, but also from a surrounding area with r pixel radius around the region (we use $r = 15$ by default) to simulate the coarse scribbles outside of the sampled regions. Finally, we perform the aforementioned color consistency optimization in each color map C_i to simulate color uncertainty, mimicking real scribbles that are coarsely drawn by artists.

Method

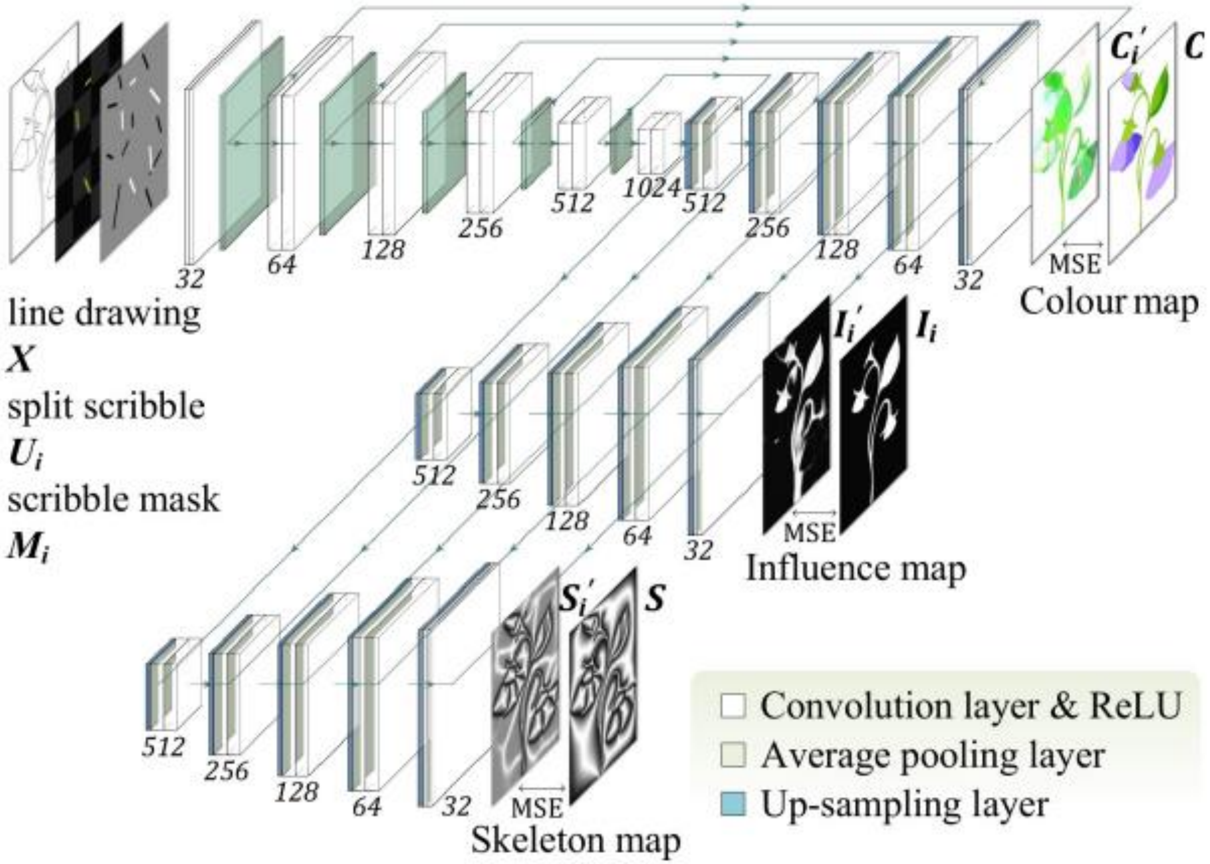


Figure 5. **Neural network architecture.** All convolutional layers use 3×3px kernels. We do not use any normalization layers. The Mean Squared Error is indicated as "MSE".

Evaluation

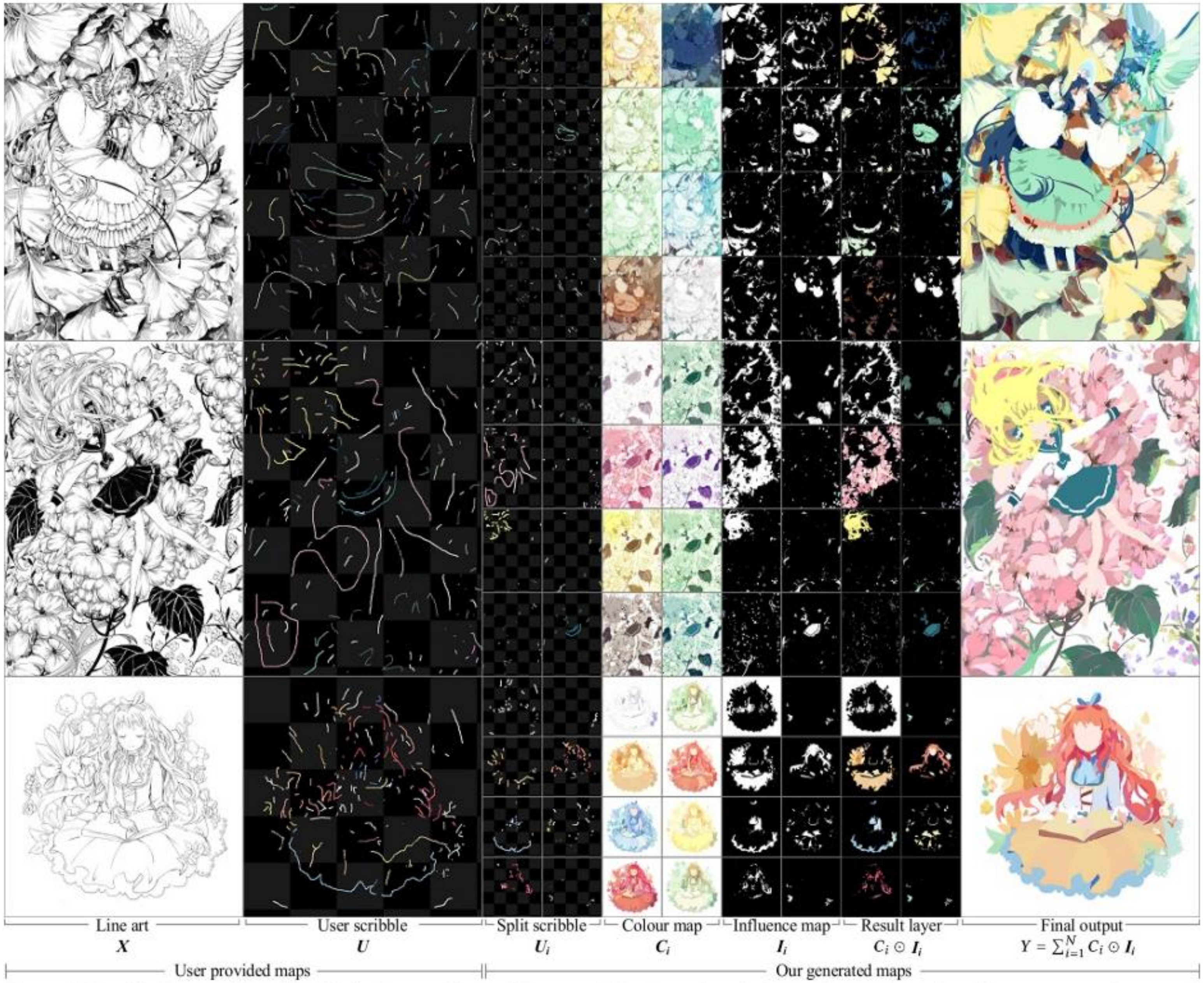
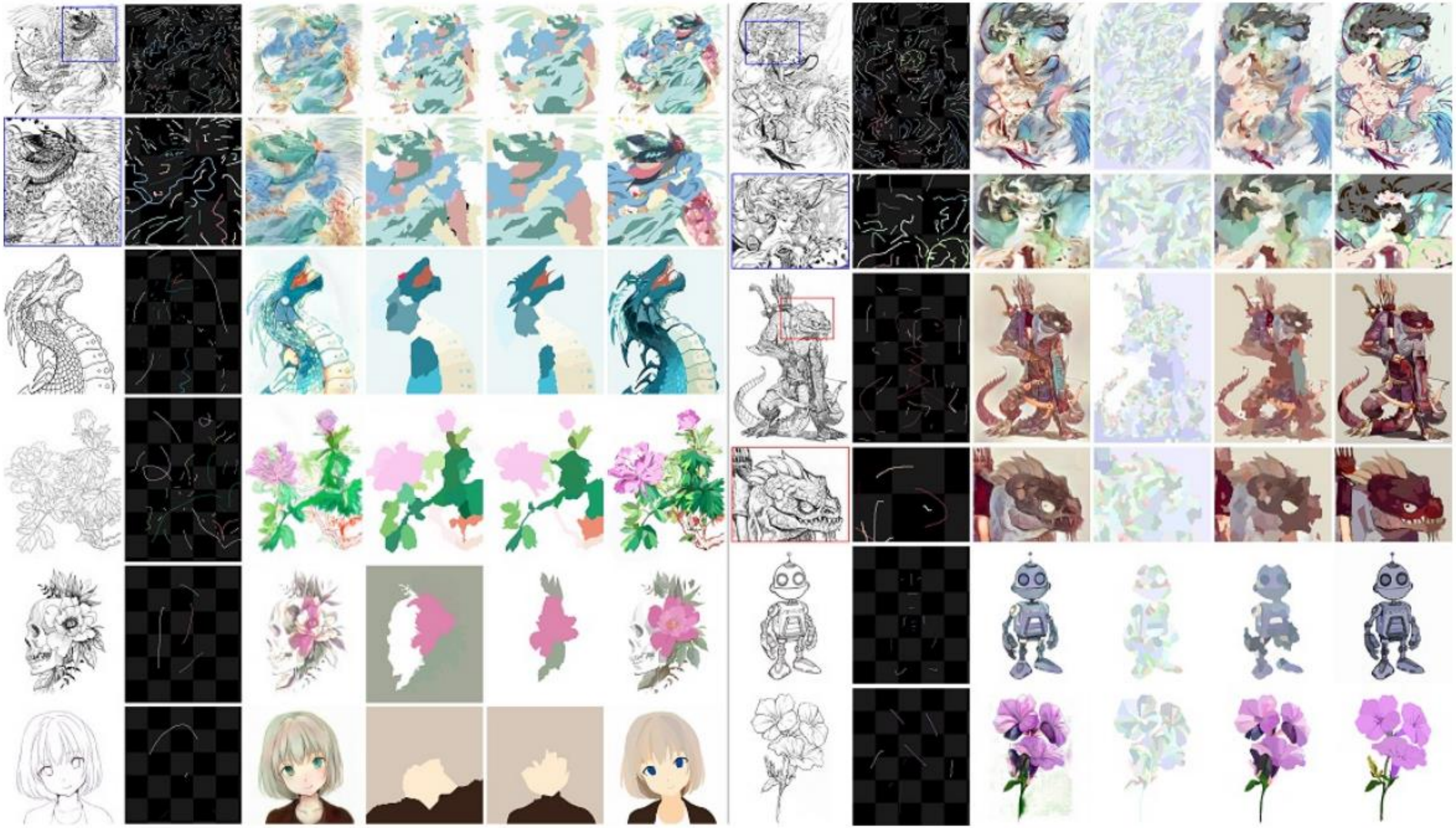


Figure 7. **Qualitative results with relatively complicated line arts.** We show a break-down of several results with our proposed approach. More examples are provided in the supplementary material. *Pea Princess, Flower with Alice, and Book Girl © used with artist permission.*



Line art Scribble Zhang [38] Qu [23] L.B. [28] Ours Line art Scribble Zhang [38] GIMP [5] [38]+[5] Ours

Figure 8. **Comparison with existing colorization methods.** Zoom in to see details. We compare our framework with [38, 23, 28] and the colorization method [38] combined with the segmentation method [5]. *Left: Flower Angel, Comollon, Wisteria Flowers, Poison Skull, and Megumi; Right: Fountain Angel, Goblin, Robot, and Azalea* © *used with artist permission.*

Evaluation

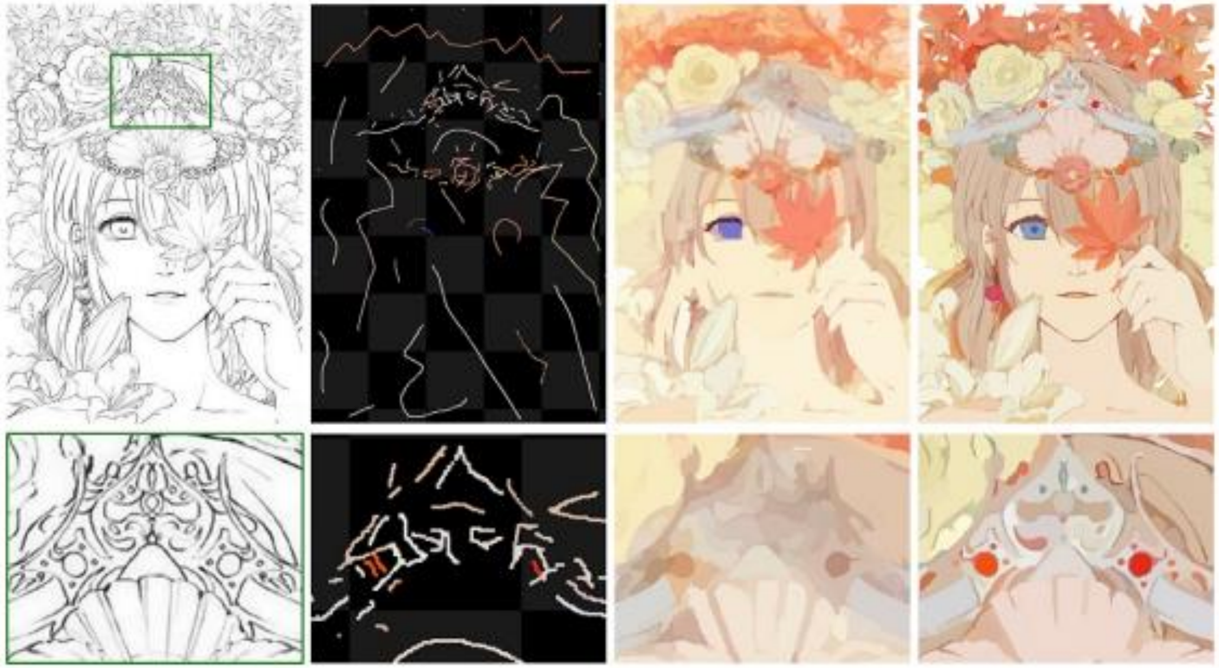


Figure 9. **Significance of Split Filling Mechanism (SFM).** We compare the results obtained from our neural architecture with (w/) or without (w/o) the SFM. *One Leaf Knows Autumn © used with artist permission.*

Evaluation

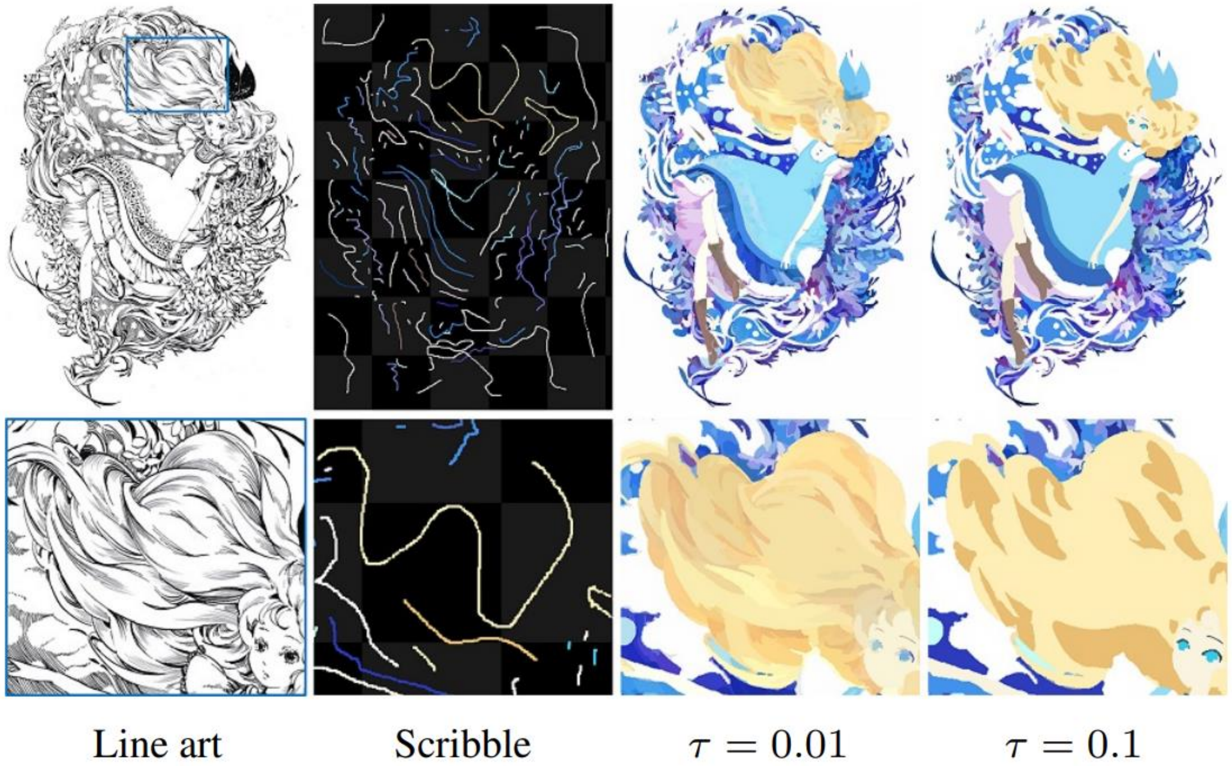


Figure 10. **Analysis of the $\boldsymbol{\tau}$ parameter.** The parameter τ controls how the framework merges adjacent regions. For higher values of τ, more regions will be merged. Different use cases may need different values of τ. *Alice's Night © used with artist permission.*

Evaluation

We calculate how many regions are filled manually by user scribbles or filled automatically by our framework. We divide all colorized regions into three categories:

Automatic regions.

Manual regions.

Semi-automatic regions.

Figure	Auto region	Semi-auto region	Manual region
Flower Mouse	513 (46.43%)	510 (46.12%)	82 (7.44%)
Pea Princess	478 (47.31%)	442 (43.73%)	91 (8.96%)
Sky with Alice	551 (55.25%)	315 (31.56%)	132 (13.19%)
Prayer	530 (60.69%)	284 (32.52%)	59 (6.79%)
Flower with Alice	447 (54.56%)	261 (31.83%)	112 (13.62%)
Tree Elves	599 (59.13%)	366 (36.18%)	48 (4.69%)
Anna in Dream	643 (61.68%)	317 (30.42%)	82 (7.90%)
Book Girl	589 (53.80%)	377 (34.45%)	129 (11.75%)
Reading Awake	396 (44.61%)	362 (40.79%)	130 (14.60%)
Overall	$53.72\% \pm 5.98\%$	$36.40\% \pm 5.44\%$	$9.88\% \pm 3.30\%$

Table 1. **Scribble analysis.** We perform an analysis of the number of scribbles used for the flat filling of different line drawings.