Manga Filling Style Conversion with Screentone Variational Autoencoder: Supplemental Material

MINSHAN XIE*, The Chinese University of Hong Kong
CHENGZE LI*, The Chinese University of Hong Kong
XUETING LIU, Caritas Institute of Higher Education
TIEN-TSIN WONG, The Chinese University of Hong Kong

ACM Reference Format:
Minshan XIE*, Chengze LI*, Xueting LIU, and Tien-Tsin WONG. 2020. Manga Filling Style Conversion with Screentone Variational Autoencoder: Supplemental Material. *ACM Trans. Graph.* 39, 6, Article 226 (December 2020), 6 pages. https://doi.org/10.1145/3414685.3417873

1 USER STUDY

1.1 Questionnaires

We have two user studies. One of them evaluates the visual quality for translation from manga to color image. The other evaluates the visual quality for translation from color comic to manga. In both user studies, 30 example images are used. The questionnaires for the two user studies are presented in Fig. 1. The statistics is presented and discussed in Section 5.6 in the paper.

1.2 T-test

At the end, in each user study for each method, we collected 300 scores. The mean score of our method is the highest among all 3 methods, as shown in Table 1 and Table 2. In particular, paired T-tests are performed to verify the significance of the statistic results. The results are shown in Table 3 and Table 4. Note that all P values are less than 0.05. This tells that our method is superior to existing ones in the perceptual quality of results.

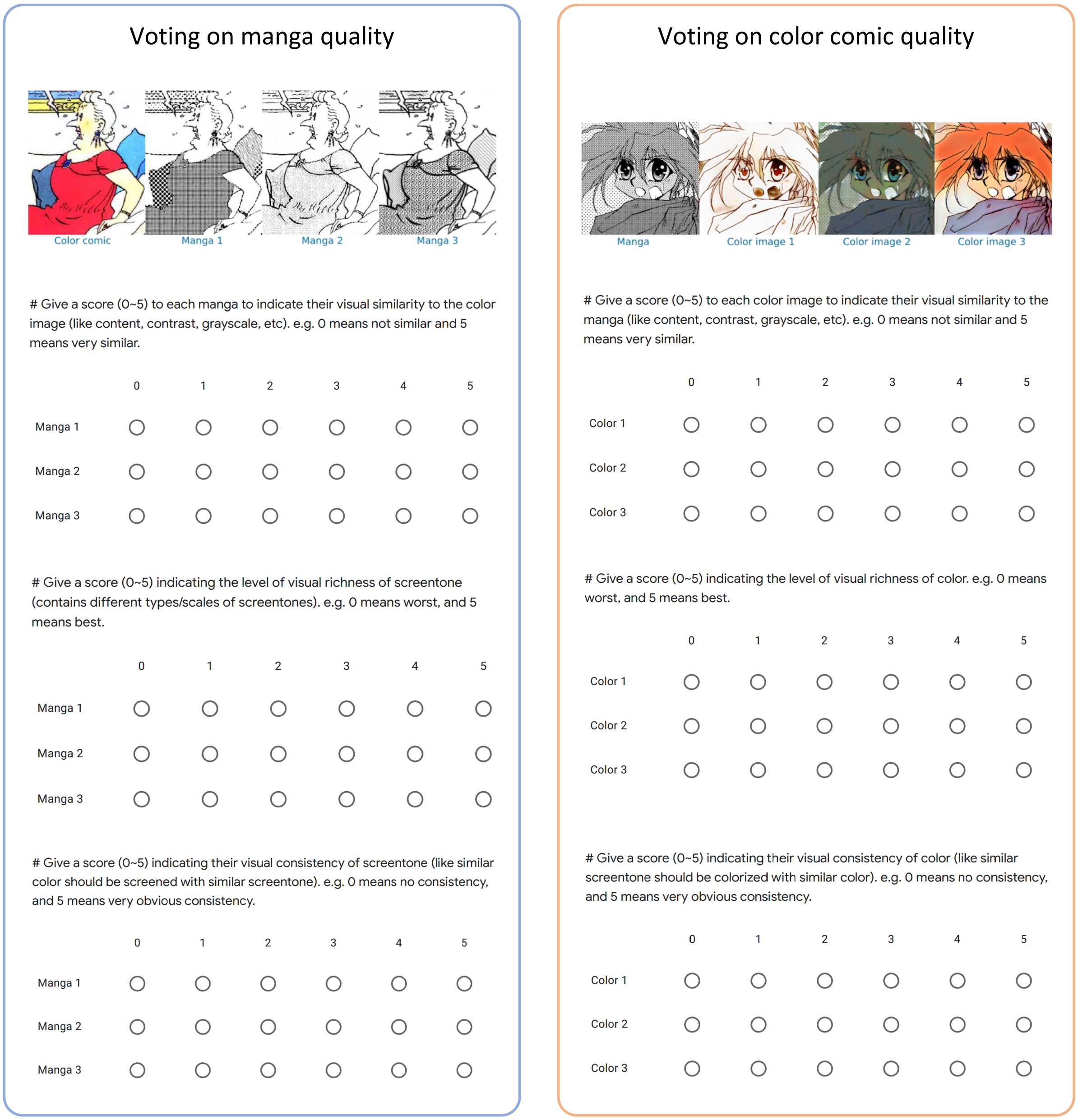
2 COMPARISON WITH SKETCH COLORIZATION

We have to clarify that it might not be fair to compare with Zhang et al. [2018] that takes as input a sketch only. The result is illustrated in Fig. 2(d). We further add an interesting comparison by providing color hints based on the tonal intensity images, as shown in Fig. 2(e). As we can see, although randomly providing color hints to the regions can generate more colorful results, it may also produce weird results. For example, regions of substantially different screentone type but similar tone, may be colorized in similar color, like the hair in first row of Fig. 2. Moreover, effects like shading and highlight, still may not be well preserved. As we can see in second

* Equal contribution.

Authors' addresses: Minshan XIE*, The Chinese University of Hong Kong, msxie@cse.cuhk.edu.hk; Chengze LI*, The Chinese University of Hong Kong, czli@cse.cuhk.edu.hk; Xueting LIU, Caritas Institute of Higher Education, tliu@cihe.edu.hk; Tien-Tsin WONG, The Chinese University of Hong Kong, ttwong@cse.cuhk.edu.hk.

0730-0301/2020/12-ART226 $15.00
https://doi.org/10.1145/3414685.3417873



Voting on manga quality

Give a score (0~5) to each manga to indicate their visual similarity to the color image (like content, contrast, grayscale, etc). e.g. 0 means not similar and 5 means very similar.

	0	1	2	3	4	5
Manga 1	○	○	○	○	○	○
Manga 2	○	○	○	○	○	○
Manga 3	○	○	○	○	○	○

Give a score (0~5) indicating the level of visual richness of screentone (contains different types/scales of screentones). e.g. 0 means worst, and 5 means best.

	0	1	2	3	4	5
Manga 1	○	○	○	○	○	○
Manga 2	○	○	○	○	○	○
Manga 3	○	○	○	○	○	○

Give a score (0~5) indicating their visual consistency of screentone (like similar color should be screened with similar screentone). e.g. 0 means no consistency, and 5 means very obvious consistency.

	0	1	2	3	4	5
Manga 1	○	○	○	○	○	○
Manga 2	○	○	○	○	○	○
Manga 3	○	○	○	○	○	○

Voting on color comic quality

Give a score (0~5) to each color image to indicate their visual similarity to the manga (like content, contrast, grayscale, etc). e.g. 0 means not similar and 5 means very similar.

	0	1	2	3	4	5
Color 1	○	○	○	○	○	○
Color 2	○	○	○	○	○	○
Color 3	○	○	○	○	○	○

Give a score (0~5) indicating the level of visual richness of color. e.g. 0 means worst, and 5 means best.

	0	1	2	3	4	5
Color 1	○	○	○	○	○	○
Color 2	○	○	○	○	○	○
Color 3	○	○	○	○	○	○

Give a score (0~5) indicating their visual consistency of color (like similar screentone should be colorized with similar color). e.g. 0 means no consistency, and 5 means very obvious consistency.

	0	1	2	3	4	5
Color 1	○	○	○	○	○	○
Color 2	○	○	○	○	○	○
Color 3	○	○	○	○	○	○

Fig. 1. Questionnaire for screened manga and colorized image evaluation in the user study.

row of Fig. 2, sometimes the colorized results with random hints may be worse and even influence the network to recognize the structures.

Method	visual similarity	screentone diversity	screentone consistency
Qu et al.[2008]	3.21	3.74	3.37
CycleGAN[Zhu et al. 2017]	3.57	3.00	3.43
Ours	4.13	4.22	4.15

Table 1. User study for translating color comic to manga.

Method	visual similarity	color diversity	color consistency
Zhang et al.[2018]	2.49	2.19	2.50
CycleGAN[Zhu et al. 2017]	3.21	3.29	3.23
Ours	4.38	4.51	4.38

Table 2. User study for translating manga to color comic.

Method	visual similarity	screentone diversity	screentone consistency
Qu et al.[2008]	7.45e-18	1.22e-8	1.16e-14
CycleGAN[Zhu et al. 2017]	7.67e-8	5.90e-33	1.37e-13

Table 3. Paired T-test Results ($\alpha = 0.05$) for translating color comic to manga.

Method	visual similarity	color diversity	color consistency
Zhang et al.[2018]	2.46e-59	3.52e-83	2.50/1.94e-53
CycleGAN[Zhu et al. 2017]	5.84e-39	5.31e-44	1.67e-38

Table 4. Paired T-test Results ($\alpha = 0.05$) for translating manga to color comic.

3 DETAILED NETWORK ARCHITECTURE

We adopt the 4-layer discriminator design of PatchGAN in both ScreenVAE model and bidirectioal translation model. Below shows the network details we adopted in our paper.

3.1 ScreenVAE

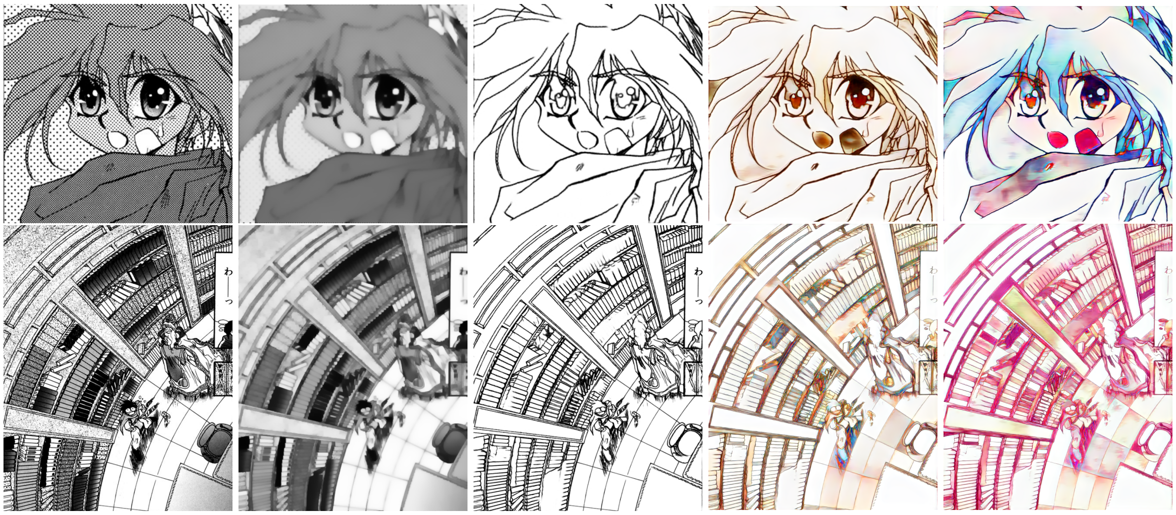
The ScreenVAE is a variational autoencoder that consists of a *ScreenEncoder* (*SE*) and a *ScreenDecoder* (*SD*). Table 5 and Table 6 listed the detail network architecture of *SE* and *SD*, respectively.

3.2 Bidirectional translation

The bidirectional translation model consists of two networks, a Screen2Color generator $G_{s\to c}$ and a Color2Screen generator $G_{c\to s}$. Table 7 and Table 8 listed the detail network architecture of $G_{s\to c}$ and $G_{c\to s}$, respectively. Table 9 shows the network architecture of the style extractor E_{st} which is used to extract the style vector of provided optional reference image. Similar to Karras. et al.[2019], we first map the style input to an intermediate latent space, which is then fed to each convolution layer in $G_{s\to c}$ through adaptive instance normalization (AdaIN).

REFERENCES

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4401–4410.



(a) Input manga (b) Tone Intensity (c) Line drawing (d) Without Hints (d) With Hints

Fig. 2. The colorized results generated by Zhang et al.[2018] with no hints or random local hints.

Part	Input → Output Shape	Layer Operations
Down-sampling	$(h, w, 2) \to (h, w, 24)$	CONV-(N24,K7x7,S1,P3),LN,ReLU
	$(h, w, 24) \to (\frac{h}{2}, \frac{w}{2}, 48)$	CONV-(N48,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 48) \to (\frac{h}{4}, \frac{w}{4}, 96)$	CONV-(N96,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 96) \to (\frac{h}{8}, \frac{w}{8}, 192)$	CONV-(N192,K3x3,S2,P1),LN,ReLU
Bottleneck	$(\frac{h}{8}, \frac{w}{8}, 192) \to (\frac{h}{8}, \frac{w}{8}, 192)$	Residual Block: CONV-(N192,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \to (\frac{h}{8}, \frac{w}{8}, 192)$	Residual Block: CONV-(N192,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \to (\frac{h}{8}, \frac{w}{8}, 192)$	Residual Block: CONV-(N192,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \to (\frac{h}{8}, \frac{w}{8}, 192)$	Residual Block: CONV-(N192,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \to (\frac{h}{8}, \frac{w}{8}, 192)$	Residual Block: CONV-(N192,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \to (\frac{h}{8}, \frac{w}{8}, 192)$	Residual Block: CONV-(N192,K3x3,S1,P1),LN,ReLU
Up-sampling	$(\frac{h}{8}, \frac{w}{8}, 192) \to (\frac{h}{4}, \frac{w}{4}, 96)$	UPSAMPLE(2),CONV-(N96,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 96) \to (\frac{h}{4}, \frac{w}{4}, 96)$	CONV-(N96,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 96) \to (\frac{h}{2}, \frac{w}{2}, 48)$	UPSAMPLE(2),CONV-(N48,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 48) \to (\frac{h}{2}, \frac{w}{2}, 48)$	CONV-(N48,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 48) \to (h, w, 24)$	UPSAMPLE(2),CONV-(N24,K3x3,S1,P1),LN,ReLU
	$(h, w, 24) \to (h, w, 1)$	CONV-(N24,K7x7,S1,P3),LN,ReLU

Table 5. Detail network architecture of ScreenEncoder *SE*.

Yingge Qu, Wai-Man Pang, Tien-Tsin Wong, and Pheng-Ann Heng. 2008. Richness-Preserving Manga Screening. *ACM Transactions on Graphics (SIGGRAPH Asia 2008 issue)* 27, 5 (December 2008), 155:1–155:8.

Lvmin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and Chunping Liu. 2018. Two-stage sketch colorization. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 261.

Part	Input → Output Shape	Layer Operations
Down-sampling	$(h, w, 4) \rightarrow (\frac{h}{2}, \frac{w}{2}, 48)$	CONV-(N48,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 48) \rightarrow (\frac{h}{4}, \frac{w}{4}, 96)$	CONV-(N96,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 96) \rightarrow (\frac{h}{8}, \frac{w}{8}, 192)$	CONV-(N192,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \rightarrow (\frac{h}{16}, \frac{w}{16}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{16}, \frac{w}{16}, 384) \rightarrow (\frac{h}{32}, \frac{w}{32}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{16}, \frac{w}{32}, 384) \rightarrow (\frac{h}{64}, \frac{w}{32}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{64}, \frac{w}{64}, 384) \rightarrow (\frac{h}{128}, \frac{w}{128}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
Up-sampling	$(\frac{h}{128}, \frac{w}{128}, 384) \rightarrow (\frac{h}{64}, \frac{w}{64}, 384)$	CONCAT,UPSAMPLE(2),CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{64}, \frac{w}{32}, 384) \rightarrow (\frac{h}{16}, \frac{w}{32}, 384)$	CONCAT,UPSAMPLE(2),CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{32}, \frac{w}{32}, 384) \rightarrow (\frac{h}{16}, \frac{w}{16}, 384)$	CONCAT,UPSAMPLE(2),CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{16}, \frac{w}{16}, 384) \rightarrow (\frac{h}{8}, \frac{w}{8}, 192)$	CONCAT,DECONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \rightarrow (\frac{h}{4}, \frac{w}{4}, 96)$	CONCAT,DECONV-(N192,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 96) \rightarrow (\frac{h}{2}, \frac{w}{2}, 48)$	CONCAT,DECONV-(N96,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 48) \rightarrow (h, w, 1)$	CONCAT,DECONV-(N48,K3x3,S2,P1),LN,ReLU
	$(h, w, 48) \rightarrow (h, w, 1)$	CONV-(N48,K7x7,S1,P3),LN,ReLU

Table 6. Detail network architecture of ScreenDecoder *SD*.

Part	Input → Output Shape	Layer Operations
Down-sampling	$(h, w, 5) \rightarrow (\frac{h}{2}, \frac{w}{2}, 48)$	CONV-(N48,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 48) \rightarrow (\frac{h}{4}, \frac{w}{4}, 96)$	CONV-(N96,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 96) \rightarrow (\frac{h}{8}, \frac{w}{8}, 192)$	CONV-(N192,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \rightarrow (\frac{h}{16}, \frac{w}{16}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{16}, \frac{w}{16}, 384) \rightarrow (\frac{h}{32}, \frac{w}{32}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{16}, \frac{w}{32}, 384) \rightarrow (\frac{h}{64}, \frac{w}{32}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{64}, \frac{w}{64}, 384) \rightarrow (\frac{h}{128}, \frac{w}{128}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
Up-sampling	$(\frac{h}{128}, \frac{w}{128}, 384) \rightarrow (\frac{h}{64}, \frac{w}{64}, 384)$	UPSAMPLE(2),CONCAT,CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{64}, \frac{w}{32}, 384) \rightarrow (\frac{h}{16}, \frac{w}{32}, 384)$	UPSAMPLE(2),CONCAT,CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{32}, \frac{w}{32}, 384) \rightarrow (\frac{h}{16}, \frac{w}{16}, 384)$	UPSAMPLE(2),CONCAT,CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{16}, \frac{w}{16}, 384) \rightarrow (\frac{h}{8}, \frac{w}{8}, 192)$	UPSAMPLE(2),CONCAT,CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \rightarrow (\frac{h}{4}, \frac{w}{4}, 96)$	UPSAMPLE(2),CONCAT,CONV-(N192,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 96) \rightarrow (\frac{h}{2}, \frac{w}{2}, 48)$	UPSAMPLE(2),CONCAT,CONV-(N96,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 48) \rightarrow (h, w, 4)$	UPSAMPLE(2),CONV-(N48,K3x3,S1,P1),LN,ReLU

Table 7. Detail network architecture of Screen2Color generator $G_{s \rightarrow c}$.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.

Part	Input → Output Shape	Layer Operations
Down-sampling	$(h, w, 4) \rightarrow (\frac{h}{2}, \frac{w}{2}, 48)$	CONV-(N48,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 48) \rightarrow (\frac{h}{4}, \frac{w}{4}, 96)$	CONV-(N96,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 96) \rightarrow (\frac{h}{8}, \frac{w}{8}, 192)$	CONV-(N192,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \rightarrow (\frac{h}{16}, \frac{w}{16}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{16}, \frac{w}{16}, 384) \rightarrow (\frac{h}{32}, \frac{w}{32}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{16}, \frac{w}{32}, 384) \rightarrow (\frac{h}{64}, \frac{w}{32}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{64}, \frac{w}{64}, 384) \rightarrow (\frac{h}{128}, \frac{w}{128}, 384)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
Up-sampling	$(\frac{h}{128}, \frac{w}{128}, 384) \rightarrow (\frac{h}{64}, \frac{w}{64}, 384)$	ADAIN,UPSAMPLE(2),CONCAT,CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{64}, \frac{w}{32}, 384) \rightarrow (\frac{h}{16}, \frac{w}{32}, 384)$	ADAIN,UPSAMPLE(2),CONCAT,CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{32}, \frac{w}{32}, 384) \rightarrow (\frac{h}{16}, \frac{w}{16}, 384)$	ADAIN,UPSAMPLE(2),CONCAT,CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{16}, \frac{w}{16}, 384) \rightarrow (\frac{h}{8}, \frac{w}{8}, 192)$	ADAIN,UPSAMPLE(2),CONCAT,CONV-(N384,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 192) \rightarrow (\frac{h}{4}, \frac{w}{4}, 96)$	ADAIN,UPSAMPLE(2),CONCAT,CONV-(N192,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 96) \rightarrow (\frac{h}{2}, \frac{w}{2}, 48)$	ADAIN,UPSAMPLE(2),CONCAT,CONV-(N96,K3x3,S1,P1),LN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 48) \rightarrow (h, w, 4)$	UPSAMPLE(2),CONV-(N48,K3x3,S1,P1),LN,ReLU

Table 8. Detail network architecture of Color2Screen generator $G_{c \rightarrow s}$.

Part	Input → Output Shape	Layer Operations
Down-sampling	$(h, w, 4) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(N48,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 64) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(N96,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 128) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(N192,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	CONV-(N384,K3x3,S2,P1),LN,ReLU
	$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (1, 1, 512)$	ADAPAVEPOOL(1,1)
Output Layer (μ)	$(1, 1, 512) \rightarrow (1, 1, 64)$	CONV-(N64,K1x1,S1,P0)
Output Layer ($log\sigma$)	$(1, 1, 512) \rightarrow (1, 1, 64)$	CONV-(N64,K1x1,S1,P0)

Table 9. Detail network architecture of style extractor E_{st}.