Supplementary Material for
Deep Edge-Aware Interactive Colorization against Color-Bleeding Effects

This supplementary material complements our paper with additional experimental results and their analysis. First, Section A presents how insensitive our edge-enhancing method is across the different real-world users. Then, qualitative results of the ablation study are presented in Section B, followed by an explanation about the analysis on our proposed metric CDR, with an example in Section C. In Section D, we provide the qualitative comparisons between our approach applied in encoder and decoder layers as well as their analysis. As an interactive colorization approach, we construct our own user interface, where users can draw scribbles with adjustable widths for edge enhancement. Section E describes this tool with the step-by-step demonstration. Furthermore, we provide additional quantitative and qualitative results of our method applied in two baselines (Zhang *et al*. [1] and Su *et al*. [2]) over various datasets, in Sections F and G. Moreover, Section H contains both quantitative results and qualitative examples of edge enhancements in sketch colorization, complementary to Section 5 in the main paper. Lastly, Section I provides the implementation details, such as the settings for training, network architecture, and hyper-parameters for edge-extracting modules in the generation of S_{pseudo}.

A. Robustness of Edge Enhancement across Users

To verify the robustness of the proposed method against ambiguous scribbles across different users, we measure the improved PSNR, LPIPS, and CDR on the enhanced colorization outputs obtained by different users. To this end, among the color-bleeding edges that were used in the user study (Section 4.4 in the main paper), we selected the scribbles for the edges that were enhanced by at least four different participants in our user study. For each edge, we calculated the improvement of these evaluation scores in the local regions near the scribbles drawn by the different users, following the same evaluation procedure in Section 4.4 in the main paper. The resulting *mean* and *standard deviation* are 3.380 ± 1.381, 0.031 ± 0.009, and 0.044 ± 0.031 for PSNR, LPIPS and CDR, respectively. These results indicate that our method achieves consistent improvement under various scribbles drawn by different users. In addition, we provide a qualitative example of these results in Fig. 1 with their evaluation scores. Our method robustly improves the color-bleeding edges given varying styles of the user scribbles.

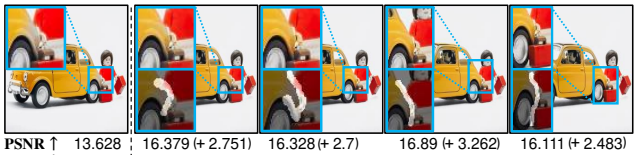

Figure 1: We visualize the initial colorization (first column) and its enhanced outputs by four different users (second to fifth column). Enlarged views of edge-enhanced regions and user-given scribbles are provided in blue boxes. Values in the brackets indicate the improved metric scores from those of the initial output. Zoom in for detail.

B. Ablation Studies on Width Augmentation and Consistency Loss

This section provides the qualitative results of ablation studies on the scribble width augmentation $w(\cdot)$ and the consistency loss $\mathcal{L}_{\text{con}}$, described in the Sections 3.2 and 3.4 in the main paper, respectively. In Fig. 2, the first column shows a binary map of user-driven scribble with its width varying from 1 to 11 pixels. The columns named $S_{\text{diff},a}$ and $S_{\text{diff},b}$ represent how the edge values are changed in *CIE ab* channels. $S_{\text{diff},a}$ and $S_{\text{diff},b}$ are obtained by subtracting $\mathcal{S}(I_{\text{init},ab})$ from $\mathcal{S}(I_{ab})$, where $\mathcal{S}(\cdot)$ approximates the edges, as described in Eq. 2 in the main paper, and $I_{\text{init},ab}$ is an initial colorized output by a backbone network and I_{ab} is a refined output by our method.

The fourth column (w/o $w(\cdot)$) presents the generated outputs of the model trained without width augmentation in a training phase, while the seventh column (Full) contains the outputs with augmentation. As the width of given scribbles w increases, especially when w becomes larger than 5, the width of changed edges $S_{\text{diff},a}$ and $S_{\text{diff},b}$ become thick as well. This results in an excessive increase of edges in ab channels, producing extremely vivid colors (*e.g.*, red) along the given boundaries. In contrast, our model trained with augmentations $w(\cdot)$ maintains the increases of enhanced edges regardless of given scribble's widths, robustly generating the plausible color corrections for all possible scribbles. This allows the users to provide their interactions without giving much effort to drawing sharp and thin scribbles.

Fig. 3 compares the qualitative results of our model trained with and without $\mathcal{L}_{\text{con}}$. As explained in Section 3.4 in the main paper, $\mathcal{L}_{\text{con}}$ enforces our model not to gener-

Figure 2: Qualitative comparisons between the generated outputs from our model trained with and without width augmentation $w(\cdot)$ for S_{pseudo} in the training. The fourth column contains the results of our model without $w(\cdot)$, and seventh column corresponds to the results of our model with all the proposed techniques, including $w(\cdot)$.

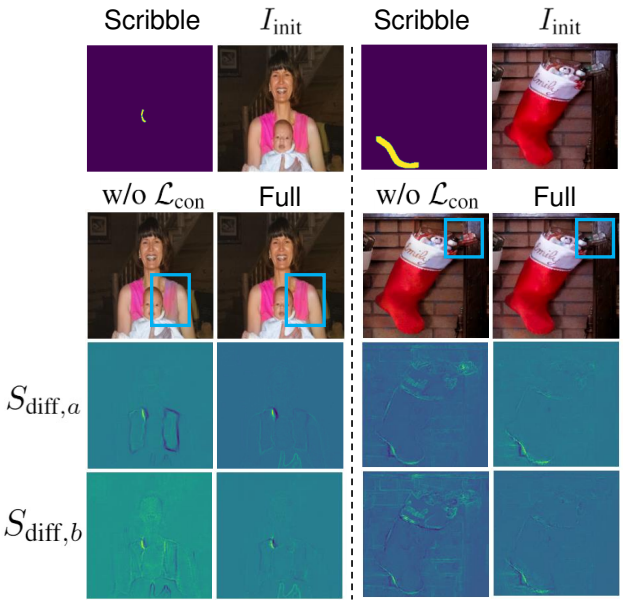

Figure 3: Qualitative comparisons between the generated outputs from our approach trained with and without $\mathcal{L}_{\text{con}}$ in the training. The column denoted as w/o $\mathcal{L}_{\text{con}}$ contains the results of our model without $\mathcal{L}_{\text{con}}$, and the column denoted as Full corresponds to the results of our model with all the proposed techniques, including $\mathcal{L}_{\text{con}}$.

ate unintentional color changes outside of the target edges. Therefore, the model trained without this objective function tends to produce unnecessary changes of colors in the wrong regions. As shown in third and fourth rows, $S_{\text{diff},a}$ and $S_{\text{diff},b}$ of the model with $\mathcal{L}_{\text{con}}$ shows more sparse changes of pixels, compared to that without $\mathcal{L}_{\text{con}}$. Ablation of $\mathcal{L}_{\text{con}}$ causes color distortions, such as washed-out colors or unintended color changes, illustrated in the images with blue bounding boxes. For example, a woman with a pink vest contains a washed-out color on her right side of the vest. This can be observed in the $S_{\text{diff},a}$, specifically in the dark boundaries of the vest (decreased edges).

C. Analysis on Cluster Discrepancy Ratio

This section provides a specific example that demonstrates the necessity of our proposed metric, *i.e.*, CDR. As mentioned in Section 4.1 in the main paper, this metric aims to cover the blind spot of two generally used metrics PSNR and LPIPS [3] in the colorization task. In Fig. 4, the first and the second columns are the ground-truth and colorized output I_1 with color-bleeding artifacts in the yellow box, respectively. I_2 is another example of colorized output with its bleeding edge enhanced by our proposed approach. Note that this is colorized with a different color from the ground-truth, enabled by providing user-interactive color hints. As I_2 contains the different colors from the ground-truth, unlike I_1, it is shown to record lower PSNR and higher LPIPS score than I_1. However, I_2 contains a *clear color edge*, while I_1 contains the color-bleeding effects, which make I_2 more visually favorable than I_1. The results of these metrics on I_2 against I_1 represent that they often underrate the quality of realistic sharp images that contain different colors from ground-truth. Our metric, however, essentially evaluates whether the colors are different across the adjacent objects and less dependent on the ground-truth colors. Therefore, the discrepancy ratio achieves a higher score for I_2 compared to I_1, showing that it is possible to robustly evaluate the color-bleeding artifacts.

Note that the sole use of our metric for evaluating colorization methods also reveals a bottleneck as mentioned in Section 4.3 in the main paper. More specifically, as our metric focuses on evaluating whether the colors are different between edges, saturated colors along the edges can be evaluated as favorable. Therefore, it is recommended to consider all of these metrics to fully evaluate the general colorization performance, including the edge preservation.

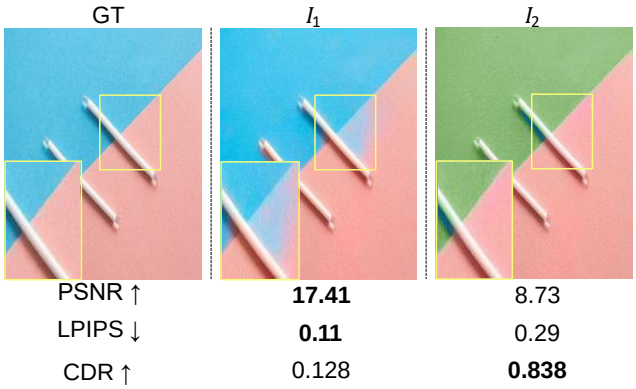

Figure 4: Quantitative comparisons of the colorized outputs, I_1 and I_2, via PSNR, LPIPS [3] and CDR. The yellow box provides the enlarged view on the regions of color-bleeding effects appearing in I_1, while enhanced in I_2 by our proposed method. Three evaluation scores for each image are presented below.

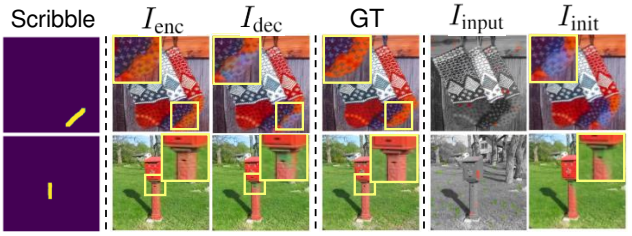

Figure 5: Qualitative comparisons of edge-enhancing network applied in encoder and decoder layers. I_{enc} and I_{dec} denote the enhanced results of the respective setting.

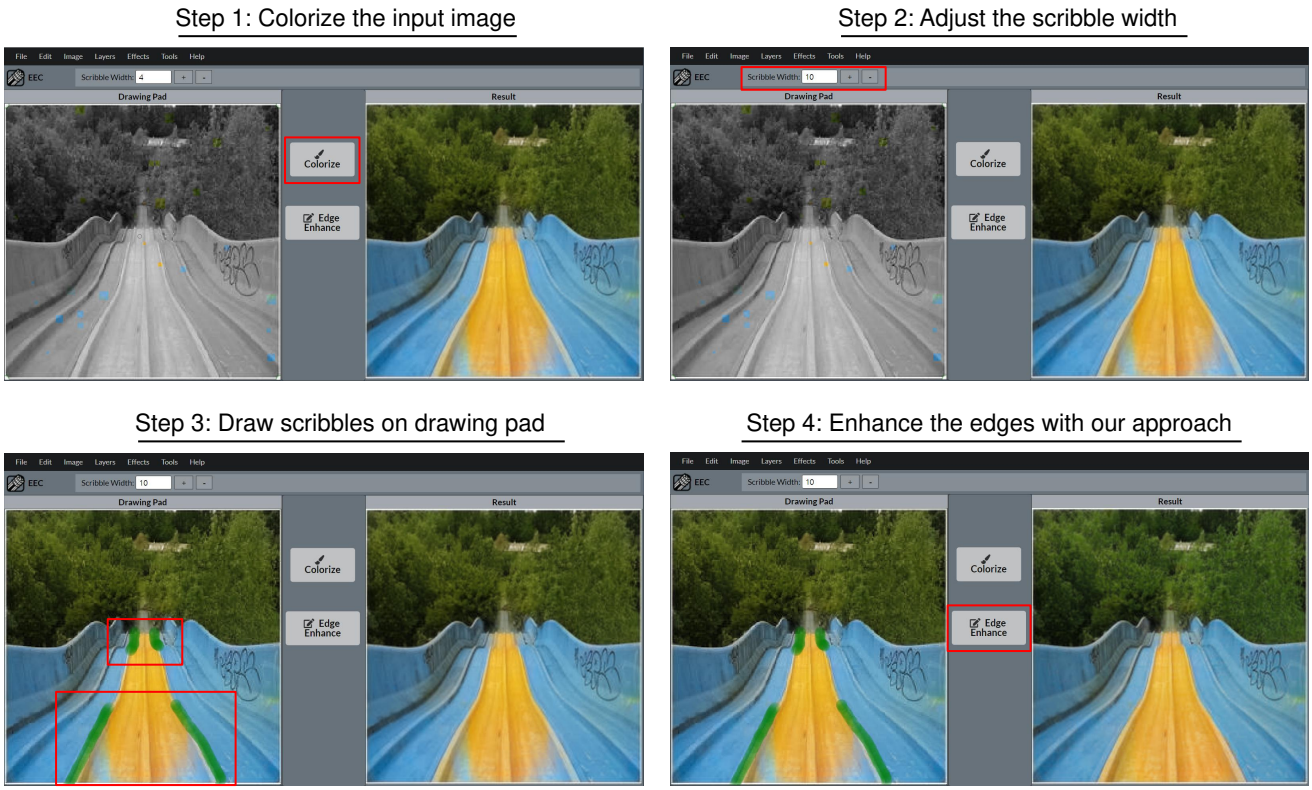

Figure 6: A step-by-step demonstration of an edge enhancement via our proposed user interface tool.

D. Selecting Layers for Edge-Enhancing Network

As briefly described in Section 3.3 in the main paper, we empirically find that applying edge-enhancing network E in the encoder layers achieves the intended results. To show the effectiveness of this choice of layers, Fig. 5 compares the results of E applied in encoder and decoder layers, I_{enc} and I_{dec}. It is shown that I_{enc} successfully preserves the edges between the different objects, correcting the colors spreading in the regions. For example, by giving a vertical scribble, wrongly spread green pixels inside the fire hydrant are corrected to a red color. However, I_{dec} merely increases the color difference along the edges without removing the green pixels inside the edges. We believe that E applied in the encoder layers refines the representations related to the edges, helping the following layers to spread the colors corresponding to the enhanced edges. In this regard, we choose encoder layers of the backbone network as an appropriate option for integrating E.

E. User-Interaction Demonstration

This section provides a step-by-step demonstration of edge enhancement through our user interface tool, as illustrated in Fig. 6. First, the user uploads and colorizes an input image (left panel) with additional color hints by clicking the *Colorize* button. Then, the user adjusts the width of the scribble and draws on color-bleeding edges of the colorized image shown in the left panel. After applying the scribbles, clicking the *Edge Enhance* button forwards the drawn scribbles into our edge-enhancing network to correct the annotated edges and show the improved result in the right panel.

F. Additional Quantitative Results

As described in Section 4 in the main paper, we apply CIC [7], DeOldify [8] and Zhang *et al*. [1] as our baseline models for unconditional colorization, and Zhang *et al*. [1], Su *et al*. [2] as our baseline for conditional colorization task. Table 1 contains the additional rows of quantitative comparisons of our model and baselines in the kernel size of 15 and 23, complementary to Table 1 in the main paper. For the qualitative results, we demonstrate that our framework is applicable to Zhang*et al*., which is the most widely used conditional colorization method, and Su *et al*., which is a recently proposed colorization approach. Utilizing our approach on these backbone networks improves the general colorization quality over the various dataset, outperforming the existing baselines. As our approach significantly improves the local color-bleeding artifacts, it improves the

Kernel Size	Method	ImageNet ctest [4]		COCO-Stuff [5]		Place205 [6]	
		LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑
K=7	CIC [7]	0.248	13.281	0.247	13.368	0.254	13.577
	DeOldify [8]	0.250	13.234	0.251	13.059	0.227	14.258
	Zhang *et al*. [1]	0.246	13.248	0.206	14.755	0.219	14.815
	+Ours	**0.217**	**13.919**	**0.192**	**15.037**	**0.211**	**15.104**
	Zhang *et al*. [1]*	0.208	14.966	0.158	17.456	0.171	17.530
	+Ours*	**0.177**	**16.041**	**0.143**	**17.953**	**0.161**	**17.906**
	Su *et al*. [2]*	0.185	16.393	0.187	15.971	0.194	17.032
	+Ours*	**0.177**	**16.507**	**0.176**	**16.188**	**0.187**	**17.098**
K=15	CIC [7]	0.221	14.377	0.237	13.937	0.219	14.841
	DeOldify [8]	0.218	14.399	0.217	14.362	0.226	14.382
	Zhang *et al*. [1]	0.209	14.662	0.204	14.975	0.218	14.962
	+Ours	**0.195**	**14.963**	**0.191**	**15.144**	**0.210**	**15.204**
	Zhang *et al*. [1]*	0.181	16.278	0.155	17.708	0.169	17.739
	+Ours*	**0.159**	**17.113**	**0.142**	**18.144**	**0.161**	**18.054**
	Su *et al*. [2]*	0.166	17.335	0.169	17.000	0.177	17.889
	+Ours*	**0.159**	**17.477**	**0.159**	**17.225**	**0.170**	**18.004**
K=23	CIC [7]	0.221	14.394	0.220	14.596	0.227	14.635
	DeOldify [8]	0.226	14.171	0.216	14.431	0.225	14.458
	Zhang *et al*. [1]	0.210	14.696	0.204	14.975	0.217	15.056
	+Ours	**0.195**	**14.987**	**0.191**	**15.180**	**0.210**	**15.282**
	Zhang *et al*. [1]*	0.164	17.195	0.154	17.806	0.169	17.864
	+Ours*	**0.147**	**17.868**	**0.141**	**18.204**	**0.161**	**18.147**
	Su *et al*. [2]*	0.154	17.945	0.156	17.611	0.164	18.567
	+Ours*	**0.148**	**18.081**	**0.148**	**17.840**	**0.158**	**18.689**
K=Full	CIC [7]	0.172	21.001	0.164	21.456	0.153	21.873
	DeOldify [8]	0.159	21.433	0.149	21.985	0.156	21.933
	Zhang *et al*. [1]	0.148	21.981	0.135	22.729	0.138	**22.846**
	+Ours	**0.147**	**22.026**	**0.134**	22.729	0.138	22.845
	Zhang *et al*. [1]*	0.086	27.202	0.080	27.681	0.087	27.697
	+Ours *	**0.085**	**27.559**	**0.078**	**27.955**	0.087	**27.935**
	Su *et al*. [2]*	0.091	26.211	0.089	26.050	0.090	27.414
	+Ours*	0.091	**26.291**	**0.088**	**26.233**	**0.089**	**27.486**

Table 1: Quantitative comparison with the baselines on 1,000 images in the ImageNet ctest [4], COCO-Stuff [5] and Place205 [6] validation set. Quantitative results in the local region show that our method effectively enhances the images.

applied baselines, especially when evaluated near the edge regions (*i.e.*, small kernel size).

Table 2 demonstrates the robustness of edge enhancing performance applied in Zhang *et al*. when the width of a given scribble varies from 1 to 9 pixel diameters. For the CDR, we evaluate it within the kernel size of 7 along the given edges. We observe that our model robustly enhances the local colorization performance in every dataset, given any scribble widths. Similar to the Table 1, the difference of global score between our model and baselines becomes minor when averaging the scores over all the spatial dimensions.

G. Additional Qualitative Results

Figs. 8 and 9 present the qualitative examples of edge enhancement applied to Zhang *et al*. [1], and Su *et al*. [2], respectively. The second and third columns of the figures contain the inference outputs of the baselines and their enhanced images using our method, respectively. Their scribbles, which are used to enhance the images, are visualized in the fourth column. Yellow boxes in the second column represent color-bleeding areas. All the images are collected from https://unsplash.com and Place205 [6].

The qualitative comparisons between our method, especially applied to Zhang *et al*. [1], and other baselines, are

Kernel Size	Datasets	Metrics	Zhang *et al.* [1]*	Scribble Width				
				1	3	5	7	9
K=7	ImageNet ctest [4]	LPIPS↓	0.208	0.182	0.178	0.174	**0.166**	0.203
		PSNR↑	14.966	15.877	16.040	16.022	**16.303**	14.862
		CDR↑	0.376	0.411	0.436	0.451	**0.471**	0.447
	COCO-Stuff [5]	LPIPS↓	0.211	0.185	0.181	0.174	**0.168**	0.180
		PSNR↑	14.964	15.790	15.938	16.074	**16.210**	15.594
		CDR↑	0.372	0.417	0.446	0.465	**0.478**	0.459
	Place205 [6]	LPIPS↓	0.223	0.205	0.201	0.193	**0.190**	0.201
		PSNR↑	15.091	15.714	15.877	16.036	**16.100**	15.540
		CDR↑	0.330	0.389	0.409	0.431	**0.438**	0.425
K=Full	ImageNet ctest [4]	LPIPS↓	0.086	0.085	0.085	0.085	**0.084**	0.087
		PSNR↑	27.202	27.555	**27.558**	27.510	27.554	27.293
		CDR↑	–	–	–	–	–	–
	COCO-Stuff [5]	LPIPS↓	0.080	0.078	0.078	0.078	0.078	0.079
		PSNR↑	27.677	**27.964**	27.959	27.953	27.939	27.880
		CDR↑	–	–	–	–	–	–
	Place205 [6]	LPIPS↓	0.087	0.087	0.087	0.087	0.087	0.087
		PSNR↑	27.697	**27.945**	27.931	27.926	27.897	27.877
		CDR↑	–	–	–	–	–	–

Table 2: Quantitative results based on scribble width. The scores for the CDR are reported only in the local regions along the edges, within the kernel size of 7.

presented in Fig. 10. While all the baselines in the figure contain the color-bleeding artifact in the regions bounded with a yellow box, our approach improves the edges, as shown in the fourth and sixth columns. Yellow boxes indicate the color-bleeding areas, and we provide the enlarged views of the areas on the right lower corner in the images. The images are selected from COCO-Stuff [5] and Place205 [6].

H. Quantitative and Qualitative Results in Sketch Colorization

In Section 5 in the main paper, we demonstrate that our approach has the potentials to enhance the edges in the sketch colorization task as well. In this study, we utilize the two datasets, Yumi's Cells [9] and Danbooru [10]. Table 3 demonstrates that applying edge-enhancing network on the colorization model, newly trained for sketch colorization task, can also improve the performance, especially along the edge regions. Additional qualitative results complementary to the Fig 7 in the main paper are illustrated in Fig. 7.

I. Implementation Details

This section provides the training details for edge enhancement, detailed architecture of the edge-enhancing network, and hyper-parameters used in S_{pseudo} generation, complementary to Section 3.5 in the main paper. Afterward, hyper-parameters of the super-pixel methods for our proposed CDR are explained as well. Additionally, the implementation details of the sketch colorization are explained, complementary to the Section 5 in the main paper.

Training Details for Edge-Enhancing Network We train and evaluate our model with a fixed size of 256 × 256 images for every dataset. We apply three edge-enhancing networks on the $5^{th}, 10^{th}$ and 17^{th} encoder layers of Zhang *et al.* [1]. In Su *et al.* [2], the instance-level colorization branch takes the patches of images cropped by their bounding boxes after the object detection module. These object-level colorized outputs are fused into a full-image colorization branch via fusion modules, predicting the final colors. For the full-image branch, we provide the pseudo-scribbles for the full image, as we do in Zhang *et al*. To accommodate our interactions into this object-level colorization branch as well, we crop the pseudo-scribbles as well as the images by their bounding boxes and give them to the branch as inputs. Therefore, our edge-enhancing networks applied in this branch take the cropped scribbles corresponding to the cropped patches of color-bleeding artifacts, generating the refined activations to be fused into the full-image branch. Both three edge-enhancing networks are applied on the $5^{th}, 10^{th}$ and 17^{th} encoder layers of instance-level and full-image network, respectively. In the training phase, we set the hyper-parameters for each loss function as $\lambda_{\text{edge}} = \lambda_{\text{con}} = 50$ and $\lambda_{\text{reg}_1} = \lambda_{\text{reg}_2} = \lambda_{\text{reg}_3} = 1$ in Zhang *et al.*, and

Figure 7: Qualitative examples of edge enhancement on sketch colorization. Yellow boxes provide a view of color-bleeding regions.

Kernel Size	Method	Yumi's Cells [9]			Danbooru [10]		
		LPIPS↓	PSNR↑	CDR↑	LPIPS↓	PSNR↑	CDR↑
K=7	Zhang *et al.* [1]*	0.240	10.738	0.231	0.322	11.970	0.191
	+Ours*	**0.226**	**11.287**	**0.298**	**0.317**	**12.335**	**0.214**
K=15	Zhang *et al.* [1]*	0.217	12.194	–	0.374	9.829	–
	+Ours*	**0.210**	**12.487**	–	**0.371**	**10.092**	–
K=23	Zhang *et al.* [1]*	0.219	12.211	–	0.290	12.146	–
	+Ours*	**0.211**	**12.531**	–	**0.286**	**12.448**	–
K=Full	Zhang *et al.* [1]*	**0.153**	19.280	–	0.204	18.698	–
	+Ours*	0.154	**19.291**	–	**0.201**	**18.975**	–

Table 3: Quantitative comparison with the baselines in the Yumi's Cells [9] and Danbooru [10] validation set. The scores for the CDR are reported only in the local regions along the edges, within the kernel size of 7.

Datasets	σ	TH_h	TH_l	TH_{gap}
ImageNet [4]	1.2	0.7	0.2	0.4
COCO-Stuff [5]	1.2	0.7	0.2	0.4
Place205 [6]	1.2	0.7	0.2	0.4
Yumi's Cells [9]	1.3	0.7	0.2	0.4
Danbooru [10]	0.7	0.8	0.2	0.5

Table 4: Hyper-parameters for the Canny edge-extractor.

$\lambda_{\text{edge}} = 50$ and $\lambda_{\text{con}} = \lambda_{\text{reg}_1} = \lambda_{\text{reg}_2} = \lambda_{\text{reg}_3} = 10$ in Su *et al*. The width of the augmentation module for the S_{pseudo} is randomly sampled from 1 to 10 pixels in the training phase. We use Adam [11] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is initially set to 0.01 and is gradually decayed for every epoch.

Edge-Enhancing Network Architecture Edge-enhancing network consists of 4 convolutional layers, each of which contains a 3×3 convolution filter with a stride of 1, ReLU [12] and Batch normalization layer [13].

Pseudo-Scribble Generation For generating the plausible approximation of real user-provided scribbles, we tune the hyper-parameters of the Canny edge extractor [14][1] on every dataset, including ImageNet [4], COCO-Stuff [5], Place205 [6], Yumi's Cells [9] and Danbooru [10], written in Table 4. We report Sigma (σ), high-threshold (TH_h), low-threshold (TH_l), and threshold gaps (TH_{gap}) of each dataset. Sigma stands for the standard deviation of the Gaussian kernel for the noise reduction step. Both TH_h and TH_l denote the threshold values for the double threshold step after the non-maximum suppression. We apply different high thresholds for $I_{\text{gt},ab}$ and $I_{\text{init},ab}$ to select highly probable edges for the bleeding artifacts. In other words, we apply a rigid criterion on the ground-truth image for extracting the edges compared to the generated outputs, resulting in severely weak edges from this comparison. We denote this threshold gap as TH_{gap}.

Cluster Discrepancy Ratio We use simple linear iterative clustering [15] (SLIC) method to assign each pixel with a cluster assignment based on its colors and textures. To focus on the color information, we run the SLIC on ab channels of

[1]Canny edge-extractor consists of 5 steps, which include noise reduction, Sobel filtering, non-maximum suppression, double threshold and edge tracking.

both ground-truth and colorized outputs. We set the number of clusters to 250, compactness to 10, and sigma to 1 for each image. After we compute each ratio from the a and b channels and average them to produce the final score.

Sketch Colorization We adjust the part of architecture and the training details of Zhang *et al*. [1] to enable its colorization with local hints in the sketch colorization task. We replace 1) the input image from gray-scale to sketch image, and 2) output channel size from 2 for ab channels to 3 for RGB outputs. To obtain the sketch image from the color image of each dataset, we first apply Gaussian blurring ($\sigma = 0.7$) to remove noisy edges and utilize a widely used edge extractor algorithm called XDoG [16], as used in Lee *et al*. [17] for the sketch colorization task. Afterward, we obtain 54,317 training images and 6,036 test images for Yumi's Cells [9], and 7,014 and 380 images for Danbooru [10]. Then, we apply the training details proposed in the original paper [1], such as providing color hints and objective functions. Note that the objective functions for color prediction are adjusted from minimizing the difference of ab channels to RGB channels between generated output and the ground-truth. To train our model on this network, we convert both generated RGB outputs images and ground-truth into *Lab* images to apply our proposed objective edge-enhancing loss.

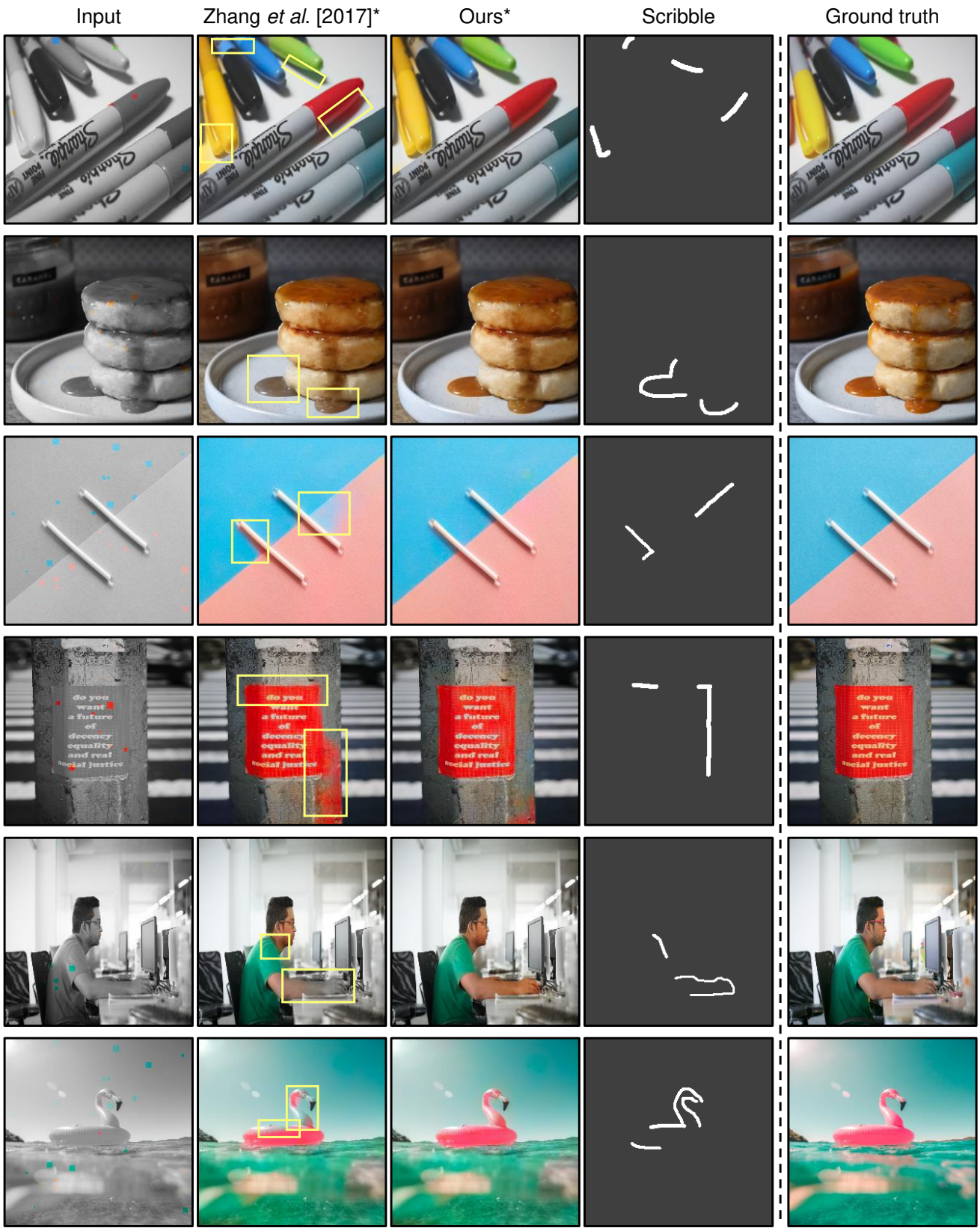

Figure 8: Qualitative results of edge-enhancement in the conditional colorization setting.

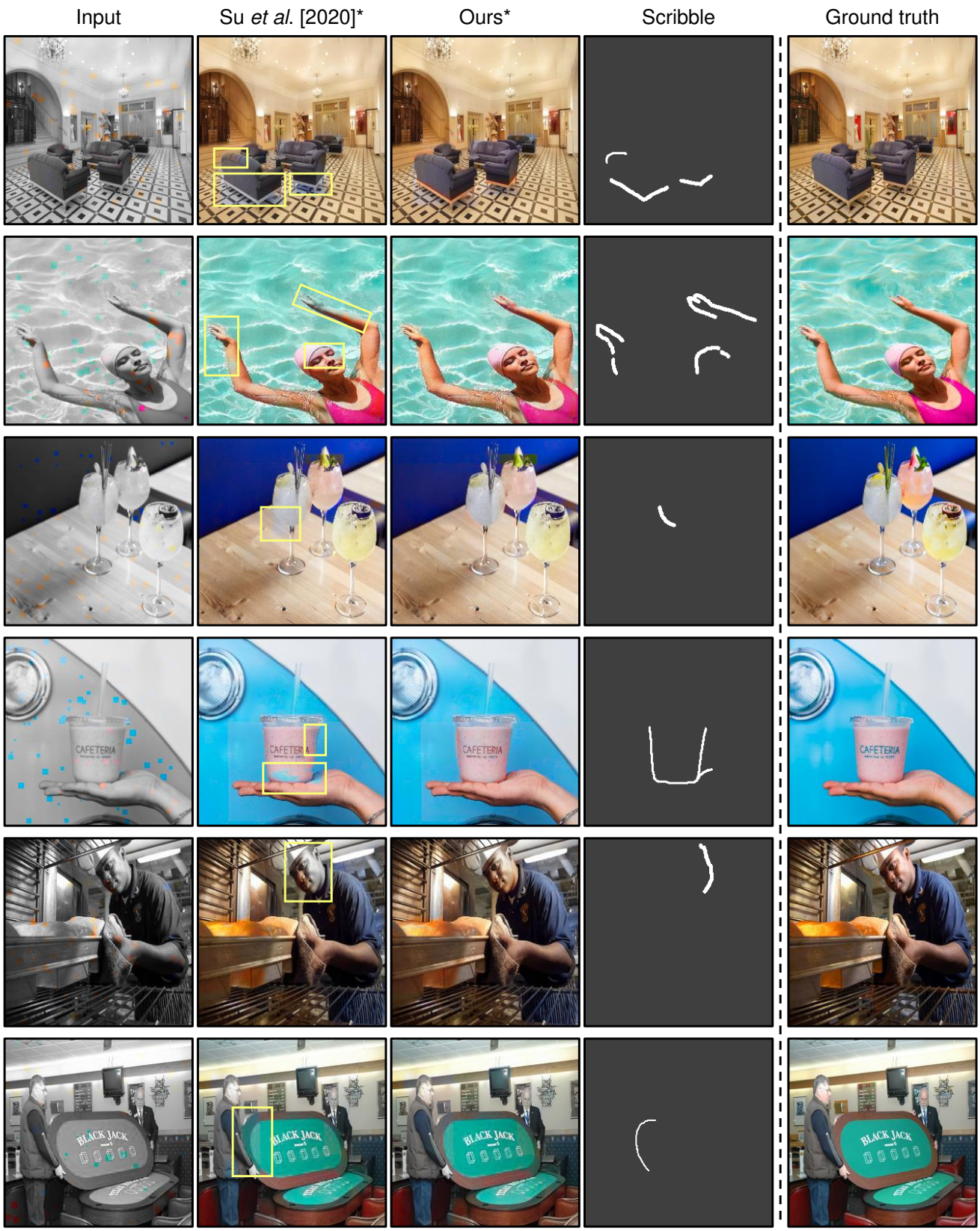

Figure 9: Qualitative results of edge-enhancement in the conditional colorization setting.

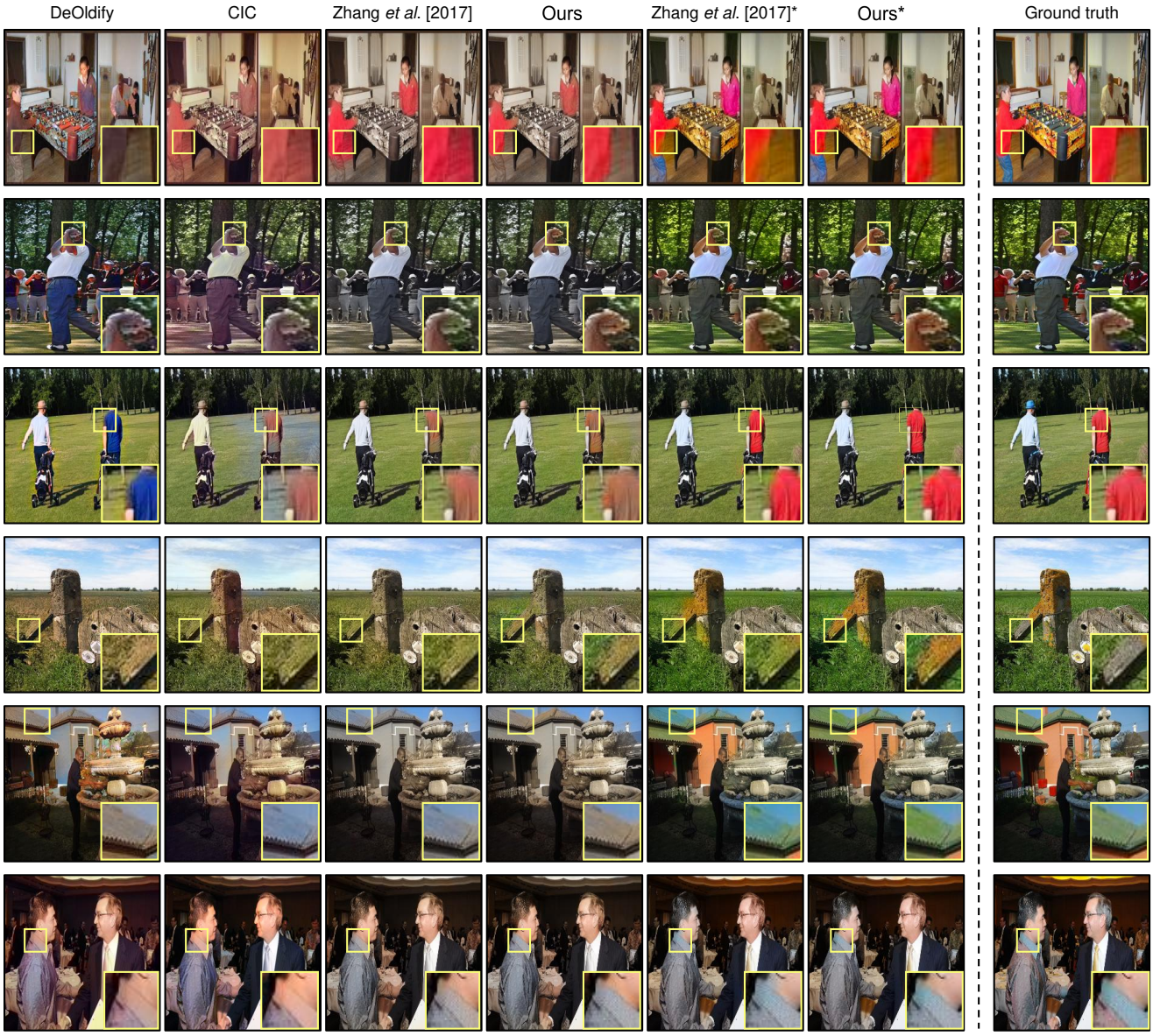

Figure 10: Qualitative comparisons between our model and other baseline models.

References

[1] 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. *Proc. the ACM Transactions on Graphics (ToG)*, 36, 2017. 1, 4, 5, 6, 7, 8

[2] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. Instance-aware image colorization. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2020. 1, 4, 5, 6

[3] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018. 3

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2009. 5, 6, 7

[5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018. 5, 6, 7

[6] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPMAI)*, 40:1452–1464, 2018. 5, 6, 7

[7] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016. 4, 5

[8] Jason Antic. deoldify. https://github.com/jantic/DeOldify, 2020. [Online; accessed 07-11-2020]. 4, 5

[9] NaverWebtoon. Yumi's cells. https://comic.naver.com/webtoon/list.nhn?titleId=651673, 2019. [Online; accessed 22-11-2019]. 6, 7, 8

[10] Aaron Gokaslan Gwern Branwen. Danbooru2017: A large-scale crowdsourced and tagged anime illustration dataset. https://www.gwern.net/Danbooru2017, 2018. [Online; accessed 22-03-2018]. 6, 7, 8

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. the International Conference on Learning Representations (ICLR)*, 2015. 7

[12] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. the International Conference on Machine Learning (ICML)*, 2010. 7

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. the International Conference on Machine Learning (ICML)*, 2015. 7

[14] J. Canny. A computational approach to edge detection. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPMAI)*, 8:679–698, 1986. 7

[15] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPMAI)*, 34:2274–2282, 2012. 7

[16] Holger WinnemöLler, Jan Eric Kyprianidis, and Sven C Olsen. Xdog: an extended difference-of-gaussians compendium including advanced image stylization. *Computers & Graphics*, 36:740–753, 2012. 8

[17] J. Lee, E. Kim, Y. Lee, D. Kim, J. Chang, and J. Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2020. 8