Supplementary material: Multimodal Style Transfer via Graph Cuts

Yulun Zhang[1], Chen Fang[2], Yilin Wang[3], Zhaowen Wang[3], Zhe Lin[3], Yun Fu[1], Jimei Yang[3]
[1]Northeastern University, [2]ByteDance AI Lab, [3]Adobe Research

1. Style Representation and Matching

1.1. Multimodal Style Representation

We visualize multimodal style representation in Fig. 1. For each style image, we extract its VGG feature (at layer Conv_4_1 in VGG-19) and cluster it into $K = 3$ clusters. Then, we conduct t-SNE [7] visualization with the cluster labels. As shown in Fig. 1, clustering results match the multimodal style distribution well. Features in the similar group would tend to have similar cluster number. These observation not only shows the multimodal style distribution, but also demonstrates that clustering is a proper way to model such a multimodal distribution.

1.2. Graph based Style Matching

We show more details about graph based style matching in Figs. 2, 3, and 4. For better understanding of the graph based style matching, we first set $K = 2$. We extract style and content features from Conv_4_1 layer in VGG-19 [9]. Due to several downsampling modules in VGG-19, the spatial resolution of the features is much smaller than that of the inputs. We label the spacial style feature pixels with their corresponding cluster labels and obtain the style cluster maps. According to the style cluster maps in Figs. 2 and 3, we find that style feature clustering grasps semantic information from style images. After style matching in pixel level, we get the content-style matching map, which also reflect the semantic information, matching the content structures adaptively. Such an adaptive matching alleviates the wash-out artifacts, when the style is very simple or has large area of unified background.

We also show style matching results in Fig. 4 and set $K = 3$, as we used in the paper by default. We can see our proposed graph based style matching still handles the matching well.

2. Experimental Results

2.1. Codes and Parameters for Comparisons

We compare with 7 state-of-the-art methods: method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], DFR [2], AvatarNet [8], and LST [5]. We obtain results using their official codes and default parameters, except for Gatys *et al.*[1] (iterations=10^3, learning rate=1). CNNMRF[2], AdaIN[3], WCT[4], DFR[5], AvatarNet[6], and LST[7] use official codes and default parameters. We have also provided the test code for our MST in the supplementary file. Readers are encouraged to try it.

2.2. Comparisons with Prior Arts

We compare MST with method by Gatys *et al.* [1], CNNMRF [4] AdaIN [3], WCT [6], DFR [2], AvatarNet [8], and LST [5]. We show extensive comparisons in Figs. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, and 19. We set $K = 3$ in MST through the whole comparison. We give further analyses in the figure captions. Those comparisons strongly demonstrate the effectiveness of our proposed MST.

[1]Gatys *et al.*: https://github.com/jcjohnson/neural-style
[2]CNNMRF: https://github.com/chuanli11/CNNMRF
[3]AdaIN: https://github.com/xunhuang1995/AdaIN-style
[4]WCT: https://github.com/Yijunmaverick/UniversalStyleTransfer
[5]DFR: https://github.com/msracver/Style-Feature-Reshuffle
[6]AvatarNet: https://github.com/LucasSheng/avatar-net
[7]LST: https://github.com/sunshineatnoon/LinearStyleTransfer

2.3. User Study

We provide more details about our user study in Figs. 20 and 21. As shown in the second row of Fig. 20, each user was provided 1 content-style pair and 6 results by different methods. These results are placed randomly, so that the user didn't know the correspondence between the result and method. Each user would make 20 decisions in the survey.

In Fig. 21, we show the statistic results. In fact, we received 2220 votes from 111 users in total. We only use the first 2000 votes (namely 100 users) to report our user study, because some users submitted their surveys very late.

The users come from different groups. About 30% of them are undergraduates, who major in computer science, electronic engineering, art, economics, media arts and so on. About 30% of them are graduates, who have professional experience in their research fields, such as computer vision and machine learning. About 30% of them are professional employees from some related companies. About 10% of them are from other communities.

2.4. Style Cluster Number

We investigate how style cluster number K affect the stylization in Fig. 22, 23, and 24. K ranges from 1 to 6. As we can see, for simple style, $K = 1$ would suffer from wash-out artifacts to some degree. This drawback indicates the necessary to treat the style feature as multimodal representation. As we enlarge K, our MST can match each content feature pixel with better style cluster and alleviate the wash-out artifacts. Moreover,

2.5. Adaptive Multi-style Transfer

We show multi-style transfer in Figs. 25 and 26. Our adaptive multi-style transfer is also similar to spatial control in previous methods [3, 6]. But, they need additional manually designed mask as input, consuming more user effort and lacks flexibility to specific content image. Instead, MST automatically allows good matching between content and style features. More analyses are given in the figure captions.

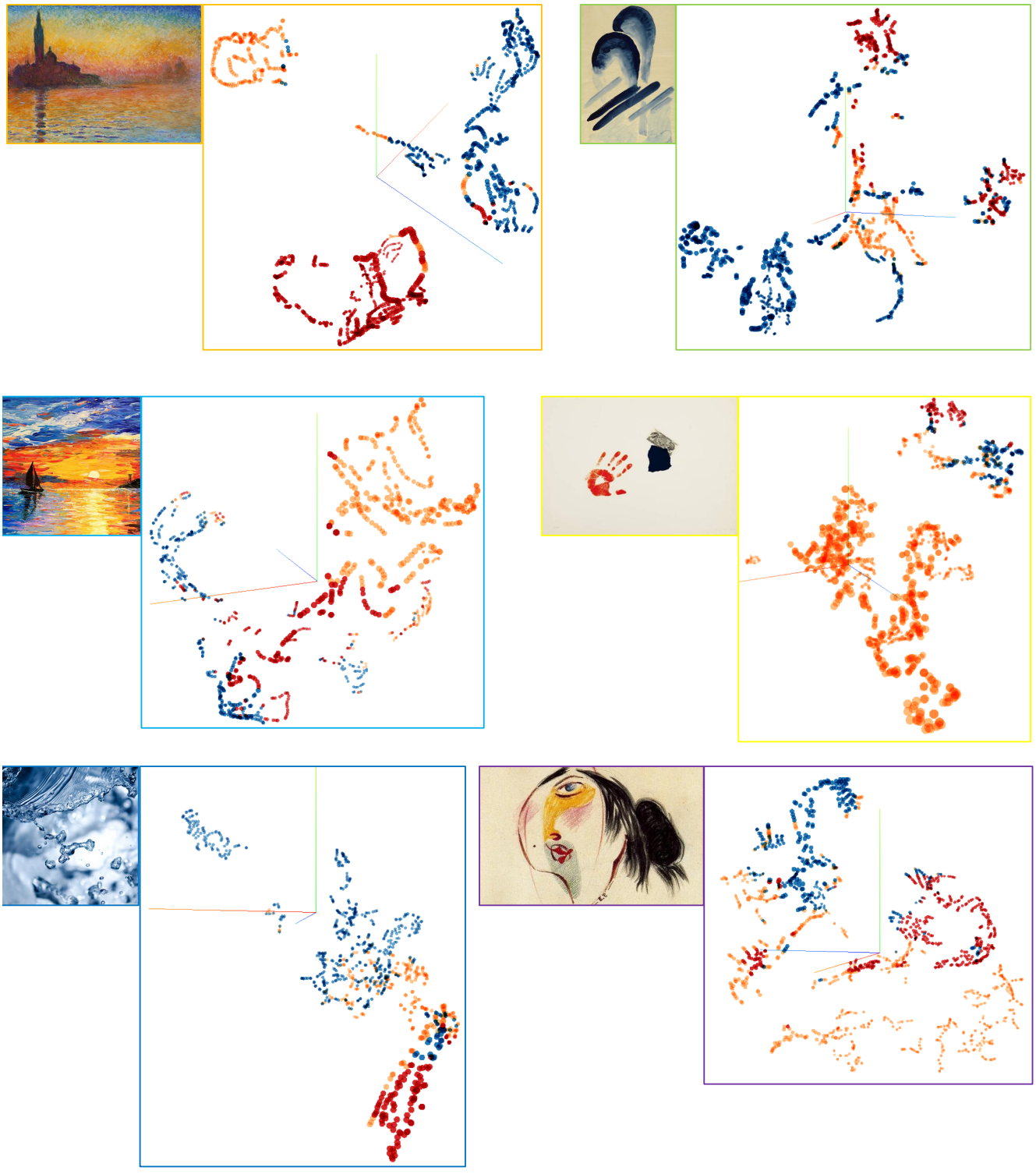
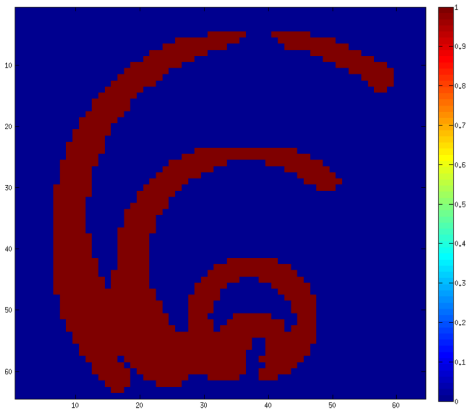


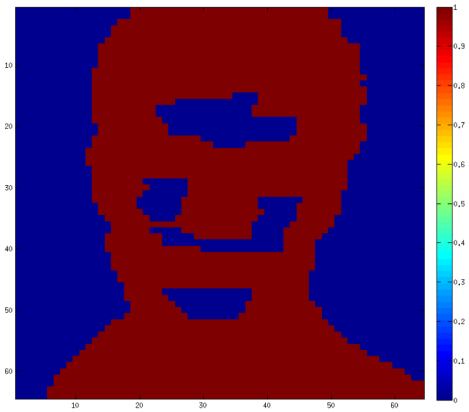
Figure 1: t-SNE [7] visualization for style features with cluster labels. For each style-visualization pair, We set $K = 3$ and label each style feature point with its corresponding cluster label. We can see that features in similar group have similar cluster labels. This observation indicates that clustering style features is a proper way for multimodal style representation.

Style



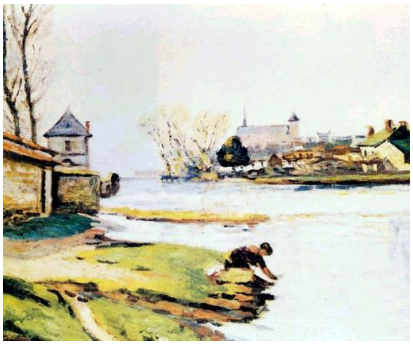
Style cluster map

Content

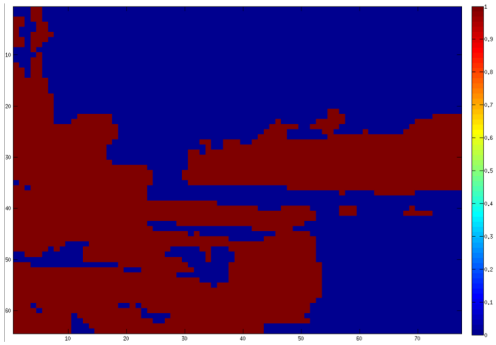


Content-style matching map

Result

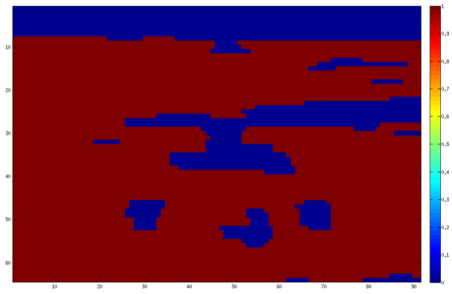


Style



Style cluster map

Content

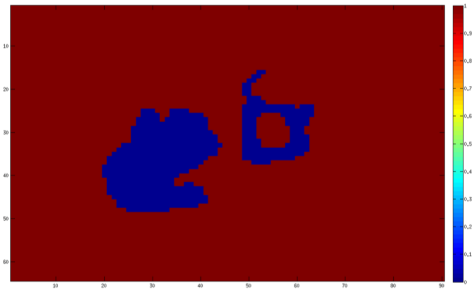


Content-style matching map

Result

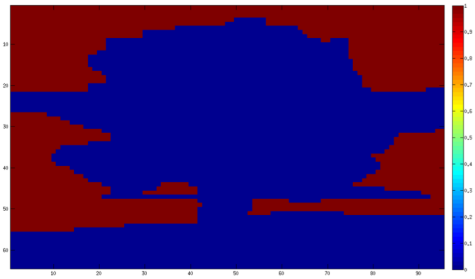
Figure 2: Graph based style matching. Style features are clustered into $K = 2$ clusters. Both the clustering and matching grasp the semantic information according to the specific structures of style and content images.

Style



Style cluster map

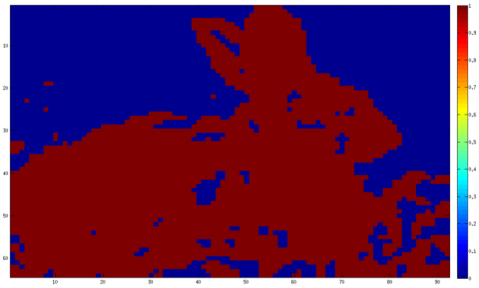
Content



Content-style matching map

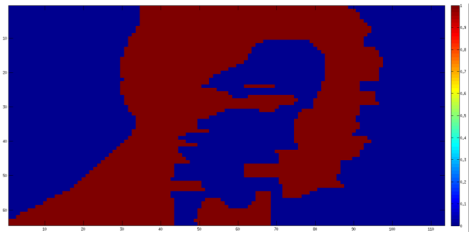
Result

Style



Style cluster map

Content



Content-style matching map

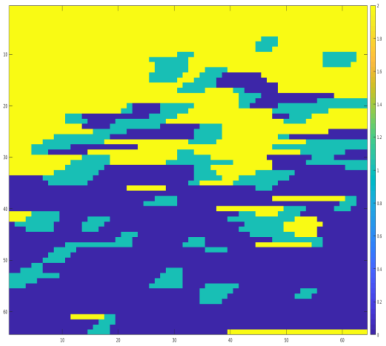
Result

Figure 3: Graph based style matching. Style features are clustered into $K = 2$ clusters. The color in 'content-style matching map' indicates that its corresponding style cluster labeled with same color in 'style cluster map'. Both the clustering and matching grasp the semantic information according to the specific structures of style and content images. For simple style (e.g., 1st row), the graph based style matching avoids wash-out artifacts. For complex style (e.g., 3rd row), it results in good semantic match according to the content and style structures.

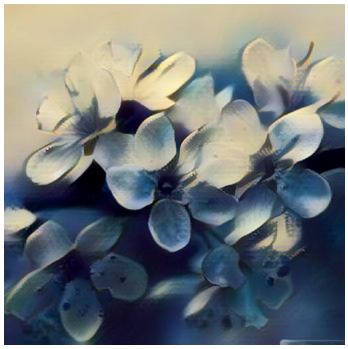
Style

Style cluster map

Content

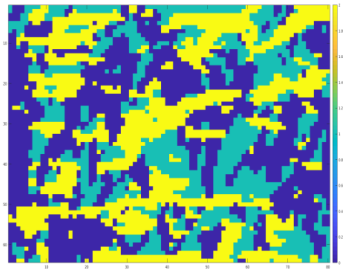


Content-style matching map



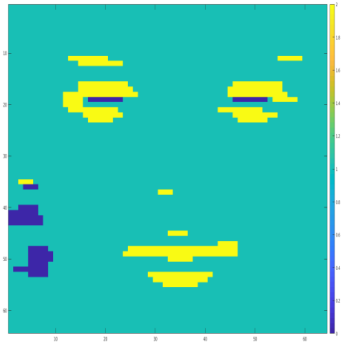
Result

Style



Style cluster map

Content



Content-style matching map

Result

Figure 4: Graph based style matching. Style features are clustered into $K = 3$ clusters. The color in 'content-style matching map' indicates that its corresponding style cluster labeled with same color in 'style cluster map'. Both the clustering and matching grasp the semantic information according to the specific structures of style and content images.

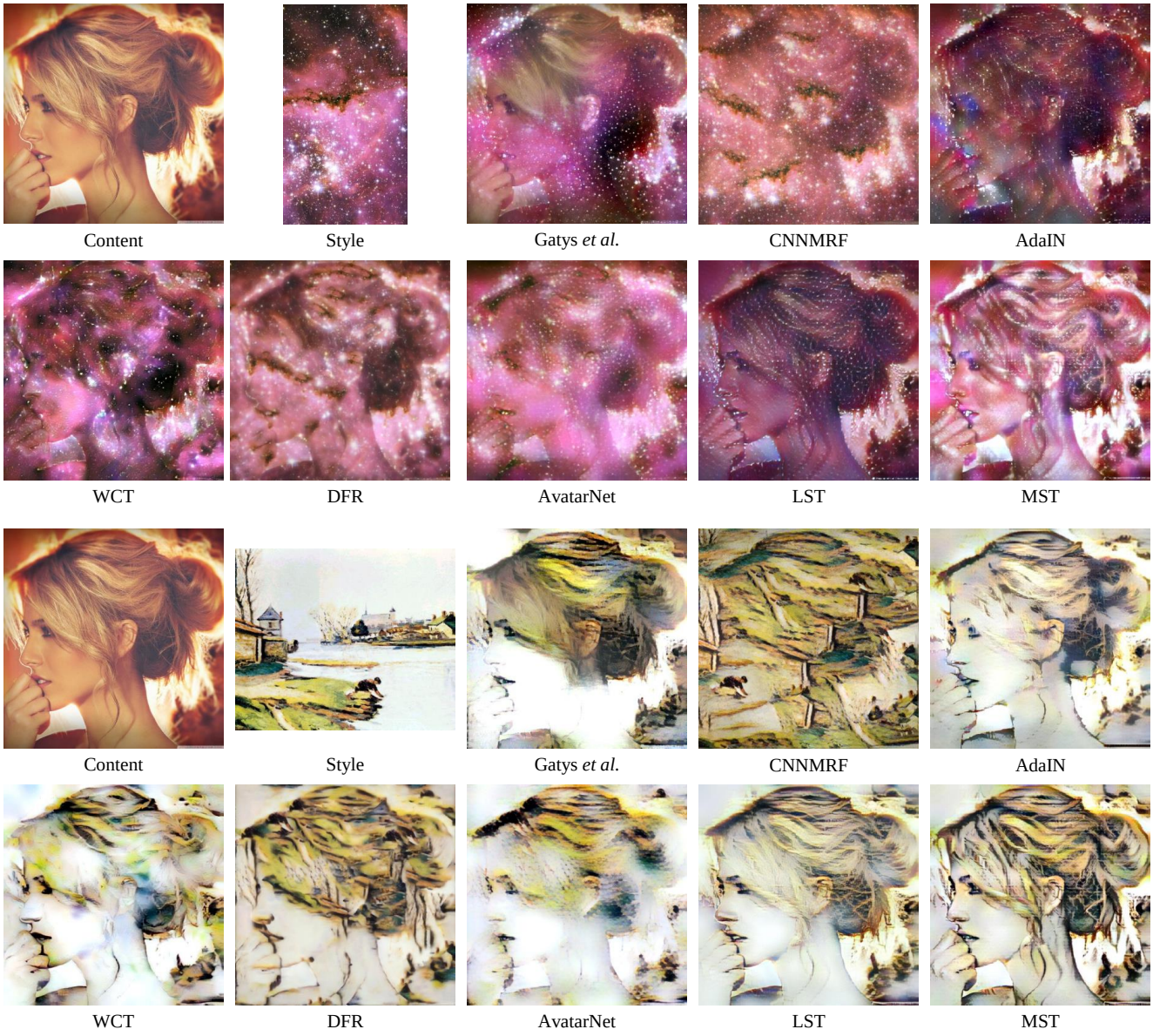


Figure 5: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. Method by Gatys *et al.* [1] would fall in local minimum. CNNMRF, AdaIN, WCT, DFR, AvatarNet, and LST may suffer from wash-out artifacts or/and content structure distortion.

Figure 6: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. Except for wash-out artifacts in 2nd comparison, some compared methods may also generate less desired strokes in the clean content background.

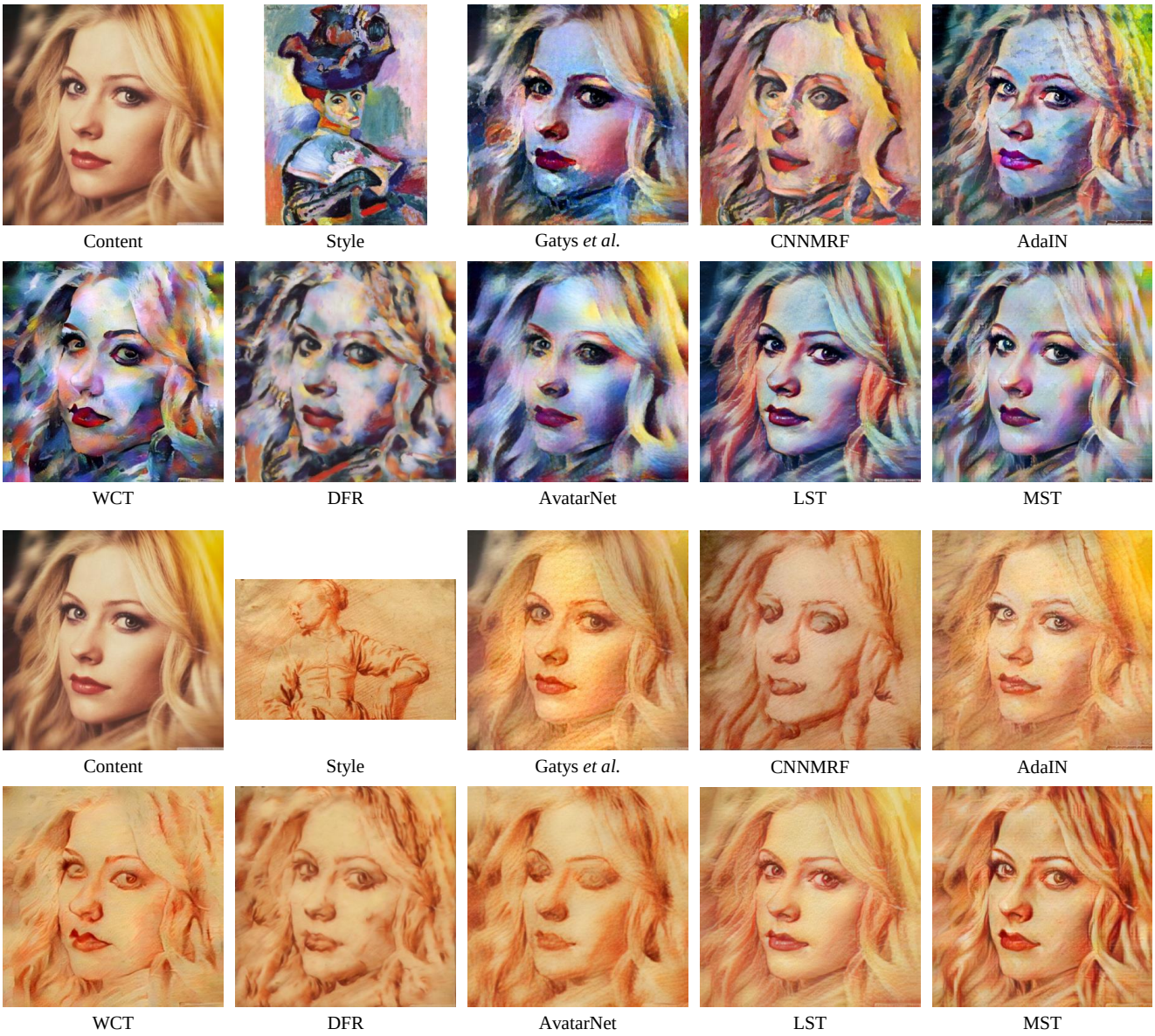


Figure 7: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. CNNMRF and AvatarNet generate unpleasing eyes. Our MST produces more visually pleasing faces.

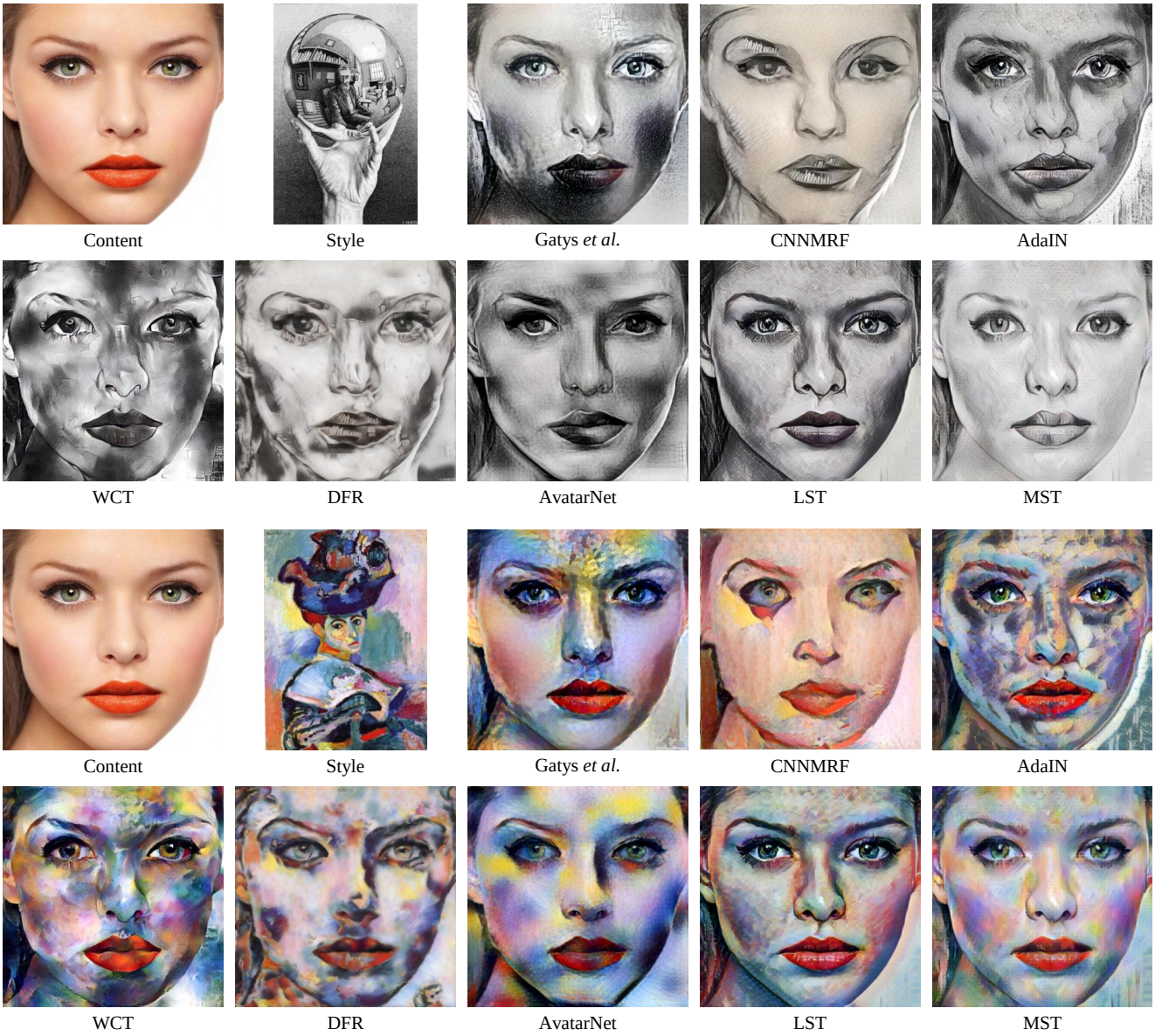


Figure 8: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. When content structures are relatively simple, compared methods cannot adaptively transfer style accordingly.

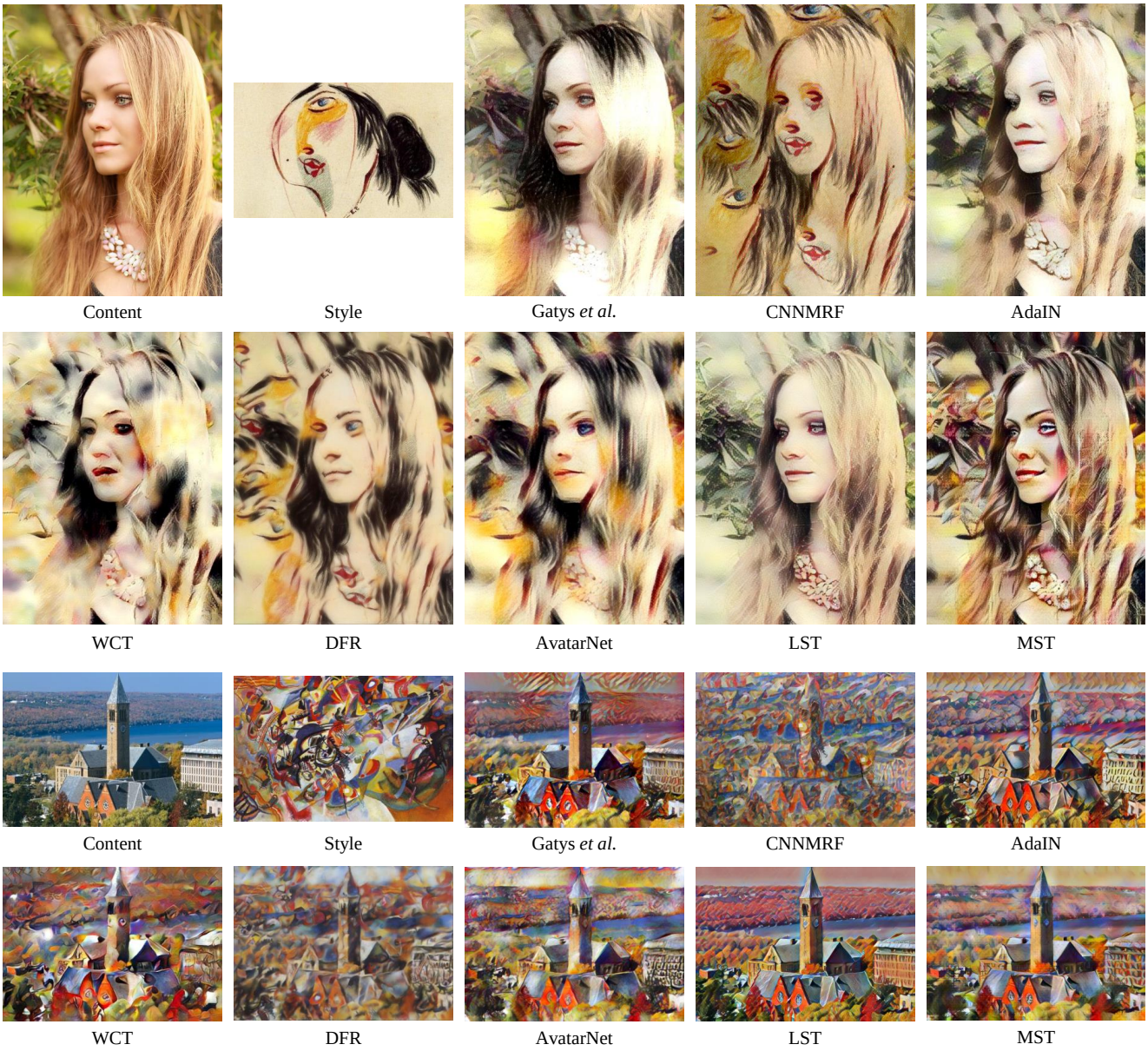


Figure 9: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. In addition to the wash-out artifacts in 1st comparison, they (except for LST) can hardly distinguish semantic content structures in the 2nd comparison.

Figure 10: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. In addition to the wash-out artifacts in 1st comparison, they can hardly distinguish semantic content structures in the 2nd comparison. It should also be noted that LST performs well in the second comparison of Fig. 9, but still suffers from wash-out artifact with another style image. However, our MST handles different style images better.

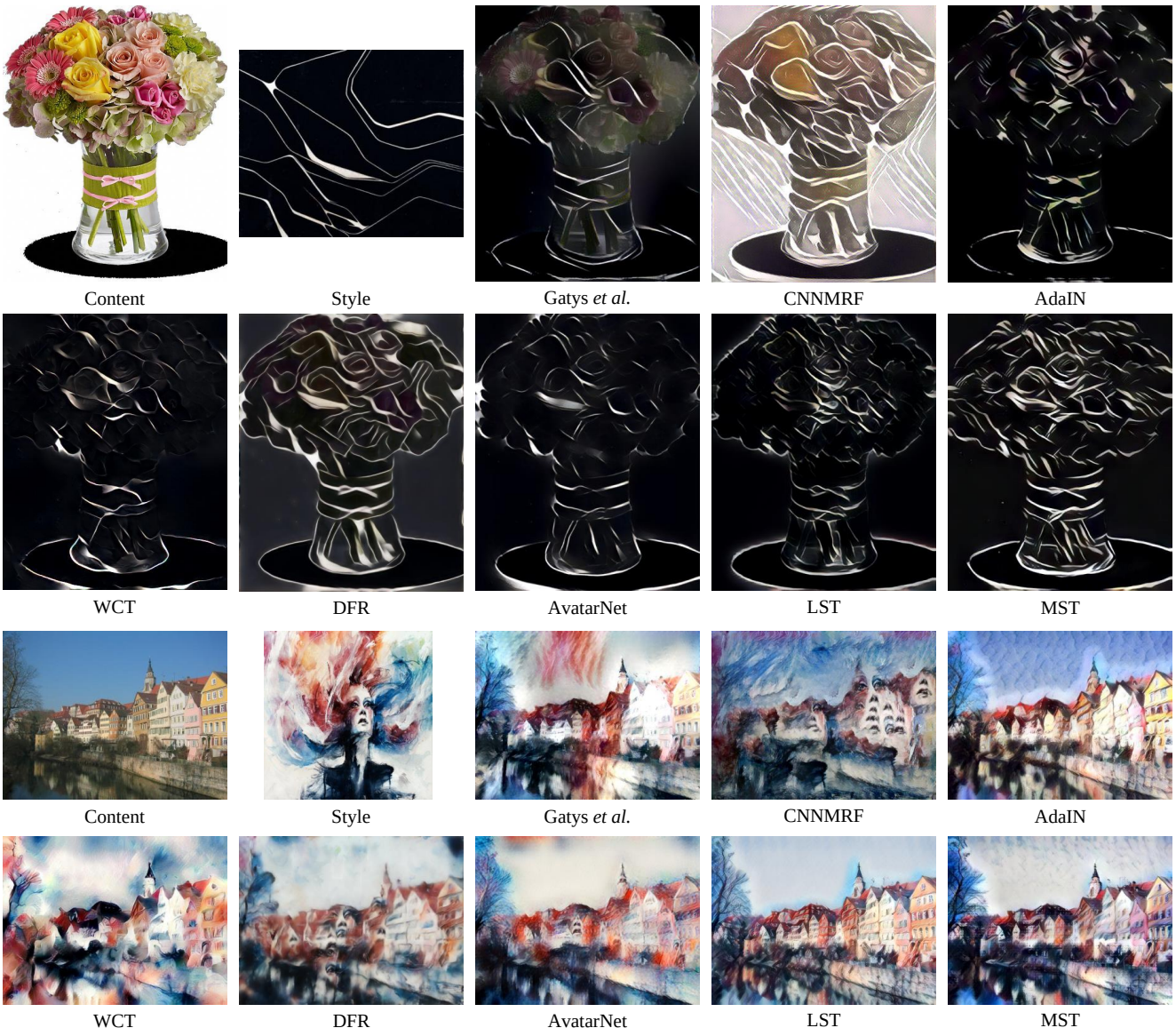


Figure 11: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. CNNMRF, DFR, and AvatarNet would copy some style patterns to the results (e.g., 2nd comparison), leading to less desired stylizations. Although LST keeps clean background in the second comparison, it also generates some halation around the building.

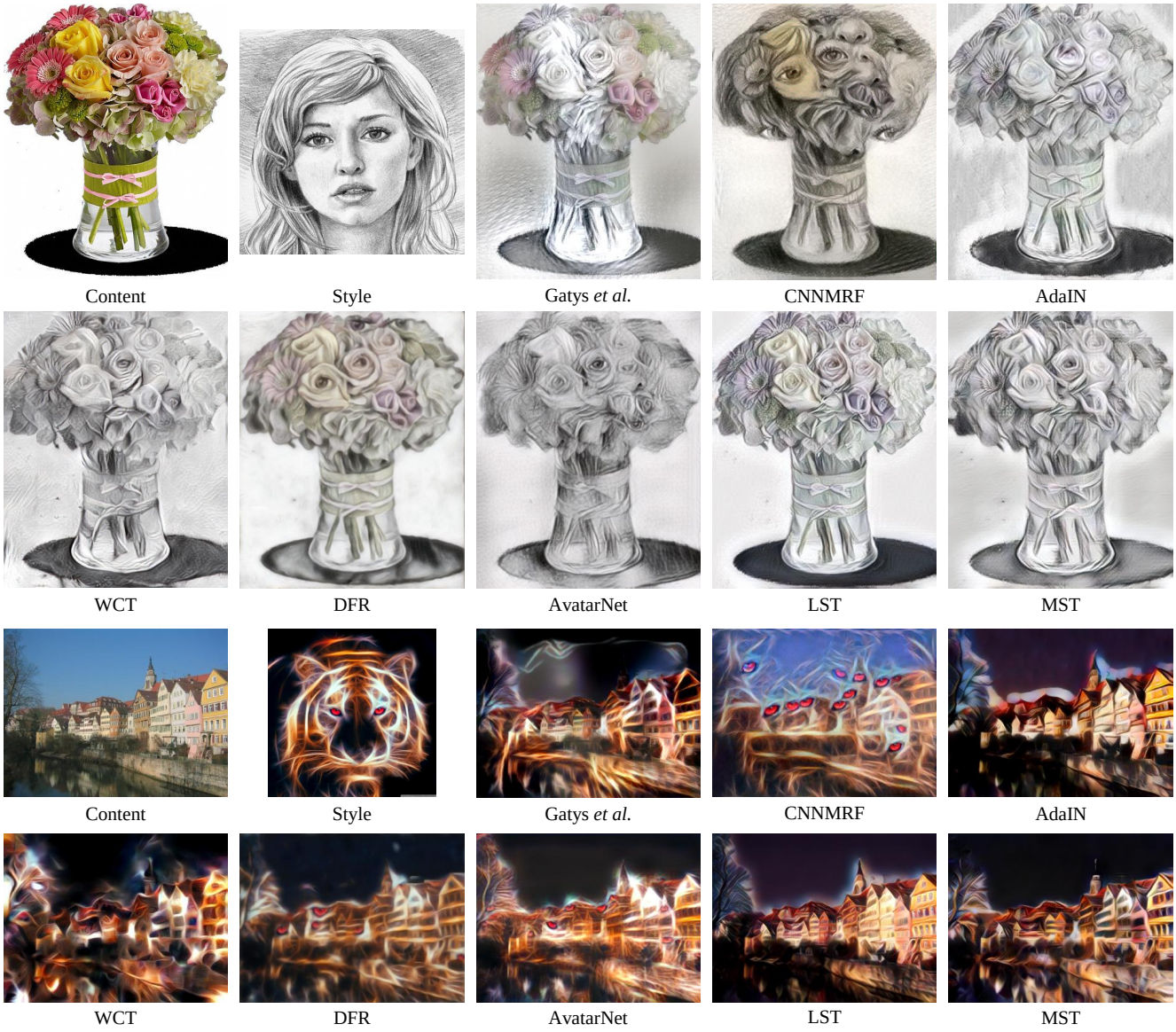


Figure 12: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. CNNMRF, DFR, and AvatarNet would copy some style patterns to the results (e.g., human eyes in the 1st comparison and tiger eyes in the 2nd comparison), leading to unpleasing stylizations.

Figure 13: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. CNNMRF, DFR, and AvatarNet would copy some style patterns to the results (e.g., tiger eyes in the 1st comparison), leading to unpleasing stylizations. In the 2nd comparison, MST generates results, being more faithful to the content structures.

Figure 14: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. In the 1st comparison, all the compared methods cannot distinguish the content background. In the 2nd comparison, CNNMRF, DFR, and AvatarNet copy the girl eyes from style image to the results.

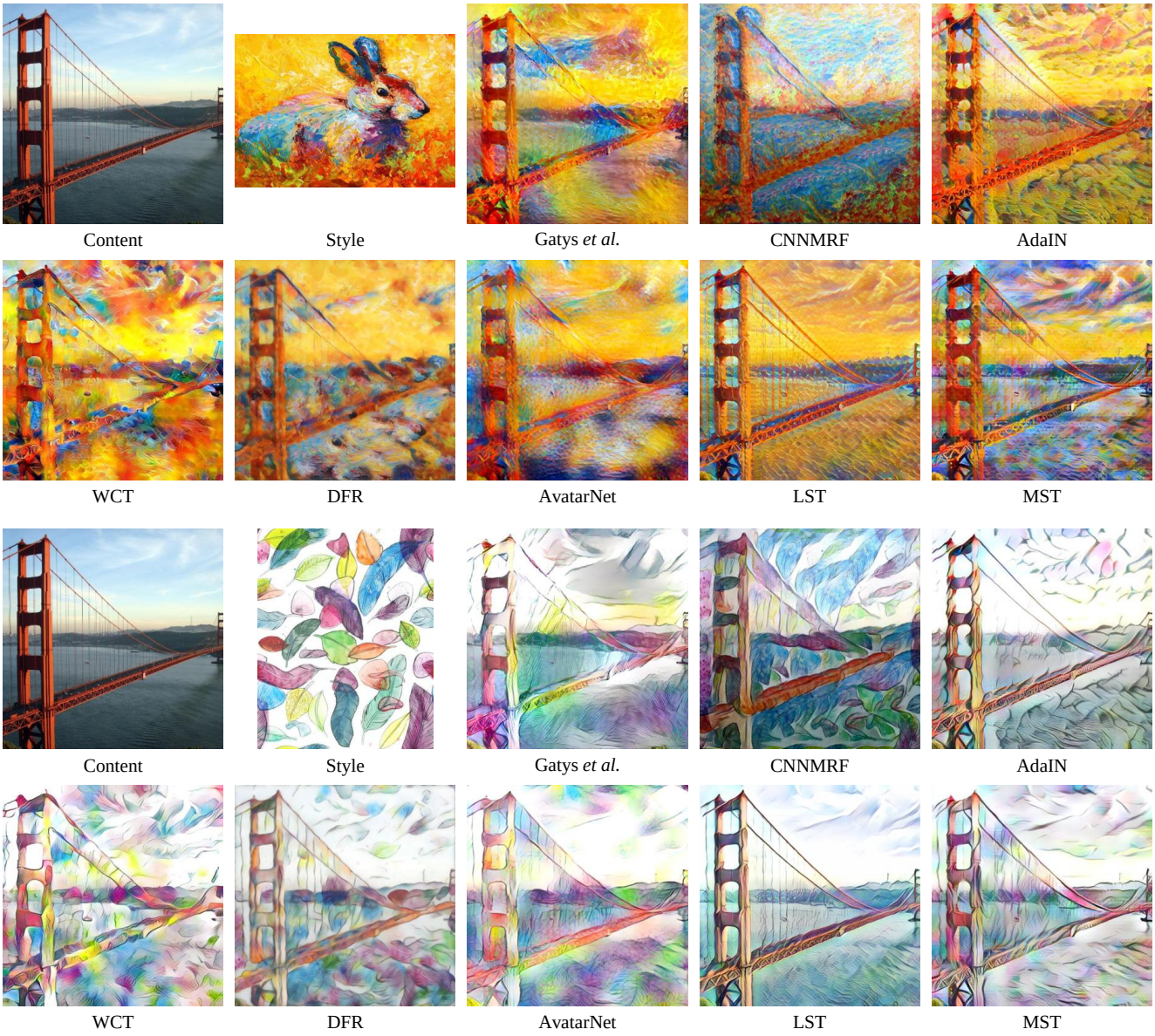


Figure 15: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. In the 1st comparison, MST reflects more semantic matching between content and style structures. In the 2nd comparison, MST preserves more structures, being more faithful to content structures.

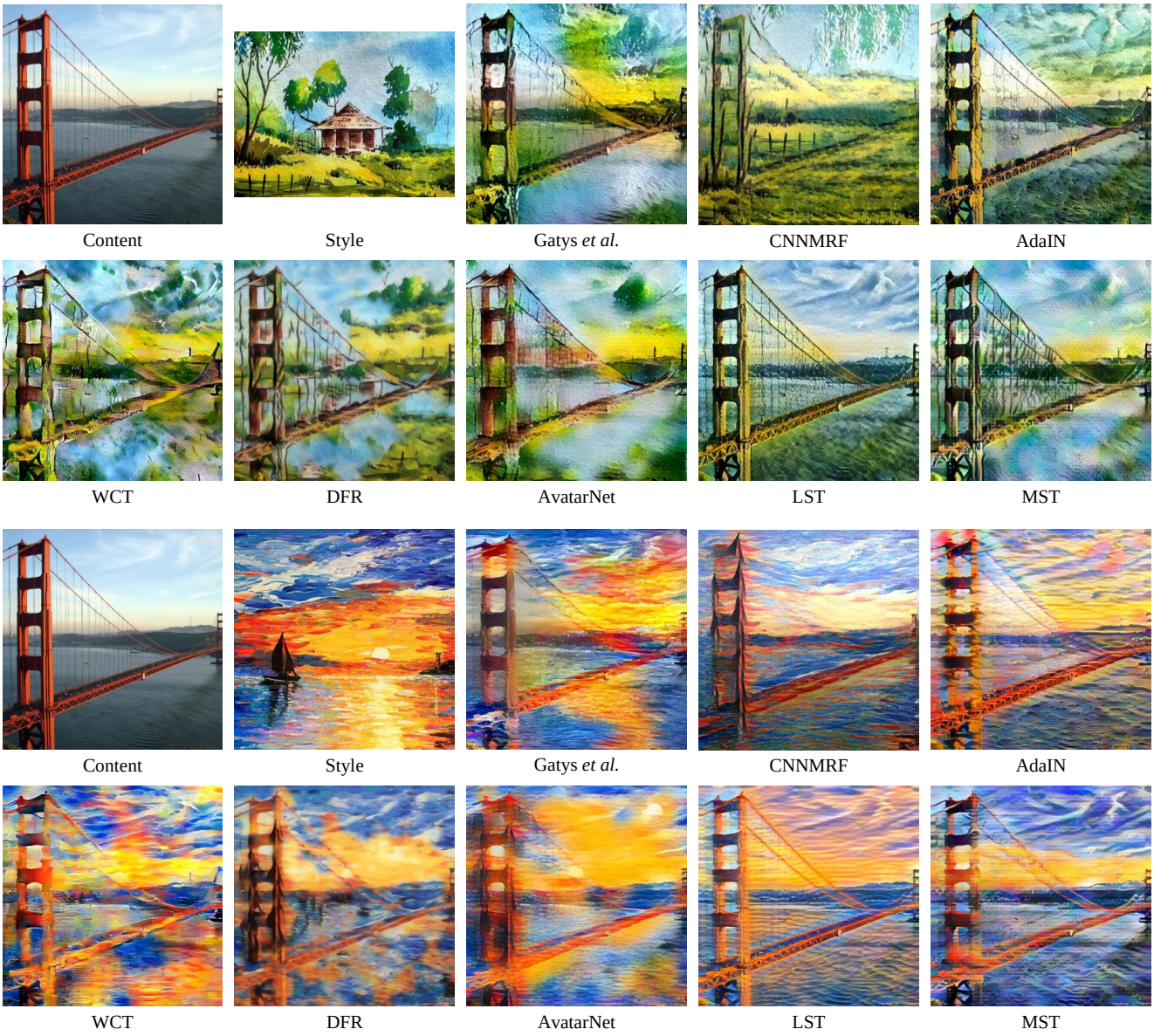


Figure 16: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. MST reflects more semantic matching between content and style structures.

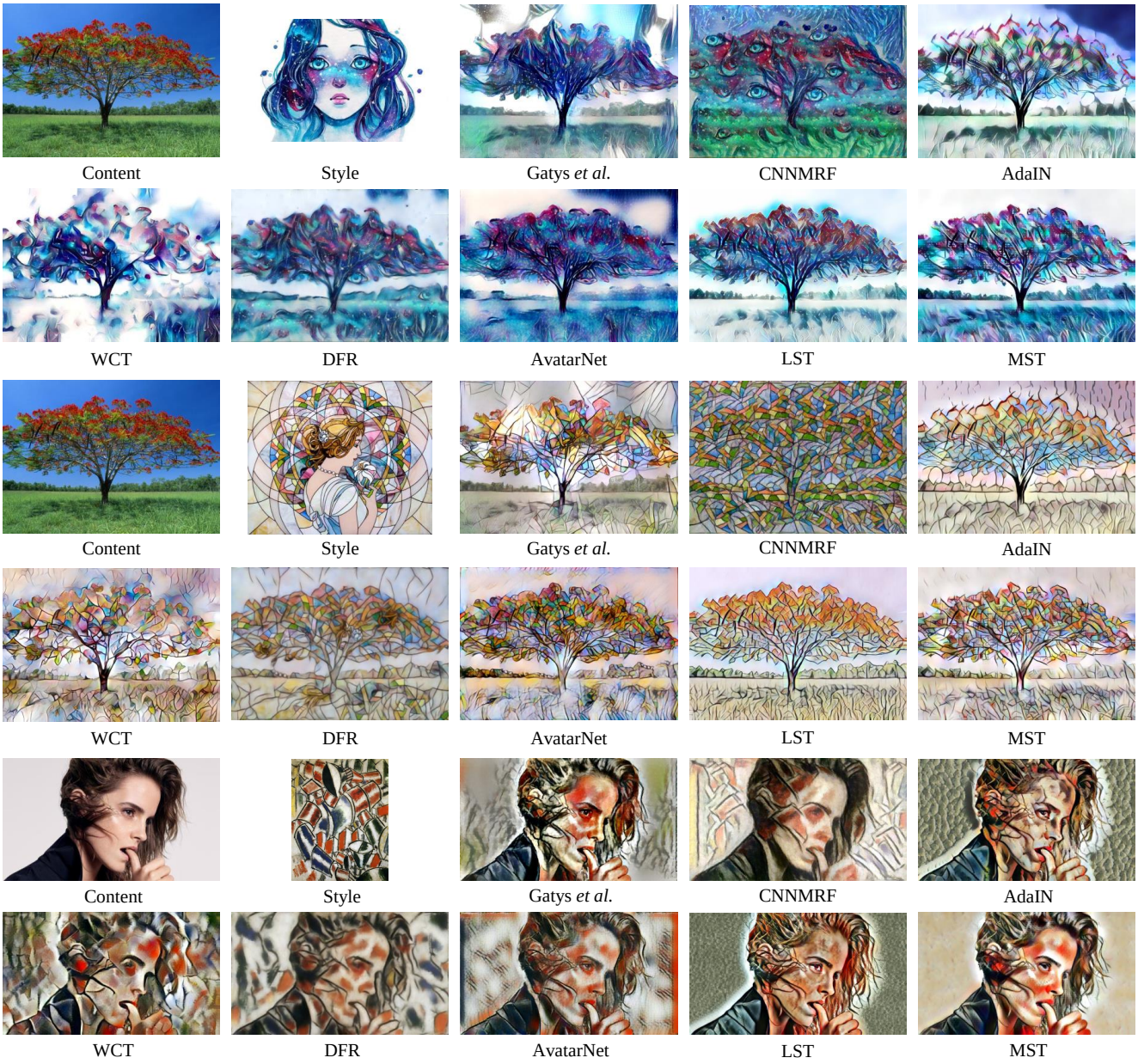


Figure 17: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. MST reflects more semantic matching between content and style structures.

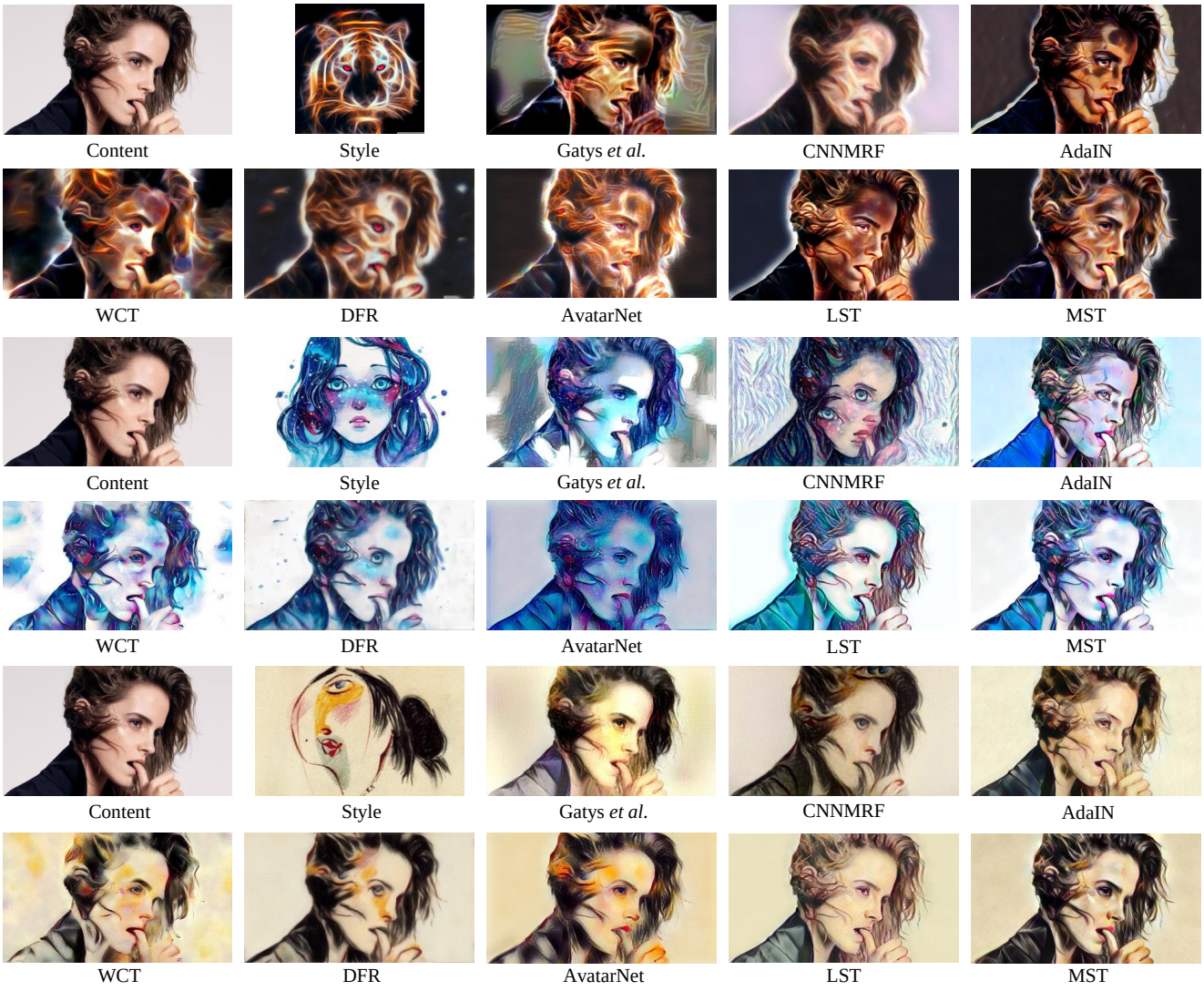


Figure 18: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. CNNMRF, DFR, and AvatarNet copy style patterns to the results.

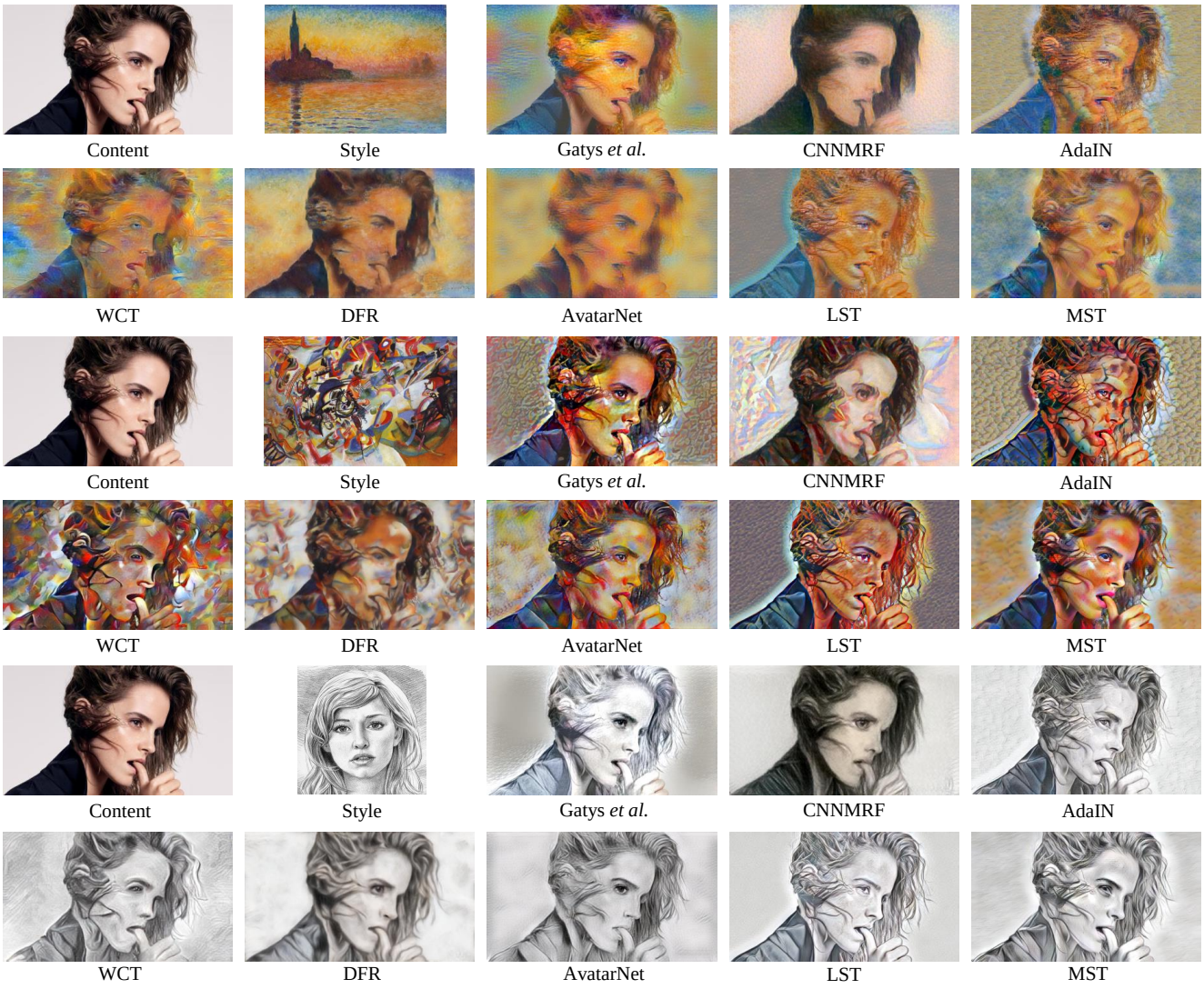
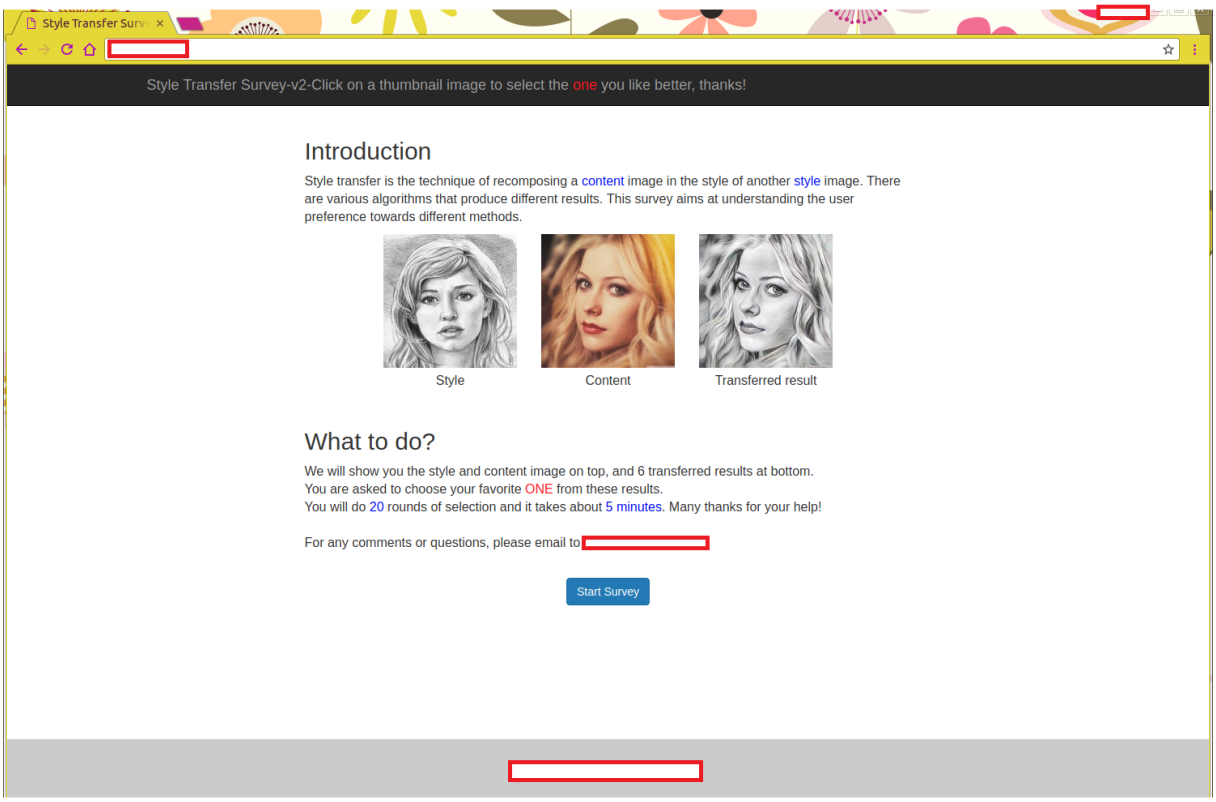


Figure 19: Visual comparison with method by Gatys *et al.* [1], CNNMRF [4], AdaIN [3], WCT [6], Deep feature reshuffle (DFR) [2], AvatarNet [8], and LST [5]. We set $K = 3$ in MST. WCT suffers from distortion heavily.



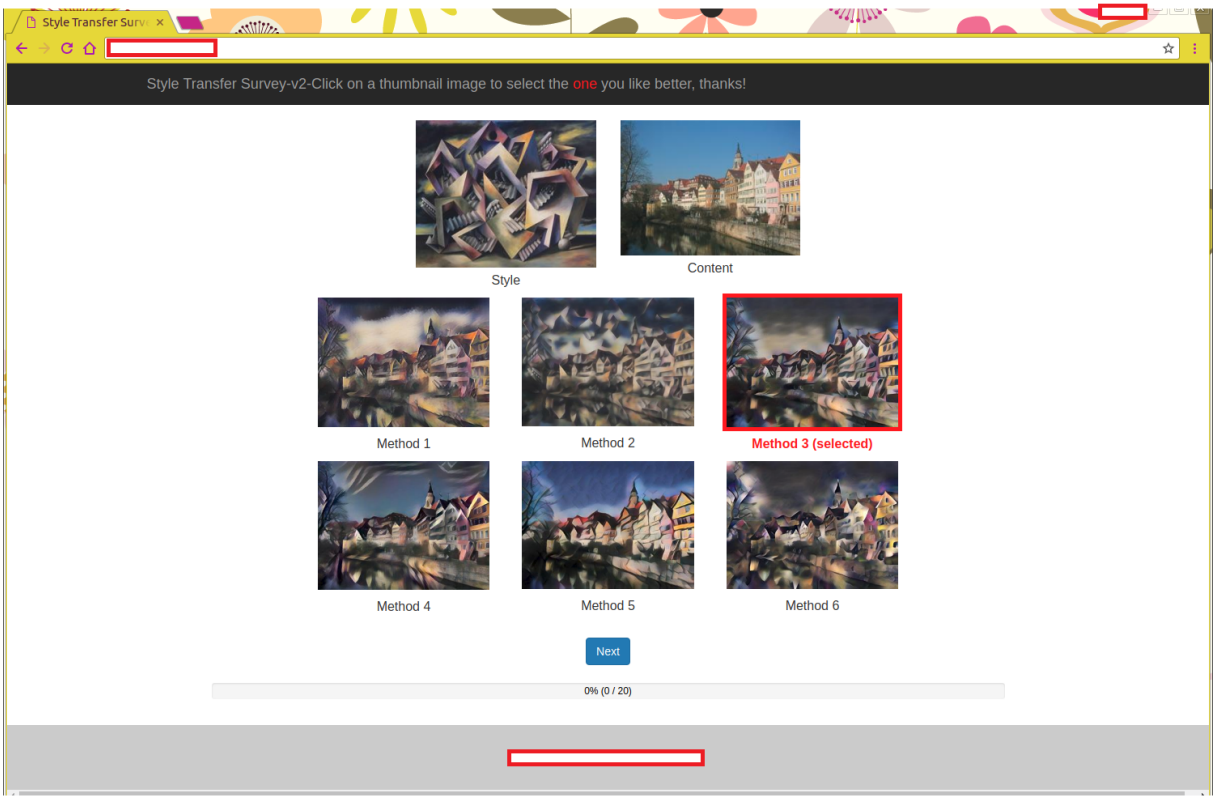
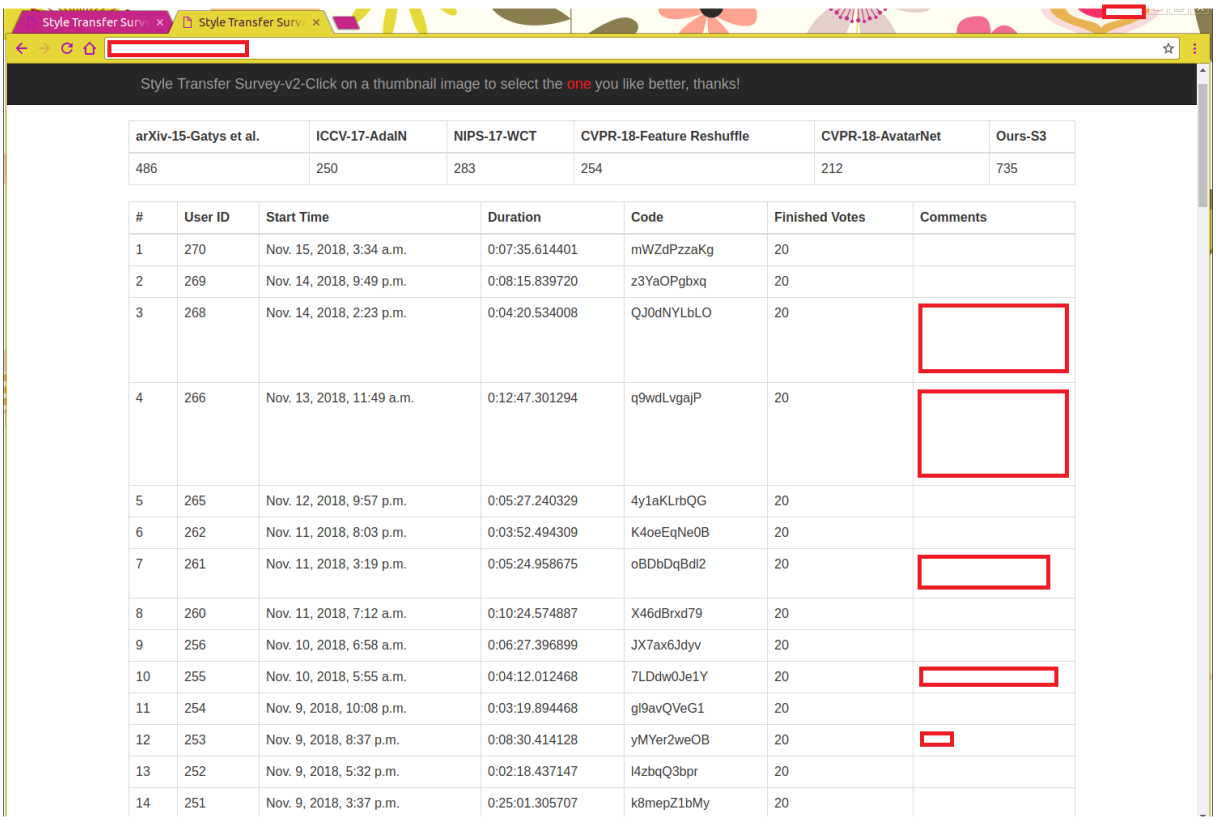
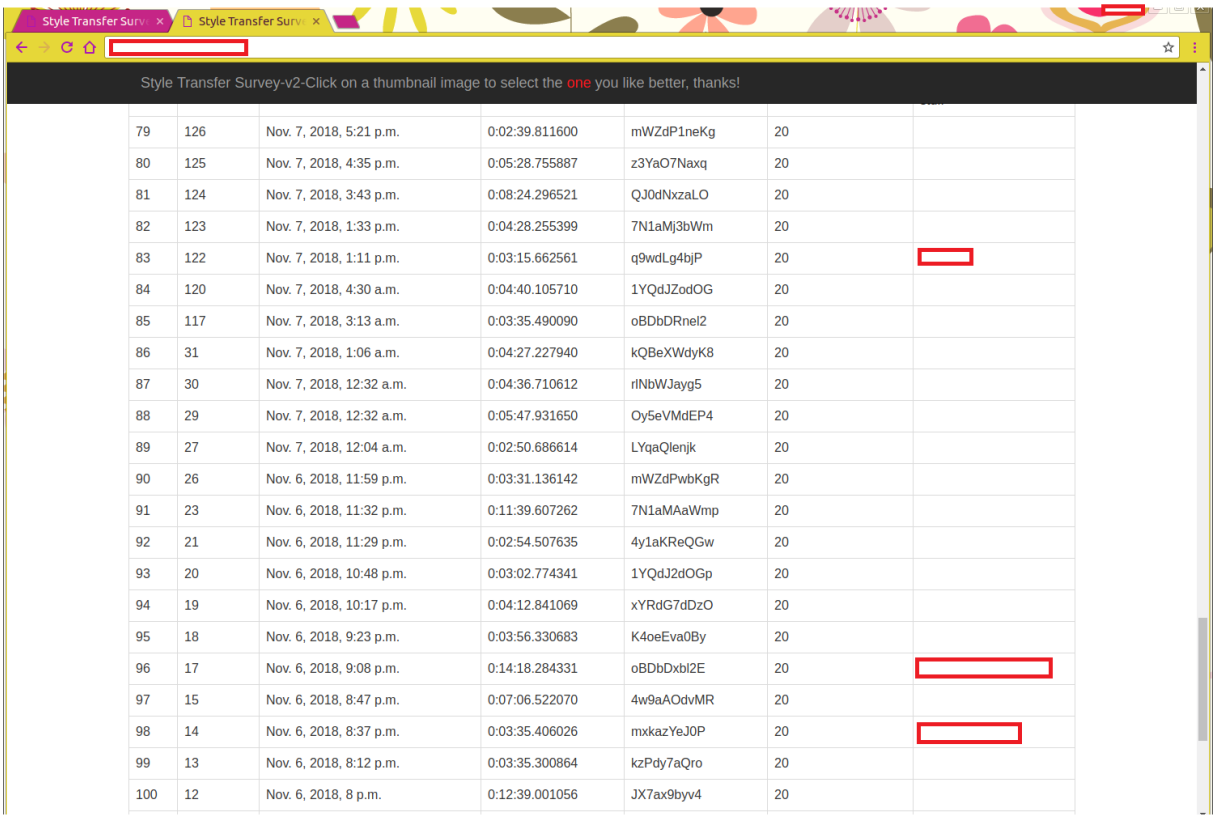


Figure 20: Details of user study. First row: the start page for the users, who're guided how to conduct this survey. Second row: one randomly selected content-style pair. Each user will select one result that (s)he likes the best. Each user would finish 20 random content-style pairs, providing 20 votes.



arXiv-15-Gatys et al.	ICCV-17-AdaIN	NIPS-17-WCT	CVPR-18-Feature Reshuffle	CVPR-18-AvatarNet	Ours-S3
486	250	283	254	212	735

#	User ID	Start Time	Duration	Code	Finished Votes	Comments
1	270	Nov. 15, 2018, 3:34 a.m.	0:07:35.614401	mWZdPzzaKg	20	
2	269	Nov. 14, 2018, 9:49 p.m.	0:08:15.839720	z3YaOPgbxq	20	
3	268	Nov. 14, 2018, 2:23 p.m.	0:04:20.534008	QJ0dNYLbLO	20	
4	266	Nov. 13, 2018, 11:49 a.m.	0:12:47.301294	q9wdLvgajP	20	
5	265	Nov. 12, 2018, 9:57 p.m.	0:05:27.240329	4y1aKLrbQG	20	
6	262	Nov. 11, 2018, 8:03 p.m.	0:03:52.494309	K4oeEqNe0B	20	
7	261	Nov. 11, 2018, 3:19 p.m.	0:05:24.958675	oBDbDqBdl2	20	
8	260	Nov. 11, 2018, 7:12 a.m.	0:10:24.574887	X46dBrxd79	20	
9	256	Nov. 10, 2018, 6:58 a.m.	0:06:27.396899	JX7ax6Jdyv	20	
10	255	Nov. 10, 2018, 5:55 a.m.	0:04:12.012468	7LDdw0Je1Y	20	
11	254	Nov. 9, 2018, 10:08 p.m.	0:03:19.894468	gl9avQVeG1	20	
12	253	Nov. 9, 2018, 8:37 p.m.	0:08:30.414128	yMYer2weOB	20	
13	252	Nov. 9, 2018, 5:32 p.m.	0:02:18.437147	l4zbqQ3bpr	20	
14	251	Nov. 9, 2018, 3:37 p.m.	0:25:01.305707	k8mepZ1bMy	20	



79	126	Nov. 7, 2018, 5:21 p.m.	0:02:39.811600	mWZdP1neKg	20	
80	125	Nov. 7, 2018, 4:35 p.m.	0:05:28.755887	z3YaO7Naxq	20	
81	124	Nov. 7, 2018, 3:43 p.m.	0:08:24.296521	QJ0dNxzaLO	20	
82	123	Nov. 7, 2018, 1:33 p.m.	0:04:28.255399	7N1aMj3bWm	20	
83	122	Nov. 7, 2018, 1:11 p.m.	0:03:15.662561	q9wdLg4bjP	20	
84	120	Nov. 7, 2018, 4:30 a.m.	0:04:40.105710	1YQdJZodOG	20	
85	117	Nov. 7, 2018, 3:13 a.m.	0:03:35.490090	oBDbDRnel2	20	
86	31	Nov. 7, 2018, 1:06 a.m.	0:04:27.227940	kQBeXWdyK8	20	
87	30	Nov. 7, 2018, 12:32 a.m.	0:04:36.710612	rlNbWJayg5	20	
88	29	Nov. 7, 2018, 12:32 a.m.	0:05:47.931650	Oy5eVMdEP4	20	
89	27	Nov. 7, 2018, 12:04 a.m.	0:02:50.686614	LYqaQlenjk	20	
90	26	Nov. 6, 2018, 11:59 p.m.	0:03:31.136142	mWZdPwbKgR	20	
91	23	Nov. 6, 2018, 11:32 p.m.	0:11:39.607262	7N1aMAaWmp	20	
92	21	Nov. 6, 2018, 11:29 p.m.	0:02:54.507635	4y1aKReQGw	20	
93	20	Nov. 6, 2018, 10:48 p.m.	0:03:02.774341	1YQdJ2dOGp	20	
94	19	Nov. 6, 2018, 10:17 p.m.	0:04:12.841069	xYRdG7dDzO	20	
95	18	Nov. 6, 2018, 9:23 p.m.	0:03:56.330683	K4oeEva0By	20	
96	17	Nov. 6, 2018, 9:08 p.m.	0:14:18.284331	oBDbDxbl2E	20	
97	15	Nov. 6, 2018, 8:47 p.m.	0:07:06.522070	4w9aAOdvMR	20	
98	14	Nov. 6, 2018, 8:37 p.m.	0:03:35.406026	mxkazYeJ0P	20	
99	13	Nov. 6, 2018, 8:12 p.m.	0:03:35.300864	kzPdy7aQro	20	
100	12	Nov. 6, 2018, 8 p.m.	0:12:39.001056	JX7ax9byv4	20	

Figure 21: Details of user study. These two rows show the votes that each method received from the users. 'Ours-S3' means our proposed MST-3. In fact, we received 2220 votes from 111 users in total. We only use the first 2000 votes to report our user study, because some users submitted their surveys very late.

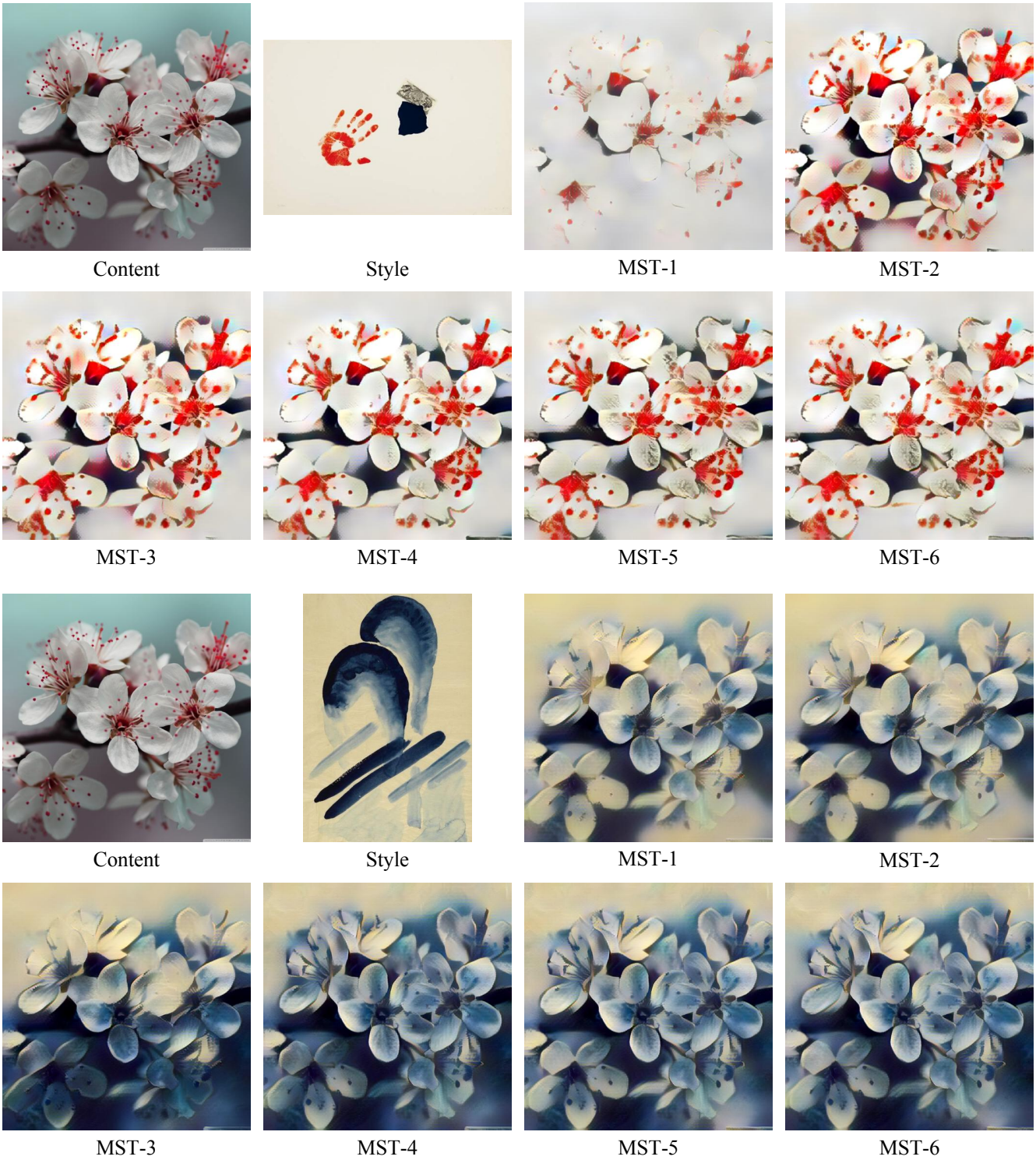


Figure 22: Style number cluster number investigation. MST-K means style features are clustered into K clusters. As we can see, for simple style, $K = 1$ would suffer from wash-out artifacts to some degree. As we enlarge K, our MST can match each content feature pixel with better style cluster and alleviate the wash-out artifacts.

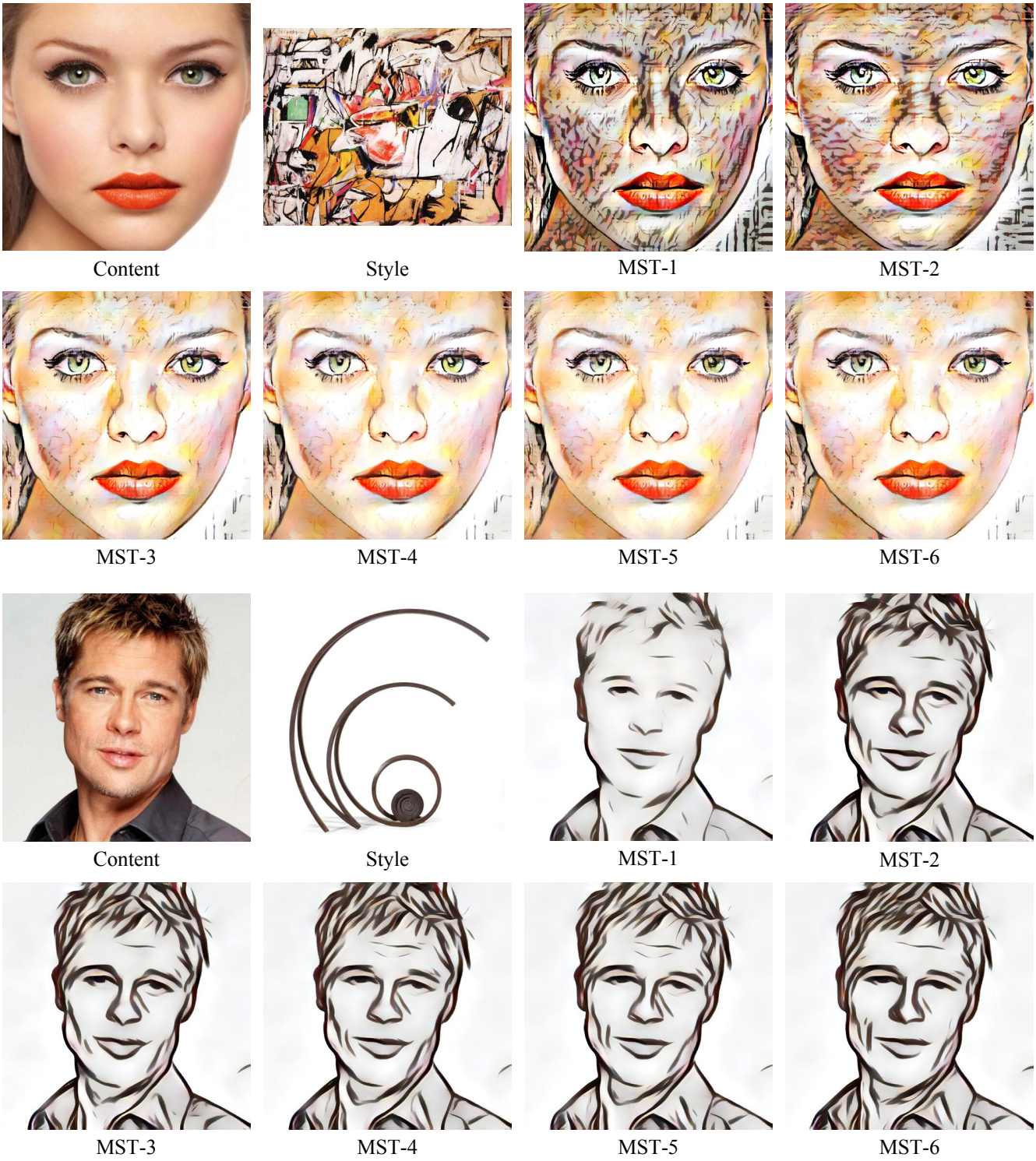


Figure 23: Style number cluster number investigation. MST-K means style features are clustered into K clusters. As we can see, for simple style, $K = 1$ would suffer from wash-out artifacts to some degree. As we enlarge K, our MST can match each content feature pixel with better style cluster and alleviate the wash-out artifacts.

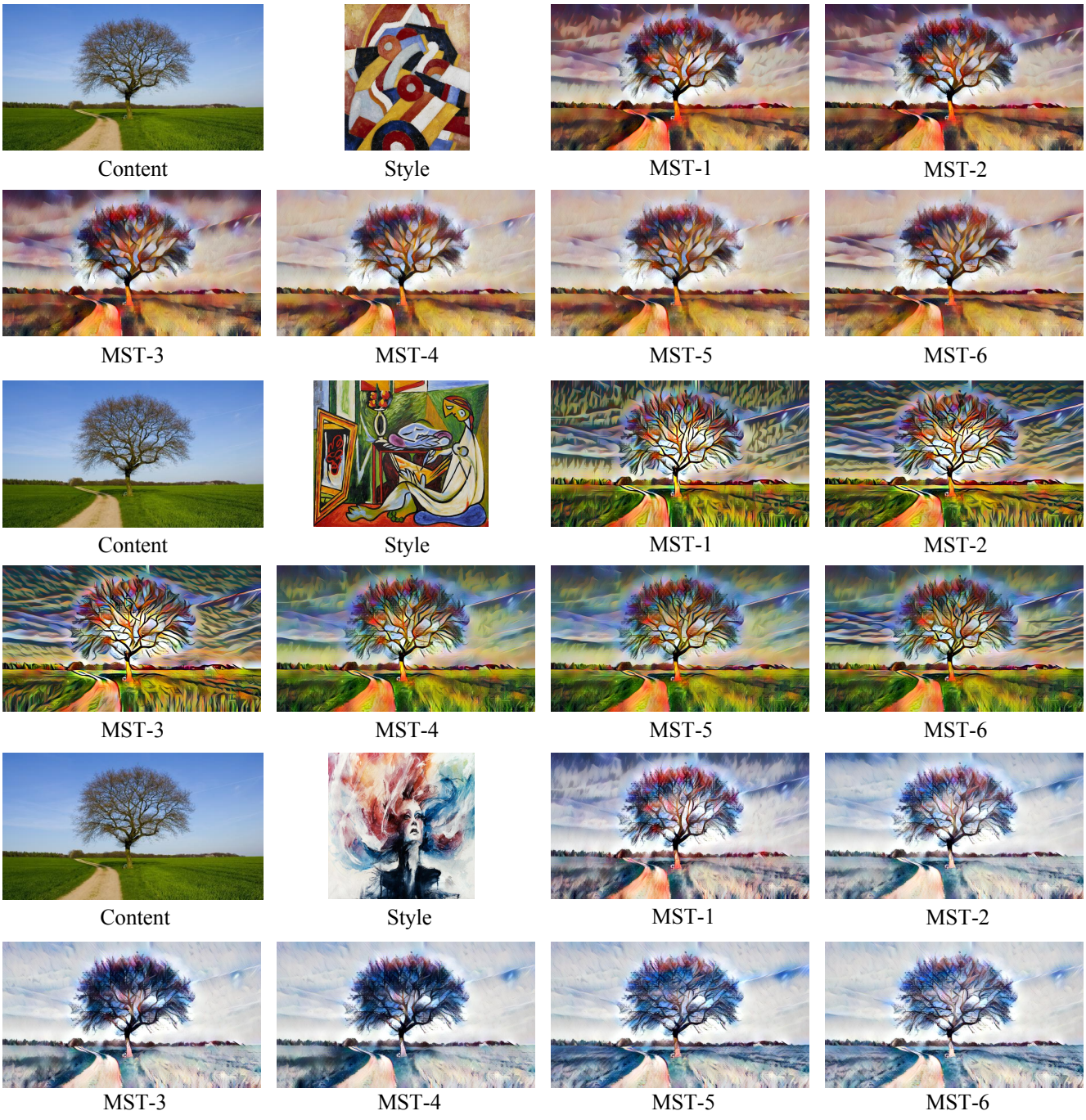


Figure 24: Style number cluster number investigation. MST-K means style features are clustered into K clusters. As we can see, for simple style, $K = 1$ would suffer from wash-out artifacts to some degree. As we enlarge K, our MST can match each content feature pixel with better style cluster and alleviate the wash-out artifacts.

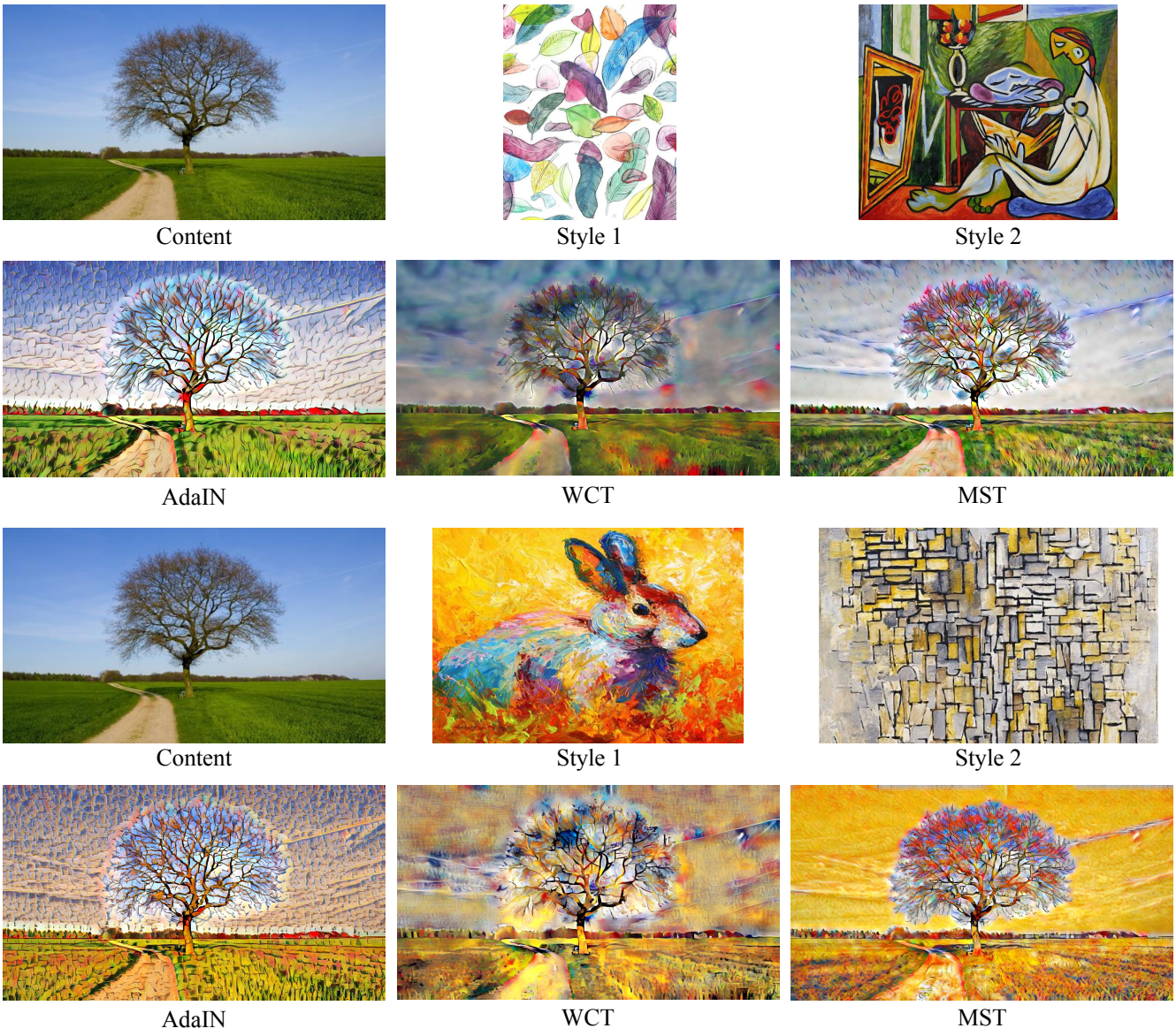
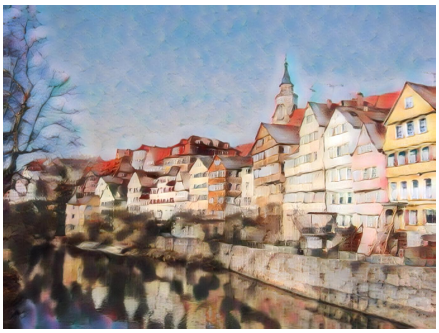


Figure 25: Multi-style transfer. We set $K = 3$ in MST. Our MST treats patterns from different style images distinctively and transfer them adaptively according to the specific content structures.

Content

Style 1

Style 2



AdaIN

WCT

MST

Content

Style 1

Style 2

AdaIN

WCT

MST

Figure 26: Multi-style transfer. We set $K = 3$ in MST. Our MST treats patterns from different style images distinctively and transfer them adaptively according to the specific content structures.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 1, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[2] S. Gu, C. Chen, J. Liao, and L. Yuan. Arbitrary style transfer with deep feature reshuffle. In *CVPR*, 2018. 1, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[3] X. Huang and S. J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 1, 2, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[4] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *CVPR*, 2016. 1, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[5] X. Li, S. Liu, J. Kautz, and M.-H. Yang. Learning linear transformations for fast arbitrary style transfer. In *CVPR*, 2019. 1, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[6] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. In *NIPS*, 2017. 1, 2, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[7] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 2008. 1, 3

[8] L. Sheng, Z. Lin, J. Shao, and X. Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *CVPR*, 2018. 1, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1