

Supplemental Material for “Visual Attribute Transfer through Deep Image Analogy”

JING LIAO, Microsoft Research Asia
YUAN YAO, Shanghai Jiao Tong University
LU YUAN, Microsoft Research Asia
GANG HUA, Microsoft Research Asia
SING BING KANG, Microsoft Research

Contents

1	Applications	3
1.1	Photo to style	3
1.2	Style to style	10
1.3	Style to photo	12
1.4	Photo to photo	14
2	Evaluations	16
2.1	Category 1: the same scene with varied views and motions	16
2.2	Category 2: the same scene, with large variations in views, colors and tones .	18
2.3	Category 3: semantically-related scene with vastly different styles	20
3	Limitations	22

1 Applications

1.1 Photo to style

In this application, our method transfers a photo to a reference artistic style. Users can easily borrow specific artwork stylization to render their own photos for sharing and entertainment.

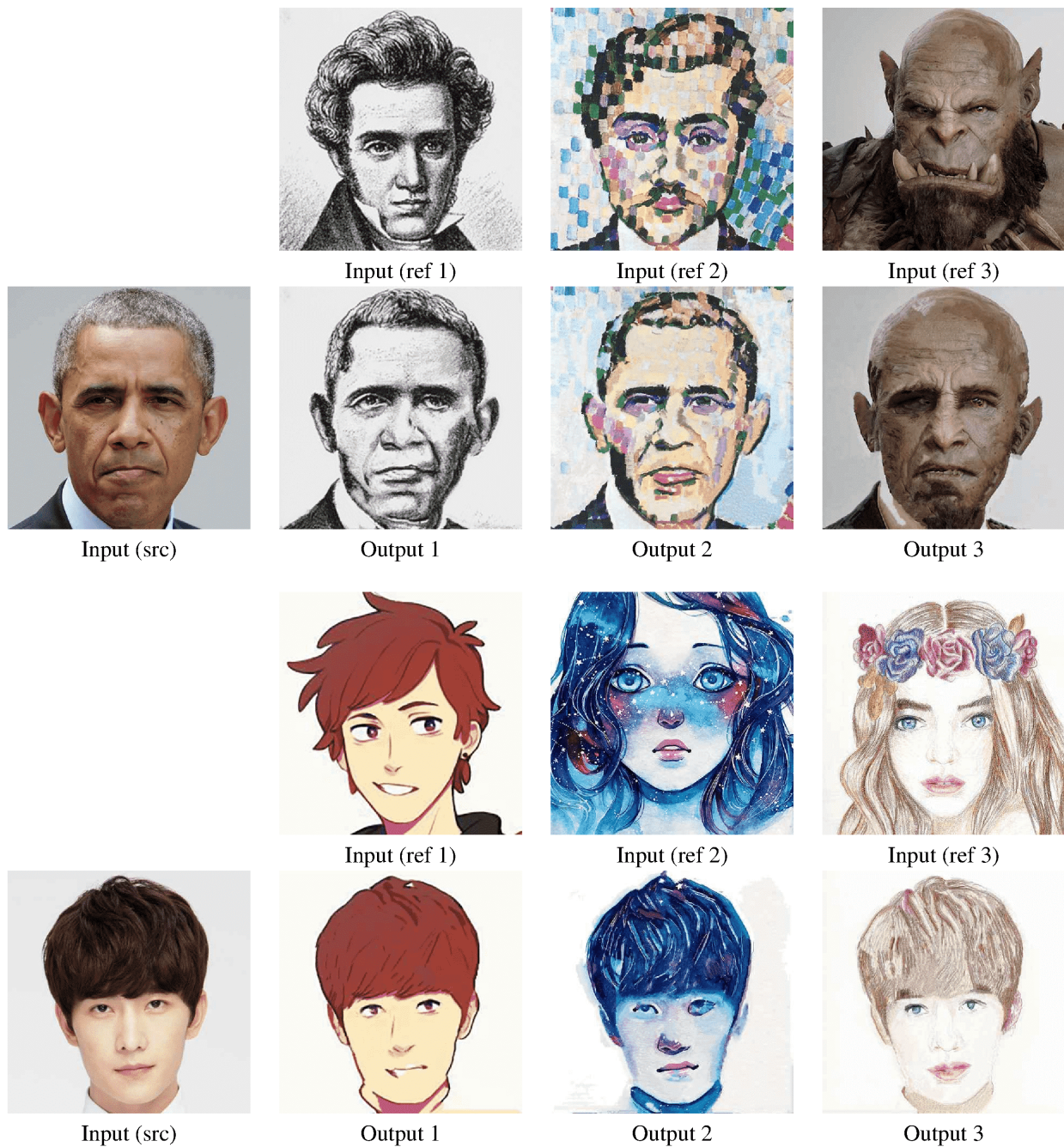


Figure 1: Photo-to-style transfer results.

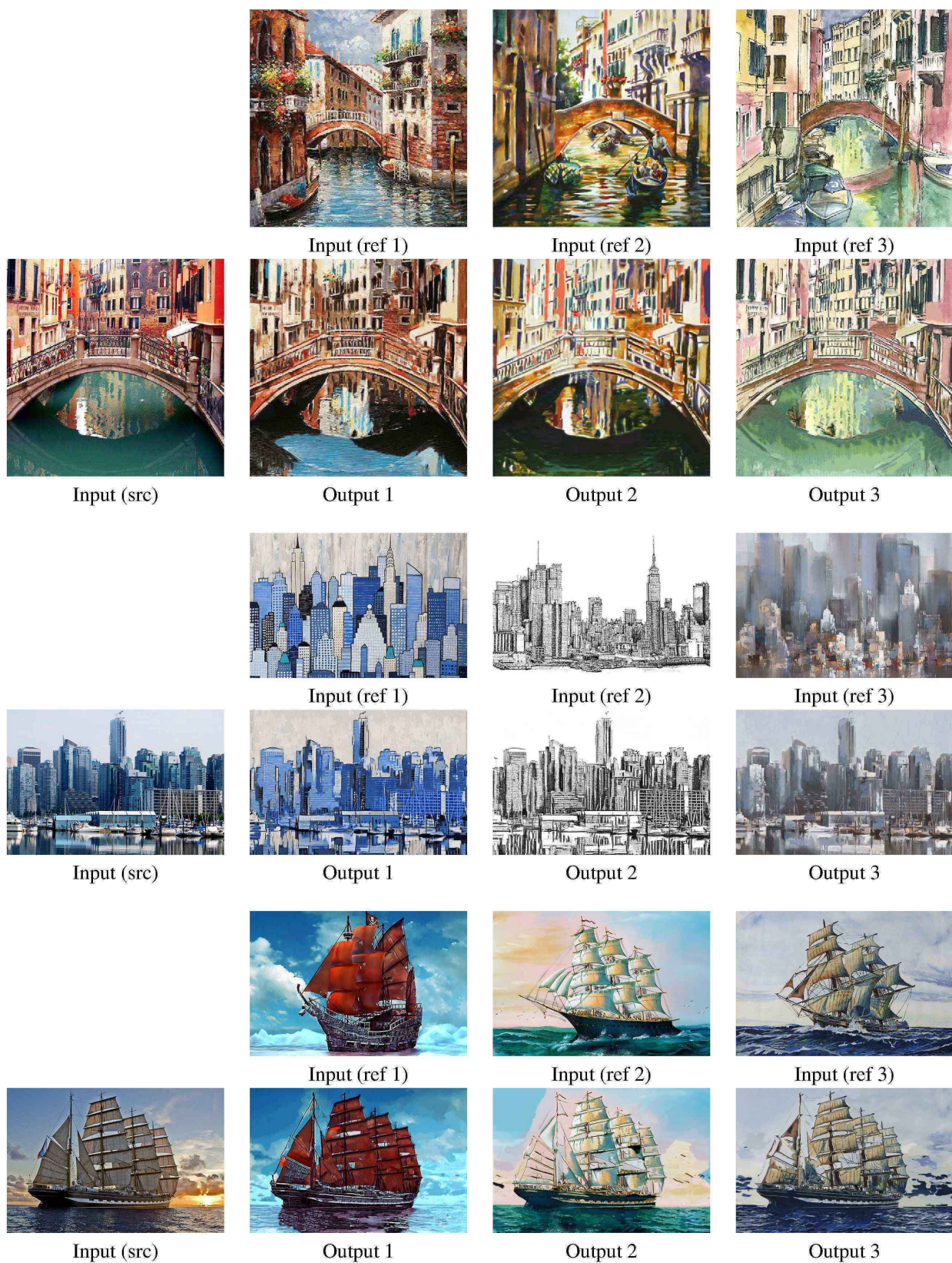


Figure 2: Photo-to-style transfer results.

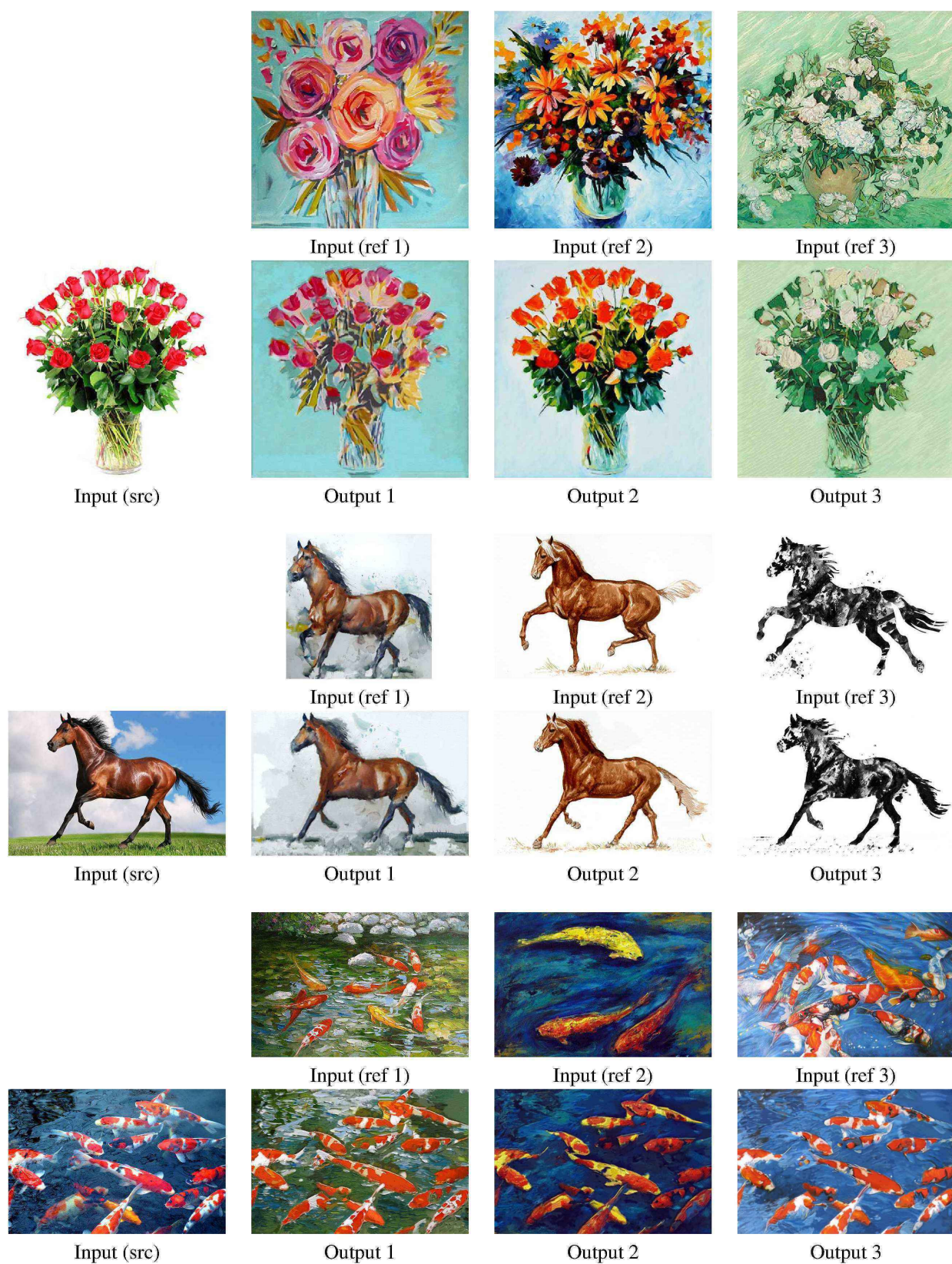


Figure 3: Photo-to-style transfer results.

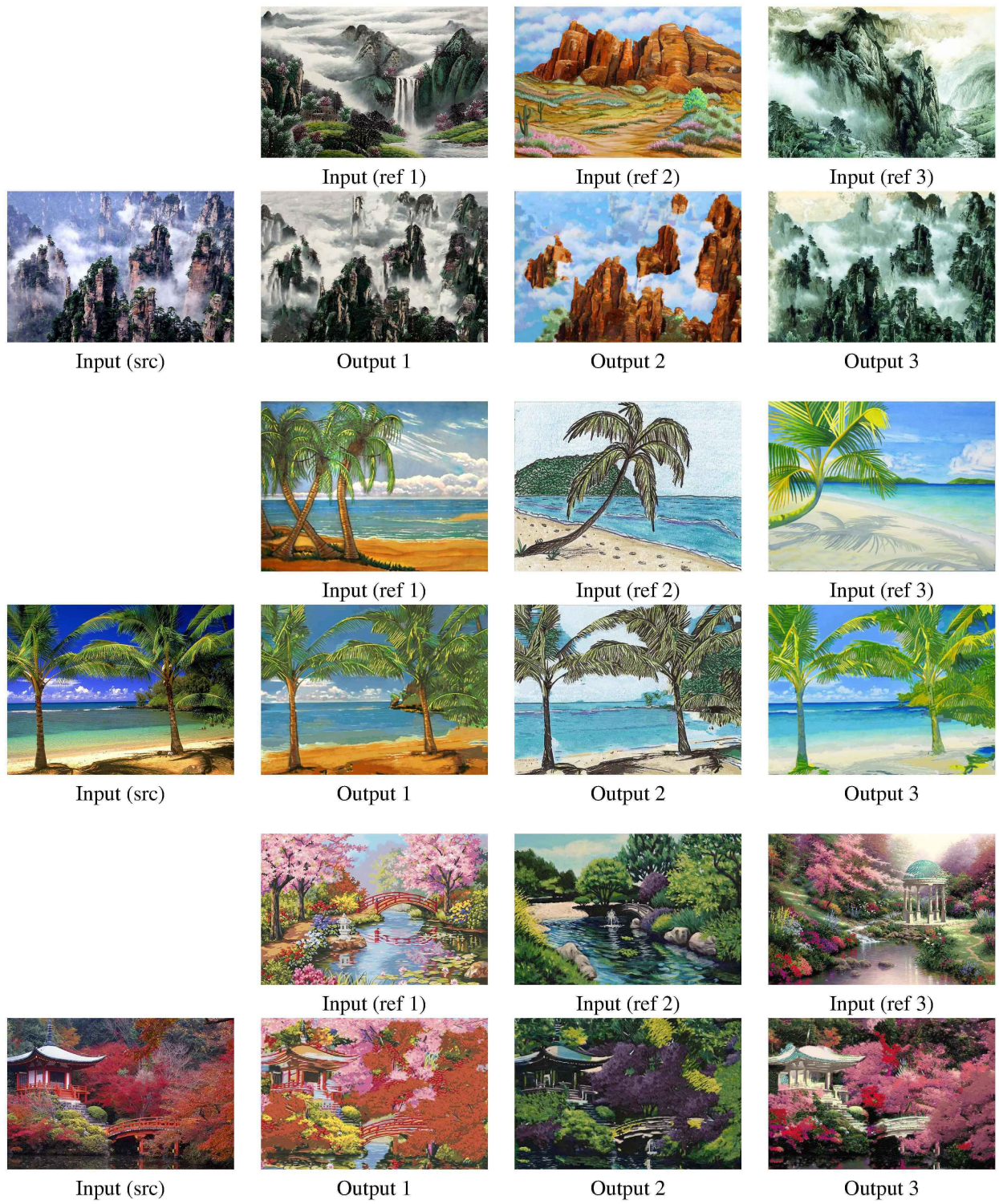


Figure 4: Photo-to-style transfer results.

Here we compare our results with other stylization methods or apps based on neural networks. our approach is capable of higher quality content-specific stylization that better preserve structures. Ours also better reproduce the reference styles.

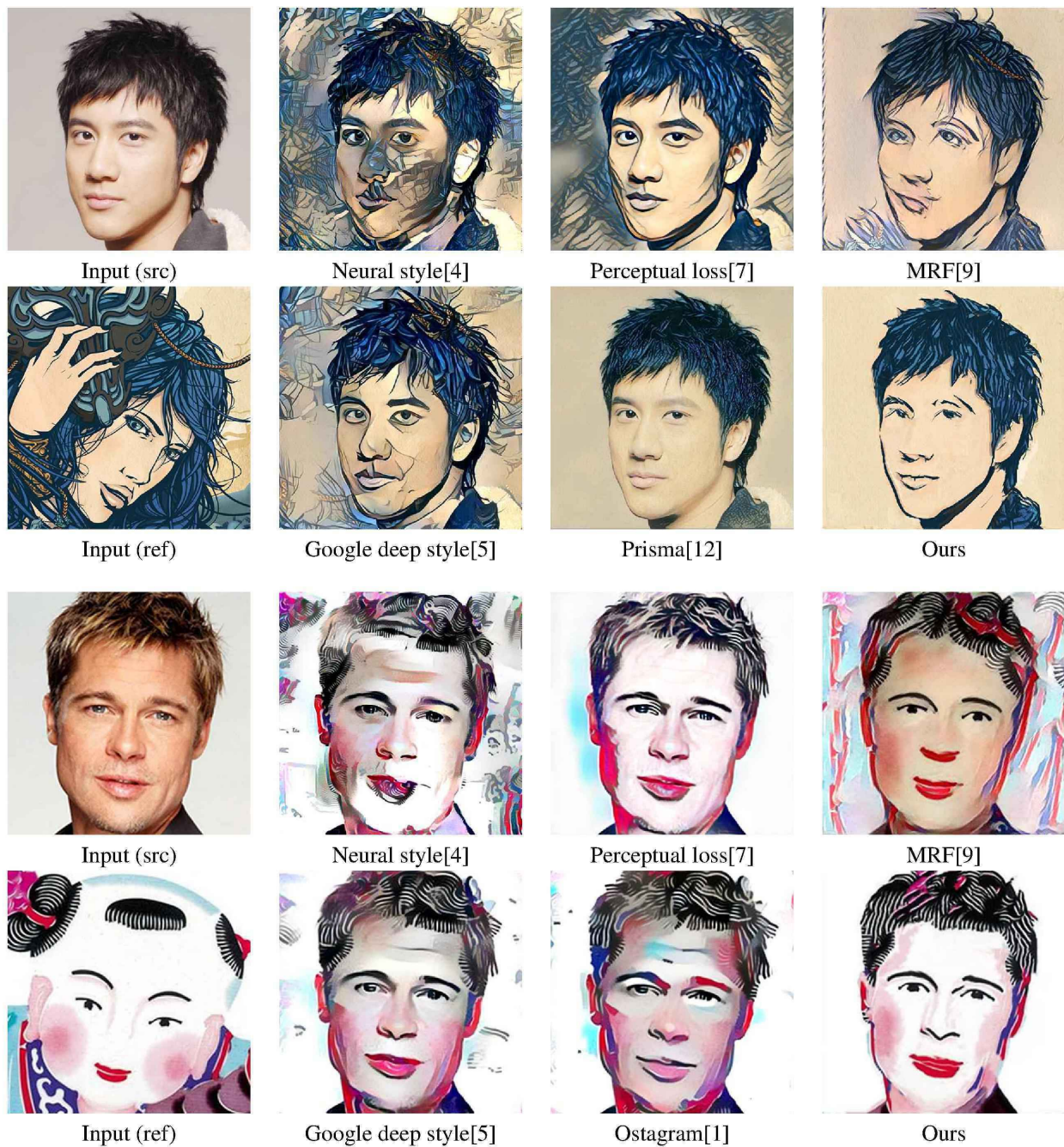


Figure 5: Comparison with other style-transfer methods and apps based on neural networks.

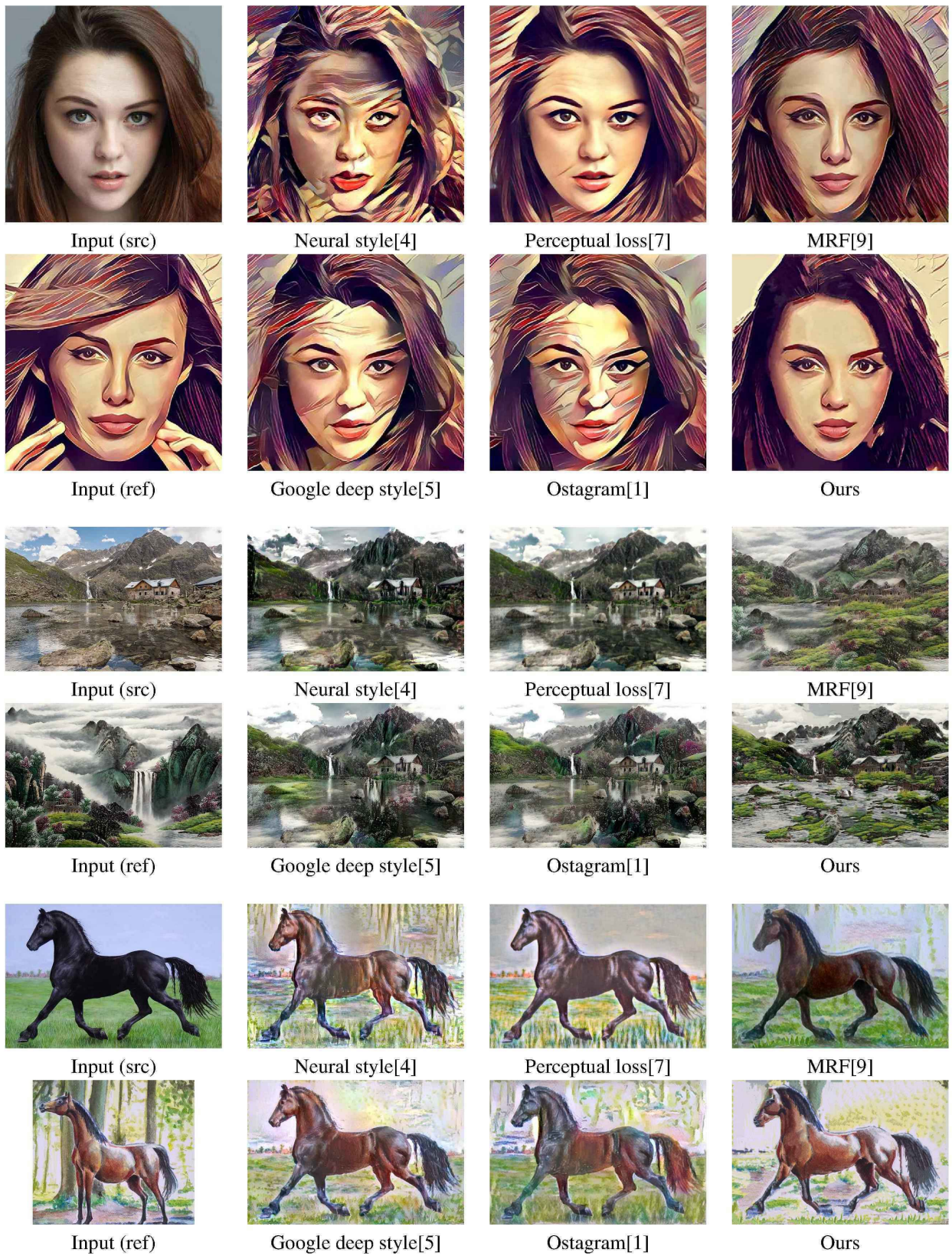


Figure 6: Comparison with other style-transfer methods based on neural networks.



Figure 7: Comparison with other style-transfer methods based on neural networks.

1.2 Style to style

When input pairs of images are two content-related artworks but with vastly different styles, our method is able to swap the styles.

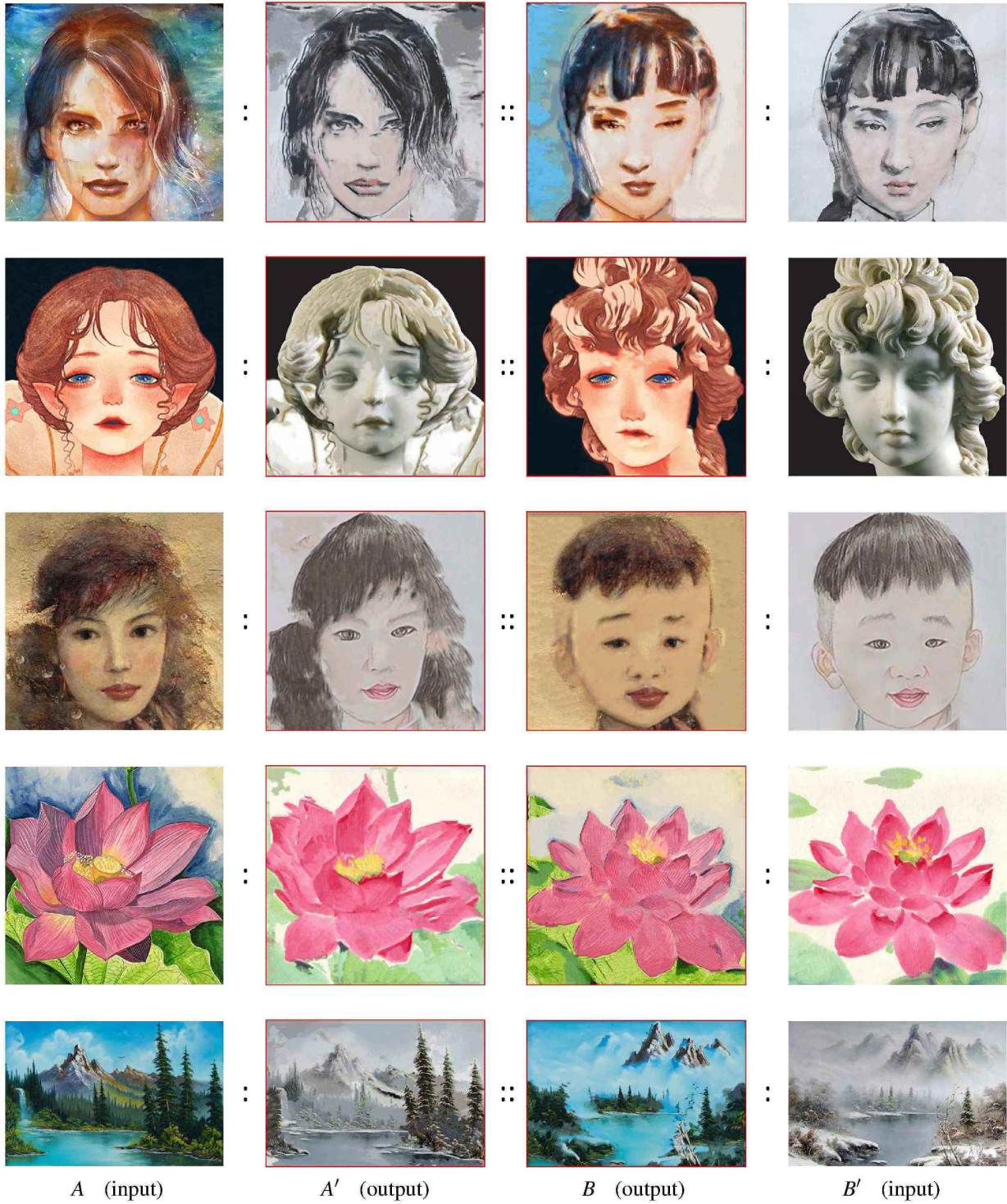


Figure 8: Style-to-style transfer results.

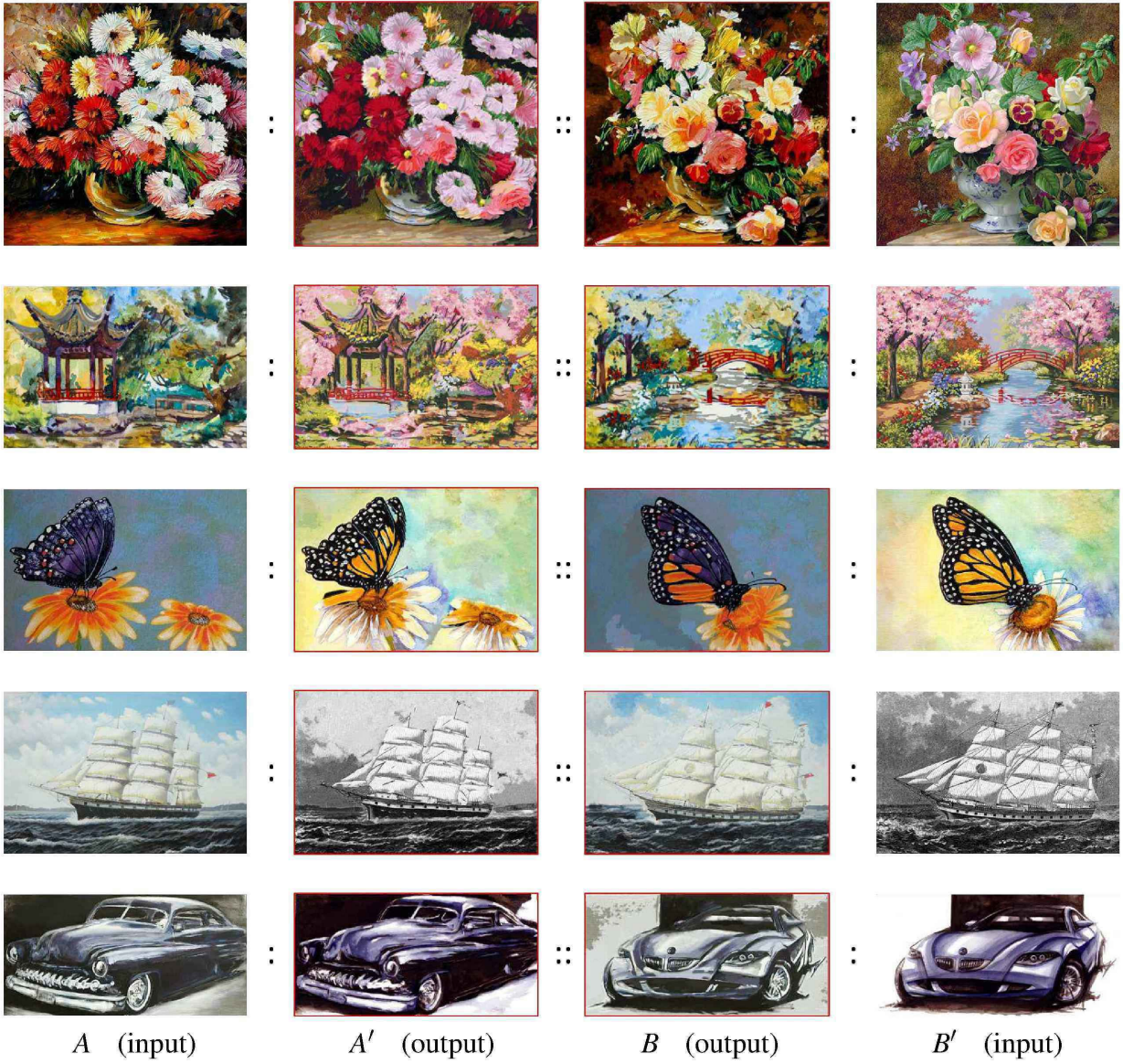


Figure 9: Style-to-style transfer results.

1.3 Style to photo

In this application, we take a piece of artworks as the the source (including sketch, CG, painting), and a photograph as the reference. Our approach is able to transfer the artwork to a real photo, with more plausible results when both images are very related.

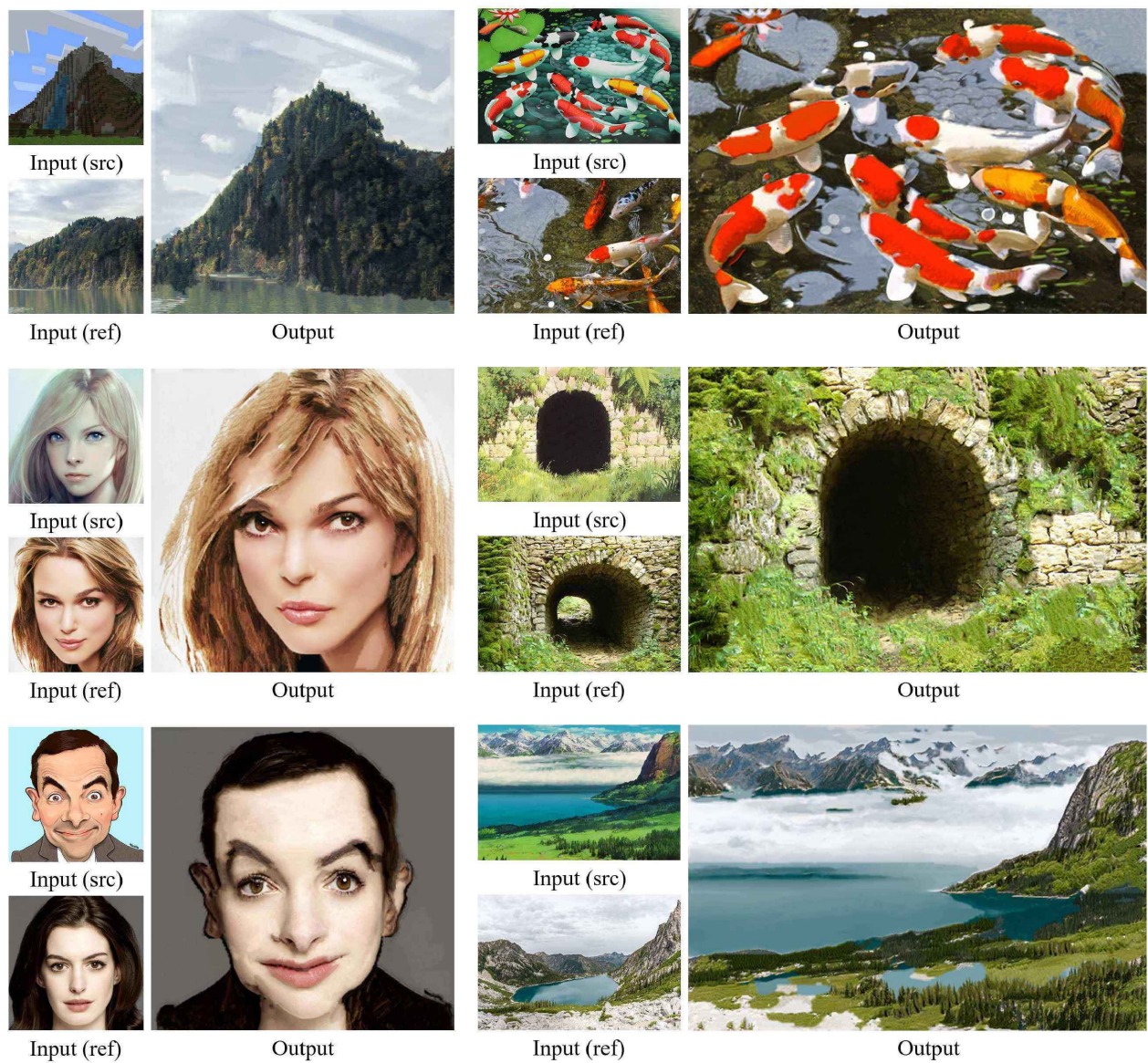


Figure 10: Results of transfer a painting or CG to a photo.

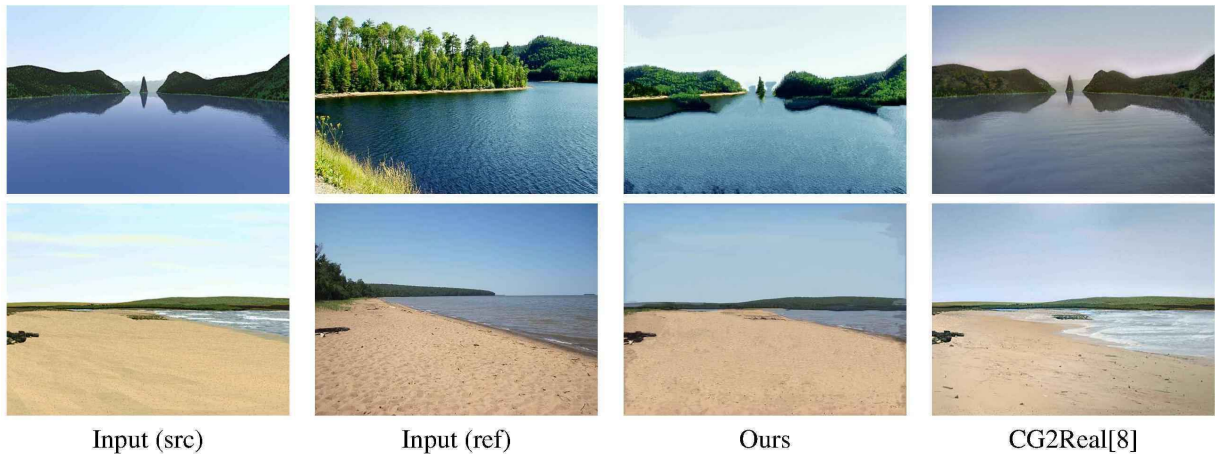


Figure 11: Comparisons of our CG-to-photo results with results of CG2Real[8] on their examples. For each group, the source CG is used by both ours and theirs, but the reference photo is only for ours. Their multiple reference photos are not given in their paper.

1.4 Photo to photo

In this application, our approach takes two photos as input and then swaps the color or tone between them in corresponding regions.

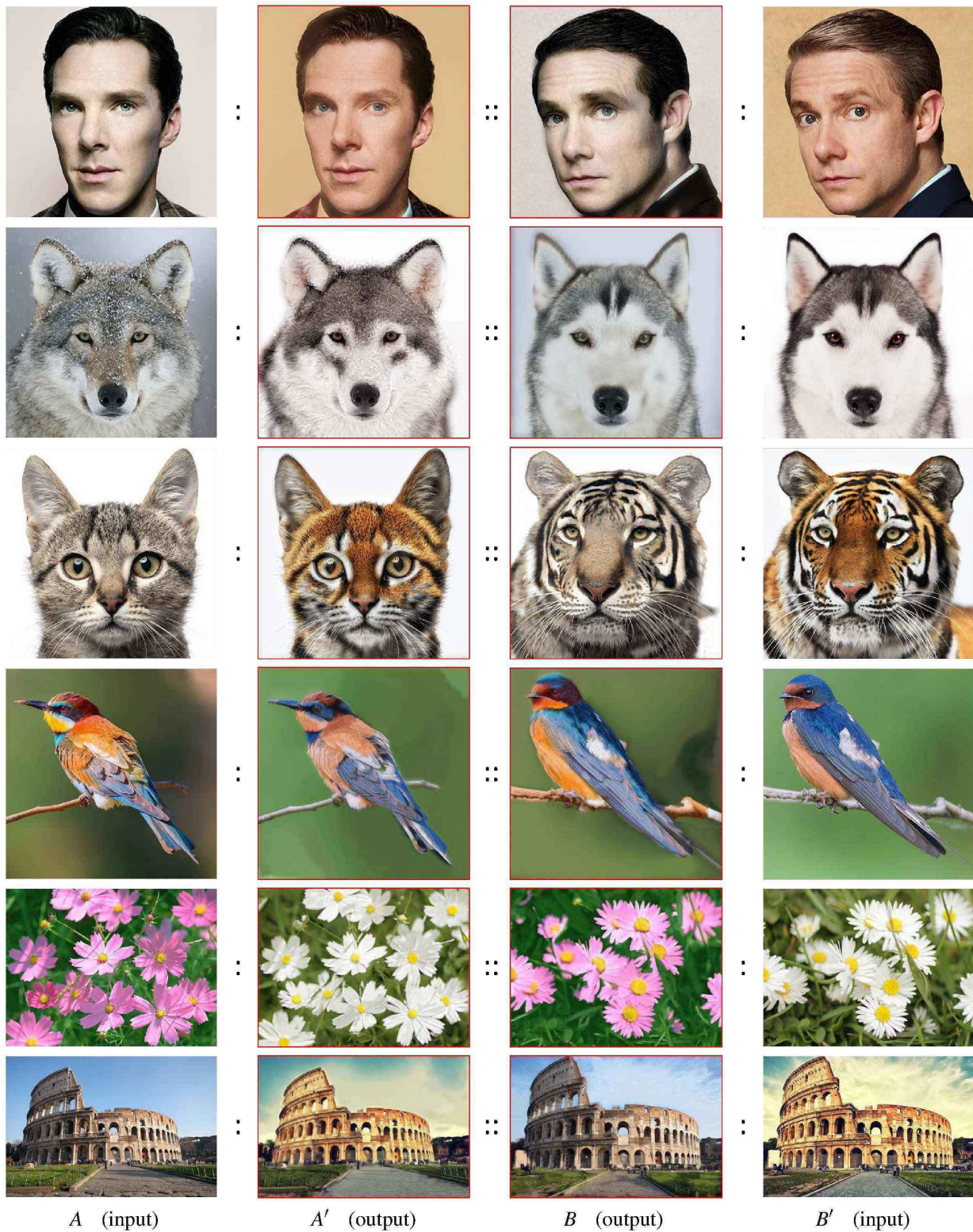


Figure 12: Results of color transfer between photographs.



Figure 13: Our time-lapse results compared with regional foremost[13] method. Ours and theirs both take the summer one as the source and other three as references. Theirs has some distortion artifacts while ours has some mismatches, as indicated by the red rectangles. To be a fair comparison, we don't use our bilateral refinement in this case.

2 Evaluations

We evaluate the matching quality of our approach and state-of-the-art methods on three various categories of data: (I) the same scene, with similar appearance but varied views or motions (*e.g.*, neighboring frames in video); (II) the same scene, with large variations in views, colors and tones (*e.g.*, two photos from different cameras or illuminations); (III) semantically-related scene with vastly different styles (*e.g.*, photograph and painting). Since the task of our paper is image reconstruction rather than motion estimation, we show the reconstruction results based on the correspondences obtained by all methods, rather than flow maps.

2.1 Category 1: the same scene with varied views and motions

In this category, we test on the datasets of Middlebury[2] and VidPair[14]. All our results are obtained with the setting $\{\alpha^L\}_{L=4,3,2,1} = \{1.0, 1.0, 1.0, 1.0\}$. And all compared other approaches use author-provided implementation with the default setting.

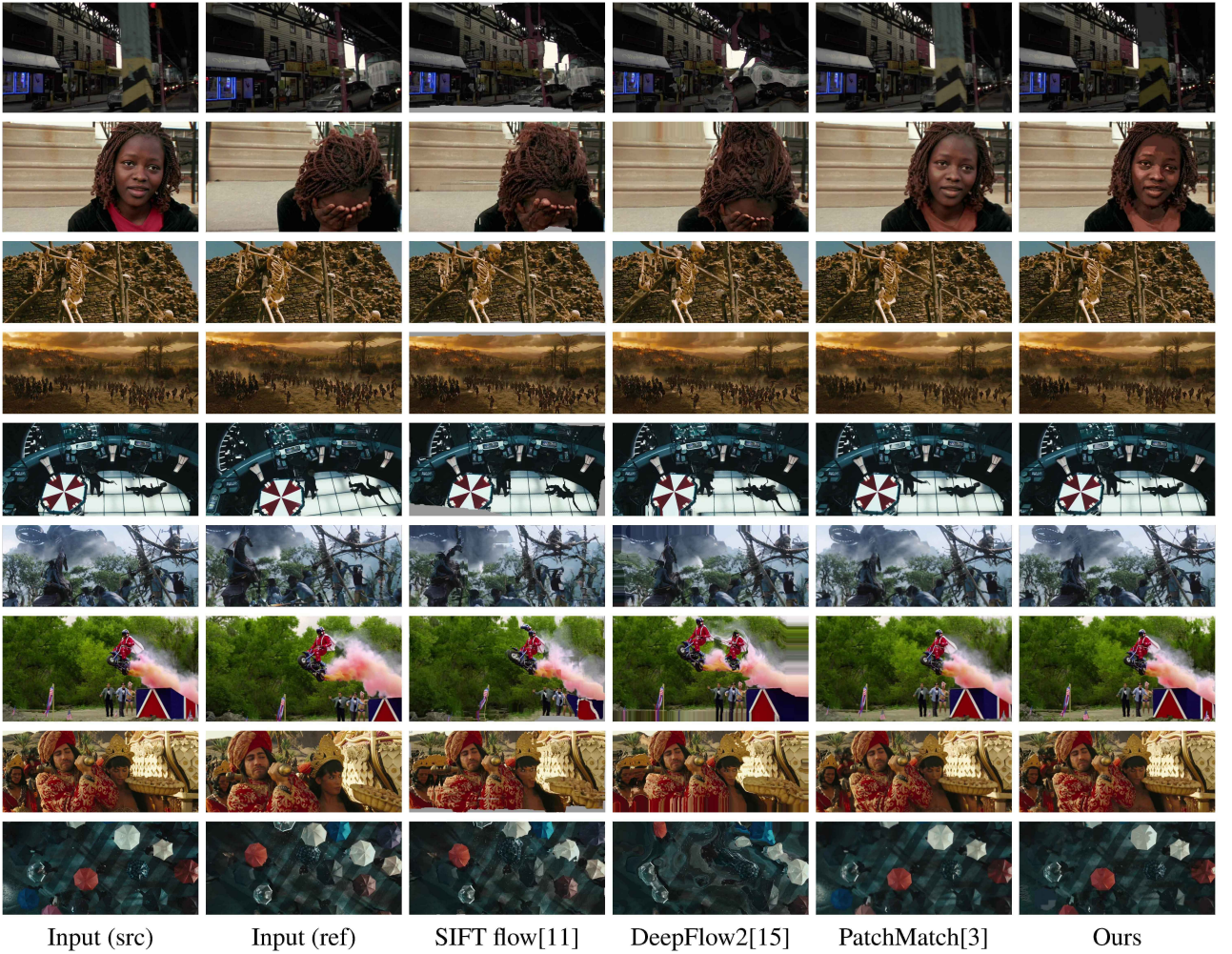


Figure 14: Comparison of different dense correspondence methods on input pairs of category 1 (VidPair[14] dataset).

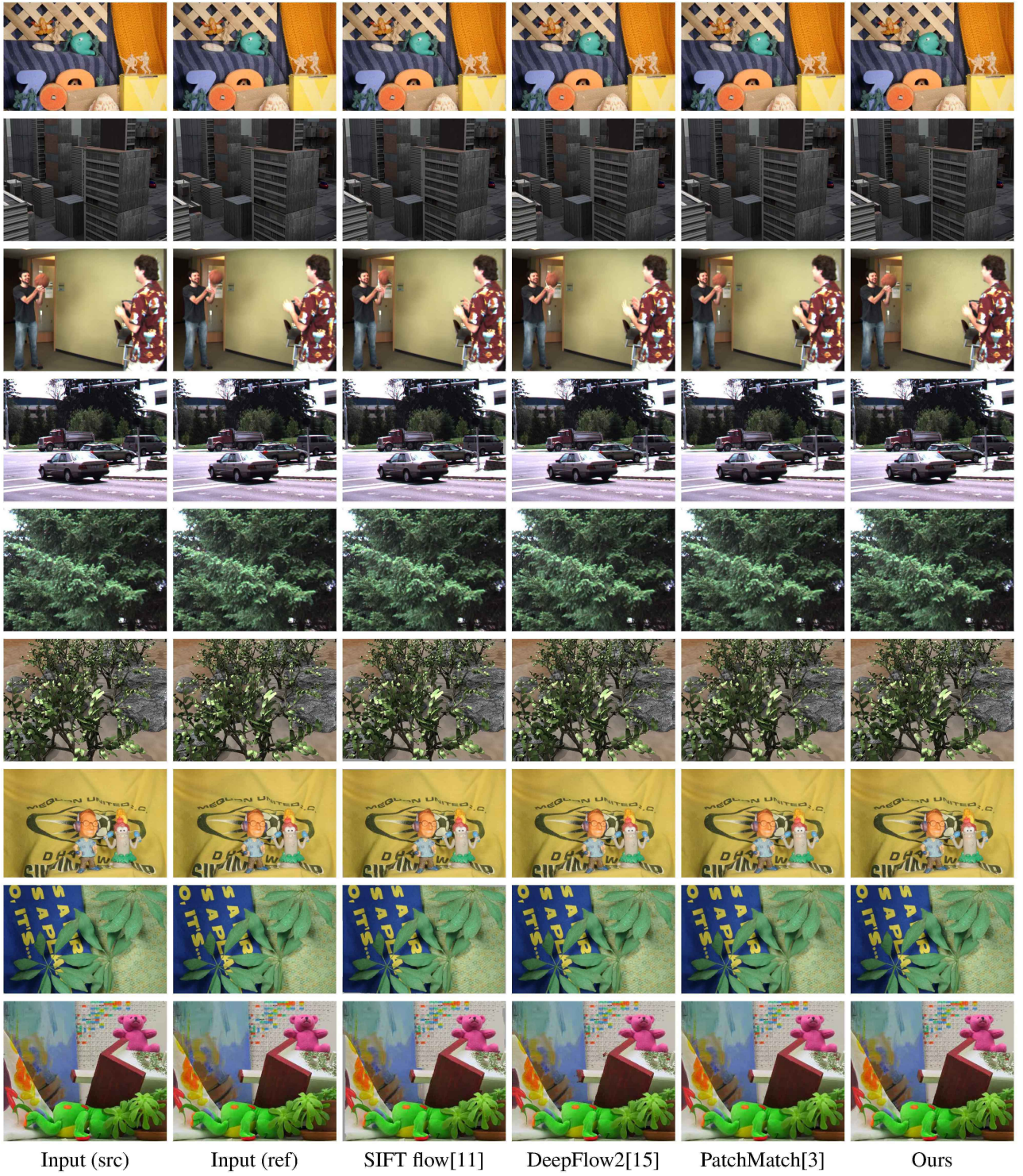


Figure 15: Comparison of different dense correspondence methods on input pairs of category 1 (Middlebury[2] dataset).

2.2 Category 2: the same scene, with large variations in views, colors and tones

In this category, we test on the data collected from the papers of SIFT flow[11], NRDC[6] and Regional foremost matching (RMF)[13]. All our results are obtained with the setting $\{\alpha^L\}_{L=4,3,2,1} = \{0.9, 0.8, 0.7, 0.2\}$. And all compared other approaches use author-provided implementation with the default setting.

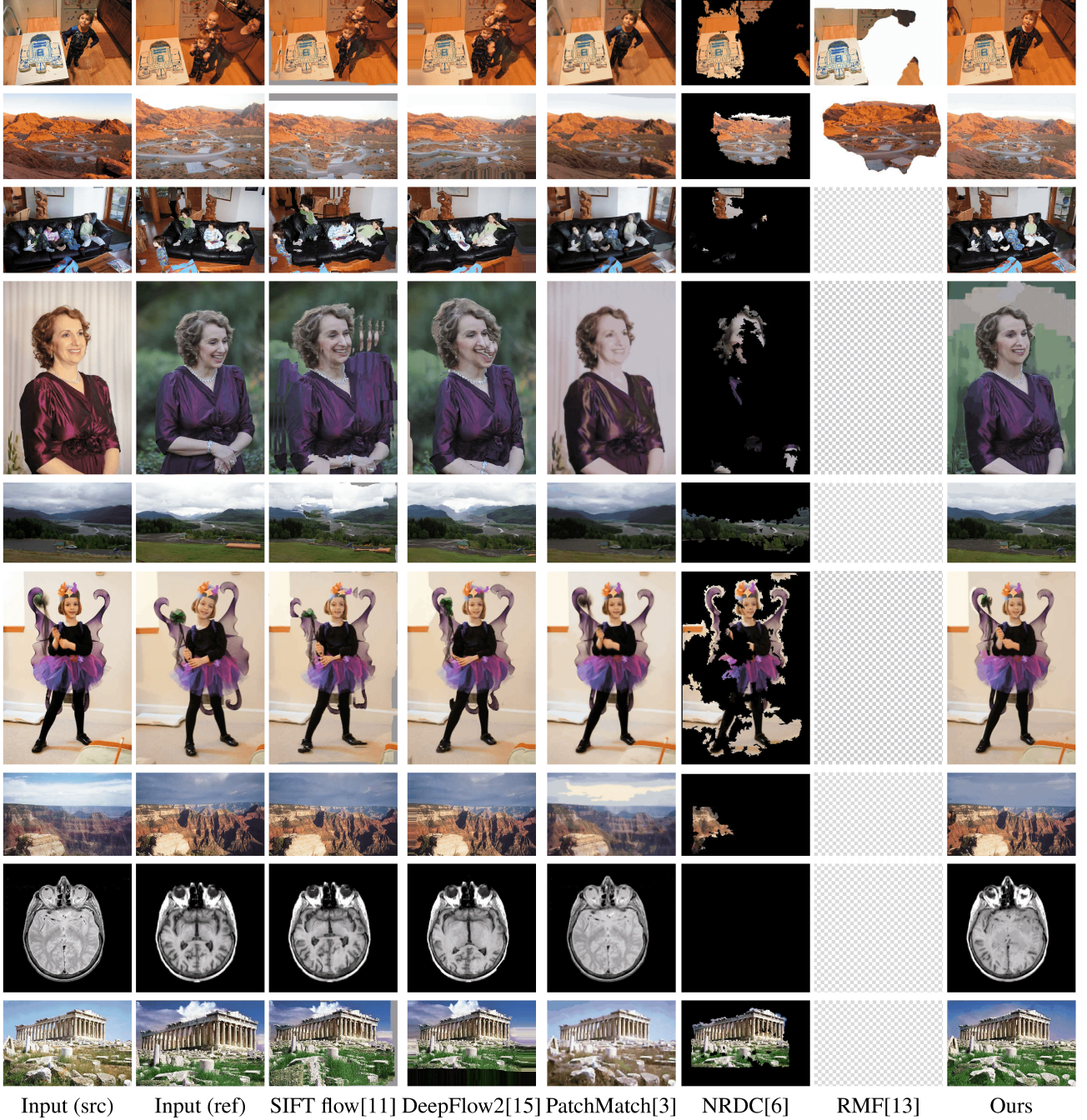


Figure 16: Comparison of different dense correspondence methods on input pairs of category 2. We do not have some results of RFM[13], and its implementation, so they are empty here.

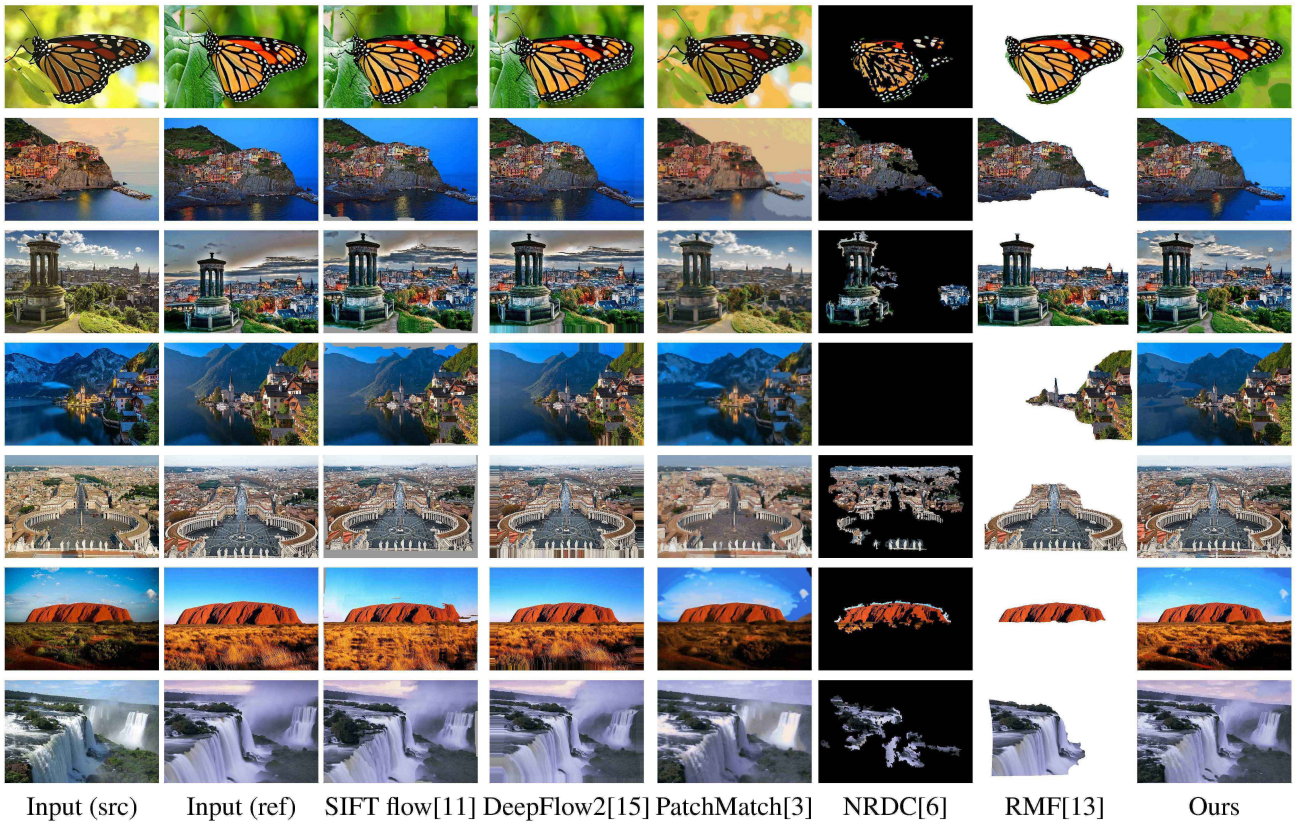


Figure 17: Comparison of different dense correspondence methods on input pairs of category 2.

2.3 Category 3: semantically-related scene with vastly different styles

In this category, we test on the image pairs collected from Internet by the same query, like bird, portrait. They are semantically related but vastly different in styles, like a painting and a photo, two different paintings, or two photos of different objects. All our results are obtained with the setting $\{\alpha^L\}_{L=4,3,2,1} = \{0.8, 0.7, 0.6, 0.1\}$. And all compared other approaches use author-provided implementation with the default setting.

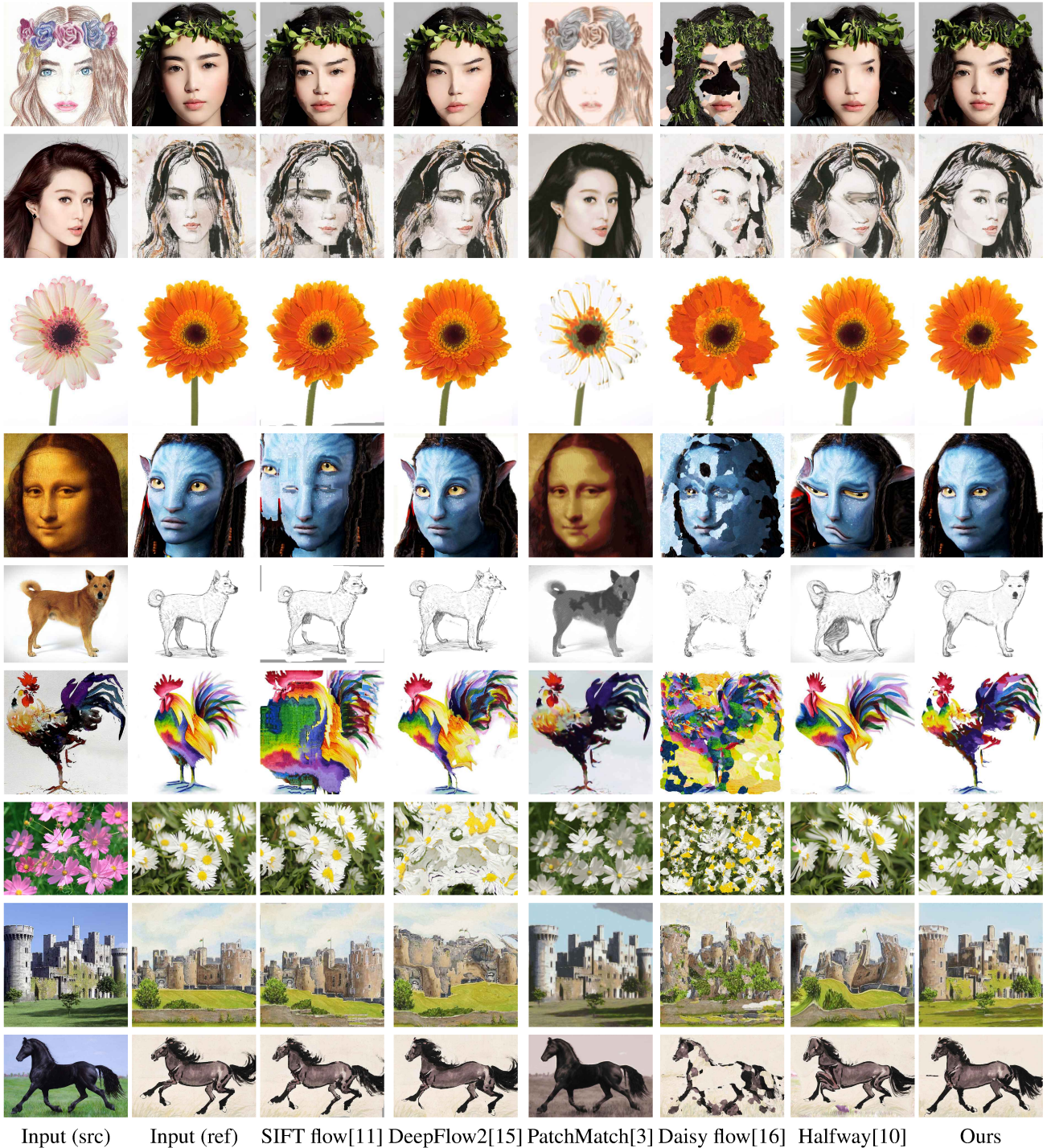


Figure 18: Comparison of different dense correspondence methods on input pairs of category 3.

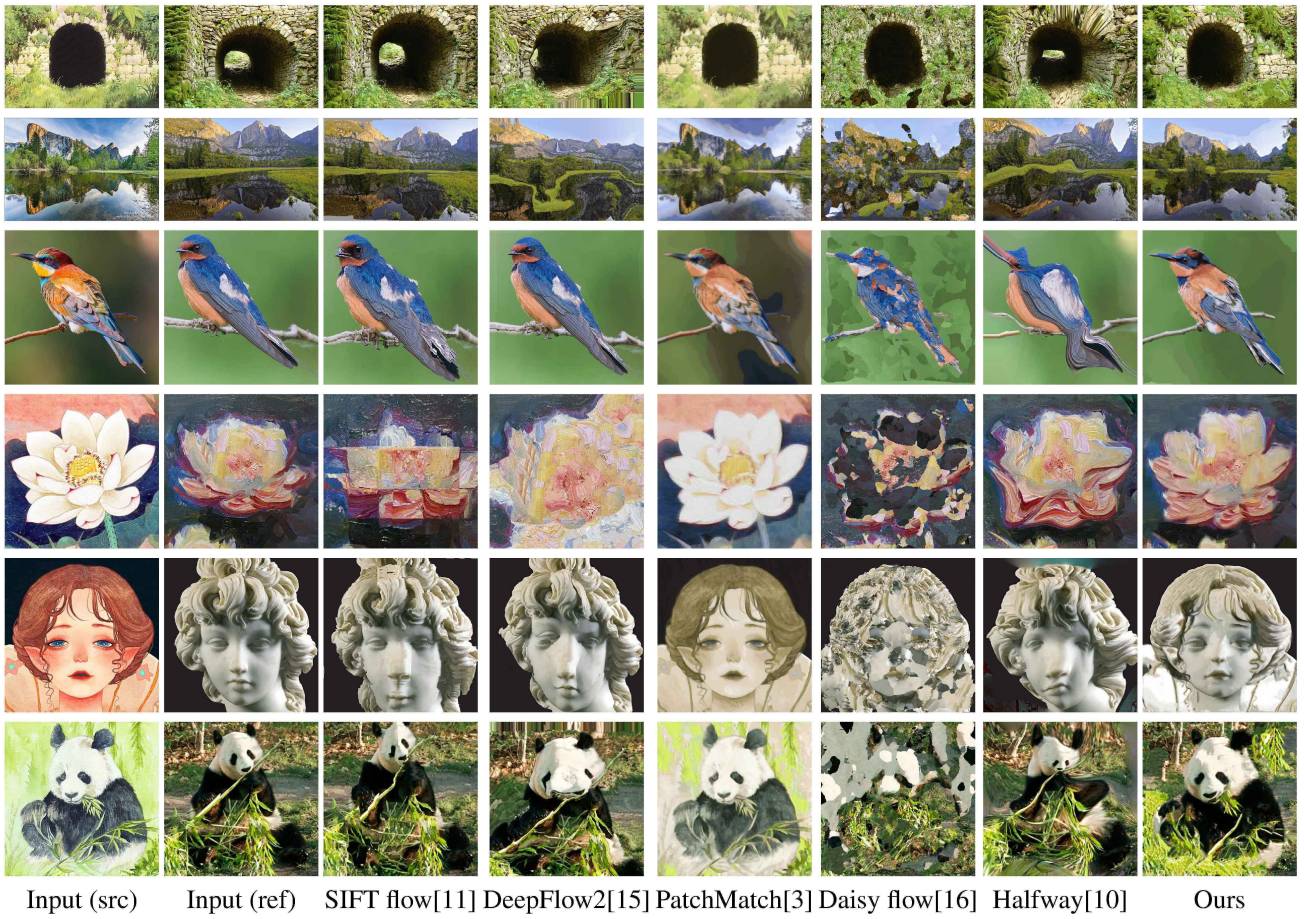


Figure 19: Comparison of different dense correspondence methods on input pairs of category 3.

3 Limitations

In this section, we give some typical failure cases to show the limitations of our work.

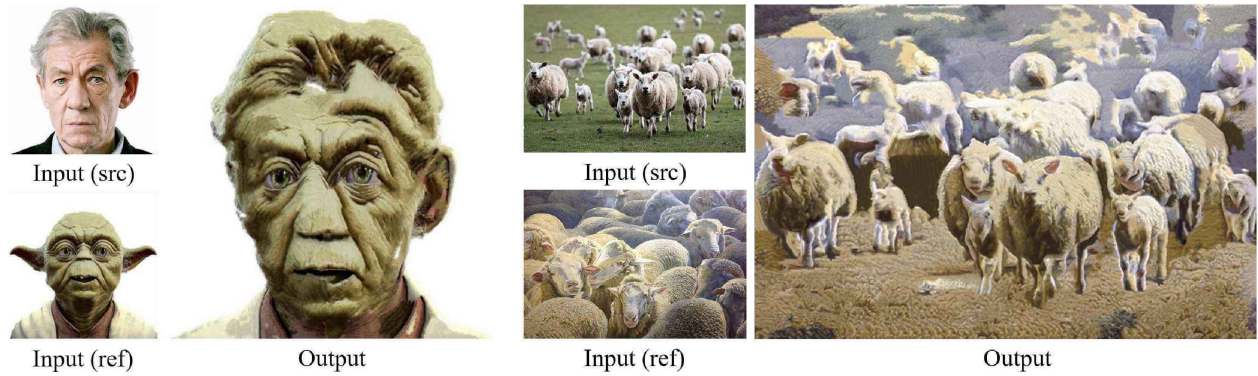


Figure 20: Failure cases of type 1. Our method fails to find correct matches for the object which is found in one image but not the other (like hairs in the left example and grass in the right example.)

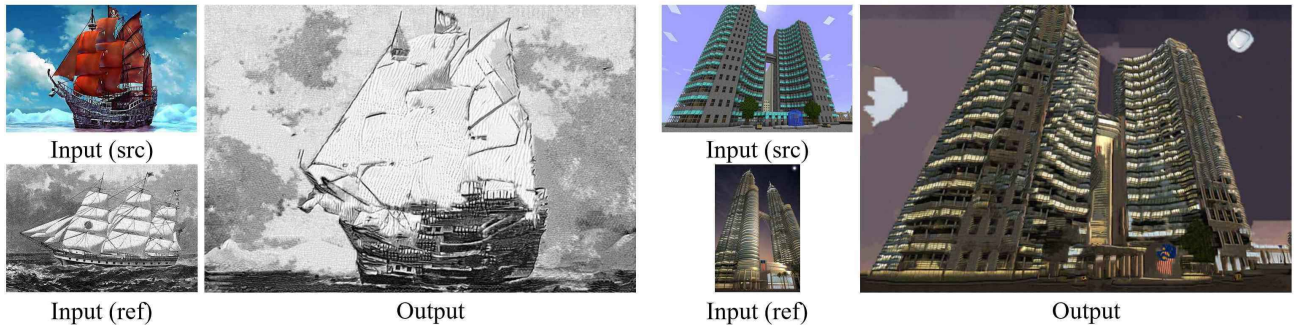


Figure 21: Failure cases of type 2. There are some structure distortions and misalignments in these examples (like sails in the left example and windows in the right one), because our NNF search does not allow patches to be either rotated or scaled.

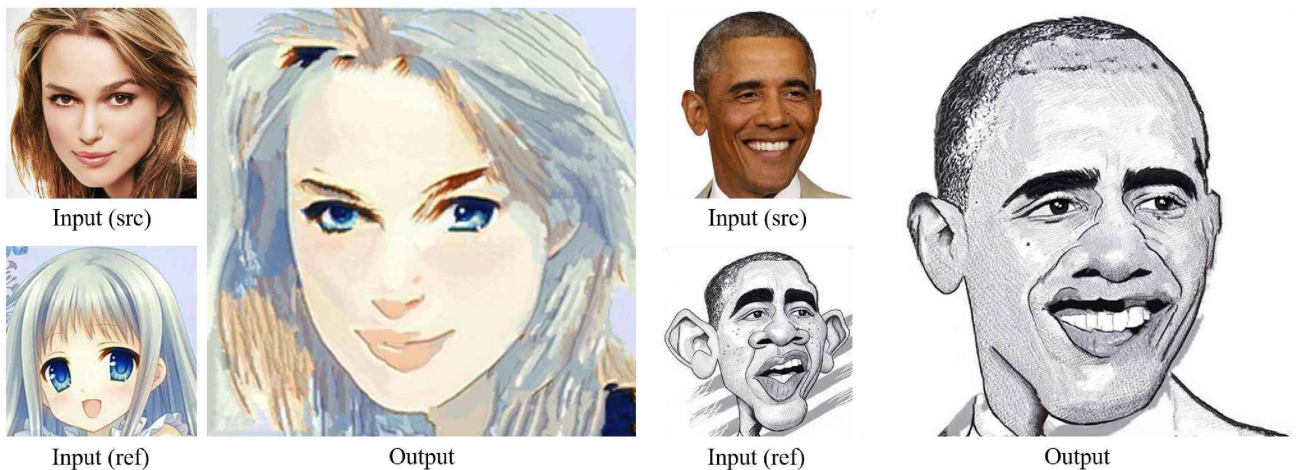


Figure 22: Failure cases of type 3. Our photo-to-style transfer cannot produce geometry style transfer, like exaggerations of the eyes and the nose, due to the assumption in our work to preserve the content structure.

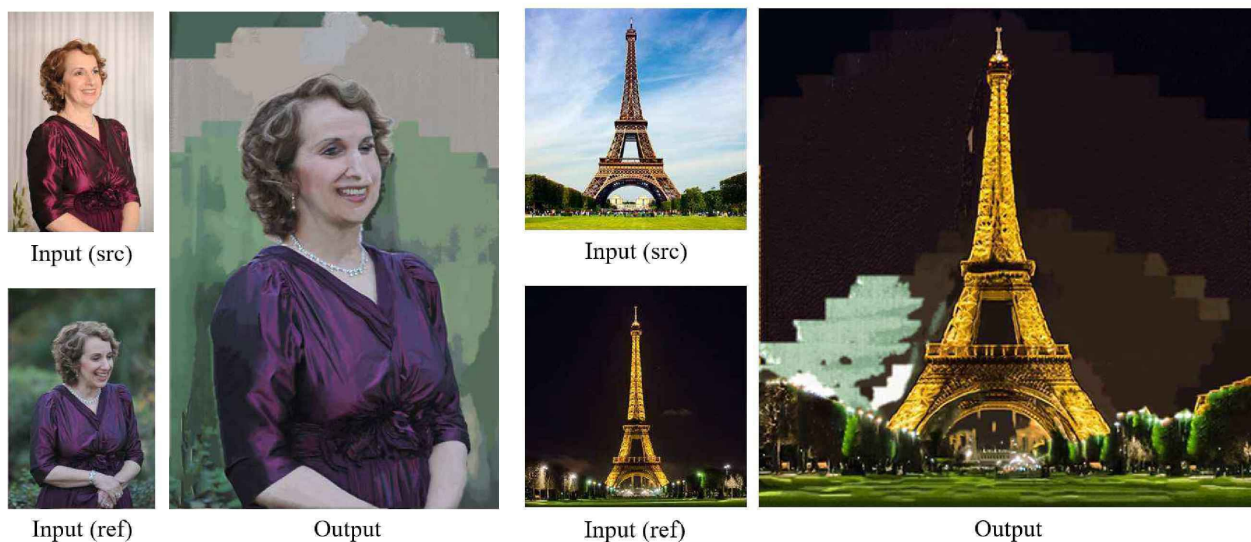


Figure 23: Failure cases of type 4. Our method may fail to find correspondence in textureless regions that have very low neural activation, like the backgrounds in these two examples.

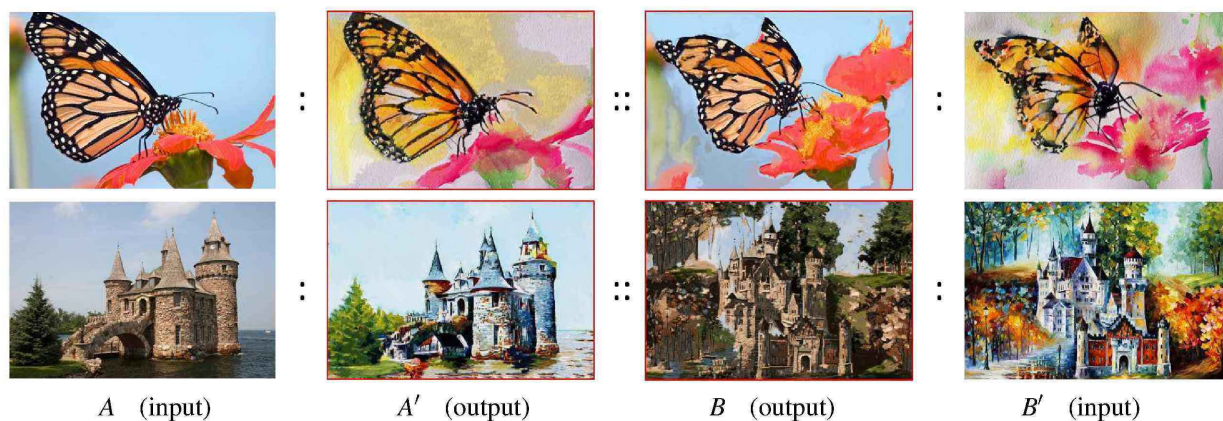


Figure 24: Failure cases of type 5. In this two examples, our method succeeds in the task of photo to style, but fails in the reversed direction. One reason is that artworks tend to lack detail in favor of creativity. Another reason is that humans are very sensitive to slight structure misalignments.

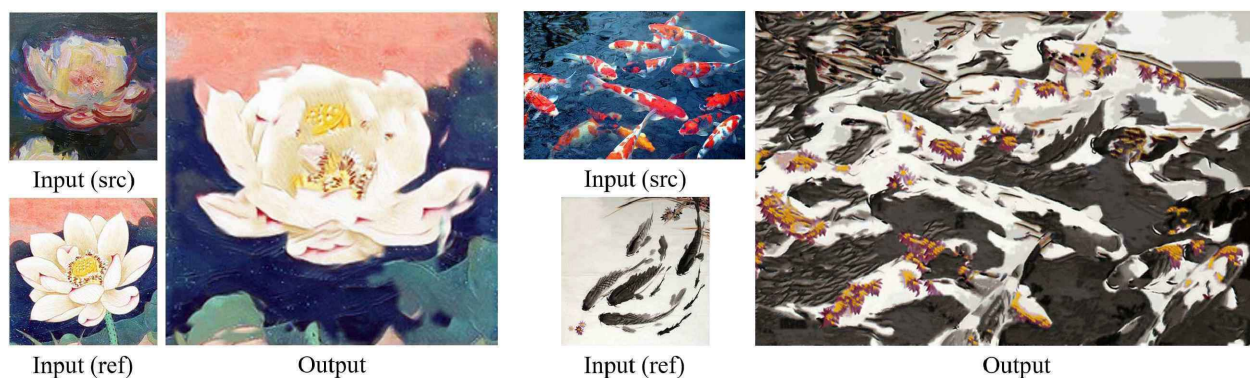


Figure 25: Failure cases of type 6. Our method may fail to find correspondences for highly abstract paintings (as the nymphs drawn by Claude Monet and the fishes in the Chinese brush painting) which are unrecognizable for a pre-trained VGG network.

Bibliography

- [1] Ostagram. <https://ostagram.ru>.
- [2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.
- [3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG*, 28(3):24, 2009.
- [4] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.
- [5] Google. Deep style. <http://www.deepstylegenerator.com/>.
- [6] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. *ACM transactions on graphics (TOG)*, 30(4):70, 2011.
- [7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *arXiv preprint arXiv:1603.08155*, 2016.
- [8] Micah K Johnson, Kevin Dale, Shai Avidan, Hanspeter Pfister, William T Freeman, and Wojciech Matusik. Cg2real: Improving the realism of computer generated images using a large collection of photographs. *IEEE Transactions on Visualization and Computer Graphics*, 17(9):1273–1285, 2011.
- [9] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. *arXiv preprint arXiv:1601.04589*, 2016.
- [10] Jing Liao, Rodolfo S Lima, Diego Nehab, Hugues Hoppe, Pedro V Sander, and Jinhui Yu. Automating image morphing using structural similarity on a halfway domain. *ACM Transactions on Graphics (TOG)*, 33(5):168, 2014.
- [11] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2011.
- [12] Inc. Prisma Labs. Prisma. <http://prisma-ai.com/>.
- [13] Xiaoyong Shen, Xin Tao, Chao Zhou, Hongyun Gao, and Jiaya Jia. Regional foremost matching for internet scene images. *ACM Transactions on Graphics (TOG)*, 35(6):178, 2016.

- [14] Shai Avidan Simon Korman. Coherency sensitive hashing (csh). <http://www.eng.tau.ac.il/~simonk/CSH/>.
- [15] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.
- [16] Hongsheng Yang, Wen-Yan Lin, and Jiangbo Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *2014 IEEE conference on computer vision and pattern recognition*, pages 3406–3413. IEEE, 2014.