Supplemental Material for EDIT: Exemplar-Domain Aware Image-to-Image Translation

Anonymous CVPR submission

Paper ID 351

1. Network Architecture

Parameter Network. Our parameter network includes the convolution layers of $VGG16$ followed by one fully connected layer and one group fully connected layer. The detailed architecture is shown in Table 1.

Generator. The generator has an encoder-decoder architecture. We notice that, the more parameters needed to be generated using the parameter network, the larger model size is required for the parameter network. As shown in Table 2, our generator begins with three down-convolution layers to extract and encode the features of input images. Afterwards, 9 residual blocks are used to process the style-related features according to exemplars.

Discriminator. The discriminator adopts the architecture of 70×70 patchGAN, which owns the ability to generate high-quality and high-resolution images. The discriminator consists of four Conv2d-IN-LeakyReLU blocks followed by one zero pad layer and one Conv2d layer. The detailed architecture is given in Table 3.

2. More Results

We provide more comparison results for painting ↔ photo, edge ↔ shoe/handbag from Figures 1 to 8. Overall, our EDIT outperforms the other state-of-the-art methods including NST [1], metaNST [6], WCT [4], cycleGAN [8], MUNIT [2], DRIT [3] and EGSC-IT [5], which is able to capture both the domain and the exemplar style information and preserve the content. Moreover, art2real [7] is specifically proposed to translate a painting to a photo. Although achieving good visual effect, this method is still lacking of the ability to control the style of generated results and suffer from abrupt changes across different semantic regions. The comparison results are shown in Figure 9. In addition, we further provide more results by our EDIT in Figures 10-14, which are semantically meaningful, structurally reasonable and visually striking.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016. 1

[2] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, pages 172–189, 2018. 1

[3] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, pages 35–51, 2018. 1

[4] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. In *NeurIPS*, pages 385–395, 2017. 1

[5] L. Ma, J. Xu, S. Georgoulis, T. Tuytelaars, and L. Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. In *ICLR*, 2019. 1

[6] F. Shen, S. Yan, and G. Zeng. Neural style transfer via meta networks. In *CVPR*, pages 8061–8069, 2018. 1

[7] Matteo Tomei, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Art2real: Unfolding the reality of artworks via semantically-aware image-to-image translation. In *CVPR*, pages 5849–5859, 2019. 1

[8] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 1

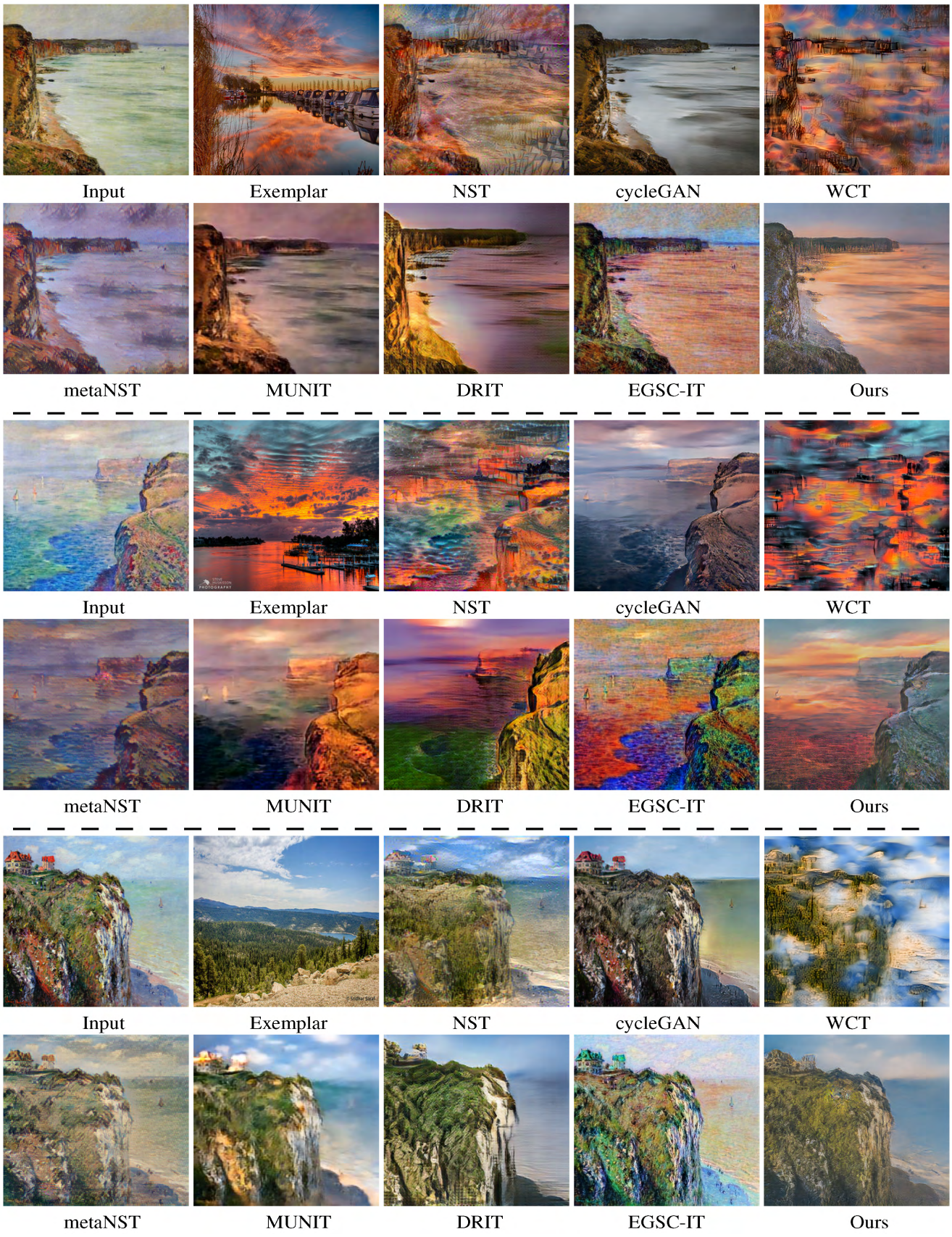


Figure 1. Visual comparisons on painting → photo.

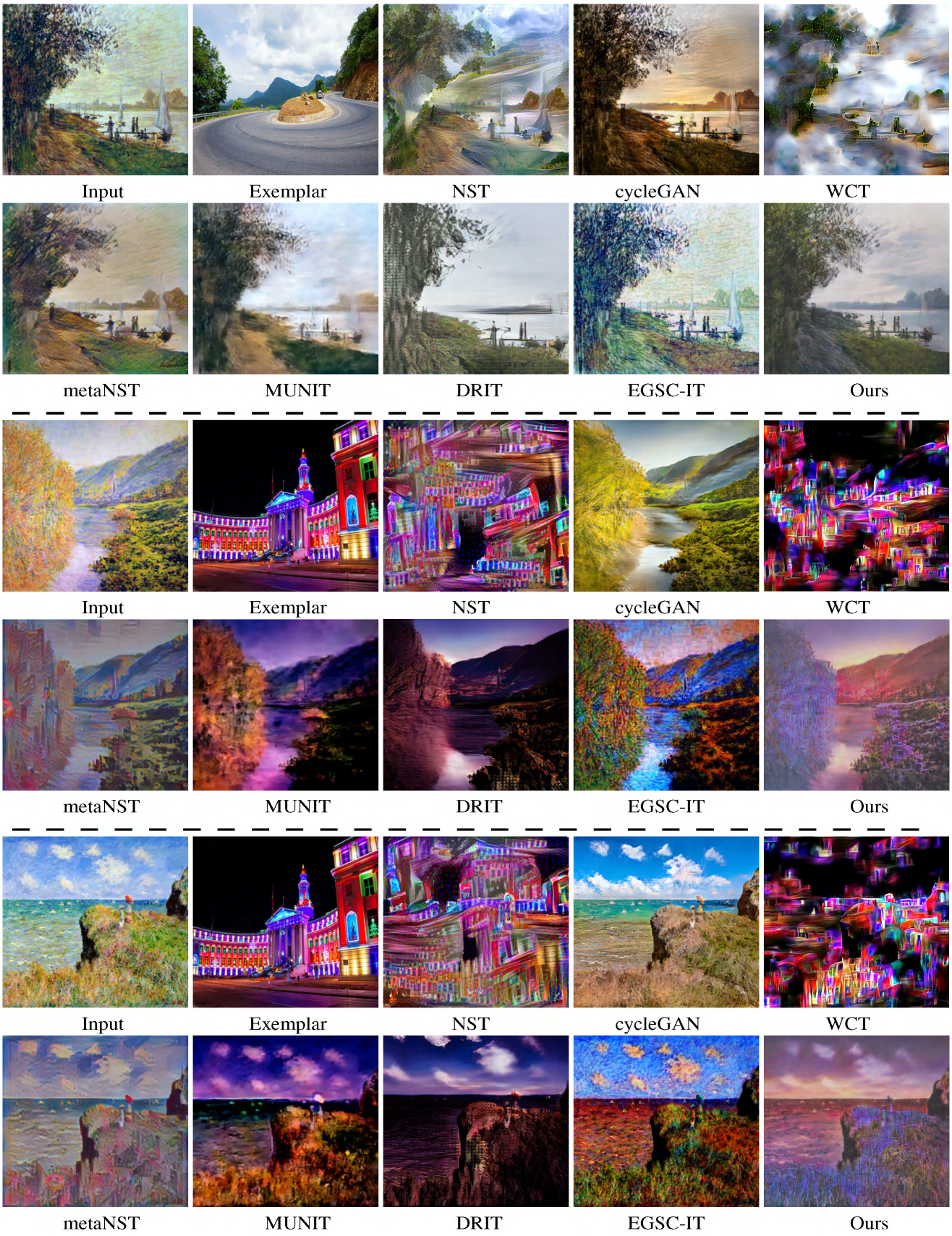


Figure 2. Visual comparisons on painting → photo.

Table 1. The parameter network architecture. The c, k, s and p in the top part stand for the channel number, the kernel size, the stride step and the padding size, respectively. The N_{neuron}, N_{group}, N_{para} and M^l_{para} in the bottom part are the number of neurons in a layer of each group, the number of group of a group fully-connected layer (N_{group}=1 if is not a group fully-connected layer), the number of layers in the generator whose parameters need to be generated and the number of parameters in the l-th layer where $1 \leq l \leq N_{para}$, respectively.

Convolution Layers

Output Size	Operator	c	k	s	p
256×256	Conv2d-ReLU	64	3	1	1
256×256	Conv2d-ReLU (relu1_2)	64	3	1	1
128×128	MaxPool2d	-	-	-	-
128×128	Conv2d-ReLU	128	3	1	1
128×128	Conv2d-ReLU (relu2_2)	128	3	1	1
64×64	MaxPool2d	-	-	-	-
64×64	Conv2d-ReLU	256	3	1	1
64×64	Conv2d-ReLU	256	3	1	1
64×64	Conv2d-ReLU (relu3_3)	256	3	1	1
32×32	MaxPool2d	-	-	-	-
32×32	Conv2d-ReLU	512	3	1	1
32×32	Conv2d-ReLU	512	3	1	1
32×32	Conv2d-ReLU (relu4_3)	512	3	1	1
16×16	MaxPool2d	-	-	-	-
16×16	Conv2d-ReLU (relu5_1)	512	3	1	1

Intermediate Processing

Input	Operation	Output Size
relu1_2,relu2_2, relu3_3,relu4_3 relu5_3	calculate mean and standard deviation	1×2944

Fully-Connected Layers

layer	N_{neuron}	N_{group}
1	$128 \times N_{para}$	1
2	M^l_{para}	N_{para}

Table 2. The generator architecture. The c, k and s in the top part is channel number, kernel size and stride step respectively. Column **Trainable** indicates whether the layer is jointly trained with the parameter network (**Trainable**=True) or generated by the parameter network (**Trainable**=False).

Name	Operator	Trainable	Repeat	c	k	s
down-conv	Conv2d-IN-ReLU	True	1	32	9	2
	Conv2d-IN-ReLU	True		64	3	2
	Conv2d-IN-ReLU	True		128	3	2
residual block	Conv2d-IN-ReLU	True	9	128	3	1
	Conv2d-IN-ReLU	False		128	3	1
up-conv	Upsample-Conv2d-IN-ReLU	True	1	128	3	1
	Upsample-Conv2d-IN-ReLU	False		64	3	1
	Conv2d-Tanh	True		32	9	1

Table 3. The discriminator architecture. The c, k, s and p is channel number, kernel size, stride step and padding size respectively.

Output Size	Operator	c	k	s	p
128×128	Conv2d-LeakyReLU	64	9	2	1
64×64	Conv2d-IN-LeakyReLUReLU	128	3	2	1
32×32	Conv2d-IN-LeakyReLU	256	3	2	1
16×16	Conv2d-IN-LeakyReLU	512	3	2	1
17×17	ZeroPad2d	-	-	-	
16×16	Conv2d	1	4	1	1

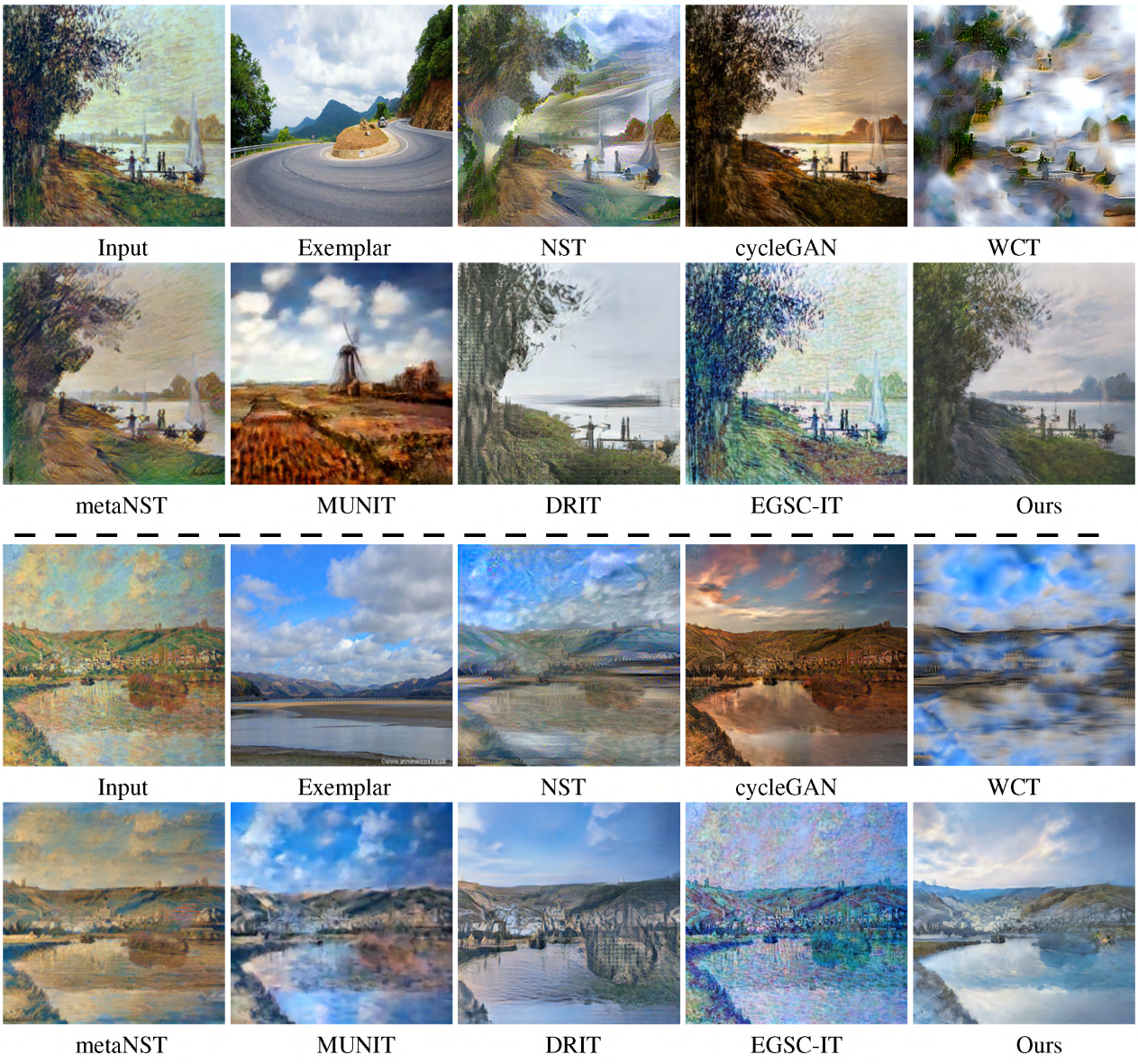


Figure 3. Visual comparisons on painting → photo.

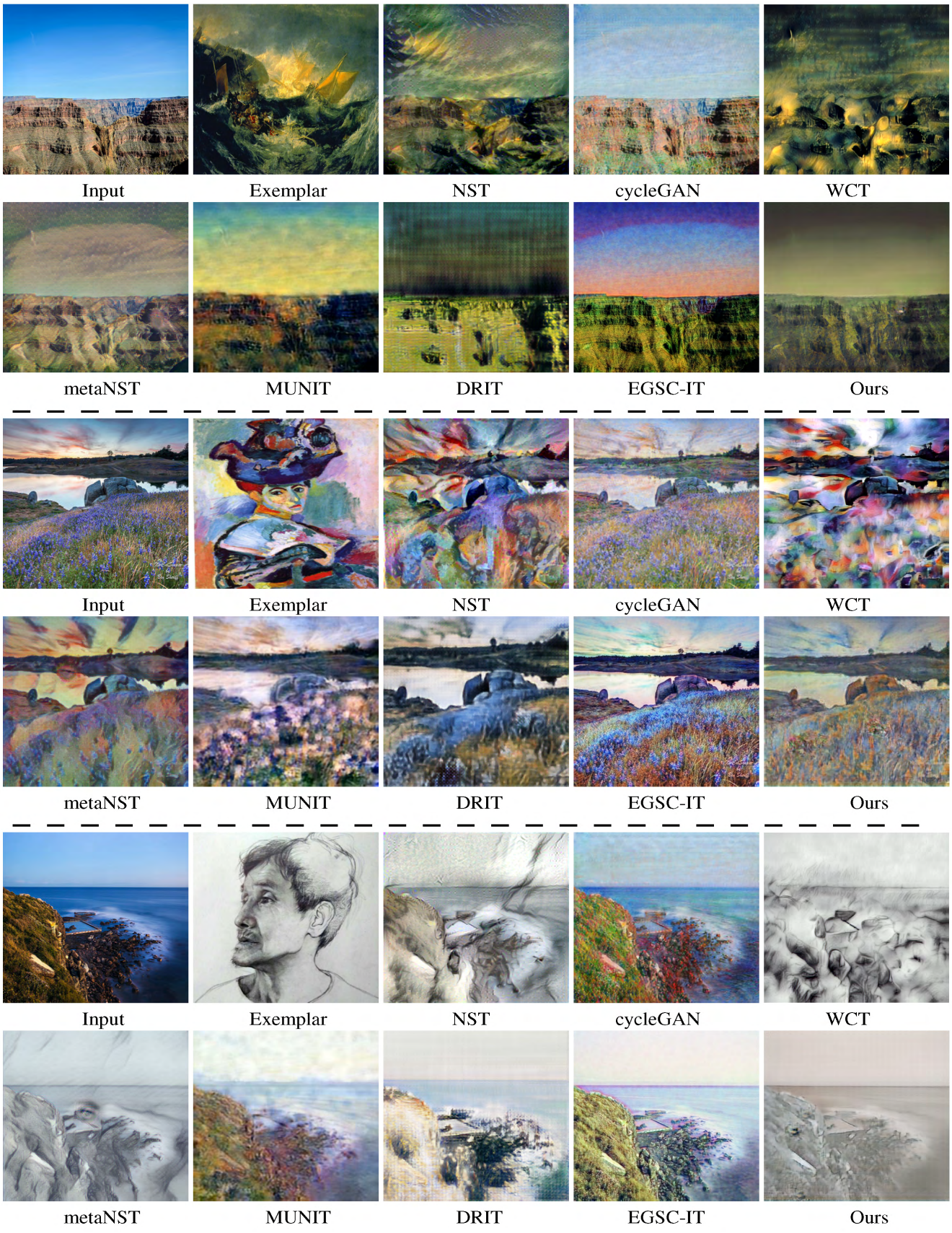


Figure 4. Visual comparisons on photo → Monet's painting

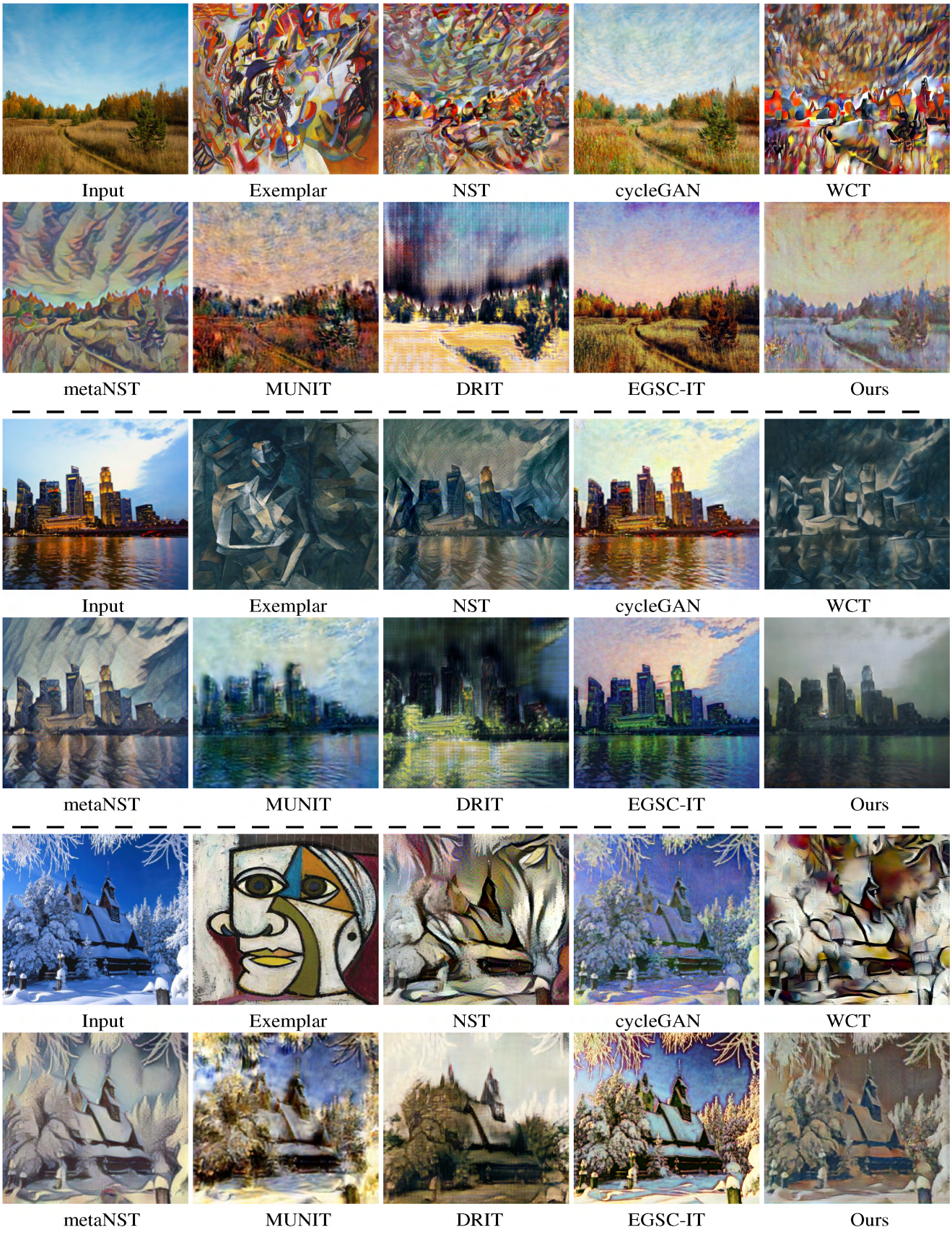
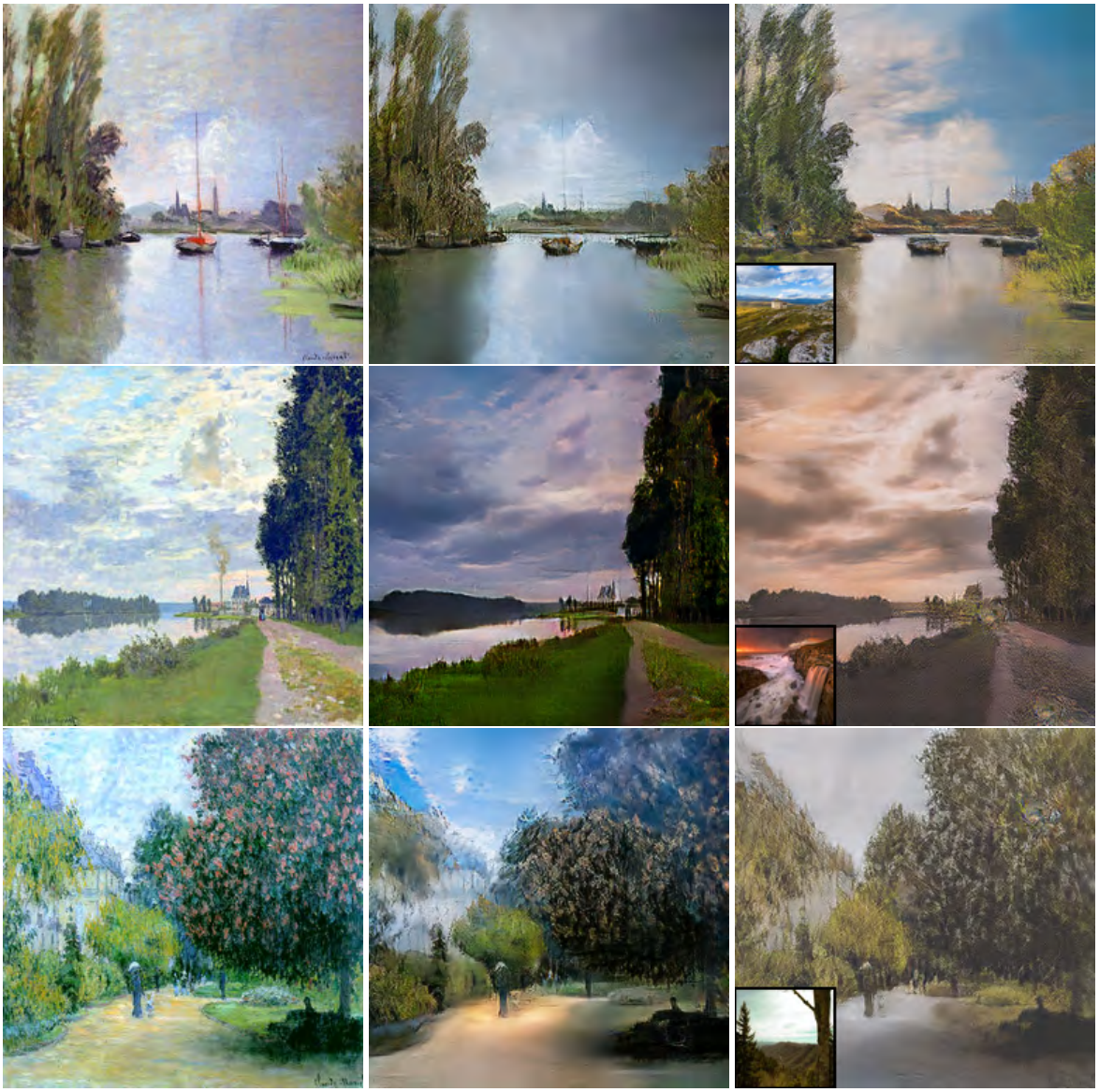


Figure 5. Visual comparisons on photo → Monet's painting

Figure 6. Visual comparisons on edge → shoe.

Figure 7. Visual comparisons on edge → handbag

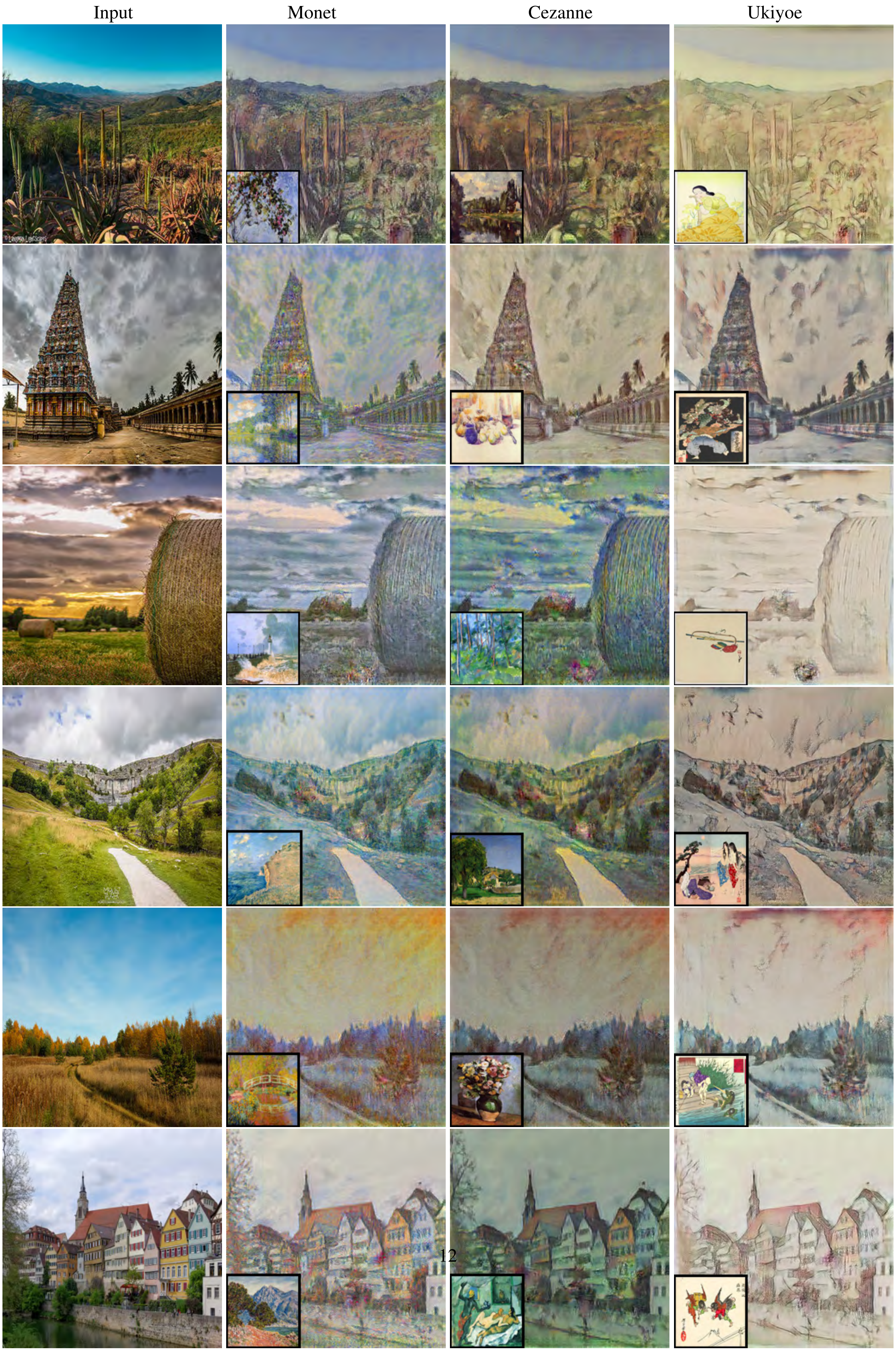
Figure 8. Visual comparisons on shoe → edge.



Input art2real Ours

Figure 9. The visual comparison between art2real and our EDIT.

Figure 10. Visual results by EDIT on painting → photo.


Input
Monet
Cezanne
Ukiyoe

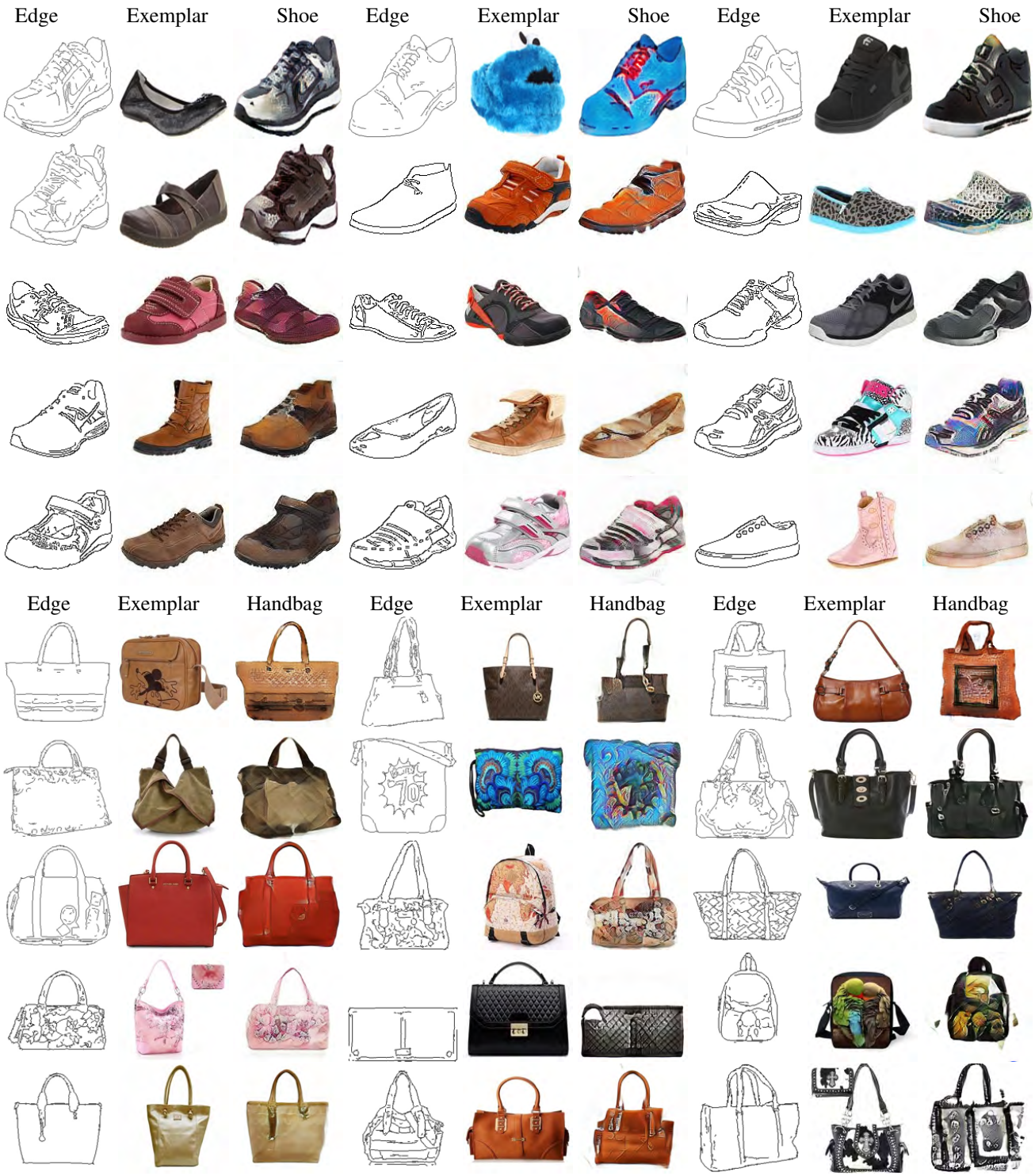


Figure 12. Visual results by EDIT on edge → shoe/handbag.

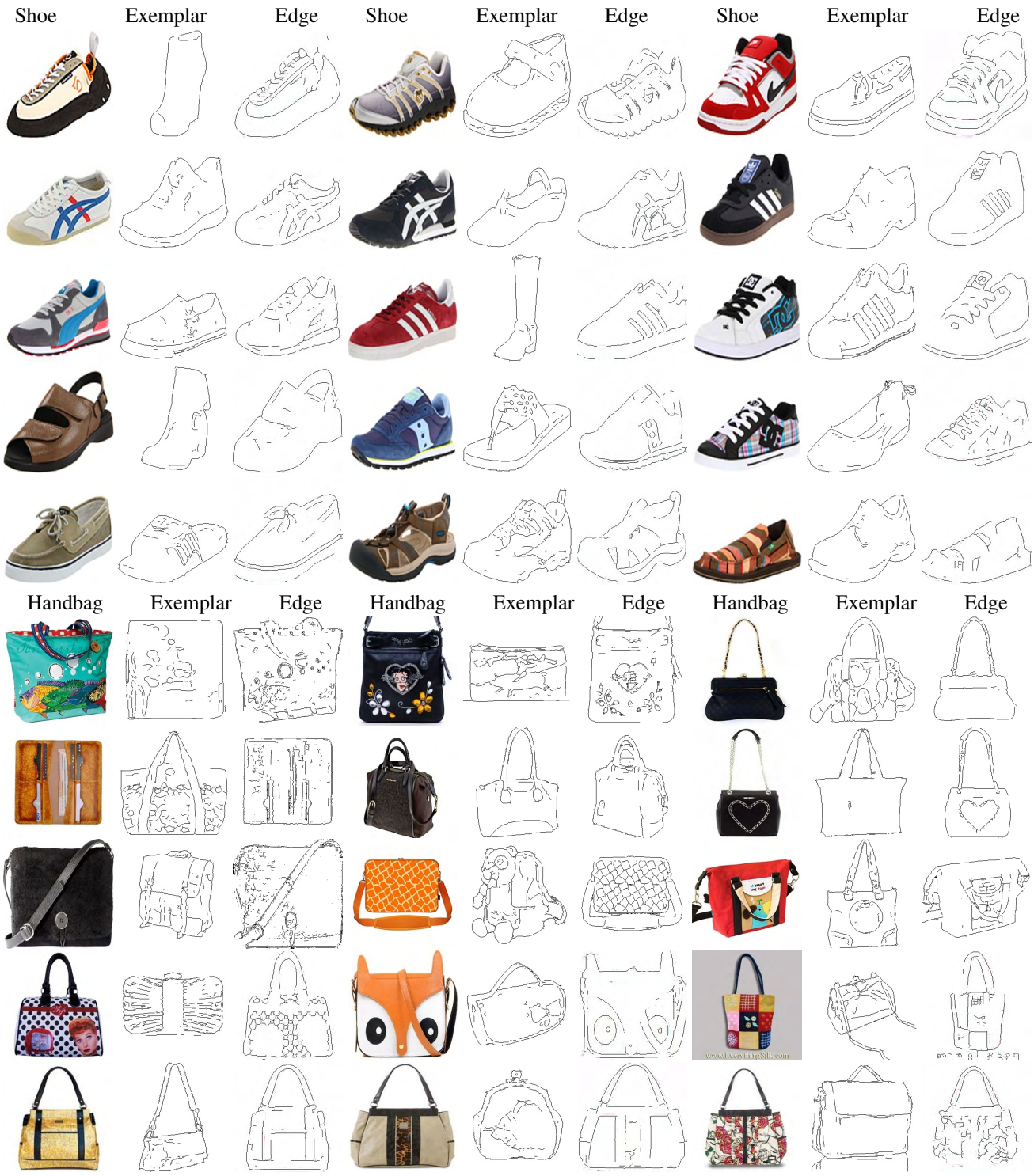


Figure 13. Visual results by EDIT on shoe/handbag → edge.

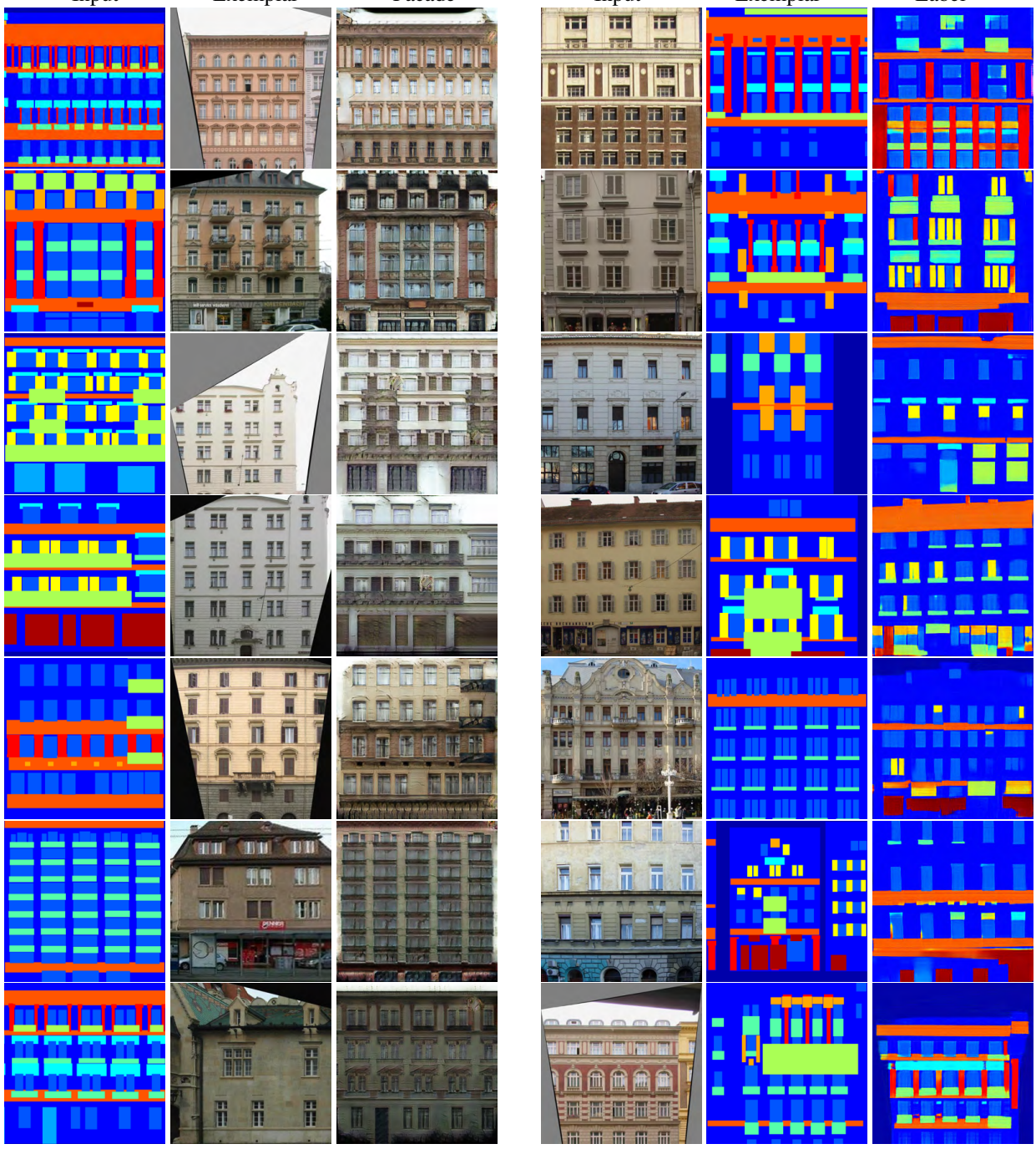


Figure 14. Visual results by EDIT on label ↔ facade.