A Generic Deep Architecture for Single Image Reflection Removal and Image Smoothing (*Supplementary Material*)

Qingnan Fan*[1] Jiaolong Yang[2] Gang Hua[2] Baoquan Chen[1,3] David Wipf[2]
[1]Shandong University [2]Microsoft Research [3]Shenzhen Research Institute, Shandong University
fqnchina@gmail.com, {jiaoyan,davidwip,ganghua}@microsoft.com, baoquan@sdu.edu.cn

1 Outline

This supplementary document provides more analyses and results that were not included in the main paper due to space limitations. The contents are organized as follows.

- **Section 2:** Detailed explanation and analysis of our synthetic data generation method for training CEILNet in the reflection removal task.
- **Section 3:** Comparison of an I-CNN network with varying convolutional layers against the full CEILNet pipeline.
- **Section 4:** Visualization of more edge maps predicted by E-CNN for both the reflection removal and image smoothing tasks.
- **Section 5:** More visual comparisons with existing deep image smoothing networks.
- **Section 6:** More visual comparisons with our baselines and existing methods on the reflection removal problem.
- **Section 7:** More visual results of our CEILNet on both synthetic and real reflection images. For the latter, our network is the only approach capable of reliably separating a series of difficult single real-world images into reflection and background layers without strong additional priors.
- **Section 8:** More visual results of our CEILNet for different smoothing filters.

*This work was done when Qingnan Fan was an intern at MSR.

2 Complete description of our synthetic reflection image generation process

Real images with ground truth background and reflection layers are difficult to obtain. In the main paper, we have proposed a novel reflection image data generation method for producing synthetic training data. Here we first elaborate on why simply mixing two images does not work well for this purpose, followed by a thorough explanation of how our more involved generation strategy operates.

2.1 The problem with naively mixing two images

To generate enough data for training a deep network, a naive approach is to simply mix a candidate background layer $\mathbf{B}$ and reflection layer $\mathbf{R}$ to create new synthetic image samples via $\mathbf{I} = \nu_1\mathbf{B} + \nu_2\mathbf{R}$, where ν_1 and ν_2 are relative scaling coefficients that sum to one to avoid overflow or image clipping (*e.g.*, 0.7 and 0.3 respectively). Indeed this exact strategy has been widely used in previous works to create ground-truth for analysis and quantitative evaluation [9, 7, 3, 5, 14]. But as a viable source of training data, this approach is problematic based on several inconsistencies with natural images:

- First, both $\mathbf{B}$ and $\mathbf{R}$ should not be scaled. In real-world images, background and reflection layers contain various levels of luminance, from the darkest to the brightest color. Scaling the images not only constrains each layer within a relatively smaller color range, but also suppresses abrupt color transitions, especially for the reflection layer.
- Second, in typical images the reflection layer will only partially cover the background. In fact, the visibility of the reflection layer depends on the relative intensity between the transmitted light from the background scene and the reflected light. Hence we often observe large regions where no reflection is visible at all, even for scenes viewed entirely through a window or other glass surface.

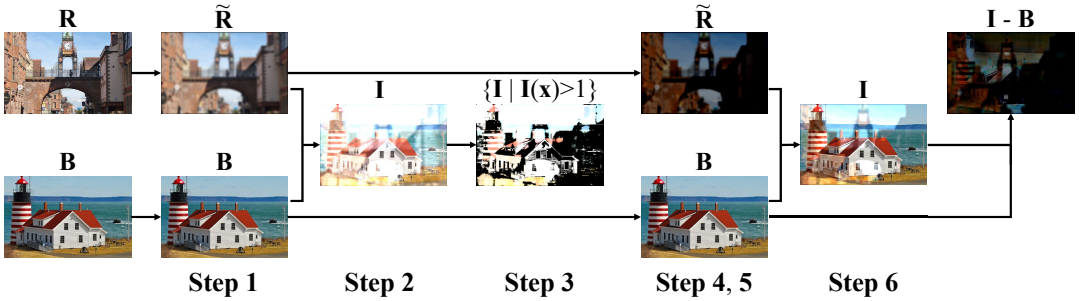
In Section 6 below we demonstrate that training with such linearly mixed images does not perform well.

2.2 Details of our alternative generation method

Our proposed pipeline is annotated in Figure 1, with an illustrative example at the top and detailed explanations of each step shown below. The crux of our method is a heuristic and simple subtraction operation: to simulate real-life reflection images, we sum up an unmodified natural image $\mathbf{B}$ and another attenuated natural image $\mathbf{R}$, which is subtracted by one single, adaptively-computed value across the whole image (see Step 4 in Figure 1).

The advantages of this subtraction strategy (which replaces any scaling operation) can be summarized as follows:

- First, the oversaturation side effects in $\mathbf{I}$ generated by directly adding up $\mathbf{B}$ and $\tilde{\mathbf{R}}$ are eliminated mostly; see the difference of generated reflection images $\mathbf{I}$ in Step 2 and Step 6 of Figure 1.
- Second, large gradients or abrupt color transitions in the original reflection layer are well maintained, as subtracting a scalar from the whole image does not affect luminance differences while the scaling operation used by naive mixing does.
- Third, strong reflections can occur as in real-world cases, especially when the given background layer is relatively dark such that oversaturation is insignificant and the subtracted part of the reflection in Step 4 (which is proportional to the mean) is small. This adaptively enables a wider color range for the reflection layer when the background is weak.
- Lastly, reflection-free regions can also naturally arise, since the subtraction and color clipping may lead to zero brightness in the reflection layer.



Randomly pick two natural images normalized to $[0, 1]$ as background layer $\mathbf{B}$ and reflection layer $\mathbf{R}$ respectively, then:

Step 1. $\tilde{\mathbf{R}} \leftarrow gauss\_blur_{\sigma}(\mathbf{R})$ with $\sigma \sim \mathcal{U}(2, 5)$

▷ *Blur the reflection layer with a Gaussian filter. The standard deviation of the Gaussian filter is randomly picked in a wide range of* $[2, 5]$ *to accommodate more diversity in the reflection layer.*

Step 2. $\mathbf{I} \leftarrow \mathbf{B} + \tilde{\mathbf{R}}$

▷ *Tentatively generate the synthesized image by directly adding up the background layer and the (blurred) reflection layer.*

Step 3. $m \leftarrow mean(\{\mathbf{I}(\mathbf{x}, c) \mid \mathbf{I}(\mathbf{x}, c) > 1, \forall \mathbf{x}, \forall c = 1,2,3\})$

▷ *Calculate the mean of all the color (R, G, B) values in the synthesized image that are out of color range (i.e., above 1).*

Step 4. $\tilde{\mathbf{R}}(\mathbf{x}, c) \leftarrow \tilde{\mathbf{R}}(\mathbf{x}, c) - \gamma \cdot (m - 1), \forall \mathbf{x}, \forall c$

▷ *Reduce the color (R, G, B) values of the reflection layer by the calculated mean (the overflowed fraction only; hence* $m - 1$*). We also magnify the subtracted part by a scalar* γ *set as 1.3 to further attenuate the reflection. We found that this heuristic for reflection layer attenuation works well in practice (and our networks are trained exclusively via data obtained this way), although there may exist other possible choices.*

Step 5. $\tilde{\mathbf{R}} \leftarrow clip_{[0,1]}(\tilde{\mathbf{R}})$

▷ *Clip the color values of the reflection layer to the valid range (to deal with negative values arisen due to the subtraction in Step. 4).*

Step 6. $\mathbf{I} \leftarrow clip_{[0,1]}\left(\mathbf{B} + \tilde{\mathbf{R}}\right)$

▷ *Generate the final image by adding up the background layer and the modified reflection layer followed by clipping the color to the valid range (to deal with values above 1).*

Output $\mathbf{I}$ as the synthesized image with $\mathbf{B}$ as the ground-truth background layer, and $\mathbf{I} - \mathbf{B}$ as the ground-truth reflection layer.

Figure 1: Our reflection image data synthesis pipeline for weakly-supervised learning. We illustrate the process with an example (*top*), followed by detailed step-by-step explanations (*bottom*).

3 Performance using an I-CNN without predicted edges

In Section 5.1 of the main paper, we described the performance of a single I-CNN (which is equivalent to our CEILNet pipeline without the E-CNN and edge supervision) as more convolution layers are stacked. Figure 2 presents the detailed results. As can be seen, using I-CNN only the performance gets saturated very quickly, and the best performing 70 layer I-CNN (PSNR 33.37 dB) lags far behind the CEILNet (PSNR 37.10 dB), even though the latter has fewer parameters (CEILNet has basically the same number of parameters as a 64 layer I-CNN).

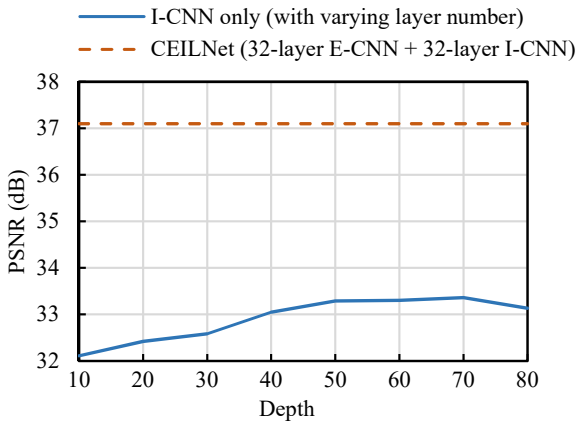


Figure 2: Performance of a simple I-CNN network with respect to network depth for the task of learning an L_0 smoothing filter [10]. When the depth is 64, the I-CNN has basically the same number of parameters as our proposed CEILNet. Clearly, the benefit of the latter is not just a larger parameter set, but rather its purposeful integration with the E-CNN.

4 Edge map visualization and analysis

In this section, we present more edge maps generated by our E-CNN in both image smoothing (Figure 3 top two rows) and reflection removal (Figure 3 bottom two rows).

In the image smoothing task, fundamental image constituents, *e.g.*, salient edges, will be maintained, and insignificant details will be diminished. In our method, the edge map of the target image is predicted by E-CNN, then used to process the input image and generate the result with I-CNN. The two examples in Figure 3 show that our E-CNN is able to remove the insignificant details, meanwhile preserving prominent edges and rendering them visually more distinct. With the guidance of the predicted edge map, I-CNN generates high fidelity smoothing results compared to the original filter (RTV [12] in these examples).

For separating reflection and background layers, edges can still play an important role to differentiate the two independent layers [4, 13, 14]. Different from image smoothing, in this task the desired edge map is from the entire image structure of the background layer only. As shown in the bottom two rows of Figure 3, our E-CNN can remove the edges belonging to reflections and I-CNN can reconstruct clean background images. Comparison of the results with and without edge prediction can be found in Section 6 below.

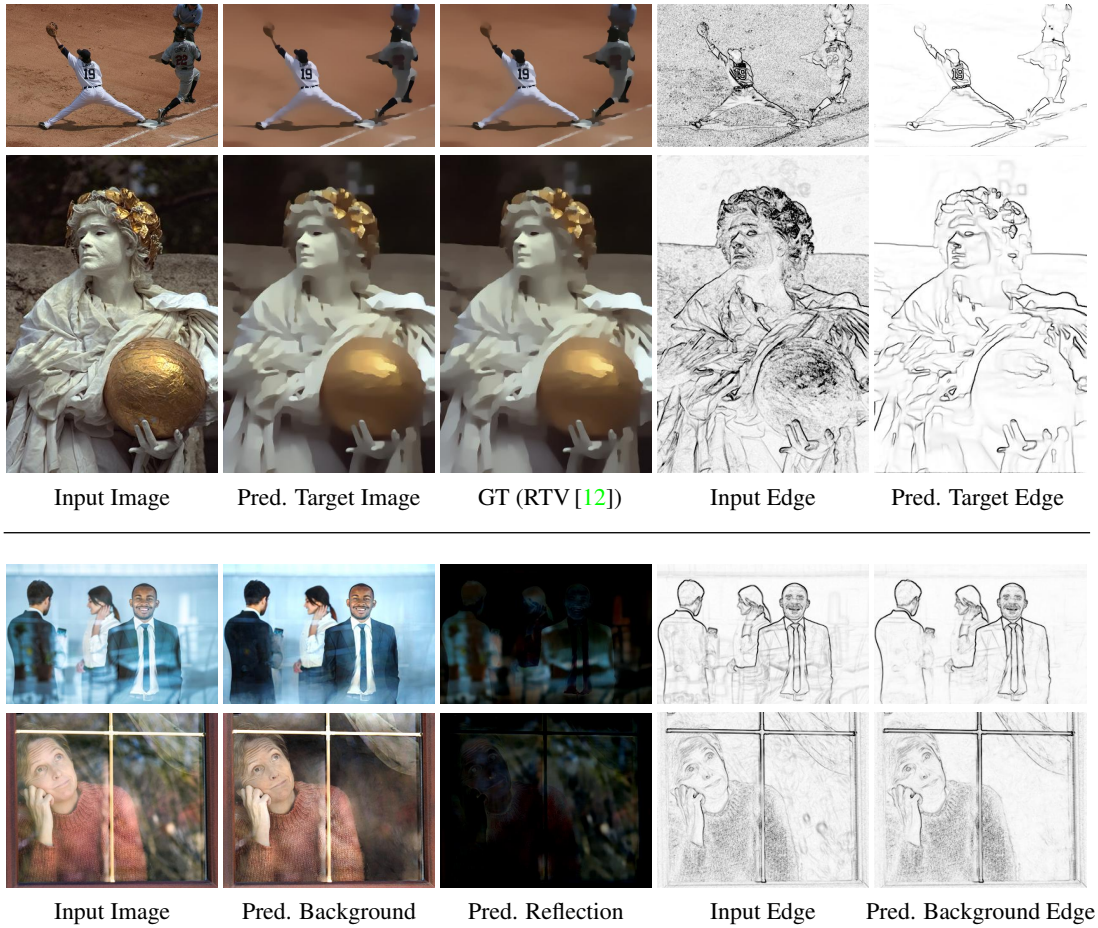


Figure 3: Visualization of the predicted edge maps in the image smoothing (top two rows) and reflection removal (bottom two rows) tasks. GT denotes ground truth.

5 Detailed comparison of different deep image smoothing networks

In the main paper, we have compared with existing deep image smoothing methods [11, 6] quantitatively and qualitatively. Here we analyze these methods and compare them with ours in more details. Additional qualitative comparisons are also provided in Figure 4.

The approach from [11] predicts the gradient map of the target image using a shallow network (3 convolution layers), and reconstructs the smoothed images by solving a separate optimization problem. One potential drawback of this method is that the optimization step completely relies on the predicted gradient map, and the gradient errors can easily spread to a surrounding area (see the boundary of "bull" and "building" cases in Figure 4). Our predicted edge map also represents sparse salient structures similar to the gradient map in [11]. But unlike [11], our image reconstruction is learned by the deep neural network, which exhibits higher robustness to errors of the predicted edge map.

The method of [6] generates smoothed images using separable recursive 1D filters based on the predicted weight maps by a CNN. Their weight map serves a similar role with a gradient map; however, it seems impracticable that one single (parameter-free) filtering technique in [6] can well approximate various existing edge-aware filters of different effects and disparate algorithm details, even if the weight map in [6] is generated by a CNN trained for each filter. As can be seen in the row of Figure 4, their learned L_0 filter turns out to have some obvious artifacts. In contrast, our filtering process is based on a network with parameters trained for each smoothing algorithm, which can generate more accurate approximations for different smoothing algorithms.

6 Detailed comparison with our baselines and previous methods on the reflection removal task

In the main paper (Figure 5), we have presented one real image result comparison between our CEILNet and two of its baselines: 1) CEILNet trained using naïvely generated reflection images and 2) I-CNN only (*i.e.*, without the predicted edge from E-CNN). We now provide more explanations and comparisons as follows. By "naïve", we mean linearly combining the background and reflection layer with two constant coefficients that sum up to 1 (see also Section 2.1). In our experiments, the coefficients vary in a wide range to account for different situations: $[0.6, 0.9]$ for $\mathbf{B}$ (thus $[0.1, 0.4]$ for $\mathbf{R}$). As shown in Figure 5, CEILNet-naïve can hardly remove real-life reflections; its results are clearly not comparable to CEILNet, which demonstrated the effectiveness of our proposed training image synthesis method. The results from an I-CNN only are better than CEILNet-naïve with more reflections removed, but are still clearly inferior compared to CEILNet. The advantage of our edge prediction is therefore also demonstrated.

The method of [5], perhaps the most closely related algorithm to ours[1], shares some similar properties as CEILNet-naïve. It assumes that (*i*) it is less likely to have abrupt color transition in the reflection layer, and (*ii*) reflection is almost everywhere in the image (*i.e.*, reflection-free regions are rare). As can been seen in the last row of Figure 5, it tends to extract reflections that are very blurry and cover the whole image. Note that the fourth example in (d) is an image collected from [5]. On such cases where the assumptions from [5] are valid, our method performs comparably with [5]. But on other cases that are more difficult (and yet more common, such as other images in Figure 5 and the images in Section 7), our method excels.

[1]The method of [8] also performs reflection removal from a single image automatically like ours. However, their method is restricted to images that contain ghost effects with two ghost layers. For reflection images beyond this limited scope (such as most of the images in our main paper and this supplementary file), their method does not work at all.

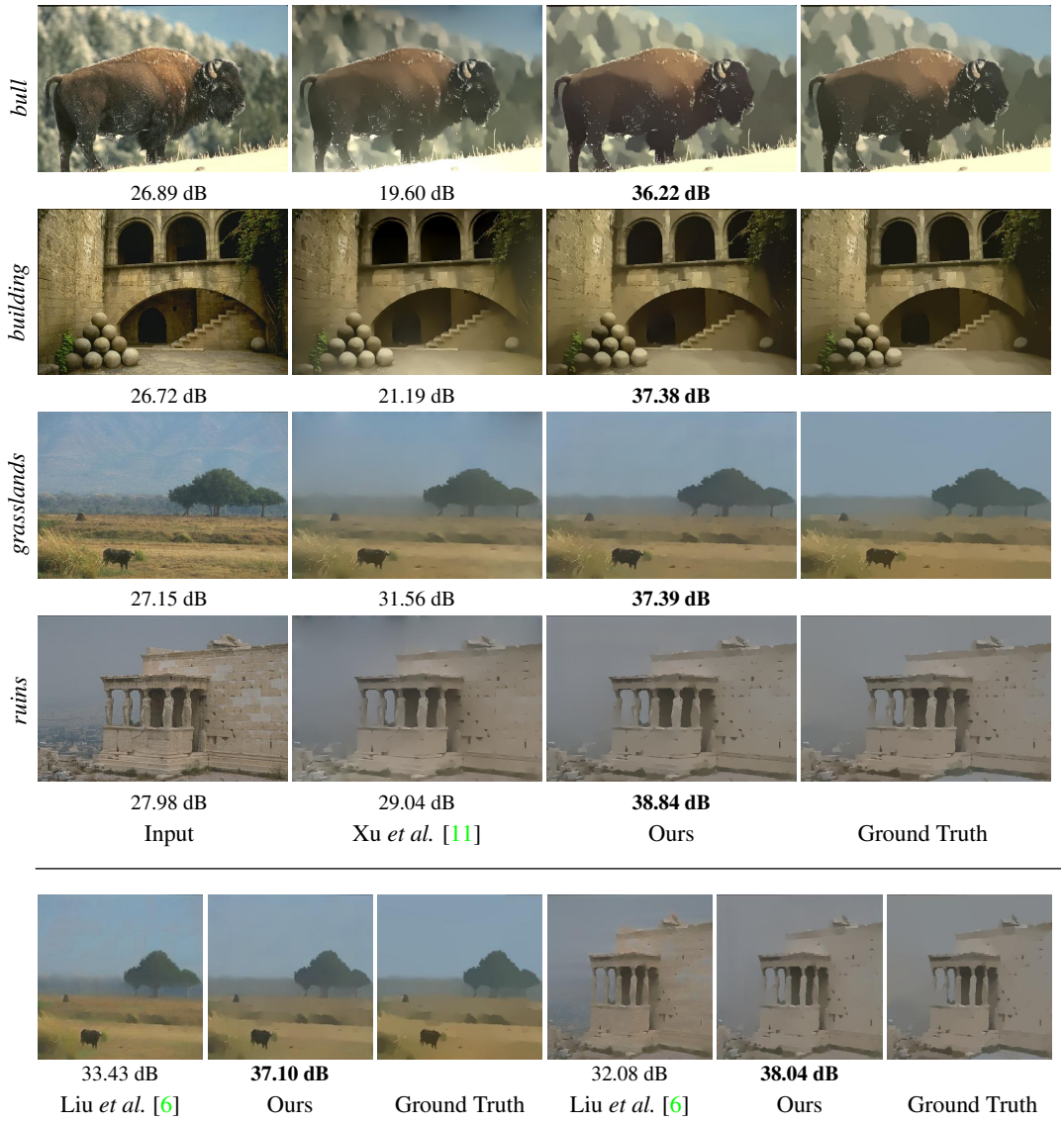


Figure 4: Qualitative comparison on the image smoothing task. All the methods are trained to approximate L_0 smoothing [10]. Top: Comparison with Xu *et al.* [11]. Bottom: Comparison with Liu *et al.* [6] on the 256×256 image size. Our results are visually much closer to the ground truth. The numbers show the PSNR values.

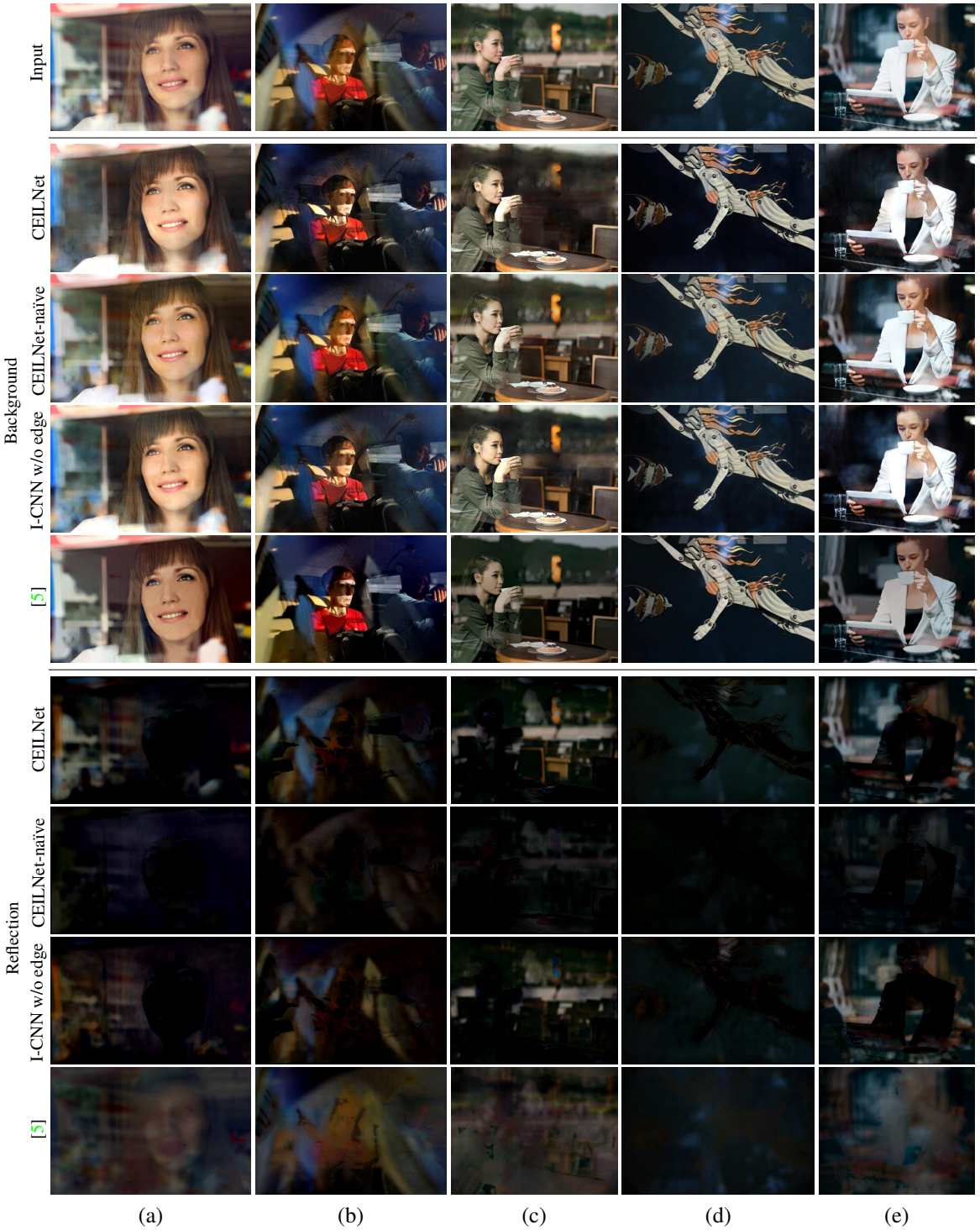


Figure 5: Qualitative comparison on real reflection images with our baselines and Li and Brown [5].

7 More results for reflection removal

This section presents more qualitative results on synthetic reflection images (Figure 6, 7) and real-world photographs (Figure 8, 9, 10, 11).

We emphasize that, because no existing database of real refection images exists with ground truth, in all cases our model was trained solely using images generated synthetically via the process described in Section 2. Moreover, given the extreme difficulty of separating a single image into background and reflection layers, no existing algorithm of any kind succeeds on these real-world cases. Nonetheless, our approach still produces reasonable results in most situations, *i.e.*, even when reflections cannot be completely removed, the recovered background images are significantly clean relative to the original.

Figure 6: More visual results of our CEILNet on synthetic reflection images. GT denotes ground truth.

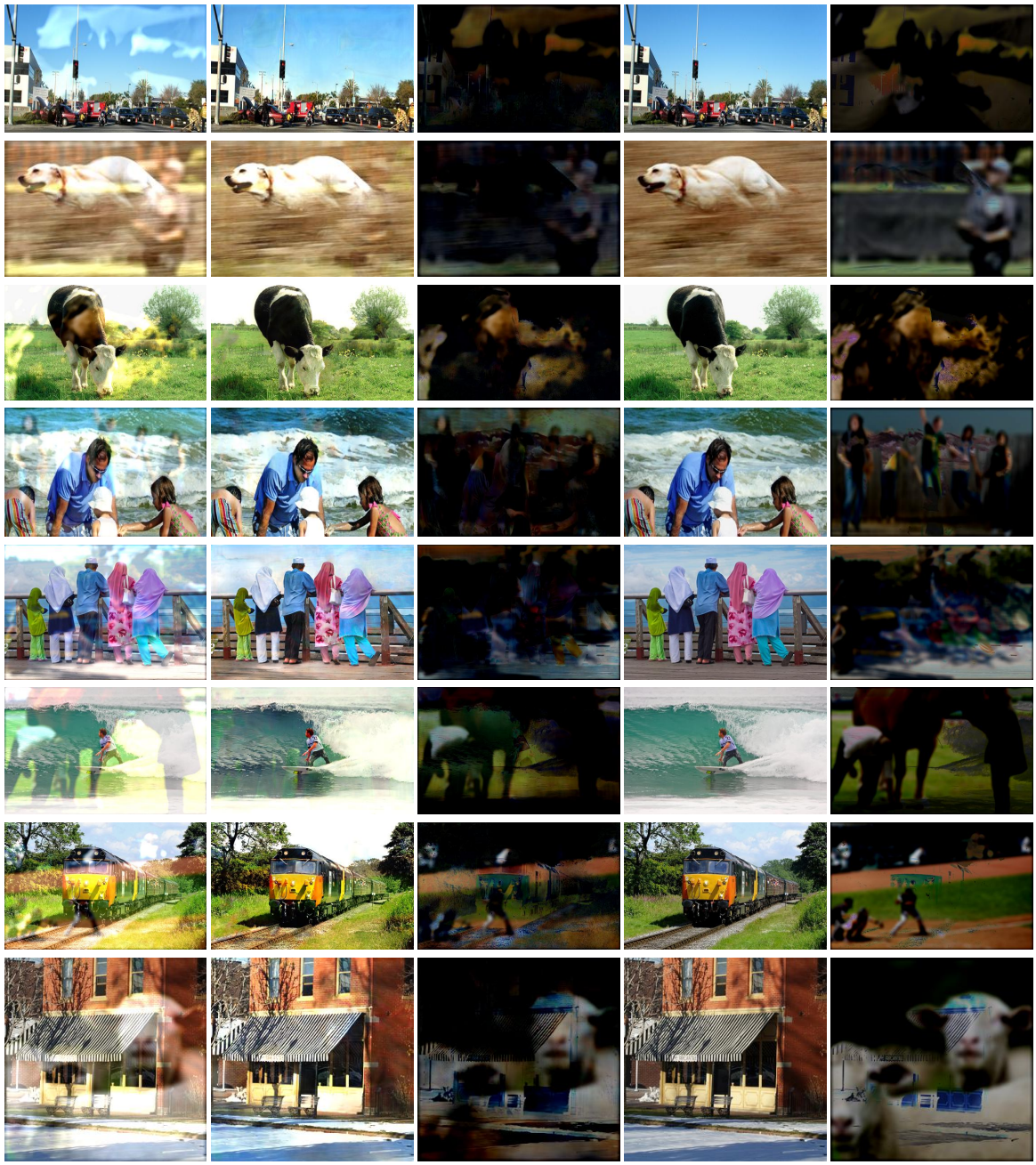
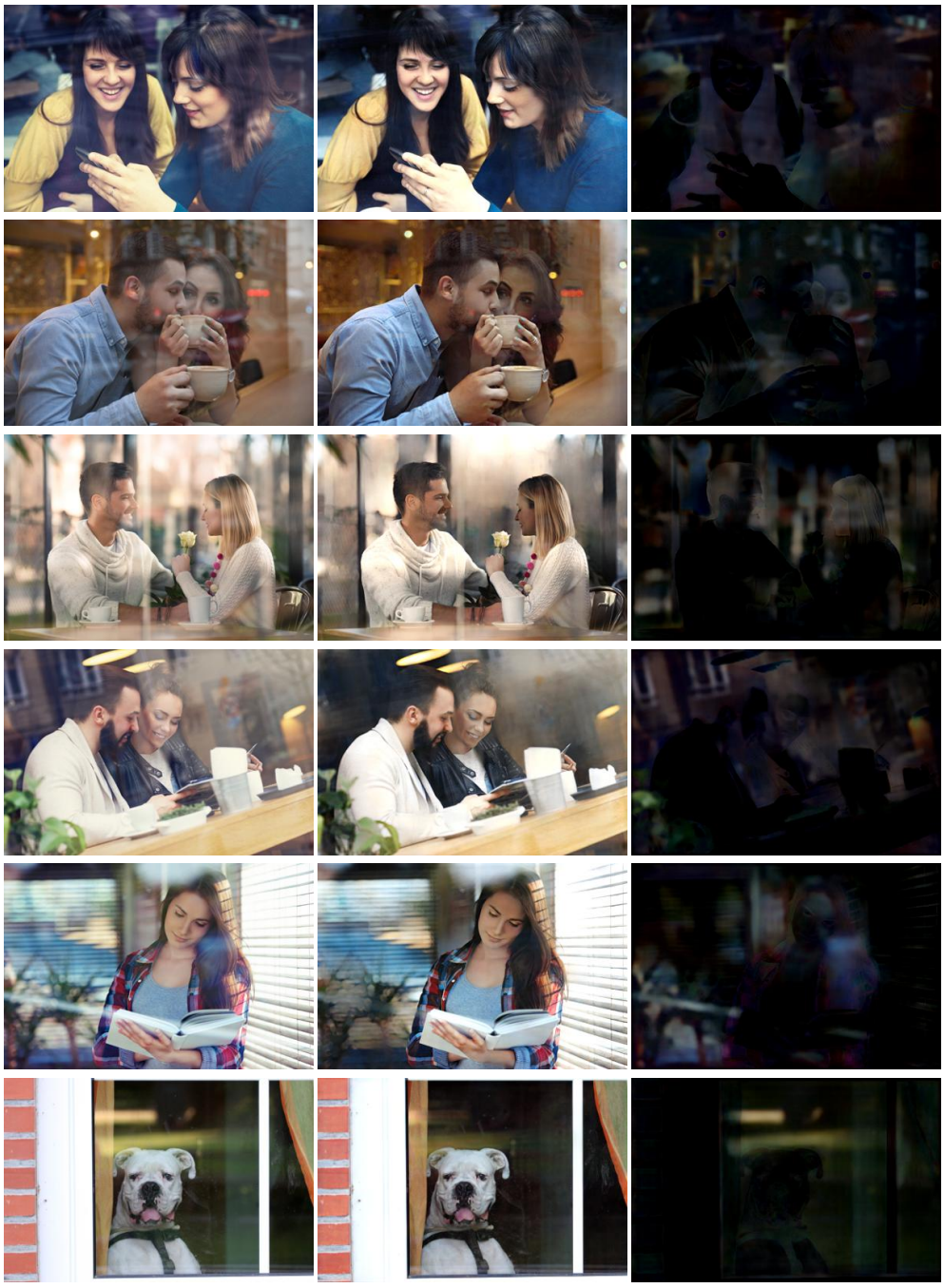
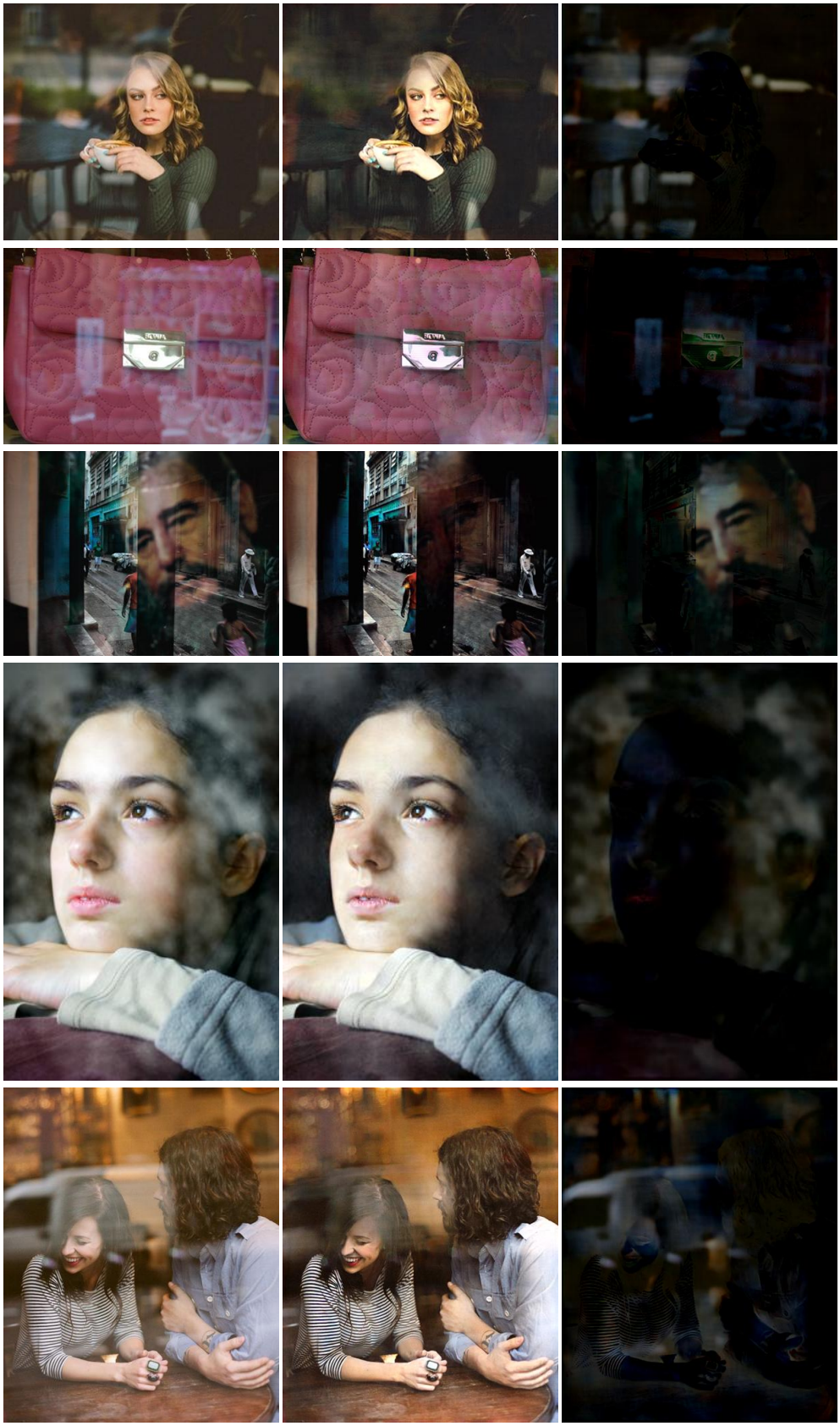


Figure 7: More visual results of our CEILNet on synthetic reflection images. GT denotes ground truth.



Input Predicted Background Predicted Reflection

Figure 8: More visual results of our CEILNet on real reflection images.



Input Predicted Background Predicted Reflection

Figure 9: More visual results of our CEILNet on real reflection images.

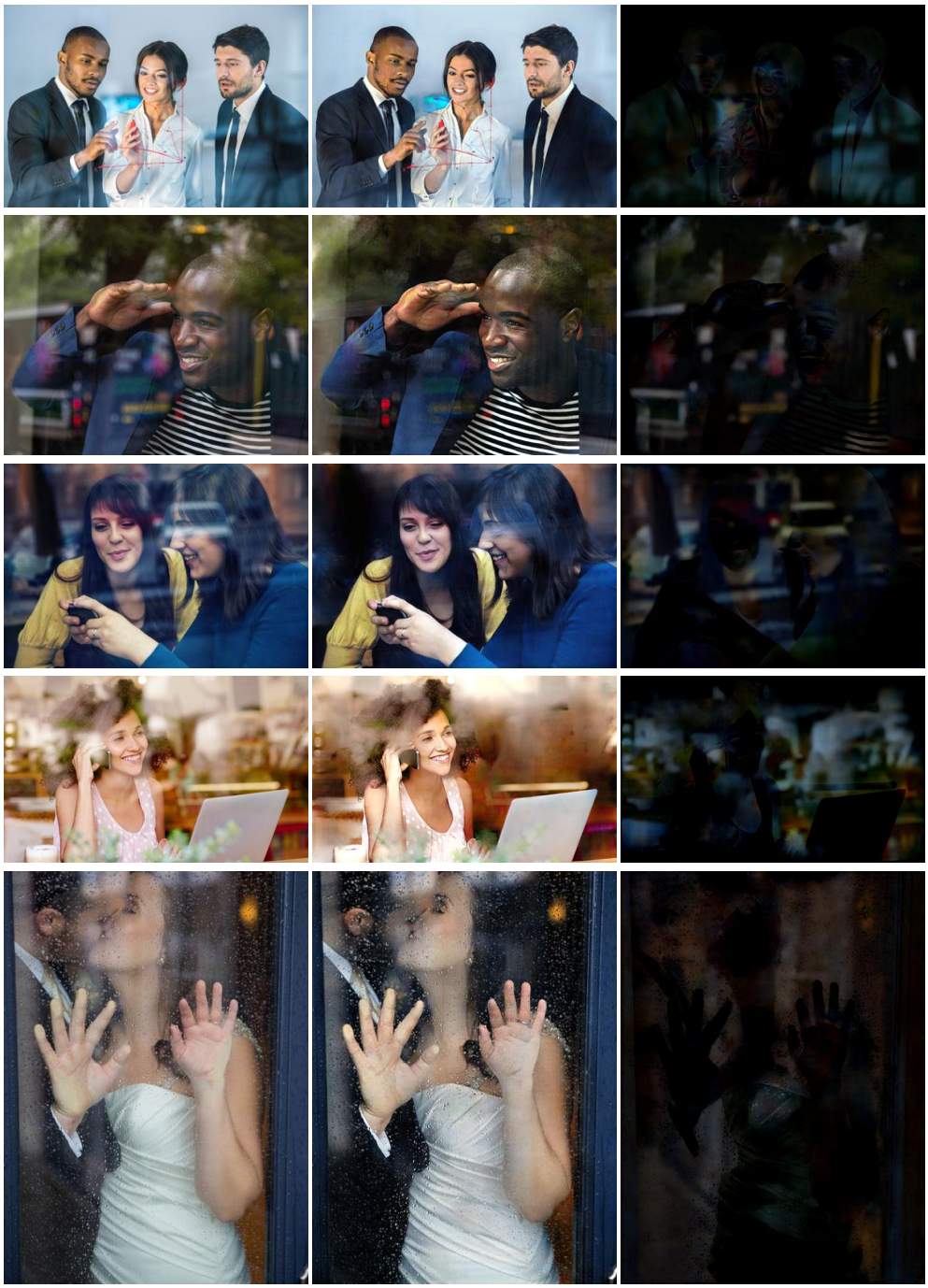
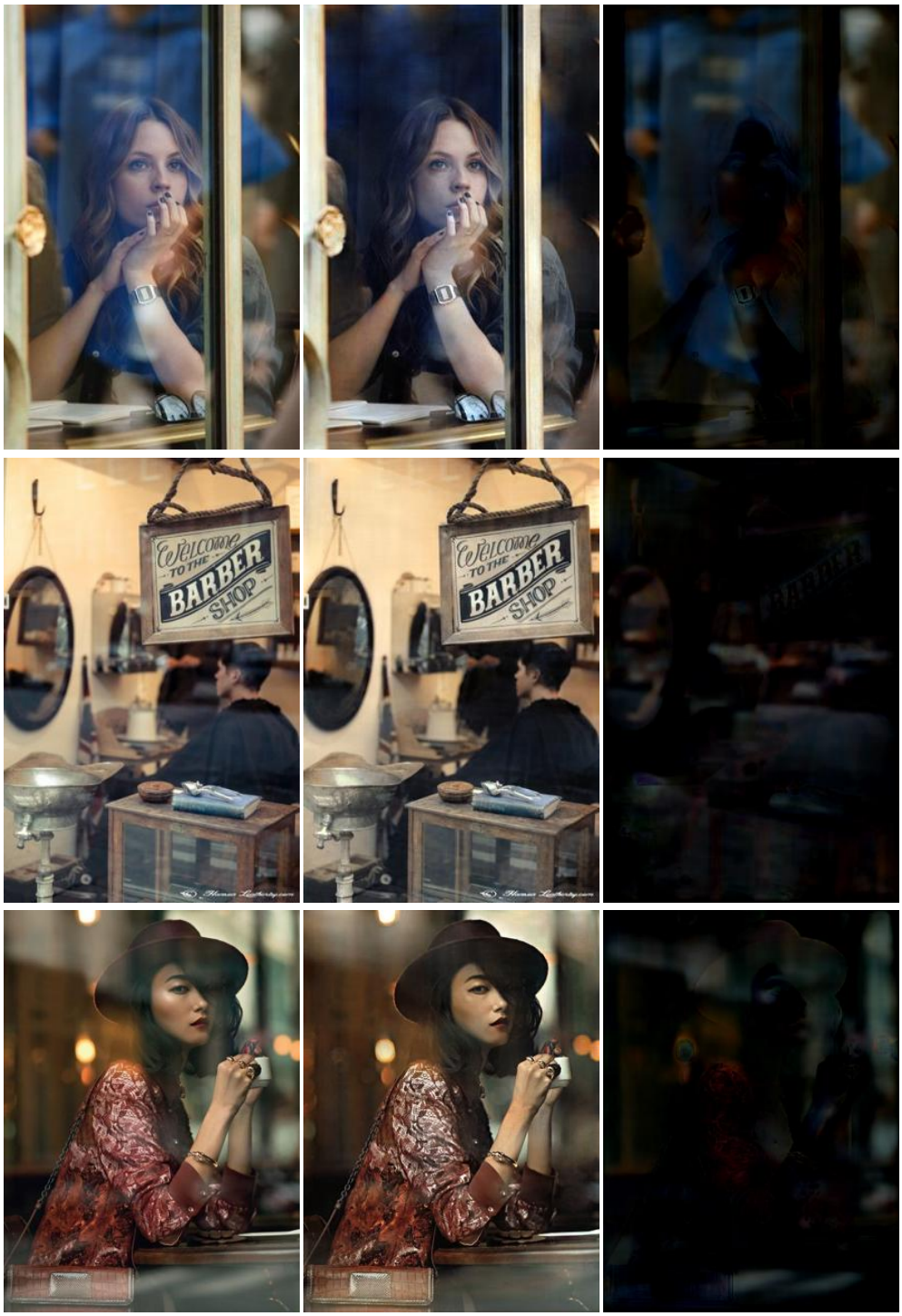


Figure 10: More results of our CEILNet on real reflection images.

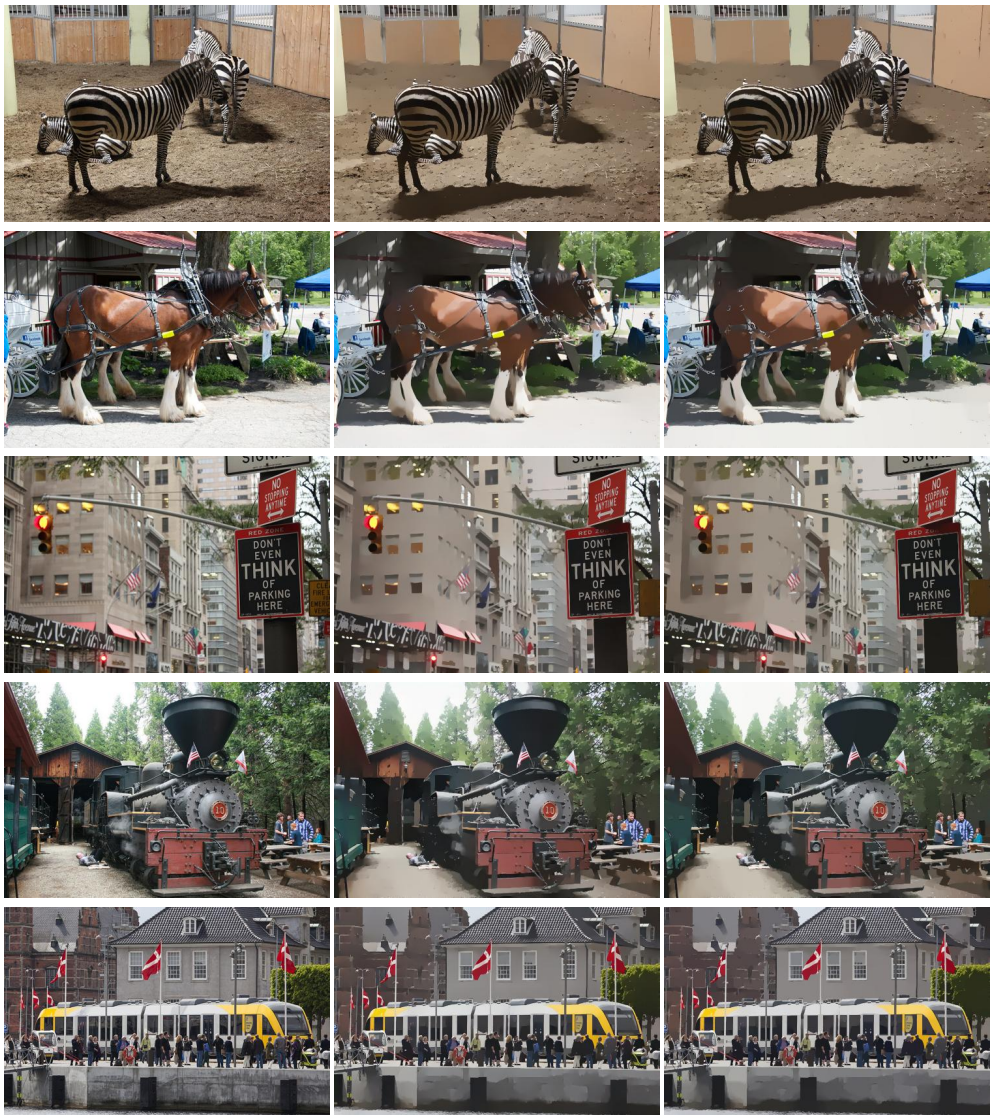


Input Predicted Background Predicted Reflection

Figure 11: More visual results of our CEILNet on real reflection images.

8 More results for image smoothing

In this section, we present more visual results for the image smoothing tasks (Figure 12, 13, 14, 15 16 for approximating L_0 [10], L_1 [1], RTV [12], RGF [15], WLS [2] filters respectively).

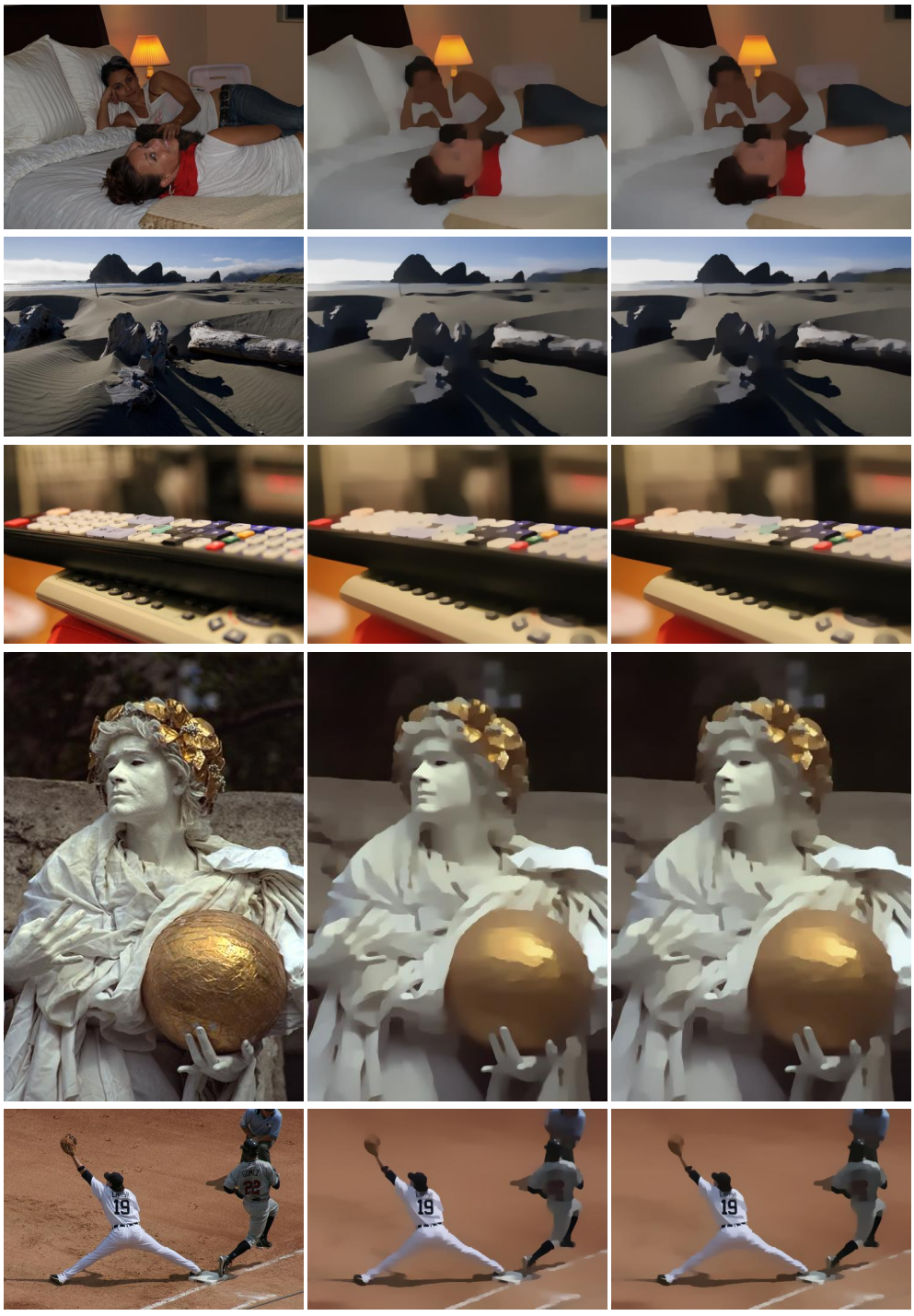


Input | Our Result | Ground Truth (L_0 [10])

Figure 12: Approximation of the L_0 image smoothing algorithm [10] using our CEILNet.

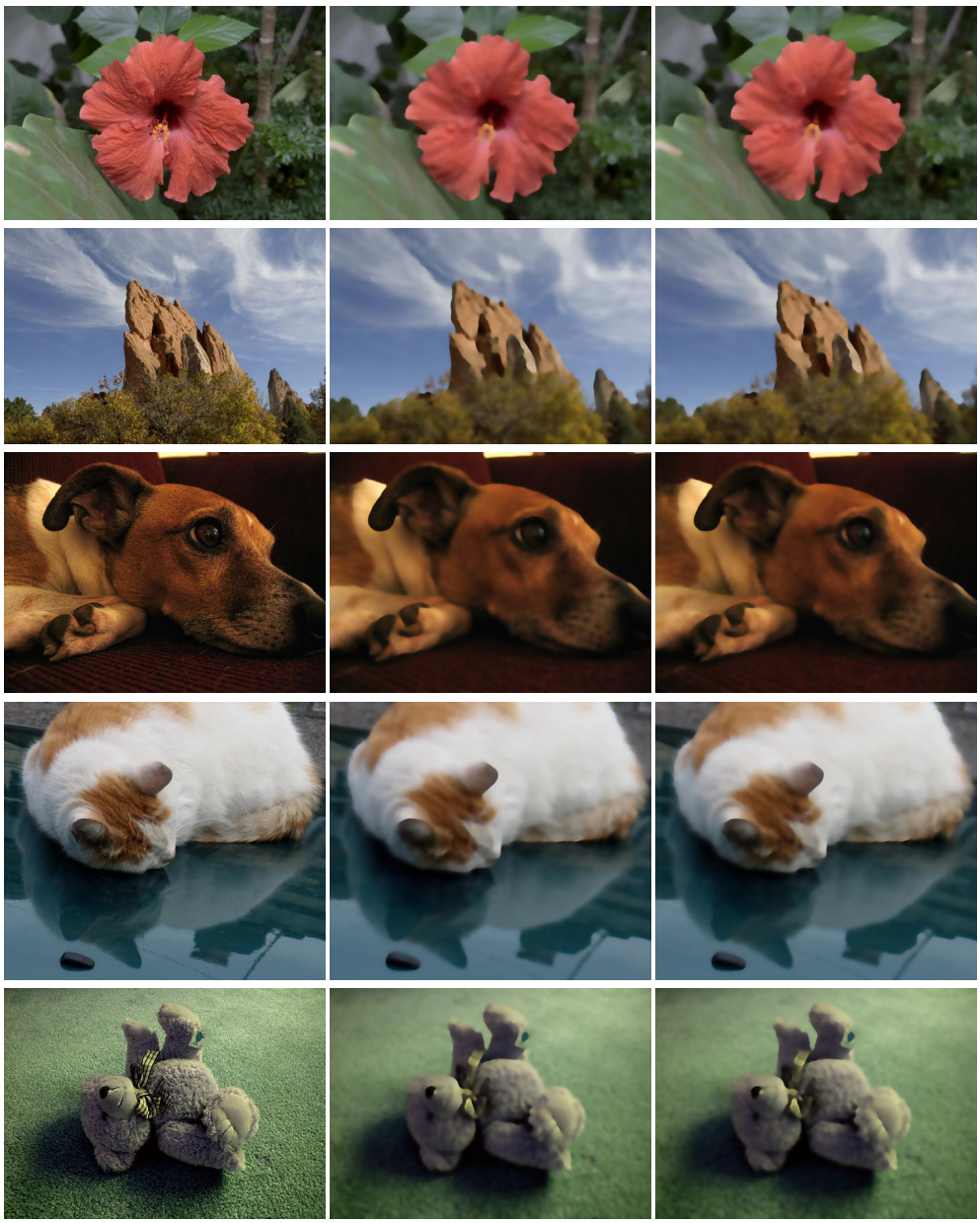
Input Our Result Ground Truth (L_1 [1])

Figure 13: Approximation of the L_1 image smoothing algorithm [1] using our CEILNet.



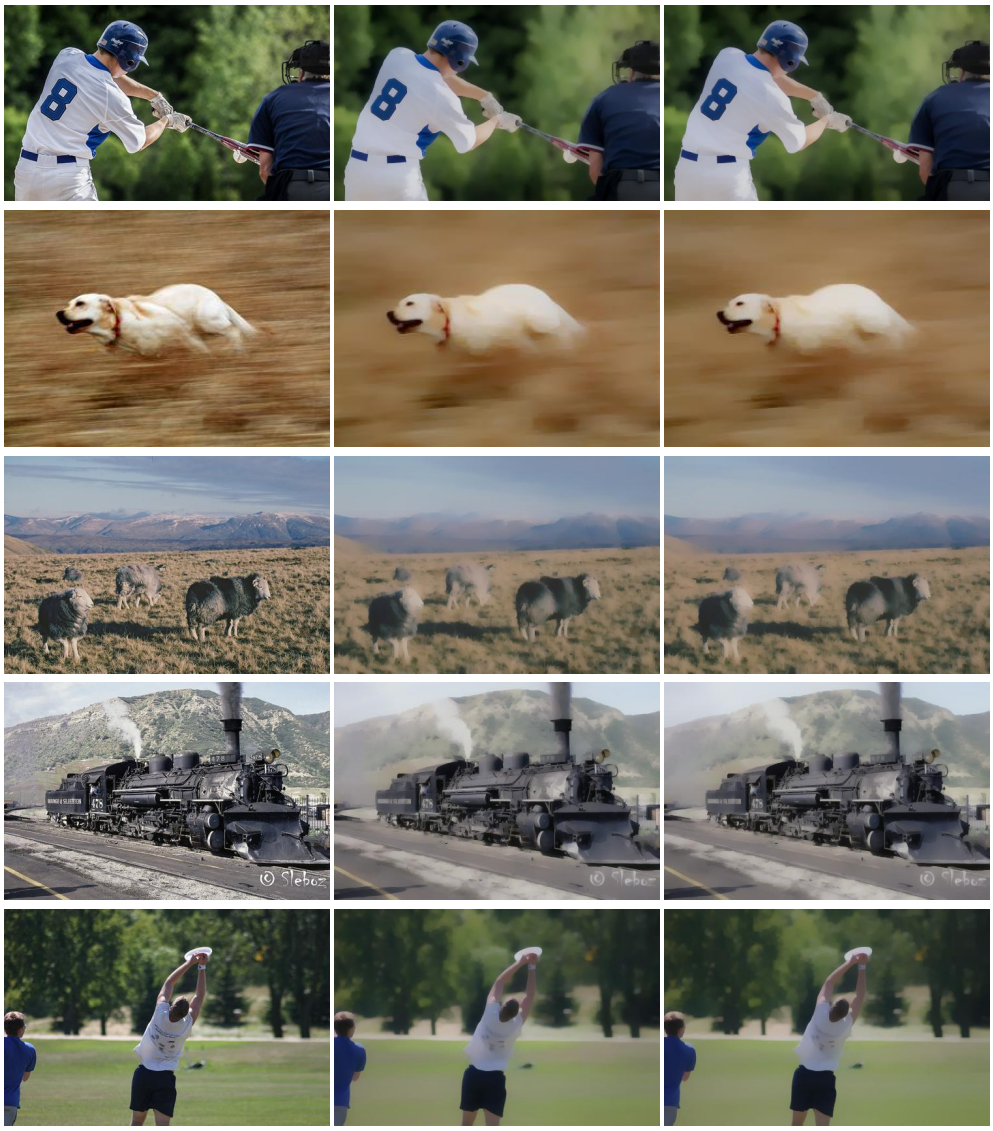
Input | Our Result | Ground Truth (RTV [12])

Figure 14: Approximation of the RTV image smoothing algorithm [12] using our CEILNet.



Input | Our Result | Ground Truth (RGF [15])

Figure 15: Approximation of the RGF image smoothing algorithm [15] using our CEILNet.



Input | Our Result | Ground Truth (WLS [2])

Figure 16: Approximation of the WLS image smoothing algorithm [2] using our CEILNet.

References

[1] S. Bi, X. Han, and Y. Yu. An L_1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *ACM Transactions on Graphics (TOG)*, 34(4):78, 2015. 15, 16

[2] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics (TOG)*, 27(3), 2008. 15, 19

[3] X. Guo, X. Cao, and Y. Ma. Robust separation of reflection from multiple images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2187–2194, 2014. 2

[4] Y. Li and M. S. Brown. Exploiting reflection change for automatic reflection removal. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2432–2439, 2013. 5

[5] Y. Li and M. S. Brown. Single image layer separation using relative smoothness. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2752–2759, 2014. 2, 6, 8

[6] S. Liu, J. Pan, and M.-H. Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *European Conference on Computer Vision (ECCV)*, 2016. 6, 7

[7] B. Sarel and M. Irani. Separating transparent layers through layer information exchange. In *European Conference on Computer Vision (ECCV)*, pages 328–341, 2004. 2

[8] Y. Shih, D. Krishnan, F. Durand, and W. T. Freeman. Reflection removal using ghosting cues. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3193–3201, 2015. 6

[9] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 246–253, 2000. 2

[10] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via L_0 gradient minimization. In *ACM Transactions on Graphics (TOG)*, volume 30, page 174, 2011. 4, 7, 15

[11] L. Xu, J. S. Ren, Q. Yan, R. Liao, and J. Jia. Deep edge-aware filters. In *International Conference on Machine Learning (ICML)*, pages 1669–1678, 2015. 6, 7

[12] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via natural variation measure. *ACM Transactions on Graphics (TOG)*, 2012. 5, 15, 17

[13] T. Xue, M. Rubinstein, C. Liu, and W. T. Freeman. A computational approach for obstruction-free photography. *ACM Transactions on Graphics (TOG)*, 34(4):79, 2015. 5

[14] J. Yang, H. Li, Y. Dai, and R. T. Tan. Robust optical flow estimation of double-layer images under transparency or reflection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1410–1419, 2016. 2, 5

[15] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In *European Conference on Computer Vision (ECCV)*, pages 815–830, 2014. 15, 18