A Late Fusion CNN for Digital Matting: Supplementary Materials

Yunke Zhang[1], Lixue Gong[1], Lubin Fan[2], Peiran Ren[2], Qixing Huang[3], Hujun Bao[1], and Weiwei Xu*[1]

[1]Zhejiang University [2]Alibaba Group [3]University of Texas at Austin

{yunkezhang, gonglx}@zju.edu.cn, {lubin.flb, peiran.rpr}@alibaba-inc.com, huangqx@cs.uteaxs.edu,{bao, xww}@cad.zju.edu.cn

1. Introduction

In this supplementary material, we provide the detailed description of the network structure of our late fusion CNN and additional image matting results on the human matting testing dataset, the composition-1k testing dataset and Internet images.

2. Network Structure Details

The structures of the segmentation network and the fusion network are shown in Tab. 1 and Tab. 2, respectively. In the segmentation network, we use DenseNet-201 [2] as our encoder, and the growth rate of the dense block is set to 32. We also adapt the technique from feature pyramid network [3], adding side losses on different output scales. Since the network structures of two decoder branches in the segmentation network are same, we only describe the details of one decoder branch in Tab.1. The fusion network is a fully convolutional network without downsampling to avoid the loss of details.

Note that we take the input image of the resolution 512×512 as an example in Tab. 1 and Tab. 2. The resolutions of output feature maps in the network are therefore computed accordingly. Since our network is fully convolutional, the RGB images whose longer side is less than or equal to 800 pixels can be input to the network.

3. Additional Testing Datasets Matting Results

From Fig. 1 to Fig. 3, we show more results of our method on the human matting testing dataset. In Fig. 4 and Fig. 5, we show more results of our method on the composition-1k testing dataset [5]. We also compare our method with two other approaches, DCNN [1] and DIM [5]. All trimaps are generated by dilating 25 pixels from the ground-truth matte. **Please zoom in on the digital copy to see the details of the mattes.**

*Corresponding author. The authors from Zhejiang University are affiliated with the State Key Lab of CAD&CG.

4. Additional Internet Images Matting Results

In Fig. 6 and Fig. 7, we show the results of our method on images collected from the Internet. Our collection contains different types of foreground objects including human portraits, human full-body images, plants, nets, plush toys, animals and glasses. Human portraits and full-body images are inferred with the model that is trained on our human image matting dataset. Other images are inferred with the model that is trained on the DIM dataset. **Please zoom in on the digital copy to see the details of the mattes.**

References

[1] D. Cho, Y.-W. Tai, and I. Kweon. Natural image matting using deep convolutional neural networks. In *The European Conference on Computer Vision (ECCV)*, pages 626–643. Springer, 2016. 1, 4, 5, 6, 7, 8

[2] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[3] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017. 1

[4] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia. Deep automatic portrait matting. In *European Conference on Computer Vision (ECCV)*, pages 92–107. Springer, 2016. 9

[5] N. Xu, B. L. Price, S. Cohen, and T. S. Huang. Deep image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 4, 2017. 1, 4, 5, 6, 7, 8

Block name	Output size	Filter size
Encoder		
conv1+BN+ReLU	$256 \times 256 \times 64$	7×7, 64, stride 2
dense_block1	$128 \times 128 \times 256$	3×3 max pool, stride 2 dense block $\times$ 6, growth rate = 32
trans1+BN+ReLU	$64 \times 64 \times 128$	1×1, 128 2×2 average pool, stride 2
dense_block2	$64 \times 64 \times 512$	dense block $\times$ 12, growth rate = 32
trans2+BN+ReLU	$32 \times 32 \times 256$	1×1, 256 2×2 average pool, stride 2
dense_block3	$32 \times 32 \times 1792$	dense block $\times$ 48, growth rate = 32
trans3+BN+ReLU	$16 \times 16 \times 896$	1×1, 896 2×2 average pool, stride 2
dense_block4	$16 \times 16 \times 1920$	dense block $\times$ 32, growth rate = 32
Decoder		
skip3+BN+ReLU	$32 \times 32 \times 128$	1×1, 128 input: skip-connection from dense_block3
deconv4+BN+ReLU	$32 \times 32 \times 128$	3×3, 128 input: the sum of (1) 2×2 bilinear upsampling from dense_block4, convoluted with a filter 1×1, 128; (2) skip3
skip2+BN+ReLU	$64 \times 64 \times 128$	1×1, 128 input: skip-connection from dense_block2
deconv3+BN+ReLU	$64 \times 64 \times 128$	3×3, 128 input: the sum of (1) 2×2 bilinear upsampling from dense_block3, convoluted with a filter 1×1, 128; (2) skip2
skip1+BN+ReLU	$128 \times 128 \times 128$	1×1, 128 input: skip-connection from dense_block1
deconv2+BN+ReLU	$128 \times 128 \times 128$	3×3, 128 input: the sum of (1) 2×2 bilinear upsampling from dense_block2, convoluted with a filter 1×1, 128; (2) skip1
skip0+BN+ReLU	$256 \times 256 \times 128$	1×1, 128 input: skip-connection from conv1
deconv1+BN+ReLU	$256 \times 256 \times 128$	3×3, 128 input: the sum of (1) 2×2 bilinear upsampling from dense_block1, convoluted with a filter 1×1, 128; (2) skip0
deconv0+BN+ReLU	$512 \times 512 \times 128$	3×3, 128 input: 2×2 bilinear upsampling from conv1
Side Loss		
side3	$512 \times 512 \times 1$	1×1, 1 input: skip-connection from deconv3 output: sigmoid+bilinear upsampling to 512×512
side2	$512 \times 512 \times 1$	1×1, 1 input: skip-connection from deconv2 output: sigmoid+bilinear upsampling to 512×512
side1	$512 \times 512 \times 1$	1×1, 1 input: skip-connection from deconv1 output: sigmoid+bilinear upsampling to 512×512
side0	$512 \times 512 \times 1$	1×1, 1 input: skip-connection from deconv0 output: sigmoid

Table 1. Architecture of the segmentation network. Note that we only show the foreground decoder branch in this table, the background decoder branch has the similar structure which is not shown here. The input of the network is set to be a $512 \times 512 \times 3$ RGB image as an example. The resolution can be changed since our network is fully convolutional. The *dense block* and *growth rate* in this table are the same as in the DenseNet architecture [2]. All convolutional layers except those in the *Side Loss* are followed by the batch-normalization layers and ReLU activation layers. All Side Loss blocks are followed by a sigmoid activation function for calculating the segmentation loss. Since the network structures of two decoder branches in the segmentation network are same, we only describe the details of one decoder branch. The output of the *side0* block is the foreground/background probability map.

Block name	Output size	Filter size
rgb_conv+BN+ReLU	$512 \times 512 \times 128$	3×3, 128
fusion_conv1+BN+ReLU	$512 \times 512 \times 256$	3×3, 256 input: concatenate FG_deconv0, BG_deconv0, rgb_conv
fusion_conv2+BN+ReLU	$512 \times 512 \times 128$	3×3, 128
fusion_conv3+BN+ReLU	$512 \times 512 \times 64$	3×3, 64
fusion_conv4+BN+ReLU	$512 \times 512 \times 64$	3×3, 64
fusion_conv5+BN+ReLU	$512 \times 512 \times 256$	3×3, 256
fusion_output	$512 \times 512 \times 1$	3×3, 1 output: sigmoid

Table 2. Architecture of the fusion network. The inputs of the network are set to be a $512 \times 512 \times 3$ RGB image and two $512 \times 512 \times 128$ decoded features, *FG_deconv0* and *BG_deconv0*, from the foreground and background decoder branches (see *deconv0+BN+ReLU* block in one decoder branch described in Tab. 1). All convolutional layers except the last one are followed by the batch-normalization layers and ReLU activation functions. The last convolutional layer in the *fusion_output* block is followed by a sigmoid activation function for calculating the fusion loss. The output of the *fusion_output* block is the fusion weight map.

Figure 1. The visual comparisons on human image matting testing dataset.

Figure 2. The visual comparisons on human image matting testing dataset.

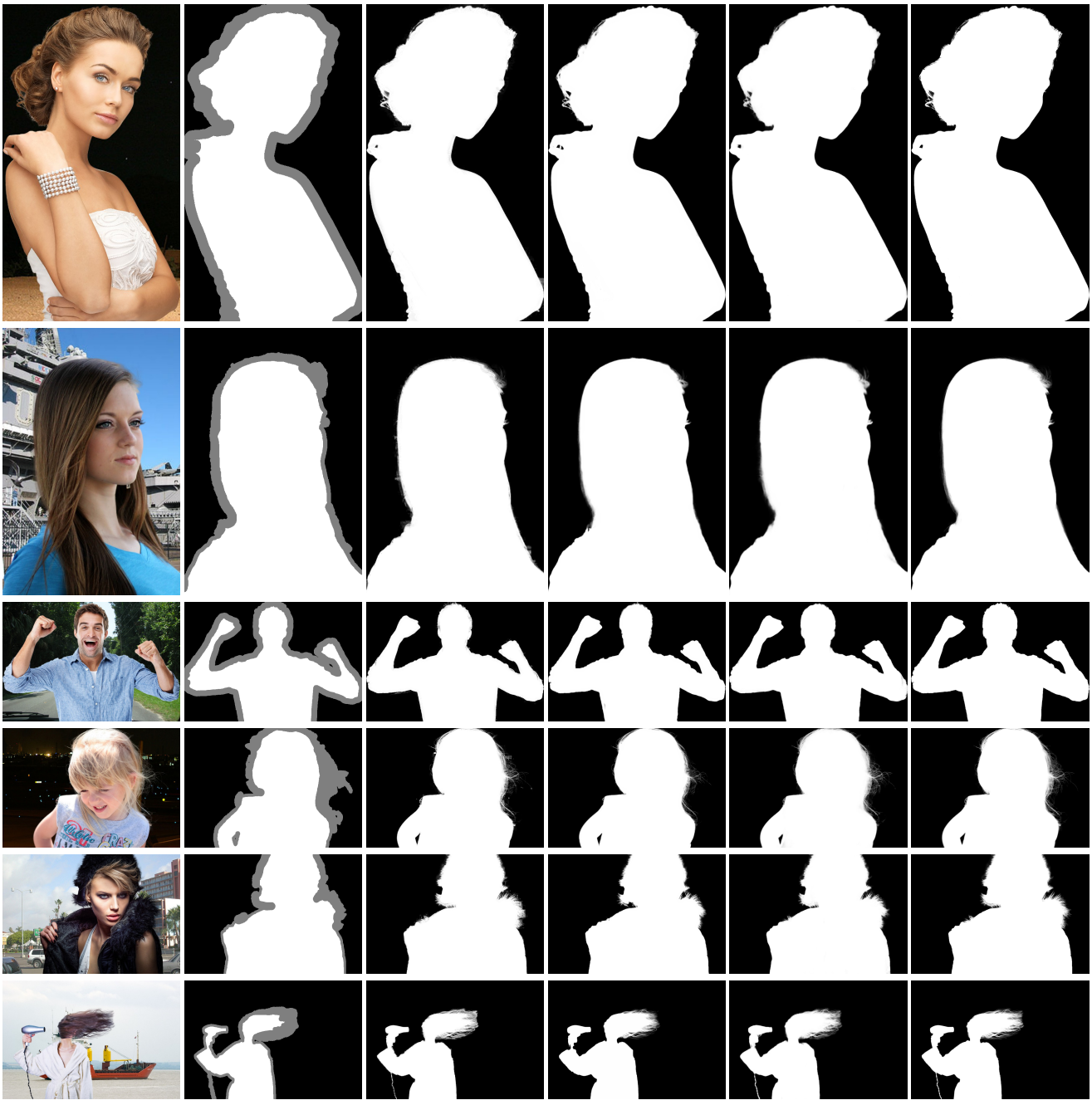

Figure 3. The visual comparisons on human image matting testing dataset.

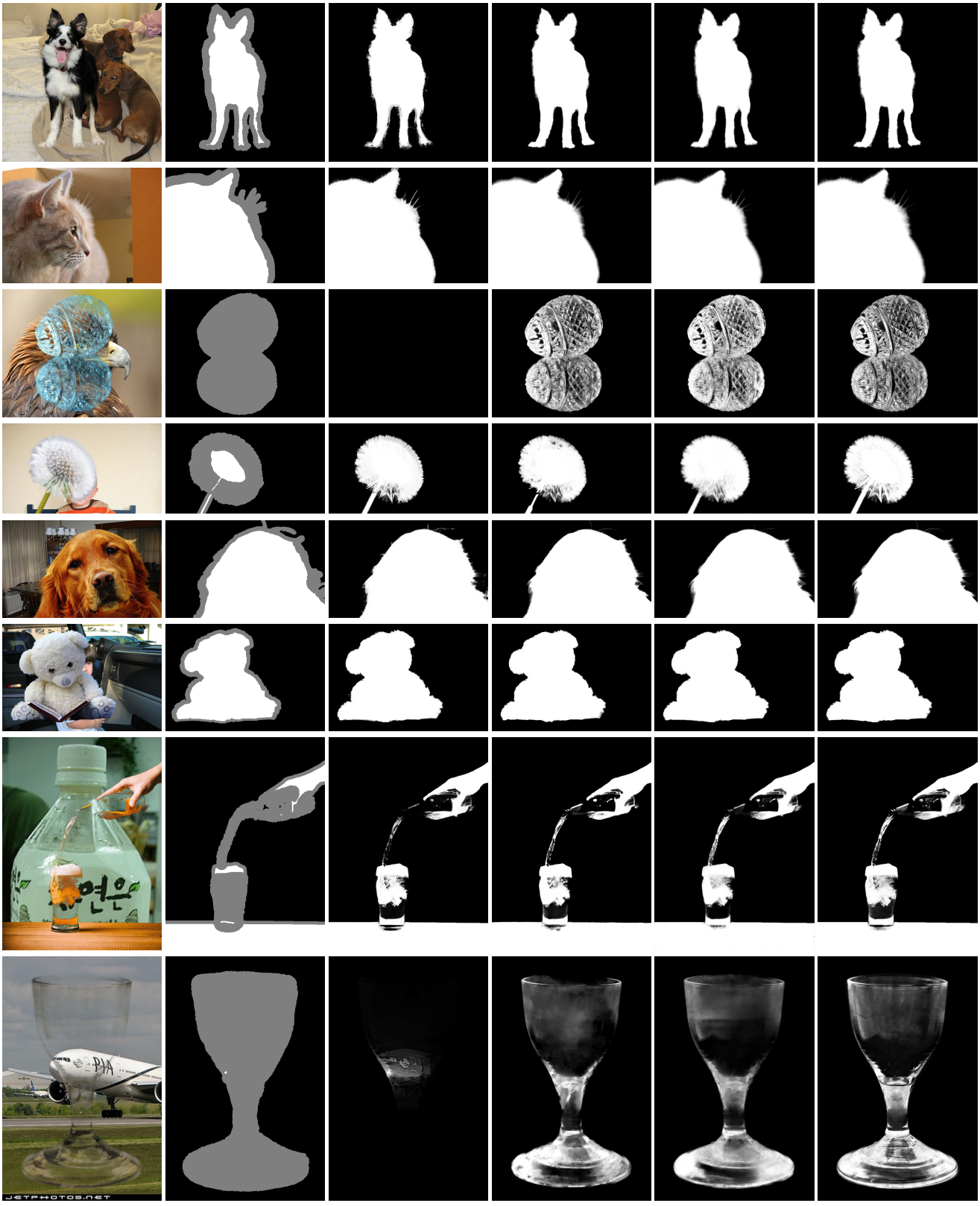

Figure 4. The visual comparisons on the composition-1k testing set.

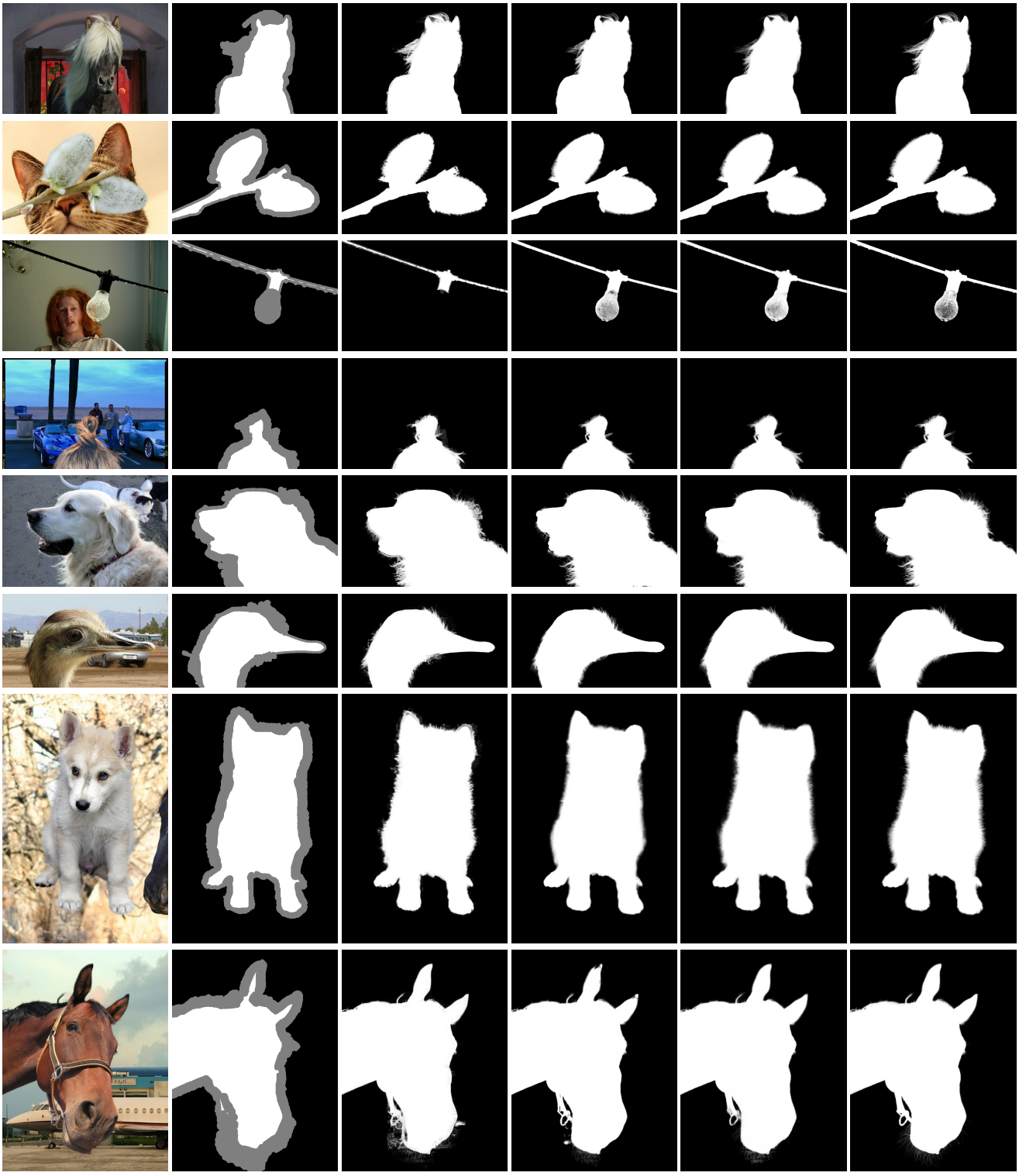

Figure 5. The visual comparisons on the composition-1k testing set.

Image Ours Image Ours

Figure 6. Matting results on images from the Internet. Images in the last row are from Deep Automatic Portrait Matting [4].

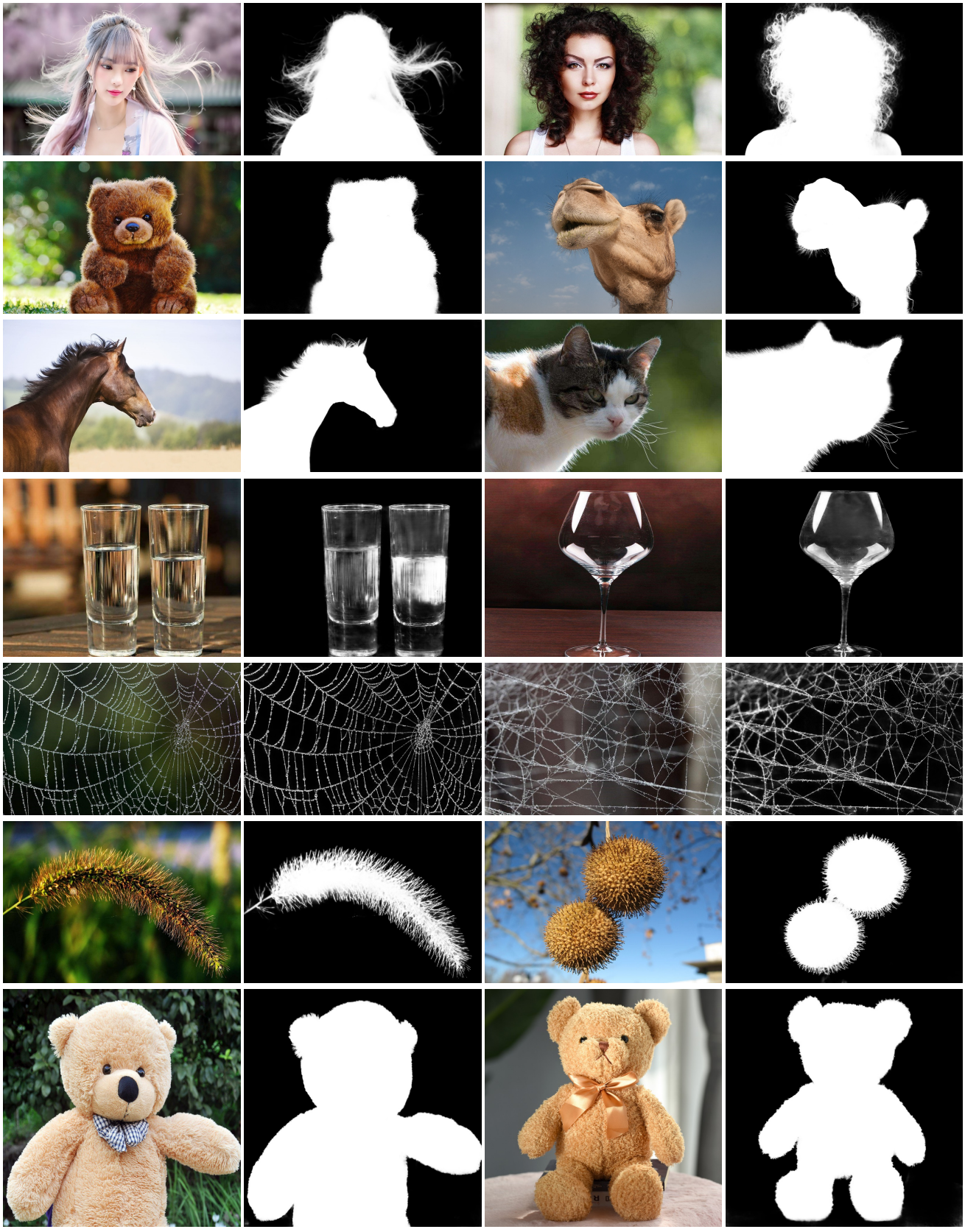

Figure 7. Matting results on images from the Internet.