Fast End-to-End Trainable Guided Filter
Supplementary Material

https://github.com/wuhuikai/DeepGuidedFilter

1. Image Processing Operators

1.1. Details of Each Operators

L_0 smoothing. L_0 smoothing [18] is effective for sharpening major edges by increasing the steepness of transition while eliminating a manageable degree of low-amplitude structures. Such an operator makes use of L_0 gradient minimization, which can identify the most important edges by penalizing the number of non-zero gradients in the image.

For generating the ground truth images, we use the official implementation of [18] with the default parameters, which could be downloaded from http://www.cse.cuhk.edu.hk/~leojia/projects/L0smoothing.

Multiscale detail manipulation. Multiscale detail manipulation (multiscale tone manipulation) [8] enhances an image by boosting features at multiple scales, which utilizes edge-preserving multiscale image decomposition based on the weighted least squares optimization framework.

Given an image, a three-level decomposition (coarse base level b and two detail levels d^1, d^2) of the **CIELAB** lightness channel is first constructed. The resulting image of manipulation can be then constructed by a non-linear combination of b, d^1 and d^2.

To generate the ground truth images, the official implementation of [8] is used, which could be obtained from http://www.cs.huji.ac.il/~danix/epd. We first generate coarse-scale, medium-scale, and fine-scale images with the default parameters. The final output is then yielded by averaging the three images.

Style transfer. Style transfer aims at transferring the photographic style of a reference image to the input image. We utilize the algorithm proposed by Aubry *et al*. [2] to generate ground truth images. Such an algorithm seeks to match both the global contrast and the local contrast between the reference image and the input image iteratively, alternating between local Laplacian filtering and histogram matching.

The official implementation of [2] is used with the default setting and the default style image. The code could be downloaded from http://www.di.ens.fr/~aubry/code/matlab_fast_llf_and_style_transfer.zip. The resulted images are grey ones, but we treat them as **RGB** images and design the network to generate outputs with three channels.

Nonlocal dehazing. The goal of image dehazing is to remove some of the effects of atmospheric absorption and scattering. Recently, Berman *et al*. [3] propose a dehazing technique that uses a nonlocal prior, named nonlocal dehazing. The algorithm could recover both the distance map and the haze-free image based on haze-lines.

We use the official implementation of [3] with default parameters to generate ground truth images. Such an implementation could be obtained from https://github.com/danaberman/non-local-dehazing. There are not too many images with heavily haze in MIT-Adobe FiveK dataset [4]. However, we find that the algorithm of [3] could enhance the visibility and contrast of all kinds of images, which enables the usage of the whole training dataset.

Image retouching. The MIT-Adobe FiveK dataset [4] contains 5,000 photos with the corresponding retouched images from five experts. We use the retouched images from expert A as the ground truth. This task measures the ability of the proposed model to learn a highly subjective image operator that requires a significant amount of learning and semantic reasoning.

1.2. Details of Dataset

The MIT-Adobe FiveK dataset [4] together with the official training/test split could be found in http://people.csail.mit.edu/vladb/photoadjust/.

1.3. Details of DGF

The architecture of $C_l(I_l)$. We deploy Context Aggregation Network (CAN) proposed by Chen *et al*. [6] as the default architecture of $C_l(I_l)$ for all the five operators. The resolution of both input images and output images is fixed at 64**s** with three channels. The concrete architecture is shown in Table 1. For all convolution layers, the stride is set to 1,

	$C_l(I_l)$								$F(I)$	
Layer	1	2	3	4	5	6	7	8	1	2
Convolution	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1	3×3	1×1
Channel	24	24	24	24	24	24	24	3	15	3
Dilation	1	1	2	4	8	16	1	1	1	1
Bias	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓
AdaptNorm	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗
Nonlinearity	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗

Table 1: **The architecture of $C_l(I_l)$ and $F(I)$ for image processing operators.**

while the padding size is set to ensure the size of output features unchanged. Following each convolution layer, a variant of batch normalization *i.e.* adaptive normalization [6] and a nonlinearity activation function leaky ReLU are applied. The negative slope of leaky ReLU is set to 0.2 by default.

The architecture of $F(I)$. The architecture of $F(I)$ is described in Table 1. The channel size of both input images and output images is 3.

The algorithm of guided filtering layer. The entire algorithm is shown in Algorithm 1. Box filter is used for implement f_{mean} as proposed by He *et al.* [12].

Algorithm 1: Guided Filtering Layer for Image Processing, adapted from [12]

Input : Low-resolution image I_l
High-resolution image I_h
Low-resolution output O_l
Radius r and Regularization term ϵ

Output: High-resolution output O_h

1 $G_l = F(I_l)\ \ G_h = F(I_h)$

2 $\bar{G}_l = f_\mu(G_l, r)$

$\bar{O}_l = f_\mu(O_l, r)$

$\overline{G_l^2} = f_\mu(G_l * G_l, r)$

$\overline{G_l O_l} = f_\mu(G_l * O_l, r)$

3 $\Sigma_{G_l} = \overline{G_l^2} - \bar{G}_l * \bar{G}_l$

$\Sigma_{G_l O_l} = \overline{G_l O_l} - \bar{G}_l * \bar{O}_l$

4 $A_l = \Sigma_{G_l O_l} / (\Sigma_{G_l} + \epsilon)$

$b_l = \bar{O}_l - A_l * \bar{G}_l$

5 $A_h = f_\uparrow(A_l)\ \ b_h = f_\uparrow(b_l)$

6 $O_h = A_h * G_h + b_h$

2. Computer Vision Tasks

2.1. Introduction to Each Task

Depth estimation from a single image. Depth estimation from a single image is first proposed by Saxena *et al.* [17], which aims at predicting the depth at each pixel of an image with monocular cues, such as texture variations, texture gradients, occlusion, known object sizes, haze, defocus, *etc*.

Saliency Object Detection. Saliency object detection is used to detect the most salient object in an input image, which is formulated as an image segmentation problem by Liu *et al.* [16]. They try to separate the salient object from the image background with multi-scale contrast, center-surround histogram, and spatial color distribution.

Semantic Segmentation. The task of semantic segmentation is labeling images, in which each pixel is assigned to one of a finite set of labels. It's first proposed by He *et al.* [13], which is solved by combining local and global information in a probabilistic framework.

2.2. Dataset for Each Task

Depth estimation from a single image. KITTI [9] contains 42,382 rectified stereo pairs from 61 scenes, with a typical image being 1242×375 pixels in size. We test on the 200 high quality disparity images provided as part of the official KITTI training set, which covers a total of 28 scenes. The remaining 33 scenes contain 30,159 images from which we keep 29,000 for training and the rest for evaluation. The list of training and test images is available at `https://github.com/mrharicot/monodepth`.

Saliency Object Detection. We use MSRA-B [15] for our experiment, which contains 5000 images with a large variation, including natural scenes, animals, indoor, outdoor, *etc*. The official training, validation and test split described in [15] is used, which could be obtained from `https://people.cs.umass.edu/~hzjiang/drfi/`.

Semantic Segmentation. The PASCAL VOC 2012 segmentation benchmark [7] involves 20 foreground object

Layer	Convolution	Dilation	ReLU	Channel
1	1×1	1	✓	64
2	1×1	1	✗	n_O

Table 2: **The architecture of $F(I)$ for computer vision tasks.** n_O represents the channel size of target images.

	1st guided filter		2nd guided filter	
Parameters	r	ϵ	r	ϵ
Depth	4	1e-2	-	-
Saliency	8	1e-2	8	1e-2
Segmentation	4	1e-2	-	-

Table 3: **The parameters of guided filtering layer for each task.**

classes and one background class. The original dataset contains 1464, 1449 and 1456 pixel-level labeled images for training, validation and test respectively. The dataset is augmented by the extra annotations provided by [11], resulting in 10582 training images. We use the 10582 augmented images for training while using the 1449 validation images for test. 50 images from the training set are used for tuning hyper-parameters.

2.3. Details of DGF

Details of $C_l(I_l)$. The network architectures we used for each task are described in main text. The input images to $C_l(I_l)$ is at high-resolution, while the output is also a high-resolution one after bilinear upsampling or deconvolution layers, noted as $O_\uparrow$.

Details of $F(I)$. The architecture of $F(I)$ is shown as Table 2.

Details of guided filtering layer. We adapt guided filtering layer to computer vision tasks as shown in Algorithm 2. For DGF$_s$, guided filter is applied as post-processing operation, while $F(I)$ is defined as the summation of **RGB** channels pixel-wisely, noted as $F_{RGB}(I)$. For DGF, the guided filtering layer is jointly trained with the entire network.

For each task, the values of r and ϵ are determined by grid search on the validation set with DGF$_s$, we then use the same parameters to train DGF. The concrete configurations are shown in Table 3. For saliency detection, a second guided filter is applied to both DGF$_s$ and DGF for better results.

2.4. Training Details

For depth estimation from a single image, we follow the same training and test procedures as MonoDepth [1] with

Algorithm 2: Guided Filtering Layer for Computer Vision, adapted from [12]

Input : High-resolution image I_h
High-resolution output $O_\uparrow$
Radius r and Regularization term ϵ
Output: Improved High-resolution output O_h

$$G_h = F(I_h)$$
$$\bar{G_h} = f_\mu(G_h, r)$$
$$\bar{O_\uparrow} = f_\mu(O_\uparrow, r)$$
$$\overline{G_h^2} = f_\mu(G_h * G_h, r)$$
$$\overline{G_h O_\uparrow} = f_\mu(G_h * O_\uparrow, r)$$
$$\Sigma_{G_h} = \overline{G_h^2} - \bar{G_h} * \bar{G_h}$$
$$\Sigma_{G_h O_\uparrow} = \overline{G_h O_\uparrow} - \bar{G_h} * \bar{O_\uparrow}$$
$$\bar{A}_h = \Sigma_{G_h O_\uparrow} / (\Sigma_{G_h} + \epsilon)$$
$$\bar{b}_h = \bar{O_\uparrow} - \bar{A}_h * \bar{G_h}$$
$$A_h = f_\mu(\bar{A}_h, r) \quad b_h = f_\mu(\bar{b}_h, r)$$
$$O_h = A_h * G_h + b_h$$

the official implementation [1] and default settings.

For salient object detection, we reimplement DSS [14] with **PyTorch** and release the code in `https://github.com/wuhuikai/DeepGuidedFilter`.

For semantic segmentation, DeepLab-v2 [5] in **PyTorch** with Resnet as the backbone is deployed. We follow the same training and test protocols as described in `https://github.com/isht7/pytorch-deeplab-resnet`.

2.5. Other Details.

For saliency object detection, a threshold is first determined on the validation set and then is used to turn the output into binary one for visualization.

References

[1] A. Adams, J. Baek, and A. Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Eurographics*, 2009.

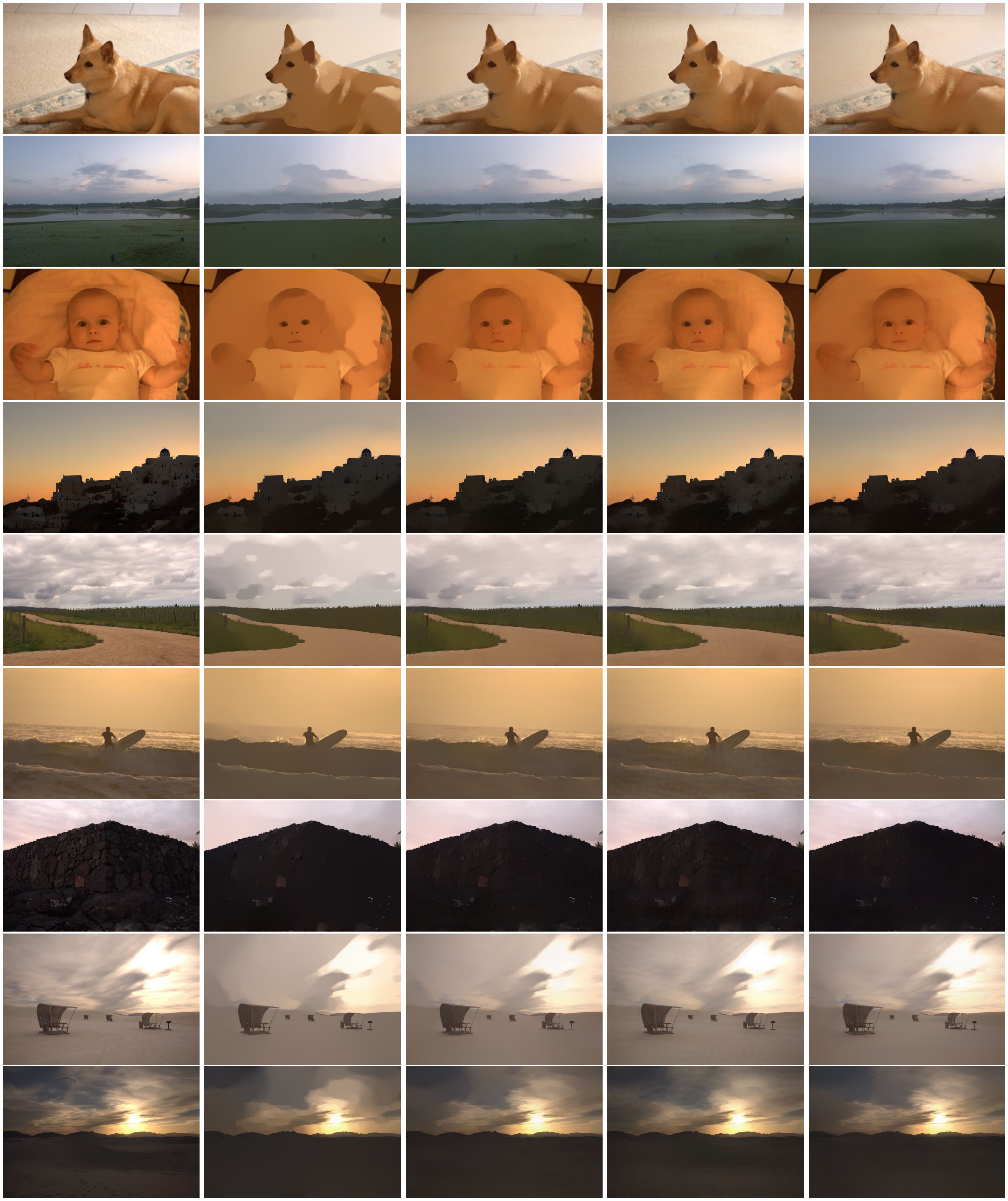
[2] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand. Fast local laplacian filters: theory and applications. *ACM TOG*, 35:1–14, 2014.

[3] D. Berman, T. Treibitz, and S. Avidan. Non-local image dehazing. In *CVPR*, 2016.

[4] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *CVPR*, 2011.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2016.

[1] `https://github.com/mrharicot/monodepth`



Input	Ground Truth	DGF (ours)	CAN [6]	DBL [10]

Figure 1: **Qualitative Results for L_0 smoothing [18].** Best viewed in color.

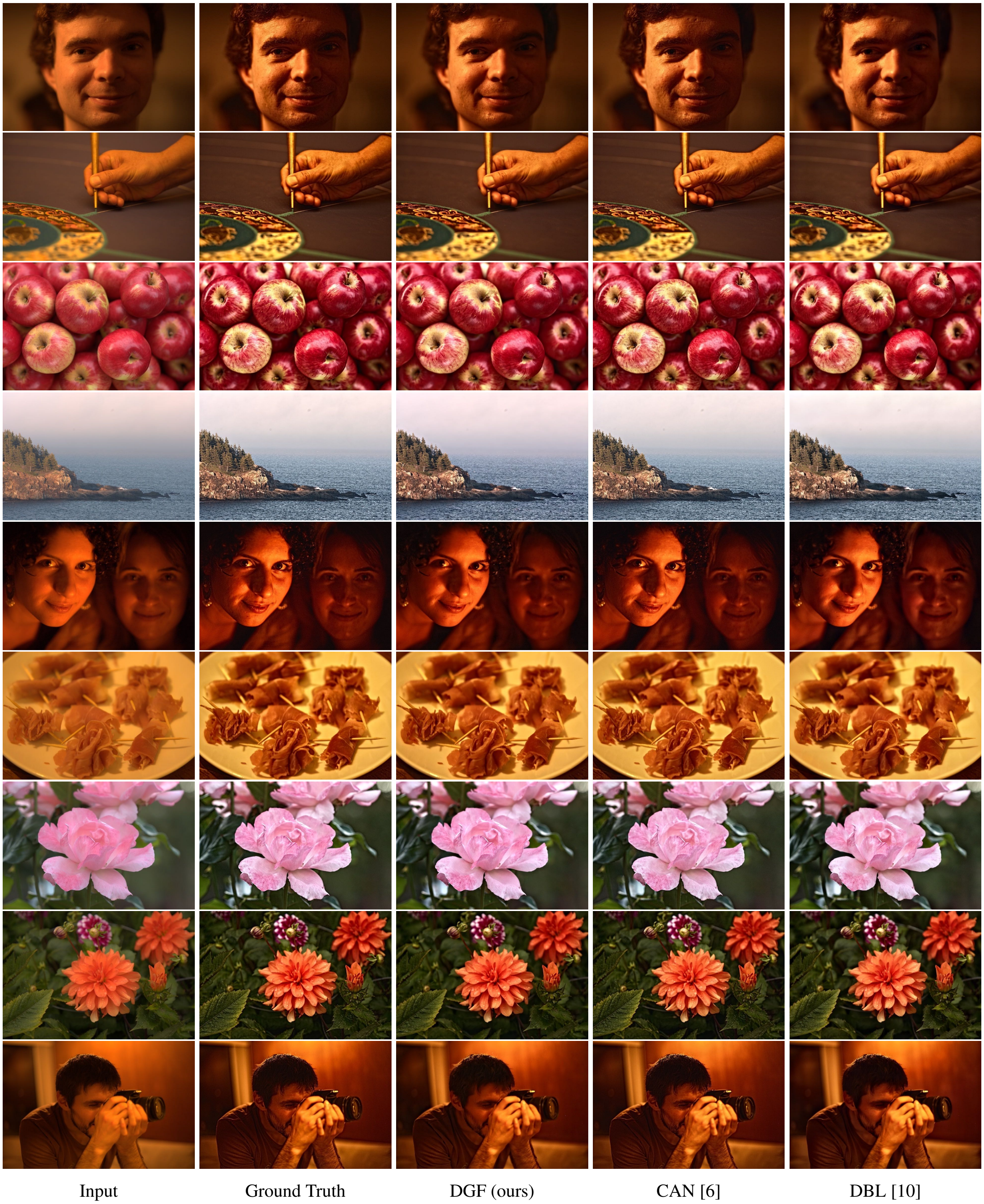
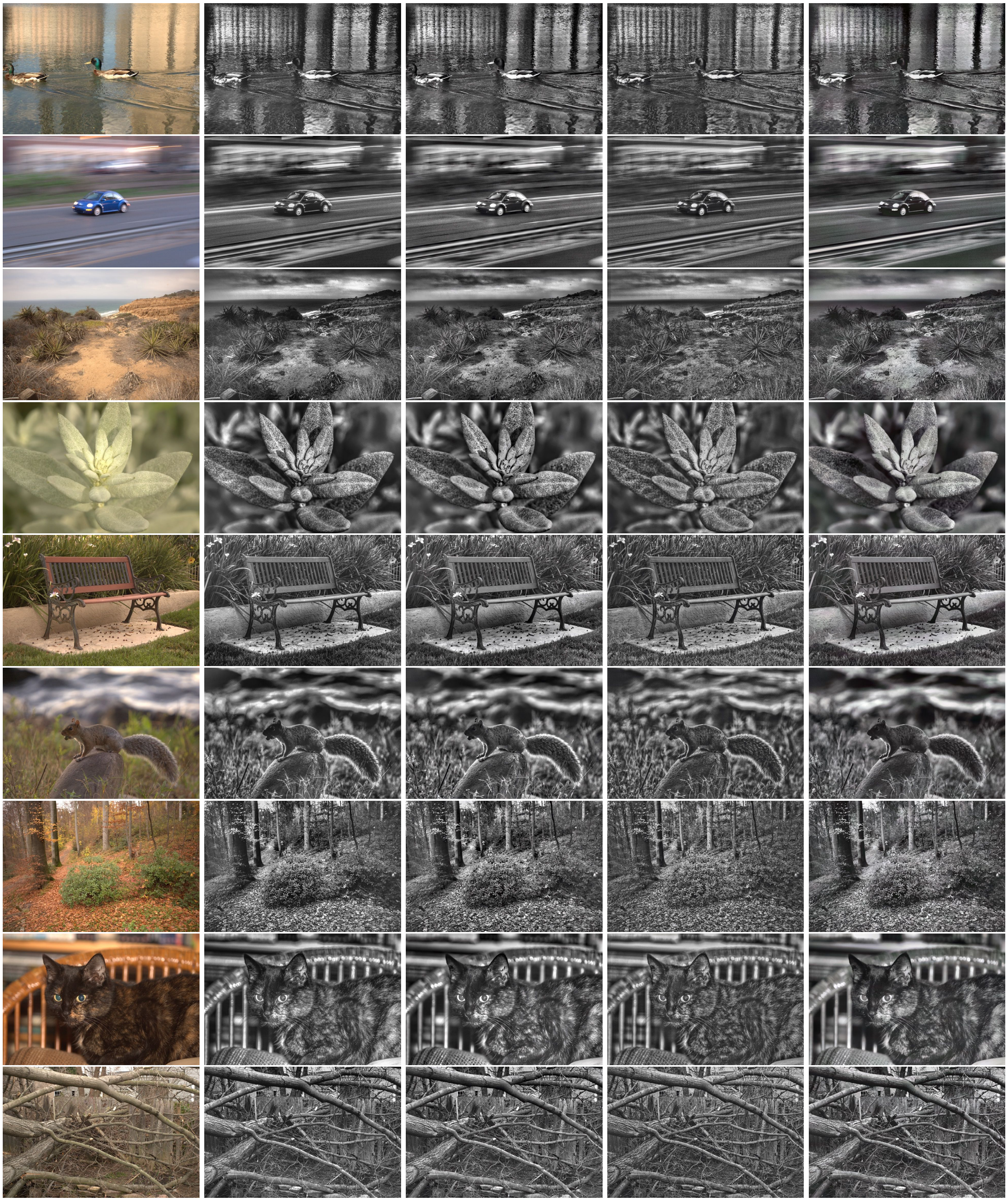
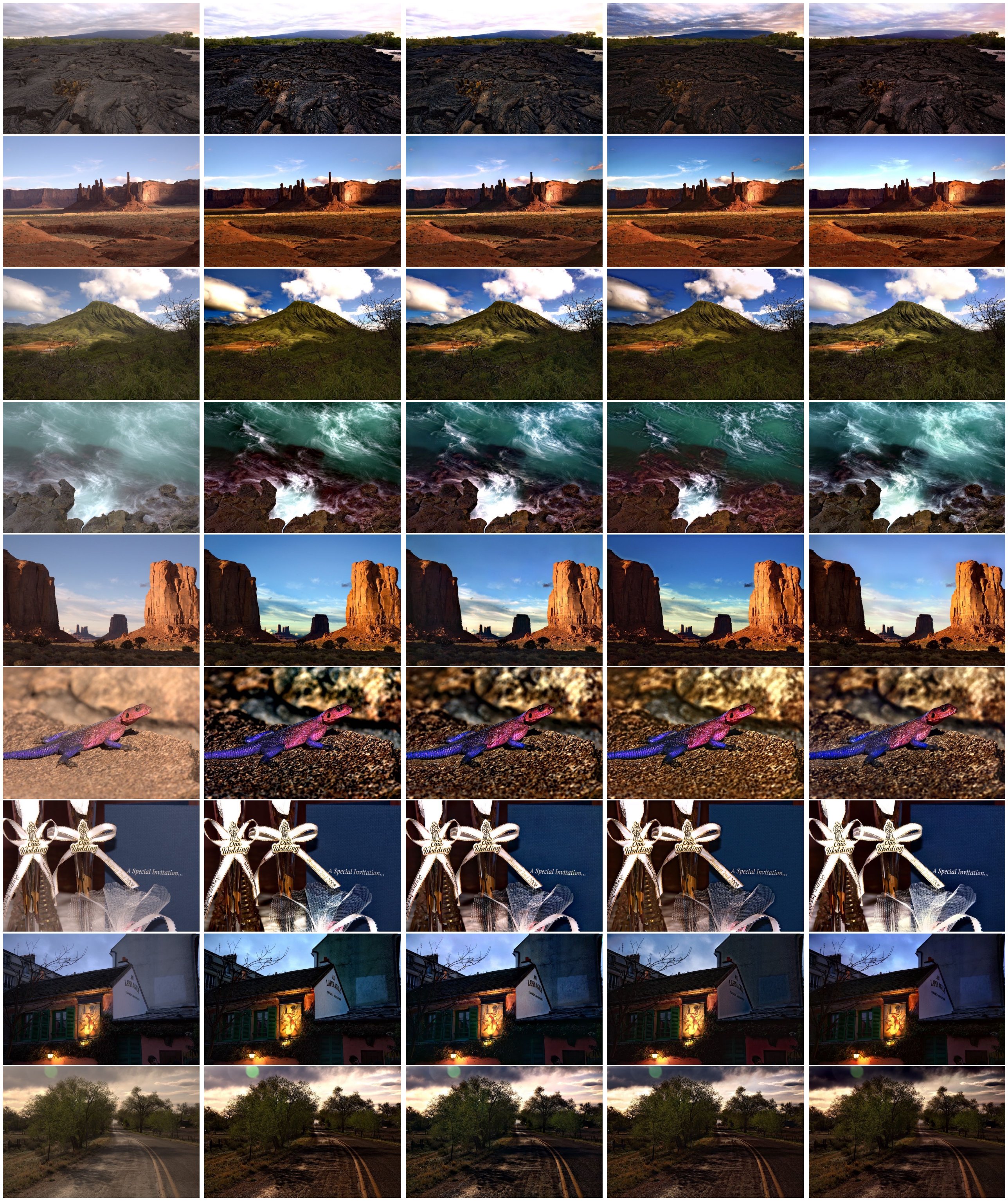


Figure 2: **Qualitative Results for multiscale detail manipulation [8].** Best viewed in color.



Input | Ground Truth | DGF (ours) | CAN [6] | DBL [10]

Figure 3: **Qualitative Results for photographic style transfer [2].** Best viewed in color.



Input Ground Truth DGF (ours) CAN [6] DBL [10]

Figure 4: **Qualitative Results for non-local dehazing [3].** Best viewed in color.

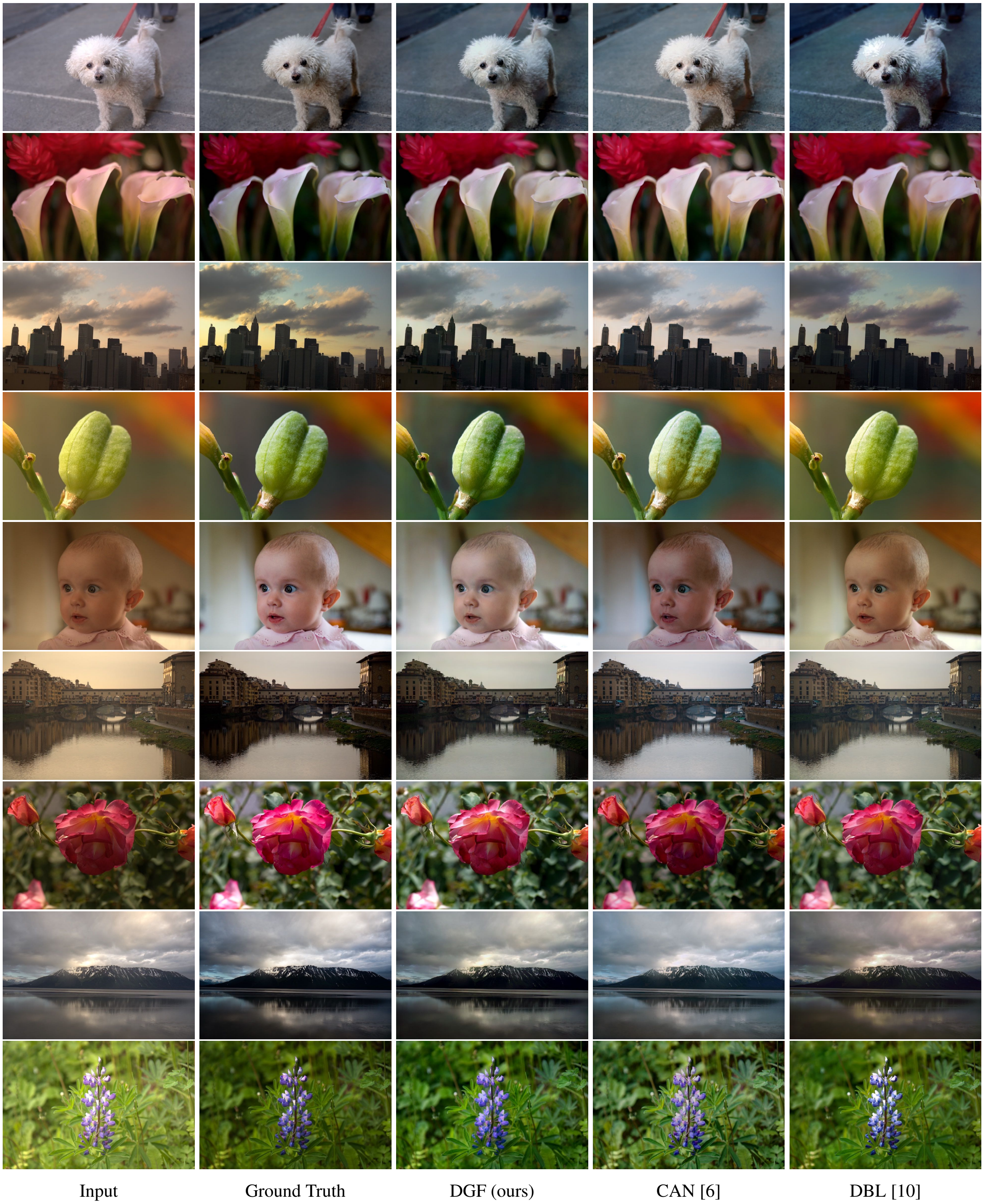


Figure 5: **Qualitative Results for image retouching learning from human annotations [4].** Best viewed in color.

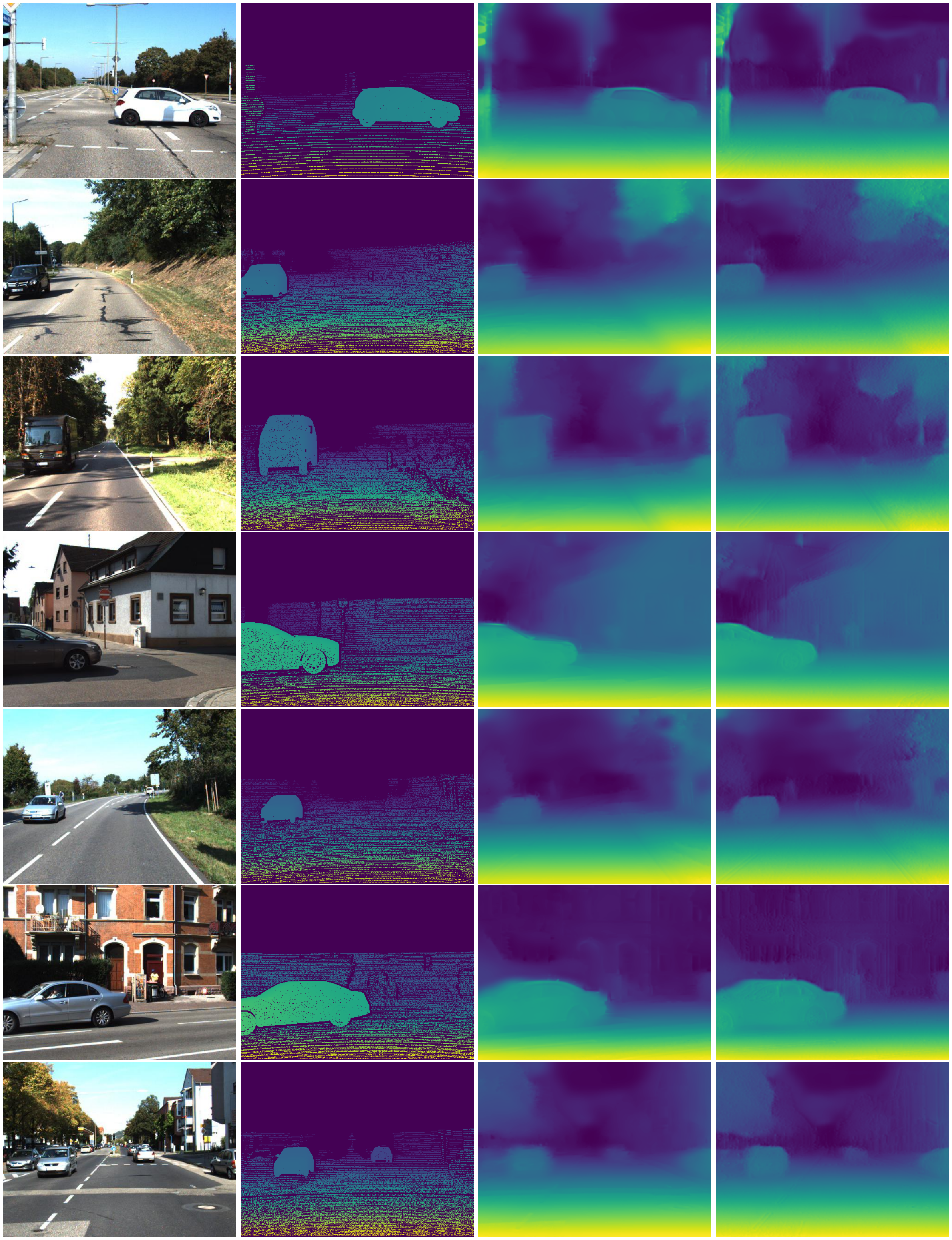
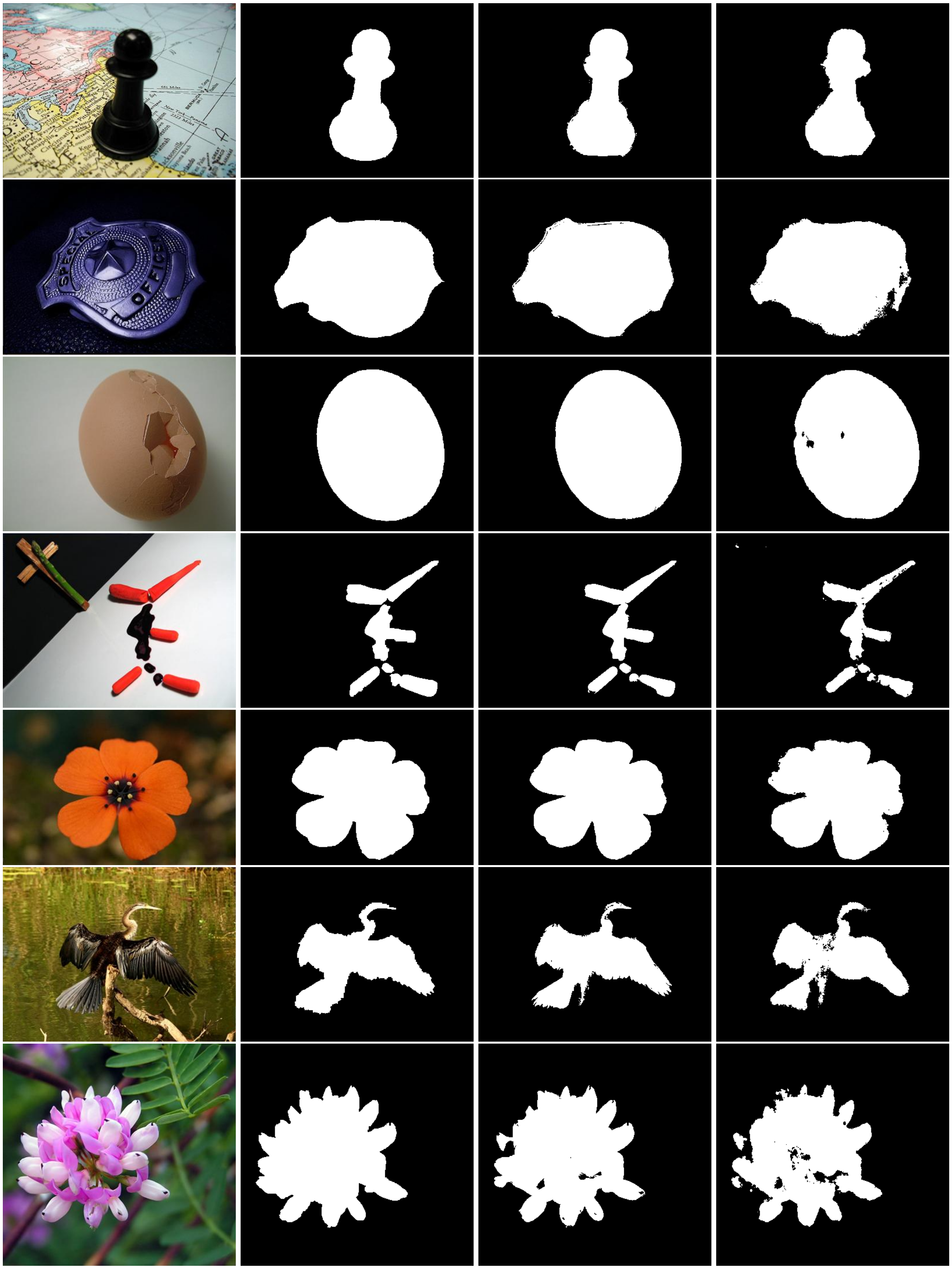


Figure 6: **Qualitative Results for depth estimation from a single image [17].** Best viewed in color.



Input Ground Truth DGF (ours) Baseline

Figure 7: **Qualitative Results for saliency object detection [16].** Best viewed in color.

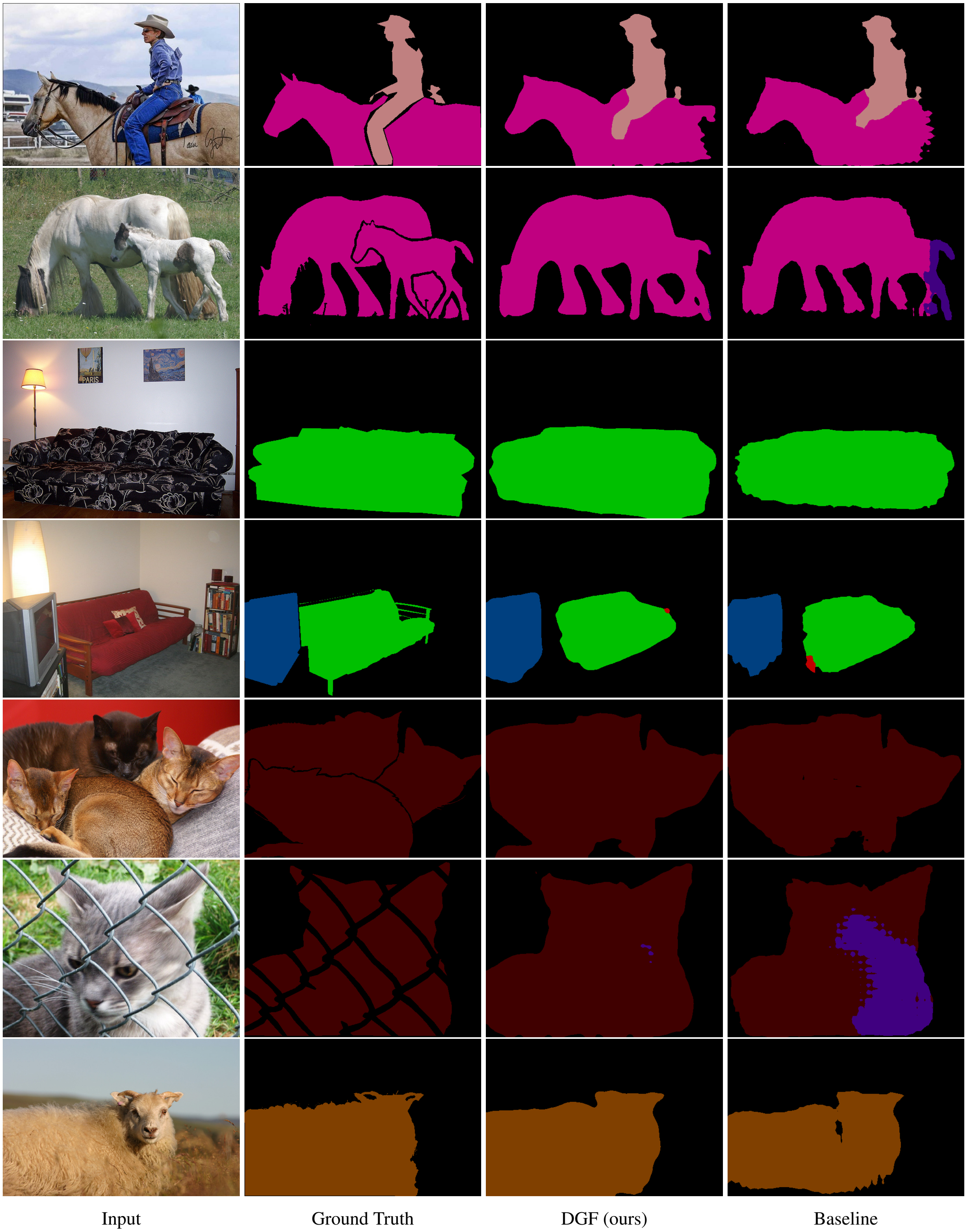


Figure 8: **Qualitative Results for semantic segmentation [13].** Best viewed in color.

[6] Q. Chen, J. Xu, and V. Koltun. Fast image processing with fully-convolutional networks. In *ICCV*, 2017.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.

[8] Z. Farbman, R. Fattal, and D. Lischinski. Edge-preserving decomposition for multi-scale tone and detail manipulation. *ACM TOG*, 27, 2008.

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[10] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand. Deep bilateral learning for real-time image enhancement. *ACM TOG*, 36(4):118, 2017.

[11] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[12] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE TPAMI*, 35(6):1397–1409, 2013.

[13] X. He, R. S. Zemel, and M. . Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *CVPR*, 2004.

[14] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. In *CVPR*, 2017.

[15] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, 2013.

[16] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. In *CVPR*, 2007.

[17] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, 2005.

[18] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l0 gradient minimization. *ACM TOG*, 30, 2011.