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 $C_i(I_i)$. We deploy Context Aggregation Network (CAN) proposed by Chen et al. [6] as the default architecture of $C_i(I_i)$ for all the five operators. The resolution of both input images and output images is fixed at 64x with three channels. The concrete architecture is shown in Table[1]. For all convolution layers, the stride is set to 1,
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 $f_{\text{mean}}$ as proposed by He et al. [12].

<table>
<thead>
<tr>
<th>Layer</th>
<th>$C_{l}(I_l)$</th>
<th>$F(I)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution</td>
<td>3 x 3</td>
<td>3 x 3</td>
</tr>
<tr>
<td>Channel</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Dilation</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Bias</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AdaptNorm</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Nonlinearity</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1: The architecture of $C_{l}(I_l)$ and $F(I)$ for image processing operators.

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

Input: Low-resolution image $I_l$
Low-resolution output $O_l$
Radius $r$ and Regularization term $\epsilon$

Output: High-resolution output $O_h$

1. $G_l = F(I_l)$, $G_h = F(I_{h})$
2. $\bar{G}_l = f_{\mu}(G_l, r)$
3. $\bar{G}_l O_l = f_{\mu}(G_l * O_l, r)$
4. $\Sigma_{G_l} = \bar{G}_l^2 - \bar{G}_l * \bar{G}_l$
5. $A_l = \Sigma_{G_l} O_l / (\Sigma_{G_l} + \epsilon)$
6. $b_l = O_l - A_l * \bar{G}_l$
7. $A_h = f_t(A_l)$, $b_h = f_t(b_l)$
8. $O_h = A_h * G_h + b_h$

Typical image being 1242 x 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 [17] involves 20 foreground object
For salient object detection, we reimplement DSS \[14\] with PyTorch and release the code in [11]. 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 [1].

2.4. Training Details

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

References

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