Supplementary Material of Blind Image Decomposition

Junlin Han[1,2] Weihao Li[1] Pengfei Fang[1,2] Chunyi Sun[2] Jie Hong[1,2]
Mohammad Ali Armin[1] Lars Petersson[1] Hongdong Li[2]

Data61-CSIRO[1] Australian National University[2]

1 Implementation Details

1.1 Training details, running time, and model size

BIDeN. We train BIDeN using a Tesla P100-PCIE-16GB GPU. The GPU driver version is 440.64.00 and the CUDA version is 10.2. We initialize weights using Xavier initialization [6]. For Task I (Mixed image decomposition across multiple domains), BIDeN (2) to BIDeN (8) takes approximately 37 hours, 50 hours, 61 hours, 71 hours, 82 hours, 91 hours, and 101 hours training time. For Task II.A (Real-scenario deraining in driving), the runtime of BIDeN is approximately 96 hours. However, BIDeN is required to perform additional mask reconstruction and source prediction tasks. By removing additional tasks from the training of BIDeN, the GPU hours can drop to 45 hours.
Double-DIP. We follow the default training setting of Double-DIP [5]. We use the official PyTorch implementation (link). We train a single image for 8000 iterations on a Tesla P100-PCIE-16GB GPU, the GPU driver version is 415.27 and the CUDA version is 10.0. The runtime for a single input image is approximately 20 minutes.
DAD. We follow the default training setting (Epoch 200, batch size 2, image crop size 256) of DAD [19]. Experiments are based on the official PyTorch implementation (link). We train DAD on a Tesla P100-PCIE-16GB GPU. The GPU driver version is 440.64.00 and the CUDA version is 10.2. DAD takes 13 hours of runtime.
MPRNet. We follow the default training setting (Epoch 250, batch size 16, image crop size 256) of MPRNet [16]. For a fair comparison, we apply the same data augmentation operations of BIDeN to MPRNet. We use the official PyTorch implementation (link) of MPRNet. We train MPRNet using 4 Tesla P100-PCIE-16GB GPU, the GPU driver version is 415.27 and the CUDA version is 10.0. The runtime of MPRNet is 20 hours and the model size of MPRNet is 41.8 MB.
Restormer. We follow the training setting used in link. Similar to MPRNet, we apply the same data augmentation operations used in BIDeN to Restormer. We reproduce our results based on a PyTorch implementation (link) of Restormer [15]. We train Restormer using a GeForce RTX 3900 GPU, the GPU driver version is 510.47 and the CUDA version is 11.6. The runtime of Restormer is approximately 8 hours and the model size is 25.3 MB.

1.2 Architecture of BIDeN

Following the naming convention used in CycleGAN [18] and perceptual loss [8], let **c3s1-k** denote a 3×3 Convolution-InstanceNorm-ReLU layer with stride 1 and k filters.

Rk denotes a residual block that contains two 3×3 convolutional layers with the same number of filters on both layer and **Rk9** denotes nine continuous residual blocks. **uk** denotes a 3×3 fractional-strided-Convolution-InstanceNorm-ReLU layer with k filters and stride $\frac{1}{2}$. Let **Ck** denotes a 4×4 Convolution-InstanceNorm-LeakyReLU (slope 0.2) layer with k filters and stride 2. Both reflection padding and zero padding are employed.

Encoder. Our multi-scale encoder E contains three branches, we name them E_{B1}, E_{B2} and E_{B3}.

E_{B1} consists of c3s2-256, Rk9, c1s1-128. E_{B2} consists of c7s1-64, c3s2-128, c3s2-256, Rk9. E_{B3} contains c15s1-64, c3s2-128, c3s2-256, c3s1-256, c3s1-256, Rk9, c1s1-128. The number of parameters is 33.908 million for the encoder.

Heads. The architecture of each head H is: c1s1-256, c1s1-256, u128, u64, c7s1-3. Each head has 0.575 million parameters.

Discriminator. The discriminator D contains two branches, D_S (Separation) and D_P (Prediction). Most weights are shared, the shared part includes C64, C128, C256.

The last layer of D_S is C512. D_P contains c1s1-512 (LeakyReLu with slope 0.2), global max pooling, c1s1-N, where N is the maximum number of source components. The Discriminator has approximately 3.028 million parameters in total. The confidence threshold of D_P is 0.

1.3 Tasks

Task I: Mixed image decomposition across multiple domains. We use linear mix as the mixing mechanism. We do not introduce additional non-balanced mixing factors or non-linear mixing as Task I is challenging enough. The mixed image z is expressed as $z = \frac{1}{L}\sum_{j=1}^{L} x_{I_j}$. The possibilities of every component to be selected vary with the maximum number of source components N. We set the possibilities to be 0.9, 0.8, 0.7, 0.6, 0.5, 0.5, 0.5 for $N = 2, 3, 4, 5, 6, 7, 8$, respectively. As the mixing is linear, the order of mixing does not matter.

Task II: Real-scenario deraining. For both Task II.A and Task II.B, the mixing mechanism is based on the physical imaging models [14,10,7,2,11] and Koschmieder's law. The model for rain streak and snow is:

$I(x) = J(x)(1 - m(x)) + A * m(x)$,

and the model for haze is:

$I(x) = J(x)t(x) + A(1 - t(x))$,

where x is the pixel of images, I is the observed intensity, J is the scene radiance, A is the global atmospheric light, and m is the mask of rain streak and snow. t denotes the transmission map. We set A between $[0.8, 1.0]$ during training, and fix $A = 0.9$ at test time.

To render the raindrop effect, we define a statistical model to estimate the location and motion of the raindrops. We employ the meta-ball model [1] for the interaction effect between multiple raindrops.

For raindrop positions, we randomly sample it over the entire scene. The raindrop radius is also randomly sampled. A single raindrop is combined with another 1 to 3 smaller raindrops to form a realistic raindrop shape. Each composite raindrop could further merge with other raindrops on the scene. The velocity along the y-axis of the raindrop is proportional to the raindrop radius. Raindrop masks are randomly selected on the time dimension for diversity. A simple refractive model [2] is employed. We create a look-up table T with three dimensions. The red and green channels together encode the texture of the raindrop, and the blue channel represents the thickness of the raindrops. Then, the texture table T is masked by the alpha mask created by the metaball model. The masked table is dubbed M. The location (x,y) of the world point that is rendered at image location (u, v) on the surface of a raindrop is modeled as:

$x = u + (R_{(u,v)} * B_{(u,v)}), y = v + (G_{(u,v)} * B_{(u,v)})$.

where $R_{(u,v)}$, $B_{(u,v)}$ denote the pixel at location (u, v) in the red and blue channels of M.

We acquire the destination pixel coordinate for location (u,v) based on the above equations and generate the distorted image. We also apply random light reduction and blur to the distorted image. For the reduction, we set the rate r between $[0.8, 0.98]$ during training, and fix rate $r = 0.9$ at test time. The reduction can be expressed as $D = rD$, where D is the distorted image and r is the rate. We use a kernel size of 3 for Gaussian blur.

At last, we merge the distorted image with the original image:

$I(x) = \alpha(x)O(x) + (1 - \alpha(x)) * D(x)$,

where x denotes the pixel of images, I is the observed intensity, O is the original image, D is the distorted image, α is the value of the raindrop mask generated by the metaball model.

The probabilities of every component to be selected are 1.0, 0.5, 0.5, 0.5 for rain streak, snow, haze, and raindrop for Task II.A. The mixing order is rain streak, snow, haze, and raindrop. We design this since 1: streak and snow do not change the atmosphere a lot, so they should come first and 2: raindrops usually occur on the top of glass lens and are observed at first, thus they should come last. In Task II.B, the probabilities of every component to be selected are 0.6, 0.5, 0.5 for rain streak, snow, and raindrop. The mixing order is rain streak $\rightarrow$ snow $\rightarrow$ raindrop.

Task III: Joint shadow/reflection/watermark removal. We use paired shadow masks, shadow images, and shadow-free images provided in ISTD [13] and SRD [12,3]. The original SRD does not offer shadow masks, we use the shadow masks generated by Cun *et al.* [3].

The algorithm of adding reflection to images [17] is expressed as:

$I(x) = T(x) + R(x) * V(x)$,

where x is the pixel of images, I is the observed intensity, T is the transmission layer, R is the reflection layer, and V denotes the vignette mask. The reflection image R is processed by a Gaussian smoothing kernel with a random kernel size, where the size is in the range of 3 to 17 pixels during training, and fixed to 11 pixels during testing.

For watermarks, we follow the watermark composition model [9]. We use the RGB watermark images to add the watermark effect. We require the BID method to reconstruct the watermark mask. The watermark composition model is:

$I(x) = J(x)(1 - w(x)) + A * w(x)$,

where x denotes the pixel of images, I is the observed intensity, J is the scene radiance, A is the global atmospheric light, and w is the watermark image. We set A between $[0.8, 1.0]$ during training, and fix $A = 0.9$ for testing.

The probabilities of every component to be selected are 0.6, 0.5, 0.5 for shadow, reflection, and watermark, respectively. The order of mixing is shadow $\rightarrow$ reflection $\rightarrow$ watermark, as watermarks are usually added during post-processing

2 Additional Results

2.1 Additional results of Task I

Detailed case results of BIDeN. When the maximum number of components N increases, the number of possible z increases rapidly. There are $2^N - 1$ possible combinations between source components, that is, $2^N - 1$ cases. We present the detailed case results of BIDeN on Task I. We show the results of $N = 2, 3, 4, 5, 6$ (BIDeN (2), BIDeN (3), BIDeN (4), BIDeN (5), BIDeN (6)) in Table 1, Table 2, Table 3, Table 4, and Table 5. These results are the extensions of Table 1 in the main paper. Note that due to the difference in precision, the PSNR results reported here are slightly different (less than 0.5% difference) from the PSNR results reported in the main paper.

Qualitative results of BIDeN. Here we present more qualitative results of BIDeN. We show the results of $N = 2, 3, 4, 5, 6$ (BIDeN (2), BIDeN (3), BIDeN (4), BIDeN (5), BIDeN (6)) in Figure 1, Figure 2, Figure 3, Figure 4, and Figure 5. The number of selected source components L and the index set I are randomly chosen. The eight source components in Task I are Fruit (A), Animal (B), Flower (C), Furniture (D), Yosemite (E), Vehicle (F), Vegetable (G), and CityScape (H).

Table 1. Detailed case results of BIDeN (2) on Task I (Mixed image decomposition across multiple domains)

Input	A	B	Acc
a	25.26	-	0.940
b	-	25.11	0.933
ab	20.09	19.93	1.000

Table 2. Detailed case results of BIDeN (3) on Task I (Mixed image decomposition across multiple domains)

Input	A	B	C	Acc
a	24.04	-	-	0.906
b	-	23.48	-	0.953
c	-	-	22.74	0.756
ab	19.07	18.78	-	0.833
ac	18.96	-	18.27	0.573
bc	-	18.31	18.25	0.763
Avg	19.01	18.54	18.26	0.723
abc	16.66	15.89	16.27	0.983

Table 3. Detailed case results of BIDeN (4) on Task I (Mixed image decomposition across multiple domains)

Input	A	B	C	D	Acc
a	23.07	-	-	-	0.896
b	-	22.61	-	-	0.886
c	-	-	21.93	-	0.770
d	-	-	-	21.88	0.933
ab	18.22	18.06	-	-	0.710
ac	18.20	-	17.41	-	0.560
ad	18.85	-	-	18.51	0.783
bc	-	16.98	17.56	-	0.710
bc	-	18.06	-	17.79	0.856
cd	-	-	18.44	18.77	0.656
Avg	18.42	17.7	17.80	18.35	0.712
abc	16.12	15.46	15.70	-	0.660
abd	16.47	15.95	-	16.40	0.676
acd	16.36	-	16.11	16.78	0.396
bcd	-	15.63	16.52	16.52	0.563
Avg	18.42	17.7	17.80	18.35	0.712
abcd	15.11	14.37	14.98	15.43	0.943

Table 4. Detailed case results of BIDeN (5) on Task I (Mixed image decomposition across multiple domains)

Input	A	B	C	D	E	Acc
a	22.75	-	-	-	-	0.840
b	-	22.02	-	-	-	0.853
c	-	-	22.30	-	-	0.553
d	-	-	-	22.52	-	0.930
e	-	-	-	-	20.77	0.923
ab	17.68	17.30	-	-	-	0.676
ac	17.60	-	17.19	-	-	0.373
ad	18.34	-	-	18.26	-	0.720
ae	18.94	-	-	-	18.36	0.700
bc	-	16.98	17.56	-	-	0.633
bd	-	17.21	-	17.60	-	0.716
be	-	17.37	-	-	17.54	0.710
cd	-	-	18.42	18.72	-	0.443
ce	-	-	18.79	-	18.34	0.556
de	-	-	-	18.18	17.94	0.796
Avg	18.14	17.21	17.99	18.19	18.04	0.632
abc	15.72	14.92	15.62	-	-	0.570
abd	16.11	15.29	-	16.15	-	0.573
abe	16.38	15.12	-	-	16.32	0.526
acd	15.91	-	15.94	16.53	-	0.320
ace	16.27	-	16.07	-	16.80	0.266
ade	16.72	-	-	16.25	16.62	0.486
bcd	-	15.04	16.46	16.32	-	0.576
bce	-	14.94	16.62	-	16.38	0.553
bde	-	15.24	-	16.07	16.15	0.736
cde	-	-	17.08	16.48	16.52	0.366
Avg	16.18	15.09	16.29	16.30	16.46	0.497
abcd	14.82	13.91	14.94	15.11	-	0.633
abce	14.97	13.80	15.06	-	15.50	0.480
abde	15.20	14.14	-	14.99	15.32	0.586
acde	15.03	-	15.23	15.11	15.62	0.180
bcde	-	13.92	15.74	15.06	15.34	0.570
Avg	14.99	13.94	15.24	15.06	15.44	0.489
abcde	14.23	13.28	14.52	14.20	14.74	0.860

Table 5. Detailed case results of BIDeN (6) on Task I (Mixed image decomposition across multiple domains)

Input	A	B	C	D	E	F	Acc
a	22.82	-	-	-	-	-	0.850
b	-	21.80	-	-	-	-	0.826
c	-	-	22.61	-	-	-	0.823
d	-	-	-	22.50	-	-	0.890
e	-	-	-	-	21.09	-	0.910
f	-	-	-	-	-	21.97	0.893
ab	17.31	16.79	-	-	-	-	0.646
ac	17.10	-	16.84	-	-	-	0.526
ad	18.14	-	-	17.66	-	-	0.613
ae	18.59	-	-	-	18.24	-	0.676
af	18.24	-	-	-	-	17.76	0.583
bc	-	16.60	17.46	-	-	-	0.613
bd	-	16.95	-	17.05	-	-	0.593
be	-	16.81	-	-	17.30	-	0.660
bf	-	16.90	-	-	-	17.04	0.510
cd	-	-	18.23	18.11	-	-	0.480
ce	-	-	18.53	-	18.16	-	0.670
cf	-	-	18.13	-	-	17.48	0.583
de	-	-	-	17.43	17.86	-	0.653
df	-	-	-	16.18	-	16.01	0.686
ef	-	-	-	-	17.42	16.74	0.673
Avg	17.87	16.81	17.83	17.28	17.79	17.00	0.654
abc	15.29	14.56	15.34	-	-	-	0.606
abd	15.83	14.95	-	15.51	-	-	0.390
abe	16.10	14.59	-	-	16.04	-	0.513
abf	15.89	14.78	-	-	-	15.61	0.333
acd	15.49	-	15.56	15.98	-	-	0.313
ace	15.88	-	15.67	-	16.55	-	0.350
acf	15.64	-	15.46	-	-	15.61	0.303
ade	16.46	-	-	15.59	16.45	-	0.330
adf	16.15	-	-	14.60	-	14.73	0.450
aef	16.44	-	-	-	15.99	15.18	0.373
bce	-	14.68	16.24	15.71	-	-	0.370
bce	-	14.43	16.43	-	16.08	-	0.500
bcf	-	14.69	16.04	-	-	15.42	0.346
bde	-	14.90	-	15.45	15.89	-	0.516
bdf	-	14.87	-	14.60	-	14.52	0.456
bef	-	14.81	-	-	15.60	15.04	0.400
cde	-	-	16.82	15.84	16.30	-	0.253
cdf	-	-	16.48	14.86	-	14.59	0.383
cef	-	-	16.64	-	15.91	15.15	0.443
def	-	-	-	14.70	15.80	14.26	0.506
Avg	15.92	14.72	16.06	15.28	16.06	16.71	0.439
abcd	14.45	13.69	14.61	14.62	-	-	0.420
abce	14.63	13.51	14.76	-	15.20	-	0.540
abcf	14.57	13.65	14.54	-	-	14.34	0.313
abde	15.02	13.79	-	14.36	15.05	-	0.323
abdf	14.90	13.80	-	13.77	-	13.81	0.373
abef	15.13	13.70	-	-	14.77	14.17	0.276
acde	14.71	-	14.88	14.60	15.34	-	0.126
acdf	14.61	-	14.67	13.97	-	13.75	0.220
acef	14.80	-	14.81	-	14.99	14.14	0.193
adef	15.13	-	-	14.81	15.00	13.50	0.293
bcde	-	13.64	15.55	14.56	14.98	-	0.296
bcdf	-	13.76	15.26	13.96	-	13.66	0.336
bcef	-	13.63	15.35	-	14.84	14.10	0.356
bdef	-	13.86	-	13.78	14.74	13.49	0.543
cdef	-	-	15.62	13.99	14.99	13.44	0.206
Avg	14.79	13.70	15.00	14.24	14.99	13.84	0.321
abcde	13.97	13.12	14.24	13.73	14.41	-	0.350
abcdf	13.93	13.22	14.05	13.38	-	13.14	0.543
abcef	14.04	13.09	14.17	-	14.23	13.41	0.346
abdef	14.30	13.25	-	13.24	14.20	12.99	0.516
acdef	14.03	-	14.28	13.40	14.37	12.94	0.140
bcdef	-	13.18	14.76	13.40	14.19	12.92	0.356
Avg	14.04	13.17	14.30	13.43	14.28	13.08	0.375
abcdef	13.54	12.86	13.82	12.95	13.80	12.57	0.846

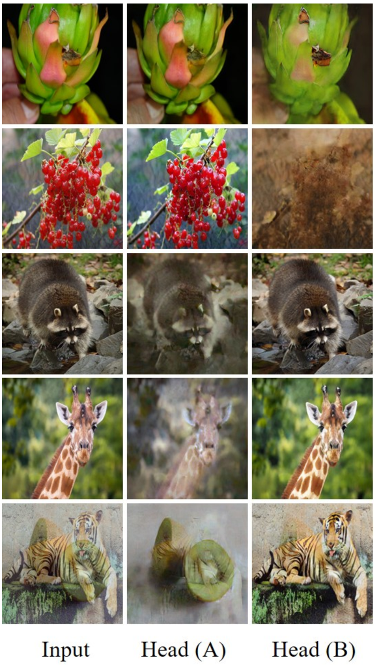

Fig. 1. Qualitative results of BIDeN (2). Fruit (A), Animal (B). Row 1-2: a. Row 3-4: b. Row 5: ab

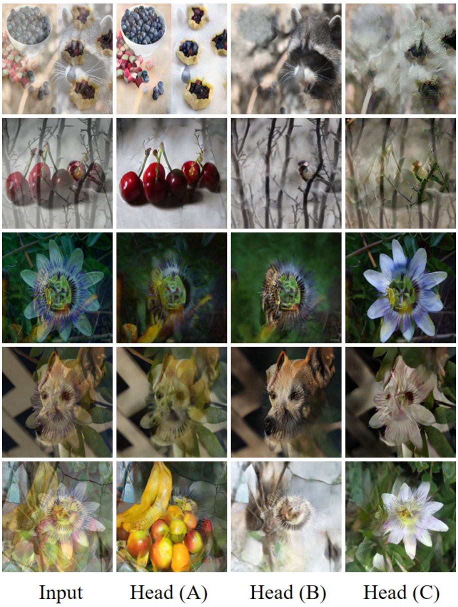

Fig. 2. Qualitative results of BIDeN (3). Fruit (A), Animal (B), Flower (C). Row 1-2: ab. Row 3-4: bc. Row 5: abc

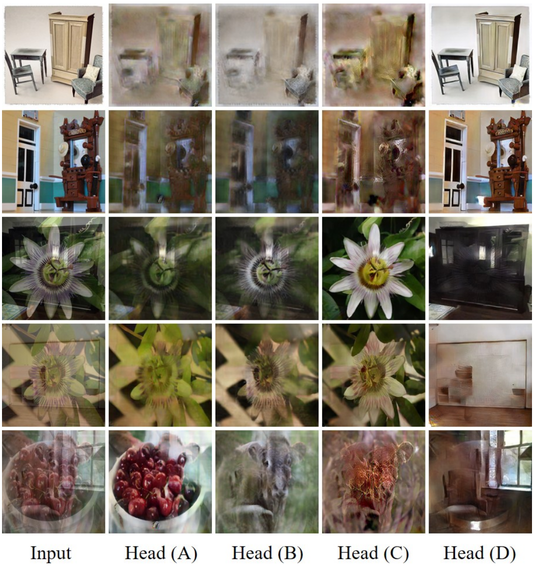

Fig. 3. Qualitative results of BIDeN (4). Fruit (A), Animal (B), Flower (C), Furniture (D). Row 1-2: d. Row 3-4: cd. Row 5: abd

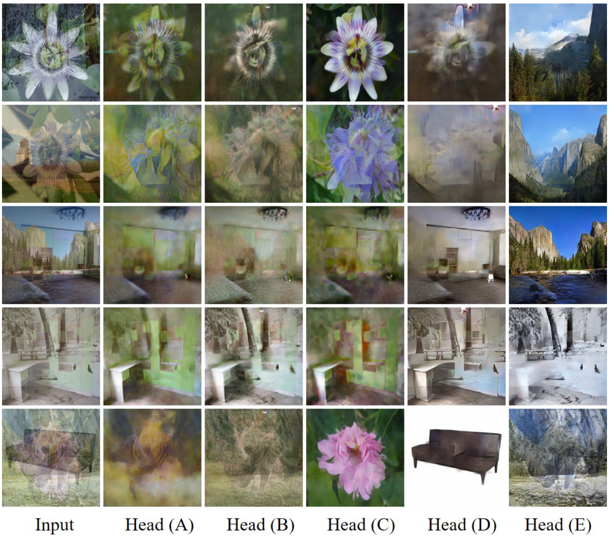

Fig. 4. Qualitative results of BIDeN (5). Fruit (A), Animal (B), Flower (C), Furniture (D), Yosemite (E). Row 1-2: ce. Row 3-4: de. Row 5: bcde

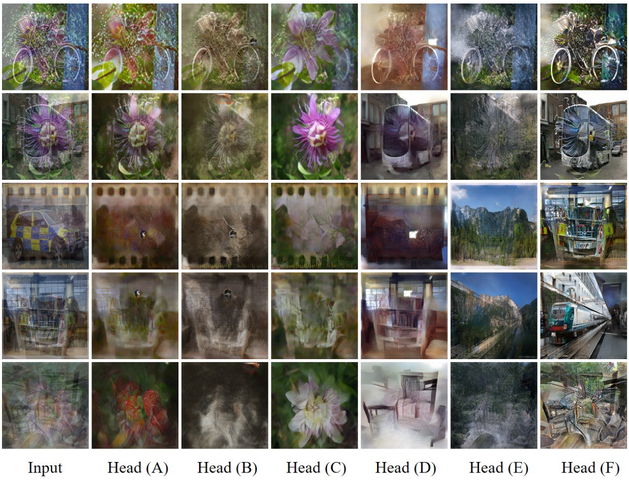

Fig. 5. Qualitative results of BIDeN (6). Fruit (A), Animal (B), Flower (C), Furniture (D), Yosemite (E), Vehicle (F). Row 1-2: cf. Row 3-4: def. Row 5: abcdef

2.2 Additional results of Task II

For Task II.A (Real-scenario deraining in driving), more qualitative results are provided. Visual examples of CityScapes/masks/transmission maps generated by BIDeN are shown in Figure 6. We present more qualitative comparisons between BIDeN and MPRNet [16] in Figure 7 and Figure 8. The comparison presents the results of 6 cases from the same scene.

For the default training setting on Task II.A, the probabilities of every component to be selected are 1.0, 0.5, 0.5, 0.5 for rain streak, snow, raindrop, and haze. Moreover, we train both BIDeN and MPRNet again setting the possibility of the rain streak component to be selected as 0.8. The quantitative results of BIDeN and the comparison between MPRNet are provided in Table 6 and Table 7. Compared to BIDeN trained under the default training setting of Task II.A, BIDeN performs better when the possibility of the rain streak component to be selected is set to 0.8.

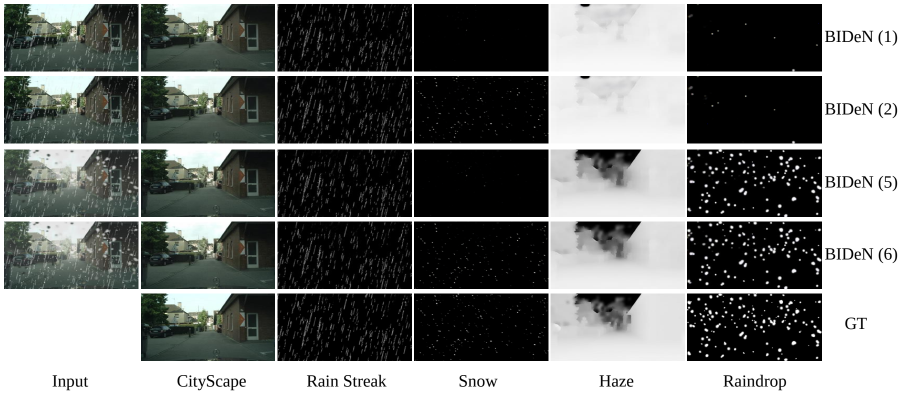

Fig. 6. CityScape, masks (Rain Streak, Snow, Raindrop), and transmission map (Haze) generated by BIDeN for case (1), case (2), case (5), and case (6). Case (1): rain streak, case (2): rain streak + snow, case (5): rain streak + moderate haze + raindrop, case (6): rain streak + snow + moderate haze + raindrop

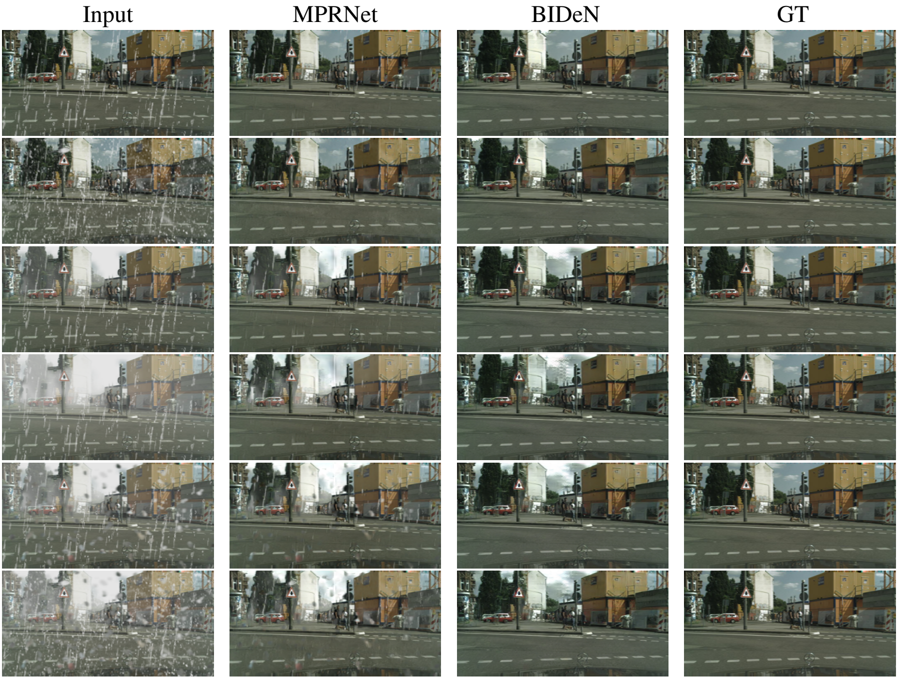

Fig. 7. Additional results of Task II.A (Real-scenario deraining in driving). Row 1-6 presents 6 cases of a same scene. The 6 cases are (1): rain streak, (2): rain streak + snow, (3): rain streak + light haze, (4): rain streak + heavy haze, (5): rain streak + moderate haze + raindrop, (6) rain streak + snow + moderate haze + raindrop. BIDeN remove all components of rain efficiently while MPRNet leaves some components that are not completely removed

Table 6. Results of BIDeN on Task II.A (Real-scenario deraining in driving). The possibility of the rain streak component to be selected is 0.8. We employ PSNR and SSIM metrics for both CityScape images, masks, and transmission maps. We report the results for 6 test cases as presented in Figure 1 of the main paper, the 6 cases are (1): rain streak, (2): rain streak + snow, (3): rain streak + light haze, (4): rain streak + heavy haze, (5): rain streak + moderate haze + raindrop, (6) rain streak + snow + moderate haze + raindrop. Note that only haze is divided into light/moderate/heavy intensities. Both training set and test set of Rain Streak, Snow, and Raindrop already consist of different intensities

Method	CityScape		Rain Streak		Snow		Haze		Raindrop		Acc ↑
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	
BIDeN (1)	33.30	0.930	31.55	0.917	-	-	-	-	-	-	1.0
BIDeN (2)	29.55	0.896	28.80	0.836	25.74	0.689	-	-	-	-	0.992
BIDeN (3)	29.38	0.919	30.98	0.907	-	-	31.11	0.956	-	-	0.994
BIDeN (4)	27.56	0.899	30.39	0.895	-	-	31.83	0.944	-	-	0.994
BIDeN (5)	27.89	0.900	30.17	0.891	-	-	30.73	0.945	22.32	0.904	0.996
BIDeN (6)	27.05	0.869	28.05	0.815	24.81	0.653	30.02	0.940	21.55	0.888	0.990

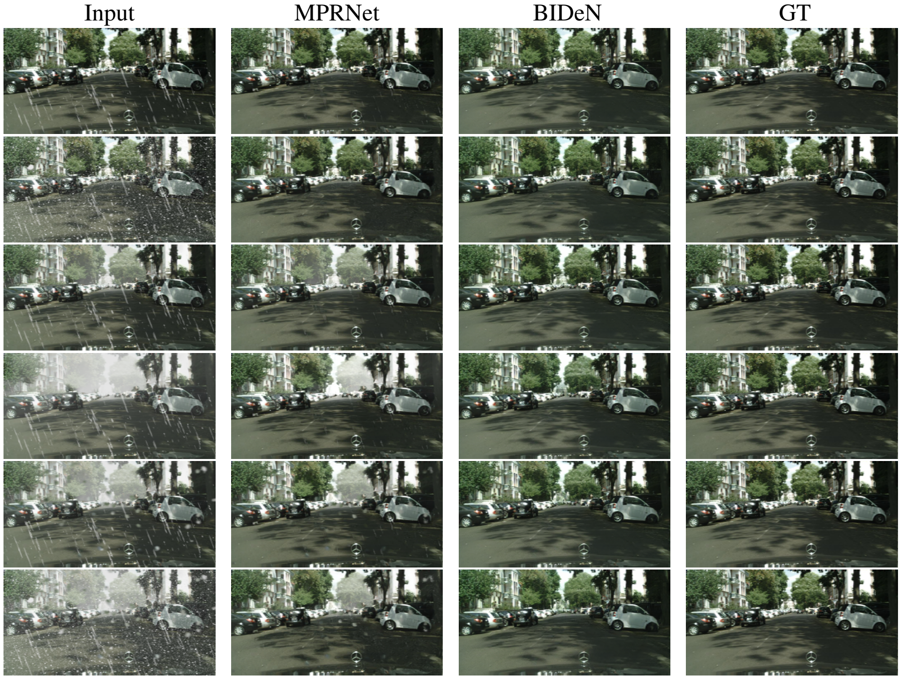

Fig. 8. Additional results of Task II.A (Real-scenario deraining in driving). Row 1-6 presents 6 cases of a same scene. The 6 cases are (1): rain streak, (2): rain streak + snow, (3): rain streak + light haze, (4): rain streak + heavy haze, (5): rain streak + moderate haze + raindrop, (6) rain streak + snow + moderate haze + raindrop. BIDeN remove all components of rain efficiently while MPRNet leaves some components that are not completely removed

Table 7. Comparison on Task II.A (Real-scenario deraining in driving) between BIDeN and MPRNet [16]. The possibility of the rain streak component to be selected is 0.8. MPRNet shows superior results for case (1) and case (2). In contrast, BIDeN is better at other cases. For the details of 6 test cases, please refer to Table 6 and Figure 1 of the main paper

Case	Input		MPRNet		BIDeN	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
(1)	25.69	0.786	33.03	0.941	33.30	0.930
(2)	18.64	0.564	30.44	0.902	29.55	0.896
(3)	17.45	0.712	23.95	0.897	29.38	0.919
(4)	11.12	0.571	17.32	0.810	27.56	0.899
(5)	14.05	0.616	20.75	0.839	27.89	0.900
(6)	12.38	0.461	19.74	0.798	27.05	0.869

2.3 Additional results of Task III

For Task III (Joint shadow/reflection/watermark removal), Figure 9 presents the qualitative comparison between BIDeN and two state-of-the-art baselines [3,4]. Though suffering from a color shift, BIDeN still shows visually pleasing, ghost-free shadow removal results. We also provide additional results of BIDeN for all cases. Results of Version one (V1) and Version two (V2) are shown in Figure 10 and Figure 11.

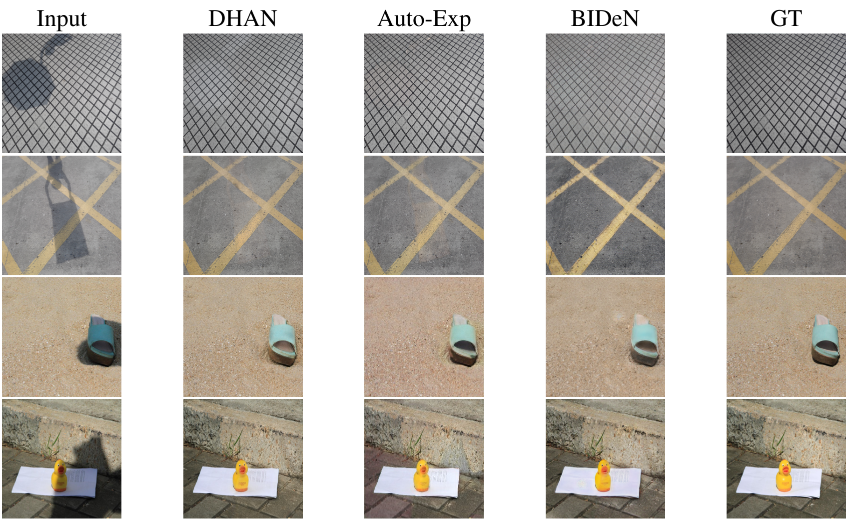

Fig. 9. Results of Task III (Joint shadow/reflection/watermark removal). Row 1-2 show the results of Version one (V1) and Row 3-4 present the results of Version two (V2). We compare BIDeN to DHAN [3] and Auto-Exposure [4] on shadow removal task. Though the performance of BIDeN is constrained by the generality of BIDeN and BID training setting, BIDeN still efficiently removes the shadow and shows competitive results. However, BIDeN suffers from a color shift whereas DHAN and Auto-Exposure better keep the fidelity of the original colors

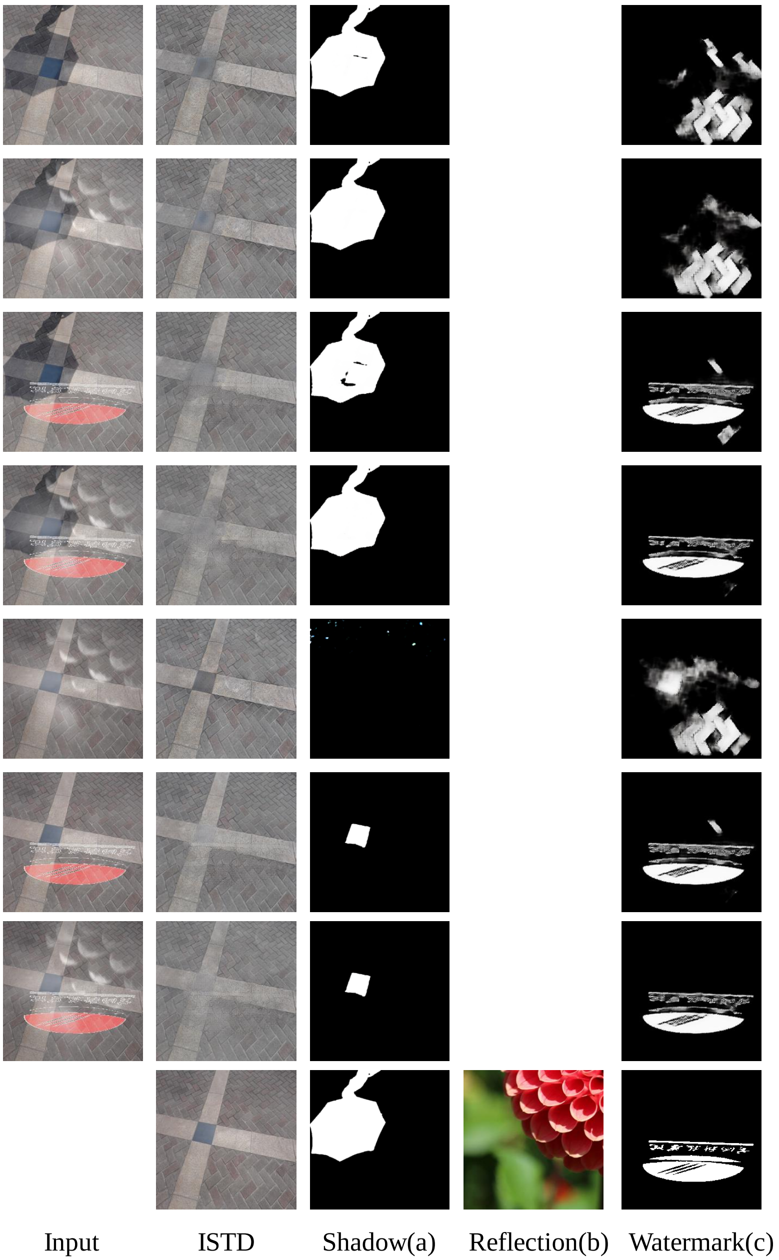

Fig. 10. All case results of Task III (Joint shadow/reflection/watermark removal), Version one (V1). ISTD images, shadow masks, and watermark masks generated by BIDeN for all cases. The order of all cases is identical to Table 4 of the main paper. The generated ISTD images suffer color shift, but all shadow/reflection/watermark are efficiently removed for all cases

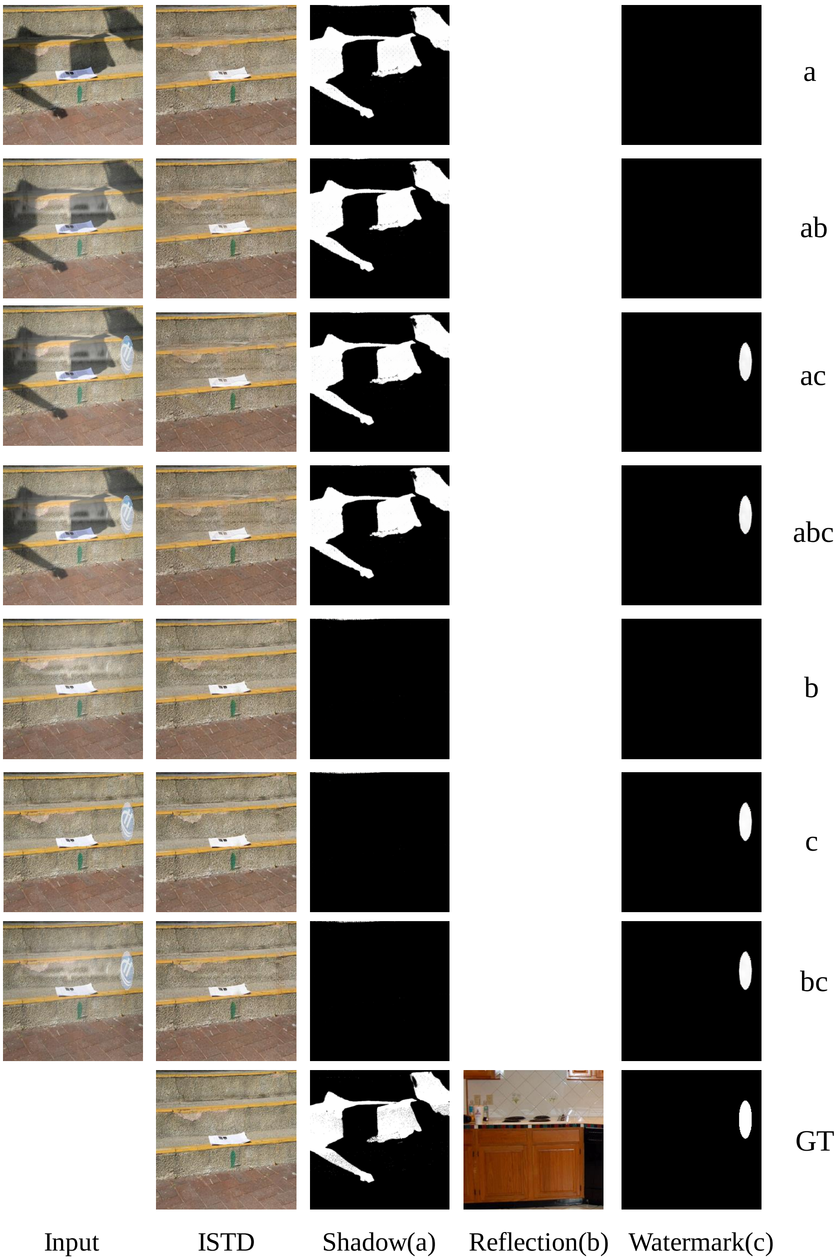

Fig. 11. All case results of Task III (Joint shadow/reflection/watermark removal), Version two (V2). ISTD images, shadow masks, and watermark masks generated by BIDeN for all cases. The order of all cases is identical to Table 4 of the main paper. All shadow/reflection/watermark are efficiently removed for all cases

References

1. Blinn, J.F.: A generalization of algebraic surface drawing. ACM transactions on graphics (TOG) **1**(3), 235–256 (1982)
2. Cohen, J., Olano, M., Manocha, D.: Appearance-preserving simplification. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques. pp. 115–122 (1998)
3. Cun, X., Pun, C.M., Shi, C.: Towards ghost-free shadow removal via dual hierarchical aggregation network and shadow matting gan. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10680–10687 (2020)
4. Fu, L., Zhou, C., Guo, Q., Juefei-Xu, F., Yu, H., Feng, W., Liu, Y., Wang, S.: Auto-exposure fusion for single-image shadow removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10571–10580 (2021)
5. Gandelsman, Y., Shocher, A., Irani, M.: " double-dip": Unsupervised image decomposition via coupled deep-image-priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11026–11035 (2019)
6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256 (2010)
7. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE transactions on pattern analysis and machine intelligence **33**(12), 2341–2353 (2010)
8. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
9. Liu, Y., Zhu, Z., Bai, X.: Wdnet: Watermark-decomposition network for visible watermark removal. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3685–3693 (2021)
10. Liu, Y.F., Jaw, D.W., Huang, S.C., Hwang, J.N.: Desnownet: Context-aware deep network for snow removal. IEEE Transactions on Image Processing **27**(6), 3064–3073 (2018)
11. Porav, H., Bruls, T., Newman, P.: I can see clearly now: Image restoration via de-raining. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 7087–7093. IEEE (2019)
12. Qu, L., Tian, J., He, S., Tang, Y., Lau, R.W.: Deshadownet: A multi-context embedding deep network for shadow removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4067–4075 (2017)
13. Wang, J., Li, X., Yang, J.: Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1788–1797 (2018)
14. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Deep joint rain detection and removal from a single image. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1685–1694 (2017). https://doi.org/10.1109/CVPR.2017.183
15. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. CVPR (2022)
16. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: CVPR (2021)
17. Zhang, X., Ng, R., Chen, Q.: Single image reflection separation with perceptual losses. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4786–4794 (2018)
18. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)

19. Zou, Z., Lei, S., Shi, T., Shi, Z., Ye, J.: Deep adversarial decomposition: A unified framework for separating superimposed images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12806–12816 (2020)