Deep Adversarial Decomposition: A Unified Framework for Separating Superimposed Images

Zhengxia Zou\textsuperscript{1}, Sen Lei\textsuperscript{2}, Tianyang Shi\textsuperscript{3}, Zhenwei Shi\textsuperscript{2}, Jieping Ye\textsuperscript{1,4}
\textsuperscript{1}University of Michigan, Ann Arbor, \textsuperscript{2}Beihang University, \textsuperscript{3}NetEase Fuxi AI Lab, \textsuperscript{4}Didi Chuxing
{zzhengxi, jpye}@umich.edu, {senlei, shizhenwei}@buaa.edu.cn, shitianyang@corp.netease.com

Abstract

Separating individual image layers from a single mixed image has long been an important but challenging task. We propose a unified framework named “deep adversarial decomposition” for single superimposed image separation. Our method deals with both linear and non-linear mixtures under an adversarial training paradigm. Considering the layer separating ambiguity that given a single mixed input, there could be an infinite number of possible solutions, we introduce a “Separation-Critic” - a discriminative network which is trained to identify whether the output layers are well-separated and thus further improves the layer separation. We also introduce a “crossroad $l_1$” loss function, which computes the distance between the unordered outputs and their references in a crossover manner so that the training can be well-instructed with pixel-wise supervision. Experimental results suggest that our method significantly outperforms other popular image separation frameworks.

1. Introduction

In the computer vision field, many tasks can be considered as image layer mixture/separation problems. For example, when we take a picture on rainy days, the image obtained can be viewed as a mixture of two layers: a rain-streak layer and a clean background layer. When we look through a transparent glass, we see a mixture of the scene beyond the glass and the scene reflected by the glass.

Separating superimposed images with a single observation has long been an important but challenging task. On one hand, it forms the foundation of a large group of real-world applications, including transparency separation, shadow removal, deraining, etc. On the other hand, it is naturally a massively ill-posed problem, where the difficulty lies not only in the absence of the mixture function but also in the lack of constraints on the output space. In recent literature, most of the above-mentioned tasks are investigated individually despite the strong correlation between them [8, 23, 48, 54, 59, 60]. In this paper, we propose a new framework for single superimposed image separation, as shown in Fig 1 which deals with all the above tasks under a unified framework.

Learning a priori for image decomposition. Given a single superimposed image, as we have no extra constraint on the output space, there could be an infinite number of possible decomposition. Previous works often integrate hard-crafted priors to apply additional constraints to their separation outputs. For example, in recent literature [17, 67], researchers introduced the “gradient exclusiveness” [67] and the “internal self-similarity” [17], where the former one emphasizes the independence of the layers to be separated in their gradient domain, and the latter one assumes that the distribution of small patches within each separate layer should be “simpler” (more uniform) than in the original mixed one. However, these hand-crafted priors may introduce unexpected bias and thus fails in complex mixture conditions. In this paper, we investigate an interesting question: can a good prior be learned from data? To answer this question, we re-examine the prior under a totally different point of view by taking advantage of the recent success of generative adversarial networks [19, 28, 47]. We introduce a “Separation-Critic” - a discriminative network $D_C$, which is trained to identify whether the output layers are well-separated. The layer separation thus can be gradually enforced by fooling the Critic under an adversarial training paradigm.

Crossroad loss function. In addition to the Critic $D_C$, we also introduce a layer separator $G$ and train it to minimize the distance between the separated outputs and their ground truth references. However, a standard $l_1$ or $l_2$ loss does not apply to our task since the $G$ may predict unordered outputs. We, therefore, introduce a “crossroad $l_1$” loss which computes the distance between the outputs and their references in a crossover fashion. In this way, the train-
Figure 1: We propose a unified framework for single mixed image separation under an adversarial training paradigm. Our method can be applied to a variety of real-world tasks, including image deraining, photo reflection removal, image shadow removal, etc.

Confronting nonlinear mixture and degradation. In some real-world image separation tasks, the mixture of images is usually beyond linear. For example, the formation of a reflection image may depend not only on the relative position of the camera to the image plane but also on lighting conditions. Besides, the degradation (e.g., over-exposure and noise) may further increase the difficulty of the separation. In these conditions, one may need injecting “imagination” into the algorithm to recover the hidden structure of degraded data. Inspired by the recent image translation methods, we further introduce two Markovian discriminators (PatchGAN) to improve the perceptual quality of the outputs.

Experimental results show that our method significantly outperforms other popular image separation frameworks. We apply our method to a variety of computer vision tasks. Without specifically tuning, we achieve the state of the art results on nine datasets of three different tasks, including image deraining, photo reflection removal, and image shadow removal. To our best knowledge, this is the first unified framework for solving these problems as most previous solutions on these tasks are separately investigated and designed.

2. Related Work

Superimposed image separation. In signal processing, a similar topic to our paper is Blind Source Separation (BSS), which aims to separate source signals from a set of mixed ones. The research of this topic can be traced back to the 1990s, where the Independent Component Analysis (ICA) was a representative of the methods at the time. The key to estimating the ICA model is the Central Limit Theorem, i.e., the distribution of a sum of two images tends toward a Gaussian distribution, under certain conditions. Some statistics-based criteria thus have been introduced to measure the independence and non-Gaussianity of the images, e.g., Kurtosis and negative entropy. The main difference between the BSS and our task is that the former one typically requires multiple mixed inputs or additional user interactions, while the latter one does not. We focus on the latter case since multiple mixed inputs or user interactions are not always available in practice. Recently, Gandelsman et al. proposed a deep learning-based method called Double-DIP that can separate superimposed images with a single observation under certain conditions. However, their method can only handle the input with regular mixed patterns.

Related applications. Many real-world tasks can be viewed as special cases of superimposed image separation:

1) Single Image Deraining. A rainy image can be simply viewed as a superposition of a clean background image and rain streaks. Some early deraining methods were designed based on low-rank constraints and sparse coding methods, where the rain-streaks are considered as high frequency noise. However, these methods usually lead to over-smoothed results. Recent deraining methods typically formulate the deraining as a deep learning based “image-to-image” translation process which is trained with pixel-wise regression loss.

2) Reflection-removal. Early reflection removal methods often require additional input images and hand-crafted priors on estimating the reflection layer. Such priors include...
Figure 2: An overview of our method. Our method consists of a separator \( G \) and several discriminators. The \( G \), which is trained under a “crossroad \( l_1 \)” loss, aims to decompose a single mixed input into two individual images. To identify whether the separation is good or not, we introduce a “Critic” \( D_C \), which is trained together with \( G \) under an adversarial training paradigm. We further use two Markovian Discriminators \((D_{M1}, D_{M2})\) to improve the perceptual quality of the outputs.

The smoothness prior \([40, 51]\), gradient sparsity constraint \([2, 36, 37]\), ghost cues \([51]\), etc. In Recent methods, the priors are usually explored by data-driven methods \([9, 44, 53, 60]\) and adversarial training/synthesis \([59, 67]\), which better handle more complex reflections.

3) Shadow-removal. To remove shadows, some early works designed physical models based on illumination invariant assumption \([10, 11]\). Later, more methods based on hand-crafted features were proposed \([1, 21, 25, 32, 52, 66, 69]\). Similar to deraining and reflection removal, recent researches on shadow removal also suggest using deep learning or adversarial training techniques, which brings additional improvements especially on complex lightening conditions \([8, 24, 30, 46, 50, 54, 68]\).

Generative Adversarial Networks (GAN). GAN has received a great deal of attention in recent years and has achieved impressive results in a variety of computer vision tasks, e.g., image generation \([7, 47]\), image style transfer \([28, 70]\), image super-resolution \([34]\), etc. A typical GAN \([19]\) consists of two neural networks: a generator \( G \) and a discriminator \( D \). The key to the GAN’s success is the idea of adversarial training where the \( G \) and \( D \) will contest with each other in a minimax two-player game and forces the generated data to be, in principle, indistinguishable from real ones. More recently, GAN has also been applied to some image separation tasks to improve perceptual quality of the recovered images, including image deraining \([38, 65]\), image reflection removal \([44, 59, 67]\) and image de-shadowing \([8, 24, 33, 54]\).

3. Methodology

We frame the training of our model as a pixel-wise regression process with the help of adversarial losses. Our method consists of an image separator \( G \), a Separation Critic \( D_C \) and two Markovian discriminators \( D_{M1} \) and \( D_{M2} \). Fig. 2 shows an overview of our method.

3.1. Crossroad \( l_1 \) Loss Function

Suppose \( \hat{x}_1 \) and \( \hat{x}_2 \) represent two individual images and \( y = f(\hat{x}_1, \hat{x}_2) \) represents their mixture. We assume the operation \( f(\cdot) \) is unknown and could be either a linear or a non-linear mixing function. Given a mixed input \( y \), our separator aims to predict two individual outputs \( x_1 \) and \( x_2 \):

\[
x_1, x_2 = G(y),
\]

that recover the two original images \( \hat{x}_1 \) and \( \hat{x}_2 \).

We train the separator \( G \) to minimize the distance between its outputs \( (x_1, x_2) \) and their ground truth \( (\hat{x}_1, \hat{x}_2) \). Note that since we can not specify the order of the two outputs for a typical image decomposition problem (especially when the \( \hat{x}_1 \) and \( \hat{x}_2 \) are from the same image domain), the standard pixel-wise \( l_1 \) or \( l_2 \) loss functions do not apply to our task. The solution to this problem is to introduce new loss functions that can deal with unordered outputs. We therefore propose a new loss function called “crossroad \( l_1 \)” loss for our task. The main idea behind is to crossly compute the distance by exchanging the order of the outputs and then take their minimum value as the final response:

\[
\begin{align*}
    l_{\text{cross}}((x_1, x_2), (\hat{x}_1, \hat{x}_2)) &= \min\{d_{1,1} + d_{2,2},  \\
    &\quad \quad d_{1,2} + d_{2,1}\}
\end{align*}
\]

where \( d_{i,j} = \|x_i - \hat{x}_j\|_1 \), \( i, j \in \{1, 2\} \). We use the standard \( l_1 \) function rather than \( l_2 \) in \( d_{i,j} \) since it encourages less blurring effect. The \( G \) can be therefore trained to minimize the loss \( L_{\text{cross}} \) on an entire dataset:

\[
L_{\text{cross}}(G) = E_{\hat{x}_1 \sim p_1(\hat{x})}\{l_{\text{cross}}((x_1, x_2), (\hat{x}_1, \hat{x}_2))\},
\]

where \( p_1(\hat{x}) \) represents the distribution of the image data, and \( i \in \{1, 2\} \).

3.2. Separation Critic

Considering the layer ambiguity, instead of applying any hand-crafted \([67]\) or statistics-based constraints \([27]\) to...
Figure 3: We compare different decomposition priors for image separation, including the “exclusion loss” [17, 67], “Kurtosis” [27], and the proposed “Separation-Critic”. For either of the three metrics, a lower score indicates a heavier mixture. In sub-figure (a), we plot the response of the three metrics given a set of mixed inputs that are synthesized based on Eq. (5). Clearly, if a metric is good enough, the response should be monotonically decreasing as \( \alpha \) increases. We also test on additional nonlinear corruptions, including (b) overexposure, (c) random gamma correction, and (d) random hue transform. The proposed Critic shows better robustness in all conditions.

our output space, we learn a decomposition prior through an adversarial training process. We therefore introduce a “Separation-Critic” \( D_C \) which is trained to distinguish between the outputs \((x_1, x_2)\) and a pair of clean images \((\hat{x}_1, \hat{x}_2)\), and is independent on the order of the two inputs. We express its objective function as follows:

\[
\mathcal{L}_{\text{critic}}(G, D_C) = E_{\hat{x}_i \sim p_i(\hat{x})} \{ \log D_C(\hat{x}_i, \hat{x}_2) \} + E_{x_i \sim p_i(x)} \{ \log(1 - D_C(x_1, x_2)) \} + E_{\hat{x}_i \sim p_i(\hat{x})} \{ \log(1 - D_C(\text{mix}(\hat{x}_1, \hat{x}_2))) \}.
\]

Note that when training the \( D_C \) with fake samples, in addition to the decomposed output \((x_1, x_2)\), we also synthesize a set of “fake” images by mixing two clean images \((x_1', x_2') = \text{mix}(\hat{x}_1, \hat{x}_2)\) with a random linear weight \( \alpha \) to enhance its discriminative ability on mixed images:

\[
x_1' = \alpha \hat{x}_1 + (1 - \alpha) \hat{x}_2, \quad x_2' = (1 - \alpha) \hat{x}_1 + \alpha \hat{x}_2.
\]

At the input end of the \( D_C \), we simply concatenate two images together in the channel dimension for modeling their joint probability distribution. The adversarial training of \( G \) and \( D_C \) is essentially a minimax optimization process, where \( G \) tries to minimize this objective while \( D_C \) tries to maximize it: \( G^* = \arg \min_G \max_{D_C} \mathcal{L}_{\text{critic}}(G, D_C) \).

In Fig. 3 we give four examples to illustrate the effectiveness of our Critic. We compare a well-trained \( D_C \) with two popular metrics for image separation, 1) the exclusion loss [17, 67], and 2) the Kurtosis [27], where the former one enforces separation of two images on the image gradient domain, and the latter one is widely used in BSS for measuring the independence (non-Gaussianity) of the recovered signals: Kurtosis \((u) = E\{u^3\} - 3(E\{u^2\})^2\). For either of the three metrics, a lower score indicates a higher degree of mixture. We mix two images by using Eq. (5) with different \( \alpha \). Clearly, if a metric is good enough, the curve should be monotonically decreasing as \( \alpha \) increases. Fig. 3 (a) shows the response above three metrics. We further add some additional nonlinear corruptions on the mixed images, including (b) random overexposure, (c) random gamma correction, and (d) random hue transform. We can see our Critic shows better robustness, especially for nonlinear degradation.

3.3. Improving Perceptual Quality

To improve the perceptual quality of the decomposed images, we further introduce another two conditional discriminators \( D_{M1} \) and \( D_{M2} \) to enhance high-frequency details. We follow Isola et al. [28] and build \( D_{M1} \) and \( D_{M2} \) as two local perception networks - that only penalize structure at the scale of patches (a.k.a the Markovian discriminator or “PatchGAN”). The \( D_{M1} \) and \( D_{M2} \) try to classify if each \( N \times N \) patch in an image is a clean image (real) or a decomposed one (fake). This type of architecture can be equivalently implemented by building a fully convolutional network with \( N \times N \) perceptive fields, which is more computationally efficient since the responses of all patches can be obtained by taking only one time of forward propagation. We express the objective of \( D_{M1} \) and \( D_{M2} \) as follows:

\[
\mathcal{L}_{M_i}(G, D_{M_i}) = E_{(\hat{x}_i, y) \sim p_i(\hat{x}, y)} \{ \log D_{M_i}(\hat{x}_i, y) \} + E_{(x, y) \sim p_i(x, y)} \{ \log(1 - D_{M_i}(x_i, y)) \},
\]

where \( i = 1, 2 \). Our final objective function is defined as follows:

\[
\mathcal{L}(G, D_C, D_{M_i}) = \mathcal{L}_{\text{cross}}(G) + \beta_C \mathcal{L}_{\text{critic}}(G, D_C) + \beta_M \sum_{i=1,2} \mathcal{L}_{M_i}(G, D_{M_i})
\]
where $\beta_C > 0$ and $\beta_M > 0$ control the balance between the different components of the objective. We aim to solve:

$$G^* = \arg\min_G \max_{D_C, D_M} \mathcal{L}(G, D_C, D_M), \ i = 1, 2. \quad (8)$$

The networks $G$, $D_C$ and $D_M$ thus can be alternatively updated in an end-to-end training process.

### 3.4. Implementation Details

We follow the configuration of the “UNet” [49] when designing the architecture of our separator $G$. We build our $D_C$, $D_{M1}$ and $D_{M2}$ as three standard FCNs with 4, 3, and 3 convolutional layers. The perceptive field of $D_{M1}$ and $D_{M2}$ is set to $N = 30$. We resize the input of $D_C$ to a relatively small size, e.g., $64 \times 64$, to capture the semantics of the whole image instead of adding more layers. We do not use batch-normalization in $G$ as it may introduce unexpected artifacts. As our default settings, we train our model for 200 epochs by using the Adam optimizer with batch_size = 2 and learning_rate = 0.0001. We set $\beta_C = \beta_M = 0$ for the first 10 epochs and set $\beta_C = \beta_M = 0.001$ for the rest epochs. For more implementation details, please refer to our supplementary material.

### 4. Experimental Analysis

We evaluate our methods on four tasks: 1) superimposed image separation, 2) image deraining, 3) image reflection removal, and 4) image shadow removal.

#### 4.1. Separating Superimposed Images

We evaluate our method on two groups of well-known datasets: 1) Stanford-Dogs [31] + VGG-Flowers [45], 2) LSUN Classroom + LSUN Church [63]. During training phase, we randomly select two images $(\hat{x}_1, \hat{x}_2)$ from one group of the datasets and then linearly mix them as $y = \alpha \hat{x}_1 + (1 - \alpha) \hat{x}_2$ with a random linear mixing factor $\alpha$ from the range of $[0.4, 0.6]$. During testing, we set the mixing factor as a constant $\alpha = 0.5$. All images are resized to 256x256 pixels. We follow the datasets’ original train/test split when performing training and evaluation. For the LSUN dataset, due to its large number of images, we only train our method for 20 epochs.

We compare our method with other two popular methods for single mixed image separation: the Double-DIP (CVPR’19) [17] and Levin’s method (TPAMI’07) [35], where the former one is an unsupervised deep learning based method, and the latter one is designed based on image statistics and requires additional user-interactions. Fig. 4 shows two typical results of the above three methods.

<table>
<thead>
<tr>
<th></th>
<th>Dogs+Flwrs.</th>
<th>LSUN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double-DIP</td>
<td>14.70 / 0.661</td>
<td>13.83 / 0.590</td>
</tr>
<tr>
<td>Levin et al.</td>
<td>10.54 / 0.444</td>
<td>10.46 / 0.366</td>
</tr>
<tr>
<td>Our method (tr. on ImageNet)</td>
<td>23.32 / 0.803</td>
<td>21.63 / 0.773</td>
</tr>
<tr>
<td>Our method (w/ default tr. set)</td>
<td>25.51 / 0.849</td>
<td>26.32 / 0.883</td>
</tr>
</tbody>
</table>

Table 1: Comparisons (PSNR/SSIM) of different methods on mixed image separation: 1) Stanford-Dogs + VGG-Flowers, 2) LSUN Classroom + LSUN Church. To test on the cross-domain generalization ability of our method, we also train on ImageNet and test on the above datasets. Higher scores indicate better.

Since Levin’s method requires heavy user interactions and the Double-DIP is extremely slow (~40 minutes / image on a GTX-1080Ti GPU), we only evaluate these two methods on 10 images that are randomly selected from each of our test set.
Figure 5: A comparison of image separation results with the standard $l_1$ loss and the proposed crossroad $l_1$ loss.

Figure 6: A comparison of the results based on different image separation priors: Zhang’s exclusion loss [67], Kurtosis [27], and the Separation Critic (ours).

Table 2: A comparison of the image separation results with standard $l_1$ loss and the proposed crossroad $l_1$ loss on Stanford-Dogs dataset [31]. PSNR/SSIM: higher scores indicate better. PI: lower scores indicate better.

<table>
<thead>
<tr>
<th>Ablation</th>
<th>PSNR</th>
<th>SSIM</th>
<th>PI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (with standard $l_1$ loss)</td>
<td>16.26</td>
<td>0.6768</td>
<td>23.34</td>
</tr>
<tr>
<td>Ours (with cross-$l_1$ loss)</td>
<td>22.74</td>
<td>0.7782</td>
<td>21.92</td>
</tr>
</tbody>
</table>

Table 3: Evaluation of decomposition results with different priors on Stanford-Dogs + VGG-Flowers datasets (with overexposure and noise). PSNR/SSIM: higher scores indicate better. PI: lower scores indicate better.

<table>
<thead>
<tr>
<th>Ablation</th>
<th>PSNR</th>
<th>SSIM</th>
<th>PI</th>
</tr>
</thead>
<tbody>
<tr>
<td>cross-$l_1$ only</td>
<td>17.34</td>
<td>0.6693</td>
<td>22.41</td>
</tr>
<tr>
<td>cross-$l_1$ + Zhang’s loss [67]</td>
<td>17.77</td>
<td>0.6840</td>
<td>23.09</td>
</tr>
<tr>
<td>cross-$l_1$ + Kurtosis [27]</td>
<td>18.02</td>
<td>0.7141</td>
<td>22.28</td>
</tr>
<tr>
<td>cross-$l_1$ + Sep-Critic</td>
<td>18.29</td>
<td>0.7231</td>
<td>22.16</td>
</tr>
<tr>
<td>cross-$l_1$ + Sep-Critic + adv-pcpt.</td>
<td>18.27</td>
<td>0.6938</td>
<td>21.97</td>
</tr>
</tbody>
</table>

4.2. Controlled Experiments

Analysis on the crossroad $l_1$ loss. To evaluate the importance of our crossroad $l_1$ loss, we replace it with a standard $l_1$ while keeping other settings unchanged. We train the above two models on the Stanford-Dogs [31]. Table 2 shows the evaluation results of the two models and Fig. 5 shows a group of visual comparisons. We can see our method clearly separates the two images while the standard $l_1$ fails to do that and encourages “averaged” outputs.

Analysis on the adversarial losses. We compare our method with different decomposition priors on Stanford-Dogs + VGG-Flowers datasets. In addition to the linear mixing inputs, we also apply overexposure and noises to increase the separation difficulties. To better evaluate the perceptual quality, we introduce another metric called Perceptual Index (PI) [3]. The PI was originally introduced as a no-reference image quality assessment method based on the low-level image statistics and is recently widely used for evaluating super-resolution results [3, 56]. In Fig. 6 and Table 3, we compare our adversarial losses with the exclusion loss [17, 67] and the Kurtosis [27]. As we can see, the integration of our adversarial losses yields noticeable improvements in the output quality. We found the exclusion loss encourages blurred output and it is hard to balance it with other losses. We also found the Kurtosis may introduce a slight color-shift on its outputs.

4.3. Application: Deraining

We conduct our deraining experiments on several datasets: Rain100H [62], Rain800 [65], and DID [64]. To better test the generalization ability of our method, we follow Zhang et al. [64] to train our method on DID [64], and then randomly sample 1,000 images from the dataset [13] as another testing set, denoted as DDN1k. Given a rainy input image, we use its clean background and the rain streak
4.4. Application: Image Reflection Removal

We test our method on two large scale datasets for reflection removal \[59, 60\]. The dataset \[60\] consists of over 50,000 images which are synthesized by mixing their transmission layers and reflection layers (linear mixture + Gaussian blur). The dataset \[59\] consists of 12,000 images with three types of reflections: “focused”, “defocused”, and “ghosting”. We achieve the best results in all experimental entries. Results reported by: † \[59\], ¶ \[60\].

Figure 8 shows a group of our deraining results on some real-world rain images. 1st row: input. 2nd row: output.
and “ghosting”, which are synthesized by using adversarial training. When we train our model, the transmission layers are used as the reference for our first output. We discard the synthesized reflection in our second output since it cannot capture ground truth reflections. We compare with several sota reflection removal methods, including the method of Zhang et al. (ICCV’18) [67], BDN (ECCV’18) [60], RmNet (CVPR’19) [59], etc. Table 6 shows the quantitative evaluations of these methods. Note that although RmNet [59] uses auxiliary images [53, 67] during training, we still achieve the best results in all experimental entries. We test on a set of real-world reflection images [67]. We train our model on the synthetic training set [67] and then evaluate on its real-world testing set. Fig. 9 and Table 7 shows some comparison results.

4.5. Application: Shadow Removal

In this experiment, we test our method on two shadow removal datasets: ISTD [54] and SRD [46]. The two datasets consist of 1,870 and 3,088 shadow/shadow-free image pairs that captured in real-world environments. We compare our methods with some sota shadow removal methods, including DSC (TPAMI19) [22], ST-CGAN (CVPR’18) [54], and ARGAN (CVPR’19) [8]. Table 8 shows the evaluation results of these methods. We do not compare ST-CGAN and ARGAN on SRD [46] because the authors did not report their accuracy on this dataset and the code has not been released yet. We follow the evaluation metric introduced by Guo et al. [21], where a lower score indicates a better result. Fig. 11 gives an comparison example of our method and DSC [22] on the above two datasets.

5. Conclusion

We propose a unified framework for single superimposed image separation - a group of challenging tasks in computer vision and signal processing field. Different from the previous methods that are either statistically or empirically designed, we shed light on the possibilities of the adversarial training for this task. Without specific tuning, our method achieves the state of the art results on multiple tasks, including image deraining, image reflection removal, and image shadow removal. In our future work, we will focus on some more general image separation tasks (e.g., on ImageNet) and also the possibility of speech signal separation by using our method.

<table>
<thead>
<tr>
<th>Method</th>
<th>ISTD [54]</th>
<th>SRD [46]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang et al. [61] (TIP’12) *†</td>
<td>15.63</td>
<td>22.57</td>
</tr>
<tr>
<td>Guo et al. [21] (TPAMI’12) *†</td>
<td>9.300</td>
<td>12.60</td>
</tr>
<tr>
<td>Gong et al. [18] (BMVC’14) *†</td>
<td>8.530</td>
<td>8.730</td>
</tr>
<tr>
<td>DeshadowNet [46] (CVPR’17) ¶†</td>
<td>7.830</td>
<td>6.640</td>
</tr>
<tr>
<td>DSC [22] (TPAMI19) ¶†</td>
<td>7.100</td>
<td>6.210</td>
</tr>
<tr>
<td>ST-CGAN [54] (CVPR’18) ¶</td>
<td>7.470</td>
<td>-</td>
</tr>
<tr>
<td>ARGAN [8] (CVPR’19) ¶</td>
<td>6.680</td>
<td>-</td>
</tr>
<tr>
<td>Our method</td>
<td>6.566</td>
<td>5.823</td>
</tr>
</tbody>
</table>
References


[58] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simon-


