BITS-NET: BLIND IMAGE TRANSPARENCY SEPARATION NETWORK

Chao Zhou, Zhaoyan Lyu, Miguel R.D. Rodrigues

Dept. Electronic and Electrical Engineering, University College London
{chao.zhou.18, z.lyu.17, m.rodrigues}@ucl.ac.uk

ABSTRACT

This research presents a new approach for blind single-image transparency separation, a significant challenge in image processing. The proposed framework divides the task into two parallel processes: feature separation and image reconstruction. The feature separation task leverages two deep image prior (DIP) networks to recover two distinct layers. An exclusion loss and deep feature separation loss are used to decompose features. For the image reconstruction task, we minimize the difference between the mixed image and the re-mixed image while also incorporating a regularizer to impose natural priors on each layer. Our results indicate that our method performs comparably or outperforms state-of-the-art approaches when tested on various image datasets.

Index Terms— blind image separation, deep image prior, deep learning, computer vision

1. INTRODUCTION

Images composed of two half-transparent layers [1] are ubiquitous in research and daily life, such as photos with reflections [1, 2, 3], double exposure photography [4], and MRI for art investigations [5]. With the advancement of deep learning techniques, algorithms for separating these overlaid images have gained momentum [2, 3, 6]. However, many of these algorithms are based on deep neural networks that are trained in a supervised manner, which requires a large dataset of paired overlaid images and their corresponding ground-truths [1, 7, 8]. Unfortunately, such datasets are not always readily available [5]. Additionally, these algorithms often rely on strong assumptions, such as one of the layers being simple, smooth, or out-of-focus [1, 2, 3]. Furthermore, distributional shifts between the training and testing samples can result in suboptimal performance of supervised learning approaches [1, 2, 3, 7, 8]. To address these challenges, this paper presents an unsupervised algorithm for separating overlaid images composed of two natural image layers.

Denote $\boldsymbol{I} \in \mathbb{R}^{h \times w \times c}$ as the overlaid image, which is comprised of two separate layers $\boldsymbol{y}_1$ and $\boldsymbol{y}_2 \in \mathbb{R}^{h \times w \times c}$. Here, h, w, and c represent the height, width, and number of channels in the image, respectively. The overlaid image can be modelled as the sum of its two underlying layers, as follows:

$$\boldsymbol{I} = \boldsymbol{y}_1 + \boldsymbol{y}_2 \tag{1}$$

The separation of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ is an underdetermined problem without additional information. To ensure a successful separation, we introduce two criteria:

1. **Separation criterion**: The features of the two layers should be disentangled, with simple patterns on each layer and minimal correlation across layers.

2. **Reconstruction criterion**: The remixed image should be as similar as possible to the original overlaid image and the separated layers should be as natural as possible.

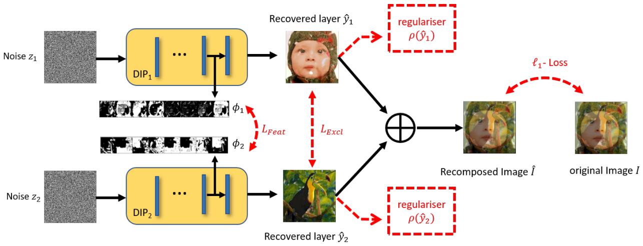

Noise z_1
DIP$_1$
Recovered layer $\hat{y}_1$
regulariser $\rho(\hat{y}_1)$
ℓ_1- Loss
ϕ_1
L_{Feat}
L_{Excl}
ϕ_2
Recomposed Image $\hat{I}$
original Image I
Noise z_2
DIP$_2$
regulariser $\rho(\hat{y}_2)$
Recovered layer $\hat{y}_2$

Fig. 1. Overall framework of BITS-Net.

These criteria form the basis of our proposed algorithm. To enforce the **Separation Criterion**, we employ separate DIP networks [9] with independent random inputs and implement an exclusion loss [1] on the recovered layers, as well as a deep feature separation loss on the feature maps of the two DIP networks. To meet the **Reconstruction Criterion**, we use an ℓ_1 reconstruction loss on the remixed image and incorporate a natural prior for each separated layer. The proposed algorithm is referred to as *BITS-Net*, which stands for Blind (Unsupervised) Image Transparency Separation Network. The overall structure of the proposed algorithm is depicted in Fig. 1. Experiment results demonstrate that the proposed unsupervised learning approach outperforms other state-of-the-art methods such as DoubleDIP [10], and is comparable in performance to supervised image separation algorithms including [1, 7].

[1] In this paper, the word *layer* refers image layers unless specified otherwise.

2. RELATED WORK

Examples of image separation problems can be found in a variety of studies such as reflection reduction, shadow removal, and others [1, 2, 8, 3, 7, 10, 11]. In this work, we focus on the challenging task of transparency image separation, which involves separating an image with intricate patterns and details that are more complex than reflections and shadows.

Most existing approaches to solving such tasks rely on supervised learning, requiring the network to be pre-trained on a large labeled dataset. For instance, Fan *et al.* [8] split the image separation task into two subtasks: one subtask that involves a supervised sub-network predicting the edge of the target image, and another that reconstructs the target image by leveraging the predicted edge maps. Zhang *et al.* [1] proposed a supervised algorithm with three loss terms: a feature loss based on a pre-trained VGG network, an adversarial loss, and an exclusion loss. There are also other methods such as the Blind Image Decomposition (BID) [7] which assumes that the overlayed layers are from known categories. On the other hand, the SILS method [11] utilizes the inherent properties of unpaired overlayed and single-layer images. However, these techniques necessitate pre-training the model on a substantial training dataset. Additionally, these methods are limited in their ability to handle more complex patterns as the losses used may not guarantee good separation results when the patterns become more intricate.

In contrast to the supervised learning approaches, DoubleDIP [10] presents an unsupervised framework for general image decomposition tasks, including image separation, segmentation, and dehazing. This approach suggests that a successful decomposition of images should fulfil three criteria: (1) The re-composed image should closely resemble the original overlaid image, (2) each separated layer should be as simple as possible, and (3) the recovered layers should be independent of each other. However, as our later experiments will demonstrate, these criteria are not sufficient to achieve high-quality image decomposition.

Alternatively, there are methodologies that make use of multiple images, such as flash and no flash pairs [12], images with different focus settings [13], and images with different reflections [14]. Nonetheless, these techniques entail the acquisition of multiple images, which are often difficult to obtain in a single shot and thus fall outside the scope of this study.

3. PROPOSED APPROACH

We break down the image separation problem into two parallel tasks: **feature separation** and **image reconstruction**. Specifically, the feature separation task focuses on separating various features and patterns into separate layers, while the image reconstruction task aims to recover each distinct layer to appear as natural as possible, and the re-mixed image to be as close as the original overlaid image. To achieve these goals, we formulate the loss function as follows:

$$L = L_{Sep} + L_{Recon} \tag{2}$$

where L_{Sep} evaluates the effectiveness of the feature separation, and L_{Recon} assesses the quality of the separated layers and the re-mixed image. We will now elaborate on the design of these two terms.

3.1. Feature Separation L_{Sep}

In order to achieve feature separation, we introduce three key notions. Firstly, each separate layer should have a simple pattern. This is accomplished through the use of a DIP network [9] for each layer. Secondly, the correlation between each layer should be low, which is ensured through the implementation of an exclusion loss [1] on the outputs from different DIP networks. Finally, the latent features of each separate layer should be mutually independent, which is achieved by implementing a deep feature separation loss on the feature maps of intermediate layers within the DIP networks. It is worth noting that although the concepts of simple patterns within each layer and minimizing the correlation between each layer have been previously introduced in the DoubleDIP framework [10], our experiments have shown that these alone are not sufficient to produce effective image separation results.

Specifically, we use a DIP network DIP_i to recover the distinct layer $\boldsymbol{y}_i$ of the mixed image, represented by $\hat{\boldsymbol{y}}_i = DIP_i(\boldsymbol{z}_i; \Theta_i)$. Here, Θ_i refers to the learnable parameters of this DIP network and $\boldsymbol{z}_i$ is a noise input. The feature maps of DIP_i are denoted as $\boldsymbol{\phi}_i = \{\boldsymbol{\phi}_{i,k}\}_{k\in[K]}$, where $\boldsymbol{\phi}_{i,k}$ represents the feature map of the k^{th} intermediate network layer of DIP_i. $[K]$ is the set $\{1, ..., K\}$ with K denoting the number of intermediate layers of DIP_i. The loss function of the feature separation task is formulated as follows:

$$\begin{aligned} L_{Sep} =& \alpha_1 \cdot Loss_{Excl}(\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2) + \\ & \alpha_2 \cdot Loss_{Feat}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \end{aligned} \tag{3}$$

where $Loss_{Excl}$ is the exclusion loss [1] that minimizes the correlation between the gradients of $\hat{\boldsymbol{y}}_1$ and $\hat{\boldsymbol{y}}_2$. $Loss_{Feat}$ is the deep feature loss, which enforces the exclusion loss on the feature maps of the intermediate network layers of the DIP networks. α_1 and α_2 are hyperparameters. The deep feature loss is defined as follows:

$$Loss_{Feat}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = \sum_{k\in\boldsymbol{\kappa}} Loss_{Excl}(\boldsymbol{\phi}_{1,k}, \boldsymbol{\phi}_{2,k}) \tag{4}$$

where $\boldsymbol{\kappa}$ is a set of pre-defined intermediate layers for computing the deep feature separation loss.

3.2. Image Reconstruction L_{Recon}

To resolve the image reconstruction task, a conventional approach would be to minimize the ℓ_1-loss between the original overlaid image $\boldsymbol{I}$ and the re-mixed image $\hat{\boldsymbol{I}}$. Meanwhile, we embed natural prior in to the reconstructed layers such that the recovered layers are close to natural imagesIn particular, we incorporate the Regularizer by Denoising (RED) approach [15] into each DIP network. RED has been shown to be an effective regularizer that can tackle any image inverse problem through the exploitation of the well-developed image denoising engine [15, 16]. Given a denoiser $f(\cdot)$, the RED regularization function is defined as follows:

$$\rho(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T(\boldsymbol{x} - f(\boldsymbol{x})) \tag{5}$$

where $\boldsymbol{x}$ is a reconstructed image to be regularized.

Finally, the loss function for image reconstruction is expressed as follows:

$$L_{Recon} = \|\boldsymbol{I} - \hat{\boldsymbol{I}}\|_1 + \lambda_1 \cdot \rho(\hat{\boldsymbol{y}}_1) + \lambda_2 \cdot \rho(\hat{\boldsymbol{y}}_2) \tag{6}$$

where $\rho(\cdot)$ represents the RED regularizer, which is defined in eq. (5). λ_1 and λ_2 are the weights of the regularizers. $\hat{\boldsymbol{y}}_1$ and $\hat{\boldsymbol{y}}_2$ are the outputs of the DIP networks and $\hat{\boldsymbol{I}}$ is the re-mixed image.

3.3. Overall Objective and Optimization

The overall training objective is combined with the aforementioned feature separation loss and the image reconstruction as follows:

$$\begin{aligned} \min_{\Theta} L = {} & \alpha_1 \cdot Loss_{Excl}(\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2) \\ & + \alpha_2 \cdot Loss_{Feat}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \\ & + \|\boldsymbol{I} - \hat{\boldsymbol{I}}\|_1 + \lambda_1 \cdot \rho(\hat{\boldsymbol{y}}_1) + \lambda_2 \cdot \rho(\hat{\boldsymbol{y}}_2) \end{aligned} \tag{7}$$

where Θ represents the learnable parameters of the overall DIP networks. The overall structure is shown in Fig. 1.

To optimize this objective function, we use the Alternating Directions Method of Multiplier (ADMM) method, which has been shown to have faster and better convergence [17, 18]. For simplicity, we denote the first three terms in Eq. (7) as L_1. By ADMM, Eq. (7) is reformulated as follows:

$$\begin{aligned} \min_{\Theta, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2} & L_1 + \lambda_1 \rho(\tilde{\boldsymbol{y}}_1) + \lambda_2 \rho(\tilde{\boldsymbol{y}}_2) \\ s.t. \quad & \tilde{\boldsymbol{y}}_1 = \hat{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2 = \hat{\boldsymbol{y}}_2 \end{aligned} \tag{8}$$

This reformulation involves introducing auxiliary variables $\tilde{\boldsymbol{y}}_i$ to split the network estimation $\hat{\boldsymbol{y}}_i$ in L_1 and ρ, where $i \in \{1, 2\}$. The ADMM solver provides iterative update rules [16] given by:

$$\begin{aligned} \hat{\boldsymbol{y}}_1^{(j+1)}, \hat{\boldsymbol{y}}_2^{(j+1)} = arg \min_{\Theta} L_1 & + \frac{\mu_1}{2}\|\tilde{\boldsymbol{y}}_1^{(j)} - \hat{\boldsymbol{y}}_1 - \boldsymbol{u}_1^{(j)}\|_2^2 \\ & + \frac{\mu_2}{2}\|\tilde{\boldsymbol{y}}_2^{(j)} - \hat{\boldsymbol{y}}_2 - \boldsymbol{u}_2^{(j)}\|_2^2 \end{aligned} \tag{9}$$

$$\begin{cases} \tilde{\boldsymbol{y}}_1^{(j+1)} = \frac{1}{\lambda_1+\mu_1}(\lambda_1 f(\tilde{\boldsymbol{y}}_1^{(j)}) + \mu_1(\hat{\boldsymbol{y}}_1^{(j+1)} + \boldsymbol{u}_1^{(j)})) \\ \tilde{\boldsymbol{y}}_2^{(j+1)} = \frac{1}{\lambda_2+\mu_2}(\lambda_2 f(\tilde{\boldsymbol{y}}_2^{(j)}) + \mu_2(\hat{\boldsymbol{y}}_2^{(j+1)} + \boldsymbol{u}_2^{(j)})) \end{cases} \tag{10}$$

$$\begin{cases} \boldsymbol{u}_1^{(j+1)} = \quad \boldsymbol{u}_1^{(j)} + \hat{\boldsymbol{y}}_1^{(j+1)} - \tilde{\boldsymbol{y}}_1^{(j+1)} \\ \boldsymbol{u}_2^{(j+1)} = \quad \boldsymbol{u}_2^{(j)} + \hat{\boldsymbol{y}}_2^{(j+1)} - \tilde{\boldsymbol{y}}_2^{(j+1)} \end{cases} \tag{11}$$

where $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ represent the Lagrange multipliers and μ_1, μ_2 are the ADMM free parameters. The variables $\boldsymbol{u}_1$ at the $(j+1)^{th}$ ADMM iteration are represented as $\boldsymbol{u}_1^{(j+1)}$. Eq. (9) is solved by training the network via a gradient descent algorithm, such as the Adam optimizer [19]. Eq. (10) is derived from the fixed point solution in [15]. Eq. (11) is a straightforward update for the Lagrange multipliers. The iterative update rules continue until convergence is achieved.

3.4. Implementation Details

The proposed algorithm is implemented using PyTorch [2] and follows a similar architecture as the DIP network presented in [9, 10]. To stabilize the learning process, the same input perturbations and data augmentation techniques outlined in [10] are employed. The network is optimized using the Adam optimizer [19] with a learning rate of 0.008. The number of ADMM iterations is set to $8,000$. The overall model converges within approximately 20 minutes when run on a server with an NVIDIA Tesla V100 GPU.

4. EXPERIMENTS

In this section, we experimentally compare the proposed method with state-of-the-art algorithms, including BIDeN [7], Zhang *et al.* [1], and DoubleDIP [10]. It should be noted that BIDeN and Zhang *et al.*'s methods are supervised. The overlaid images used in our experiments are generated by combining two images randomly selected from the Set5 and Set14 datasets [20] using the mixing function specified in Eq. (1). We provide both qualitative and quantitative evaluations, with the latter measured by widely used metrics in the literature [10, 21], namely peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). The default setup published on GitHub is used to generate unmixed images by BIDeN, Zhang*et al.*, and DoubleDIP. Our proposed algorithm is fine-tuned through grid search, with $\alpha_1 = 0.01$, $\alpha_2 = 0.001$, $\lambda_1 = \lambda_2 = 0.5$, and $\mu_1 = \mu_2 = 0.5$.

[2] Code will be made available upon acceptance.

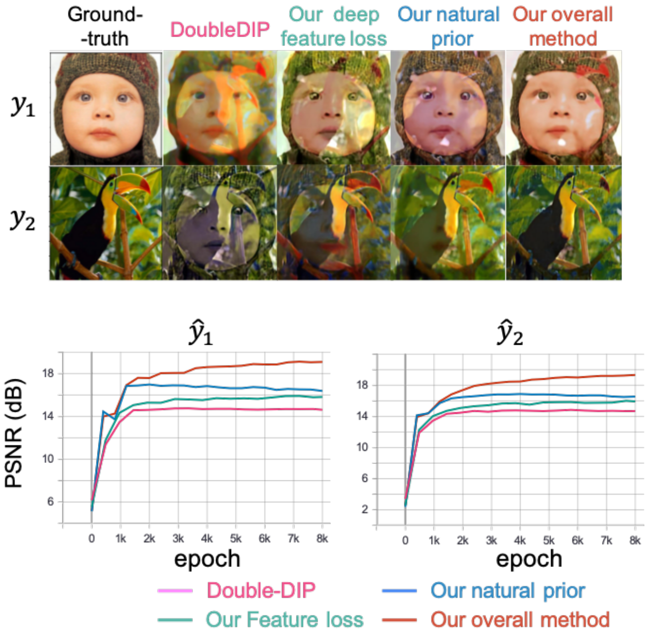
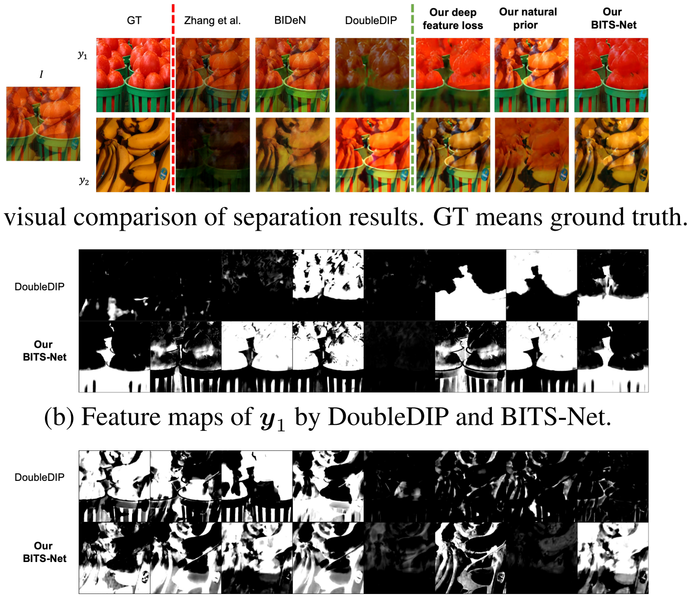


Fig. 2. Ablation study.

In order to better showcase the contribution of BITS-Net, we conduct an ablation study as shown in Fig. 2. The upper figures show the separated layers obtained using different components of BITS-Net, while the lower figures display the corresponding training curves. Different methods are color-coded, with red indicating the overall BITS-Net results, green representing BITS-Net without natural prior, blue for BITS-Net without deep feature loss, and pink indicating the use of Double-DIP algorithm only. The results show that the overall BITS-Net method produces the best visual and numerical outcomes when separating the baby-bird overlaid image. Removing either deep feature loss or natural prior leads to a decline in performance, but all methods still perform better than DoubleDIP.

We compare our proposed BITS-Net with state-of-the-art methods on overlaid banana+tomato images. As shown in Fig. 3 and Table. 1, BITS-Net outperforms the competitors with significant improvements in PSNR and SSIM, as well as visually more natural and better-separated features. Notably, our unsupervised method also outperforms supervised methods such as Zhang *et al.* and BIDeN. Compared to DoubleDIP, BITS-Net learns cleaner, sharper, and better-separated features, as shown in Fig.3(b) and (c). Table. 1 shows improvements of up to 9.15 dB in PSNR and 0.32 in SSIM.



(a) visual comparison of separation results. GT means ground truth.

(b) Feature maps of $\boldsymbol{y}_1$ by DoubleDIP and BITS-Net.

(c) Feature maps of $\boldsymbol{y}_2$ by DoubleDIP and BITS-Net.

Fig. 3. Qualitative comparisons: (a) the separation results by various methods. The quantitative comparisons are shown in Table. 1. (b) the feature maps of iamge layer $\boldsymbol{y}_1$ from the second-to-last network layer of the DoubleDIP and our BITS-Net. (c) the feature maps of image layer $\boldsymbol{y}_2$.

Table 1. Quantitative comparisons of the experiment presented in Fig. 3.

Metrics	Zhang *et al* [1]	BIDeN [7]	DoubleDIP [10]	BITS-Net
PSNR of $\boldsymbol{y}_1$	13.71	14.81	12.12	**18.46**
PSNR of $\boldsymbol{y}_2$	9.30	15.73	12.11	**18.45**
SSIM of $\boldsymbol{y}_1$	0.59	0.52	0.58	**0.71**
SSIM of $\boldsymbol{y}_2$	0.44	0.56	0.63	**0.76**

5. CONCLUSION

In this work, we study the single image transparency separation problem by dividing it into two parallel tasks: (1) the feature separation task and (2) the image reconstruction task. In particular, for the feature separation task, we (a) deploy separated DIP networks and (b) impose exclusion loss and deep feature separation loss to ensure the recovered layers have simple patterns, small correlations, and independent latent features. For the image reconstruction task, we (a) enforce ℓ_1 loss on the re-mixed image and (b) impose an explicit regularizer to promote the natural recovery of each layer. The overall model is optimized by the ADMM algorithm for better stability. Experiments show that our proposed unsupervised BITS-Net method outperforms other state-of-the-art approaches, including supervised ones.

6. REFERENCES

[1] Xuaner Zhang, Ren Ng, and Qifeng Chen, “Single image reflection separation with perceptual losses,” in *Pro-*

ceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4786–4794.

[2] Renjie Wan, Boxin Shi, Haoliang Li, Ling-Yu Duan, and Alex C Kot, "Reflection scene separation from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2398–2406.

[3] Zhixiang Chi, Xiaolin Wu, Xiao Shu, and Jinjin Gu, "Single image reflection removal using deep encoder-decoder network," *arXiv preprint arXiv:1802.00094*, 2018.

[4] N Dombrowski and A Levy, "Double-exposure photography," *Nature*, vol. 202, no. 4931, pp. 521–521, 1964.

[5] Z Sabetsarvestani, Barak Sober, Catherine Higgitt, Ingrid Daubechies, and MRD Rodrigues, "Artificial intelligence for art investigation: Meeting the challenge of separating x-ray images of the ghent altarpiece," *Science advances*, vol. 5, no. 8, pp. eaaw7416, 2019.

[6] Ofer Springer and Yair Weiss, "Reflection separation using guided annotation," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 1192–1196.

[7] Junlin Han, Weihao Li, Pengfei Fang, Chunyi Sun, Jie Hong, Mohammad Ali Armin, Lars Petersson, and Hongdong Li, "Blind image decomposition," *arXiv preprint arXiv:2108.11364*, 2021.

[8] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf, "A generic deep architecture for single image reflection removal and image smoothing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3238–3247.

[9] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Deep image prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9446–9454.

[10] Yosef Gandelsman, Assaf Shocher, and Michal Irani, "Double-dip: Unsupervised image decomposition via coupled deep-image-priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11026–11035.

[11] Yunfei Liu and Feng Lu, "Separate in latent space: Unsupervised single image layer separation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 11661–11668.

[12] Amit Agrawal, Ramesh Raskar, Shree K Nayar, and Yuanzhen Li, "Removing photography artifacts using gradient projection and flash-exposure sampling," in *ACM SIGGRAPH 2005 Papers*, pp. 828–835. 2005.

[13] Yoav Y Schechner, Nahum Kiryati, and Ronen Basri, "Separation of transparent layers using focus," *International Journal of Computer Vision*, vol. 39, no. 1, pp. 25–39, 2000.

[14] Byeong-Ju Han and Jae-Young Sim, "Reflection removal using low-rank matrix completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5438–5446.

[15] Yaniv Romano, Michael Elad, and Peyman Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[16] Gary Mataev, Peyman Milanfar, and Michael Elad, "Deepred: Deep image prior powered by red," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[17] Stephen Boyd, Neal Parikh, and Eric Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Now Publishers Inc, 2011.

[18] Shaozhe Tao, Daniel Boley, and Shuzhong Zhang, "Convergence of common proximal methods for l1-regularized least squares," in *International Joint Conference on Artificial Intelligence*, 2015.

[19] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] Radu Timofte, Vincent De Smet, and Luc Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1920–1927.

[21] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.