"Double-DIP" : Unsupervised Image Decomposition via Coupled Deep-Image-Priors

Yossi Gandelsman Assaf Shocher Michal Irani
Dept. of Computer Science and Applied Mathematics
The Weizmann Institute of Science, Israel
Project Website: *www.wisdom.weizmann.ac.il/∼vision/DoubleDIP/*

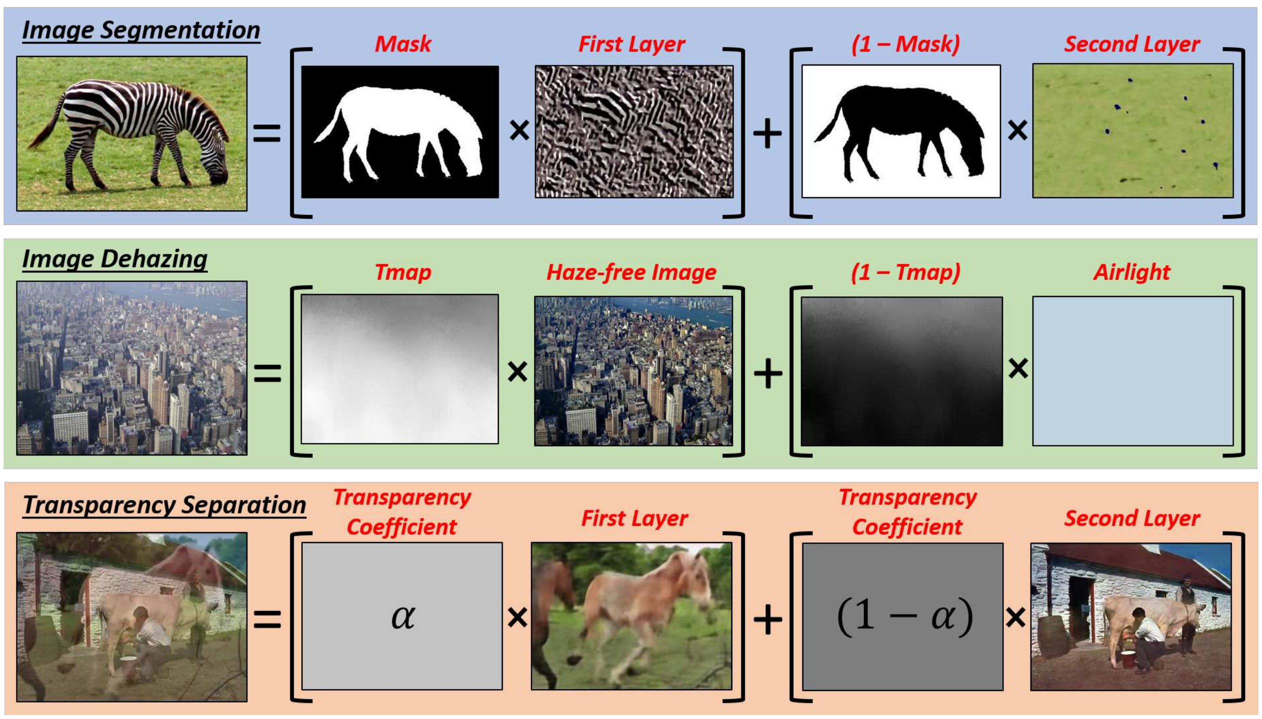

Figure 1: **A unified framework for image decomposition.** An image can be viewed as a mixture of "simpler" layers. Decomposing an image into such layers provides a unified framework for many seemingly unrelated vision tasks (e.g., segmentation, dehazing, transparency separation). Such a decomposition can be achieved using "Double-DIP".

Abstract

Many seemingly unrelated computer vision tasks can be viewed as a special case of image decomposition into separate layers. For example, image segmentation (separation into foreground and background layers); transparent layer separation (into reflection and transmission layers); Image dehazing (separation into a clear image and a haze map), and more. In this paper we propose a unified framework for unsupervised layer decomposition of a single image, based on coupled "Deep-image-Prior" (DIP) networks. It was shown [38] that the structure of a single DIP generator network is sufficient to capture the low-level statistics of a single image. We show that coupling multiple such DIPs provides a powerful tool for decomposing images into their basic components, for a wide variety of applications. This capability stems from the fact that the internal statistics of a mixture of layers is more complex than the statistics of each of its individual components. We show the power of this approach for Image-Dehazing, Fg/Bg Segmentation, Watermark-Removal, Transparency Separation in images and video, and more. These capabilities are achieved in a totally unsupervised way, with no training examples other than the input image/video itself. [1]

1. Introduction

Various computer vision tasks aim to decompose an image into its individual components. In image/video segmen-

[1]Project funded by the European Research Council (ERC) under the Horizon 2020 research & innovation programme (grant No. 788535)

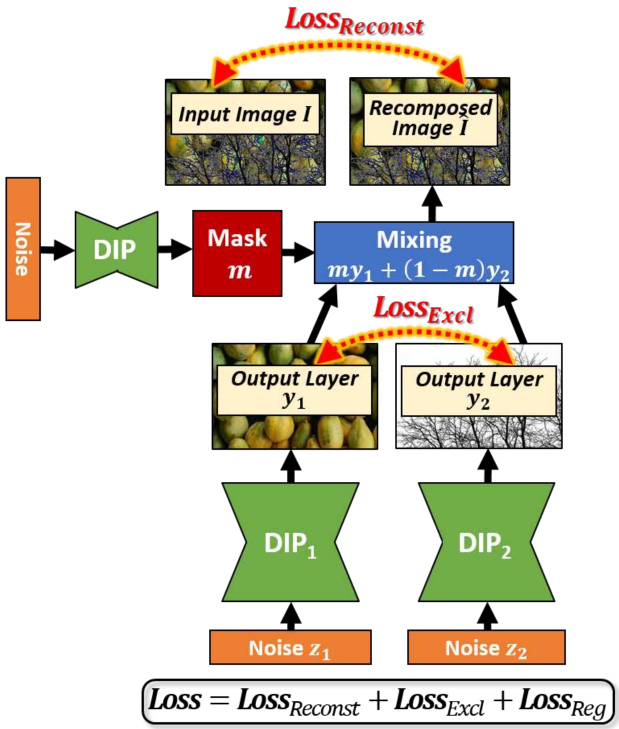

Figure 2: **Double-DIP Framework.** *Two Deep-Image-Prior networks (DIP_1 & DIP_2) jointly decompose an input image I into its layers (y_1 & y_2). Mixing those layers back according to a learned mask m, reconstructs an image $\hat{I} \approx I$.*

tation, the task is to decompose the image into meaningful sub-regions, such as foreground and background [1, 5, 17, 24, 31]. In transparency separation, the task is to separate the image into its superimposed reflection and transmission [37, 32, 26, 14]. Such transparency can be a result of accidental physical reflections, or due to intentional transparent overlays (e.g., watermarks). In image dehazing [6, 23, 18, 30, 8], the goal is to separate a hazy/foggy image into its underlying haze-free image and the obscuring haze/fog layers (airlight and transmission map). Fig. 1 shows how all these very different tasks can be casted into a single unified framework of *layer-decomposition*. What is common to all these decompositions is the fact that the *distribution of small patches* within each separate layer is "simpler" (more uniform) than in the original mixed image, resulting in *strong internal self-similarity*.

Small image patches (e.g., 5x5, 7x7) have been shown to repeat abundantly inside a single natural image [19, 41]. This strong internal patch recurrence was exploited for solving a large variety of computer vision tasks [9, 13, 16, 15, 19, 36, 29, 7, 11]. It was also shown that the empirical entropy of patches inside a *single image* is much smaller than the entropy in a *collection of images* [41]. It was further observed by [5] that the *empirical entropy* of small image regions composing a segment, is smaller than the *empirical cross-entropy* of regions across different segments within the same image. This observation has been successfully used for *unsupervised* image segmentation [5, 17]. Finally, it was observed [6] that the distribution of patches in a *hazy image* tends to be more diverse (weaker internal patch similarity) than in its underlying *haze-free* image. This observation was exploited by [6] for blind image dehazing.

In this paper we combine the power of the internal patch recurrence (its strength in solving unsupervised tasks), with the power of Deep-Learning. We propose an *unsupervised* Deep framework for decomposing a single image into its layers, such that the distribution of "image elements" within each layer is "simple". We build on top of the "Deep Image Prior" (DIP) work of Ulyanov *et al*. [38]. They showed that the structure of a *single* DIP generator network is sufficient to capture the low-level statistics of a *single* natural image. The input to the DIP network is *random noise*, and it trains to *reconstruct a single image* (which serves as its sole output training example). This network was shown to be quite powerful for solving inverse problems like denoising, super-resolution and inpainting, in an unsupervised way.

We observe that when employing *a combination of multiple DIPs* to reconstruct an image, those DIPs tend to "split" the image, such that the patch distribution of each DIP output is "simple". Our approach for *unsupervised multi-task layer decomposition* is thus based on a combination of multiple (two or more) DIPs which we coin "Double-DIP". We demonstrate the applicability of this approach to a wide range of computer vision tasks, including Image-Dehazing, Fg/Bg Segmentation of images and videos, Watermark Removal, and Transparency Separation in images and videos.

Double-DIP is *general-purpose* and caters many different applications. *Special-purpose* methods designed for one specific task may outperform Double-DIP on their own challenge. However, to the best of our knowledge, this is the first framework that is able to handle well such a large variety of image-decomposition tasks. Moreover, in some tasks (e.g., image dehazing), Double-DIP achieves comparable and even better results than leading methods.

2. Overview of the Approach

Observe the illustrative example in Fig. 3a. Two different textures, X and Y, are mixed to form a more complex image Z which exhibits layer transparency. The distribution of small patches and colors inside each pure texture is *simpler* than the distribution of patches and colors in the combined image. Moreover, the similarity of patches across the two textures is very weak. It is well known [12] that if X and Y are two *independent* random variables, the entropy of their sum $Z = X + Y$ is larger than their individual entropies: $max\{H(X), H(Y)\} \leq H(Z)$. We leverage this fact to separate the image into its natural "simpler" components.

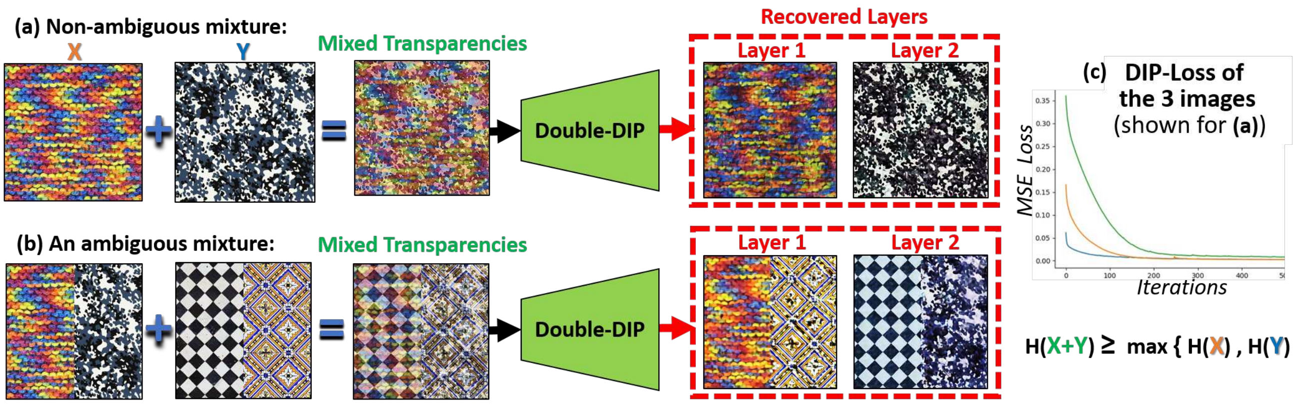

Figure 3: **The complexity of mixtures of layers vs. the simplicity of the individual components.** *(See text for explanation).*

2.1. Single DIP vs. Coupled DIPs

Let's see what happens when a DIP network is used to learn *pure images* versus *mixed images*. The graph in Fig. 3.c shows the MSE Reconstruction Loss of a *single* DIP network, as a function of time (training iterations), for each of the 3 images in Fig. 3.a: (i) the orange plot is the loss of a DIP trained to reconstruct the texture image X, (ii) the blue plot – a DIP trained to reconstruct the texture Y, and (iii) the green plot – a DIP trained to reconstruct their superimposed mixture (image transparency). Note the larger loss and longer convergence time of the mixed image, compared to the loss of its individual components. In fact, the loss of the mixed image is larger than the *sum* of the two individual losses. We attribute this behavior to the fact that the distribution of patches in the mixed image is more complex and diverse (larger entropy; smaller internal self-similarity) than in any of its individual components.

While these are pure textures, the same behavior holds also for mixtures of natural images. The *internal self-similarity* of patches inside a single natural image tends to be much stronger than the patch similarity *across different images* [41]. We repeated the above experiment for a large collection of *natural images*: We *randomly sampled* 100 pairs of images from the BSD100 dataset [27], and mixed each pair. For each image pair we trained a DIP to learn the mixed image and each of the individual images. The same behavior exhibited in the graph of Fig. 3.c repeated also in the case of natural images – interestingly, with an even larger gap between the loss of the mixed image and its individual components (see graph in the project website).

We performed a similar experiment for non-overlapping image segments. It was observed [5] that the *empirical entropy* of small regions composing an image segment is smaller than their *empirical cross-entropy* across different segments in the same image. We randomly sampled 100 pairs of images from the BSD100 dataset. For each pair we generated a new image, whose left side is the left side of one image, and whose right side is the right side of the second image. We trained a DIP to learn the mixed image and each of the individual components. The graph behavior of Fig.3.c repeated also in this case (see project website).

We further observe that when *multiple* DIPs train to *jointly* reconstruct a single input image, they tend to "split" the image patches among themselves. Namely, *similar* small patches inside the image tend to all be generated by a *single* DIP network. In other words, each DIP captures different components of the internal statistics of the image. We explain this behavior by the fact that a single DIP network is *fully convolutional*, hence its filter weights are shared across the entire spatial extent of the image. This promotes self-similarity of patches in the output of each DIP.

The simplicity of the patch distribution in the output of a single DIP is further supported by the denoising experiments reported in [38]. When a DIP was trained to reconstruct a *noisy* image (high patch diversity/entropy), it was shown to generate along the way an *intermediate clean* version of the image, before overfitting the noise. The clean image has higher internal patch similarity (smaller patch diversity/entropy), hence is simpler for the DIP to reconstruct.

Building on these observations, we propose to decompose an image into its layers by *combining multiple (two or more) DIPs*, which we call "Double-DIP". Figs. 3.a,b show that when training 2 DIP networks to jointly recover the mixed texture transparency image (as the sum of their outputs), each DIP outputs a coherent layer on its own.

2.2. Unified Multi-Task Decomposition Architecture

What is a good image decomposition? There are infinitely many possible decompositions of an image into layers. However, we suggest that a *meaningful* decomposition satisfies the following criteria: (i) the recovered layers, when recombined, should yield the input image. (ii) Each of the layers should be as "simple" as possible, namely, it should have a strong internal self-similarity of "image elements". (iii) The recovered layers should be as independent of each other (uncorrelated) as possible.

These criteria form the basis of our general-purpose Double-DIP architecture, illustrated in Fig. 2. The first criterion is enforced via a *"Reconstruction Loss"*, which measures the error between the constructed image and the input image (see Fig. 2). The second criterion is obtained by employing *multiple DIPs* (one per layer). The third criterion is enforced by an *"Exclusion Loss"* between the outputs of the different DIPs (minimizing their correlation).

Each DIP network (DIP_i) reconstructs a different layer y_i of the input image I. The input to each DIP_i is randomly sampled uniform noise, z_i. The DIP outputs, y_i=$DIP_i(z_i)$, are mixed using a weight mask m, to form a reconstructed image $\hat{I} = m \cdot y_1 + (1-m) \cdot y_2$, which should be as close as possible to the input image I.

In some tasks the weight mask m is simple and known, in other cases it needs to be learned (using an additional DIP). The learned mask m may be uniform or spatially varying, continuous or binary. These constraints on m are task-dependant, and are enforced using a task-specific *"Regularization Loss"*. The optimization loss is therefore:

$$Loss = Loss_{Reconst} + \alpha \cdot Loss_{Excl} + \beta \cdot Loss_{Reg} \quad (1)$$

where $Loss_{Reconst} = \|I - \hat{I}\|$, and $Loss_{Excl}$ (the Exclusion loss) minimizes the correlation between the *gradients* of y_1 and y_2 (as defined in [40]). $Loss_{Reg}$ is a task-specific mask regularization (e.g., in the segmentation task the mask m has to be as close as possible to a binary image, while in the dehazing task the t-map is continuous and smooth). We further apply guided filtering [22] on the learned mask m to obtain a refined mask.

Inherent Layer Ambiguities: Separating a superposition of 2 pure uncorrelated textures is relatively simple (see Fig. 3.a). There are no real ambiguities other than a constant global color ambiguity $c : I = (y_1 + c) + (y_2 - c)$. Similarly, pure non-overlapping textures are relatively easy to segment. However, when a single layer contains *multiple independent regions*, as in Fig. 3.b, the separation becomes ambiguous (note the switched textures in the recovered output layers of Fig. 3.b). Unfortunately, such ambiguities exist in almost any natural indoor/outdoor image.

To overcome this problem, initial "hints" are often required to guide the Double-DIP. These hints are provided *automatically* in the form of *very crude* image saliency [20]. Namely, in the first few iterations, DIP_1 is encouraged to train more on the salient image regions, whereas DIP_2 is guided to train more on the non-salient image regions. This guidance is relaxed after a few iterations.

When more than one image is available, this ambiguity is often resolved on its own, *without* requiring any initial hints. For example, in *video transparency*, the superposition of 2 video layers changes from frame to frame, resulting in different mixtures. The *statistics* of each layer, however, remains the same throughout the video (despite its dynamics) [34, 33]. This means that *a single DIP* suffices to represent all the frames of *a single video layer*. Hence, Double-DIP can be used to separate video sequences into 2 dynamic layers, and can often do so with no initial hints.

Figure 4: **Foreground/Background Image Segmentation.** *(Please see many more results in the project website)*

Optimization: The architecture of the individual DIPs is similar to that used in [38]. As in the basic DIP, we found that adding extra non-constant noise perturbations to the input noise adds stability in the reconstruction. We gradually increase noise perturbations with iterations. We further enrich the training set by transforming the input image I and the corresponding random noise inputs of all the DIPs using 8 transformations (4 rotations by 90° combined with 2 mirror reflections - vertical and horizontal). Such an augmentation was also found to be useful in the unsupervised *internal learning* of [35]. The optimization process is done using ADAM optimizer [25], and takes a few minutes per image on Tesla V100 GPU. In the case of video, the runtime grows sub-linearly with the number of frames, since all frames are used to train the same DIP.

3. Segmentation

Fg/Bg segmentation can be viewed as decomposing an image I into a foreground layer y_1 and background layer y_2, combined by a binary mask $m(x)$ at every pixel x:

$$I(x) = m(x)y_1(x) + (1 - m(x))y_2(x) \quad (2)$$

This formulation naturally fits our framework, subject to y_1 and y_2 complying to natural image priors and each being

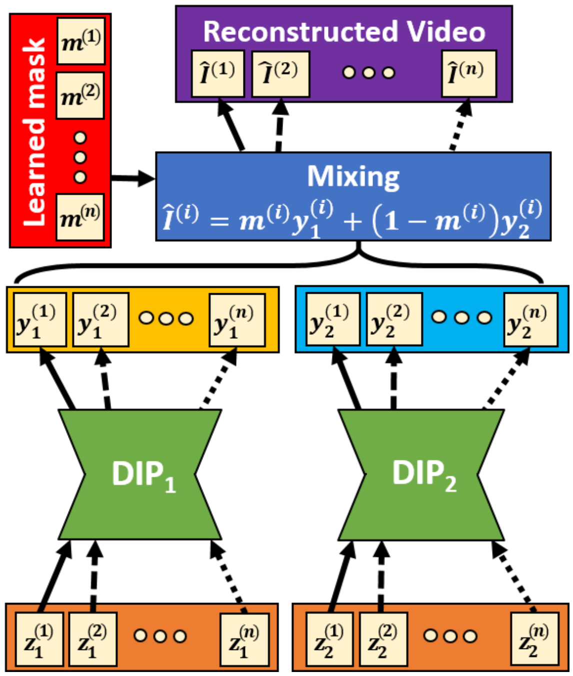

Figure 5: **Video Decomposition using Double-DIP.**

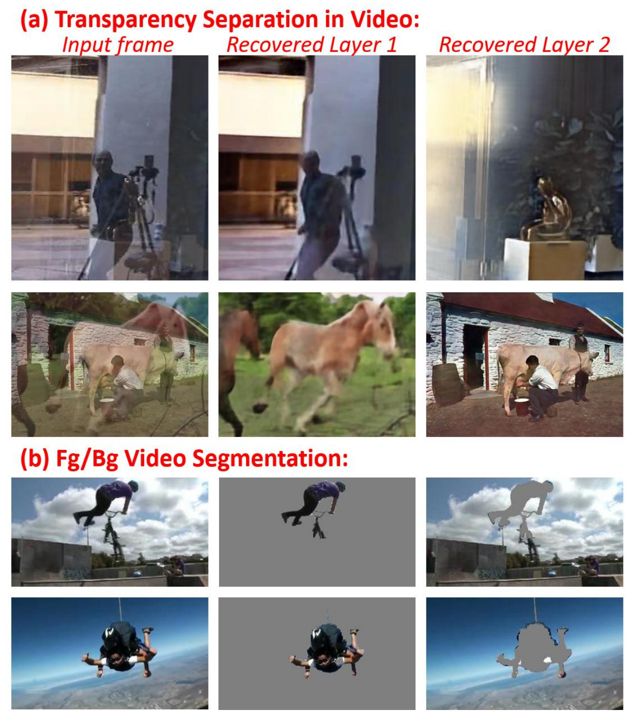

Figure 6: **Video Layer Separation via Double-DIP.** *Double-DIP exploits the fact that all frames of a single dynamic video layer share the same patches. This promotes: (a) video transparency separation, and (b) Fg/Bg video segmentation. (See full videos in the project website).*

'simpler' to generate than I. This requirement is verified by [5] that defines a 'good image segment' as one which can be easily composed using its own pieces, but is difficult to compose using pieces from other parts of the image.

The Zebra image in top row of Fig. 1 demonstrates the decomposition of Eq. 2. It is apparent that the layers y_1 and y_2, generated by DIP_1 and DIP_2, each complies with the definition of [5], thus allowing to obtain also a good segmentation mask m. Note that DIP_1 and DIP_2 automatically *filled-in* the 'missing' image parts in each output layer.

In order to encourage the learned segmentation mask $m(x)$ to be binary, we use the following regularization loss:

$$Loss_{Reg}(m) = (\sum_x |m(x) - 0.5|)^{-1} \tag{3}$$

While Double-DIP does not capture any semantics, it is able to obtain high quality segmentation based solely on unsupervised layer separation, as shown in Fig. 4. Please see many more results in the project website. Other approaches to segmentation, such as semantic-segmentation (eg., [21]) may outperform Double-DIP, but these are supervised and trained on many labeled examples.

Video segmentation: The same approach can be used for Fg/Bg video segmentation, by exploiting the fact that sequential video frames share internal patch statistics [34, 33]. Video segmentation is cast as 2-layer separation as follows:

$$I^{(i)}(x) = m^{(i)}(x)y_1^{(i)}(x) + (1 - m^{(i)}(x))y_2^{(i)}(x) \quad \forall i \tag{4}$$

where i is the frame number. Fig. 5 depicts how *a single DIP is shared by all frames* of a separated video layer: $y_1^{(1)}, ..., y_1^{(n)}$ are all generated by DIP_1, $y_2^{(1)}, ..., y_2^{(n)}$ are generated by DIP_2, $m^{(1)}, ..., m^{(n)}$ are all generated the mask DIP. The similarity across frames in each separated video-layer strengthens the tendency of a single DIP to generate a consistently segmented sequence. Fig. 6.b shows example frames from 2 different segmented videos (full videos can be found in the project website).

We *implicitly* enforce temporal consistency in the segmentation mask, by imposing temporal consistency on the *random noise* inputted to the mask DIP in successive frames:

$$z_m{}^{(i+1)}(x) = z_m{}^{(i)}(x) + \Delta z_m^{(i+1)}(x) \tag{5}$$

where $z_m^{(i)}$ is the noise input at frame i to the DIP that generates the mask. These noises change gradually from frame to frame by $\Delta z_m^{(i)}$ (which is a random uniform noise with variance significantly lower than that of $z_m^{(i)}$).

4. Transparent Layers Separation

In the case of image reflection, each pixel value in image $I(x)$ is a convex combination of a pixel from the transmission layer $y_1(x)$ and the corresponding pixel in the reflection layer $y_2(x)$. This again can be formulated as in Eq. 2, where $m(x)$ is the reflective mask. In most practical cases, it is safe to assume that $m(x) \equiv m$ is a uniform mask (with

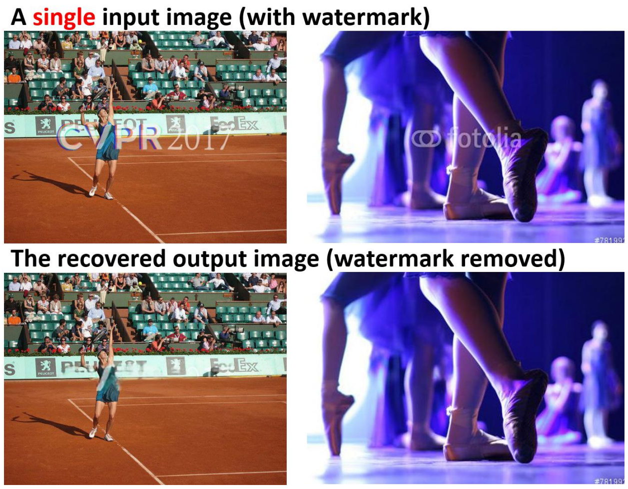

Figure 7: **Watermark removal from a single image.** *Ambiguity is mostly resolved by a rough bounding-box around the watermark. In the tennis image (provided by [14] as a tough image) part of the "V" of "CVPR" remains.*

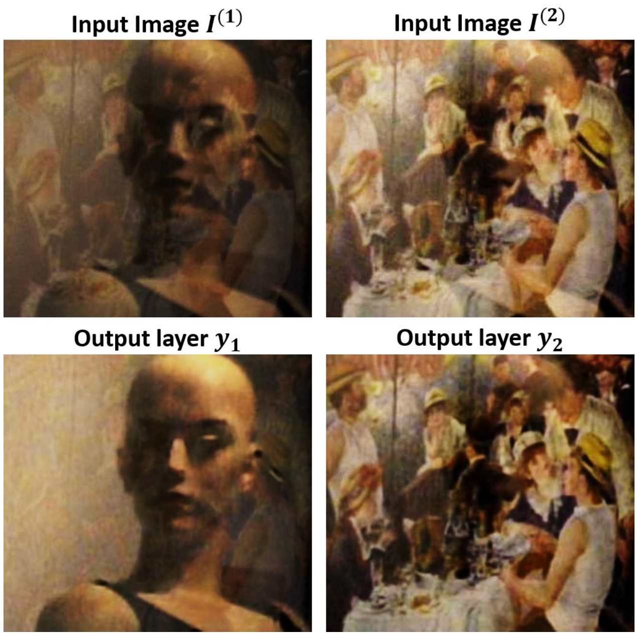

Figure 8: **Layer ambiguity is resolved when two different mixtures of the same layers are available.**

an unknown constant $0 < m < 1$). Double-DIP can be used to decompose an image I to its transparent layers. Each layer is again constructed by a separate DIP. The constant m is calculated by a third DIP. The Exclusion loss encourages minimal correlation between the recovered layers.

Fig. 3.a shows a successful separation in a *simple case*, where each transparent layer has a relatively uniform patch distribution. This however does not hold in general. Because each pixel in I is a mixture of 2 values, the inherent layer ambiguity (Sec. 2.2) in a single transparent image is much greater than in the binary segmentation case.

Ambiguity can be resolved using *external training*, as in [40]. However, since Double-DIP is unsupervised, we resolve this ambiguity when 2 different mixtures of the same layers are available. This gives rise to coupled equations:

$$\begin{cases} I^{(1)}(x) = m^{(1)} y_1(x) + (1 - m^{(1)}) y_2(x) \\ I^{(2)}(x) = m^{(2)} y_1(x) + (1 - m^{(2)}) y_2(x) \end{cases} \tag{6}$$

Since the layers y_1, y_2 are shared by both mixtures, one Double-DIP suffices to generate these layers using $I^{(1)}, I^{(2)}$ simultaneously. The different coefficients $m^{(1)}, m^{(2)}$ are generated by the same DIP using 2 random noises, $z_m^{(1)}, z_m^{(2)}$. See such an example in Fig. 8 (real transparent images).

Video Transparency Separation: The case of a static reflection and dynamic transmission can be solved in a similar way. This case can be formulated as a set of equations:

$$I^{(i)}(x) = m^{(i)} y_1^{(i)}(x) + (1 - m^{(i)}) y_2(x) \tag{7}$$

where i is the frame number, and y_2 is the static reflection (hence has no frame index i, but could have varying intensity over time, captured by $m^{(i)}$). Applying Double-DIP to a separate transparent video layers is done similarly to video segmentation (Sec. 3). We employ one DIP for each video layer, y_1, y_2 and one more DIP to generate $m^{(i)}$ (but with a modified noise input per each frame, as in the video segmentation). Fig. 6.a shows examples of video separation. For full videos see the project website.

Watermark removal: Watermarks are widely-used for copyright protection of photos and videos. Dekel *et al.* [14] presented a watermark removal algorithm, based on recurrence of the same watermark in many different images. Double-DIP is able to remove watermarks shared by very few images, often only one.

We model watermarks as a special case of image reflection, where layers y_1 and y_2 are the clean image and the watermark, respectively. This time, however, the mask is not a constant m. The inherent transparent layer ambiguity is resolved by one of two practical ways: (i) when only one watermarked image is available, the user provides a crude hint (bounding box) around the location of the watermark; (ii) given a few images which share the same watermark (2-3 typically suffice), the ambiguity is resolved on its own.

When a single image and a bounding box are provided, the learned mask $m(x)$ is constrained to be zero outside the bounding box. This hint suffices for Double-DIP to perform reasonably well on this task. See examples in Fig. 7. In fact, the Tennis image in Fig. 7 was provided by [14] as an example image immune to their watermark-removal method.

When multiple images contain the same watermark are available, no bounding-box is needed. E.g., if 3 images share a watermark, we use 3 Double-DIPs, which share DIP_2 to output the common watermark layer y_2. Independent layers $y_1^{(i)}, i{=}1,2,3$, provide the 3 clean images. The

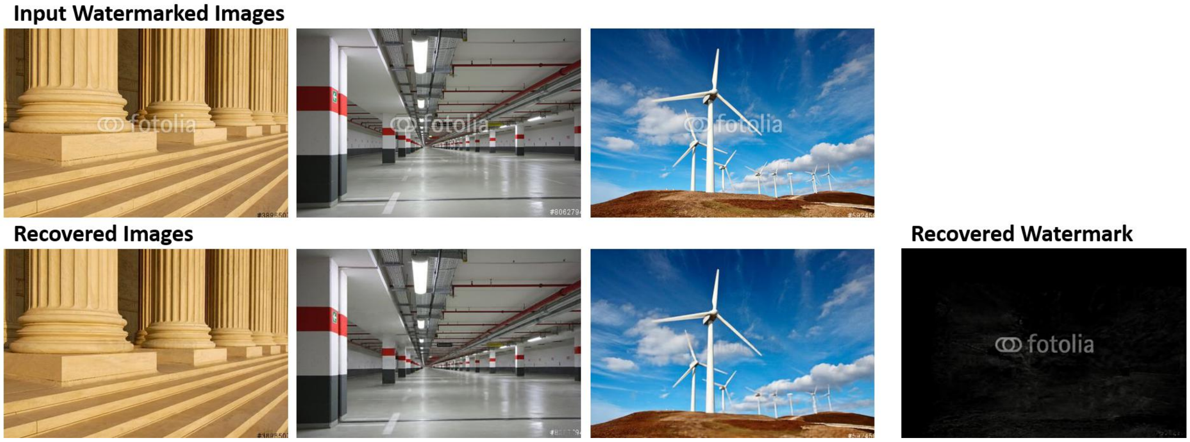

Figure 9: **Multi-image Watermark removal.** *Since the 3 images share the same watermark, the layer ambiguity is resolved.*

opacity mask m, also common to the 3 images, is generated by another shared DIP. Example is shown in Fig. 9.

5. Image Dehazing

Images of outdoor scenes are often degraded by a scattering medium (e.g., haze, fog, underwater scattering). The degradation in such images grows with scene depth. Typically, a hazy image $I(x)$ is modeled [23]:

$$I(x) = t(x)J(x) + (1 - t(x))A(x) \tag{8}$$

where $A(x)$ is the Airlight map (A-map), $J(x)$ is the *haze-free* image, and $t(x)$ is the *transmission* (t-map), which exponentially decays with scene depth. The goal of image dehazing is to recover from a hazy image $I(x)$ its underlying *haze-free* image $J(x)$ (i.e., the image that would have been captured on a clear day with good visibility conditions).

We treat the dehazing problem as a layer separation problem, where one layer is the haze-free image ($y_1(x)$=$J(x)$), the second layer is the A-map ($y_2(x)$=$A(x)$), and the mixing mask is the t-map ($m(x)$=$t(x)$).

Handling non-uniform airlight: Most single-image *blind* dehazing methods (e.g. [23, 28, 18, 6, 8]) assume a *uniform* airlight color $\mathcal{A}$ for the entire image (i.e., $A(x) \equiv \mathcal{A}$). This is true also for deep network based dehazing methods [10, 30], which train on *synthesized* datasets of hazy/non-hazy image pairs. The uniform airlight assumption, however, is only an approximation. It tends to break, e.g. in outdoor images captured at dawn or dusk, when the sun is positioned close to the horizon. The airlight color is affected by the non-isotropic scattering of sun rays by haze particles, which causes the airlight color to vary across the image. When the uniform airlight assumption holds, estimating a single uniform airlight color $\mathcal{A}$ from the hazy I is relatively easy (e.g., using the dark channel prior of [23], or the patch-based prior of [6]), and the challenging part remains the t-map estimation. However, when the uniform

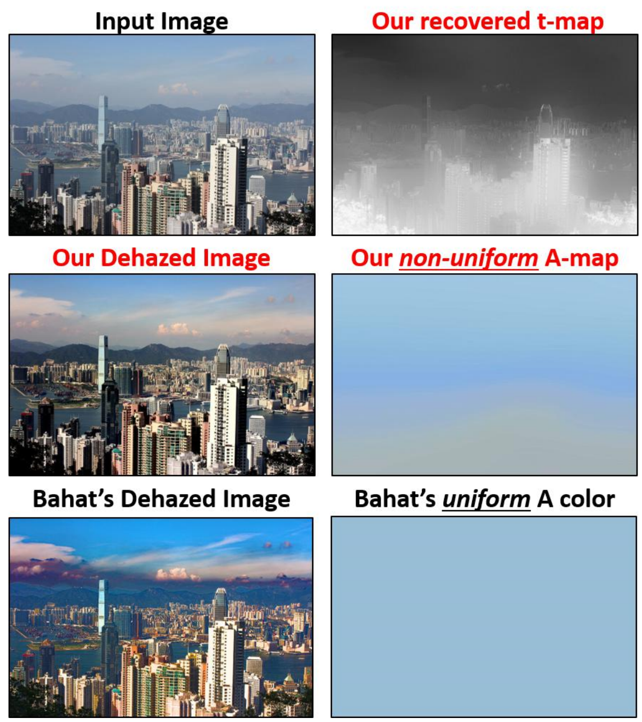

Figure 10: **Uniform vs. Non-Uniform Airlight recovery.** *The uniform airlight assumption is often violated (e.g., in dusk and dawn). Double-DIP allows to recover a non-uniform airlight map, yielding higher-quality dehazing.*

airlight assumption breaks, this produces dehazing results with distorted colors. Estimating a *varying* airlight, on the other hand, is a very challenging and ill-posed problem. Our Double-DIP framework allows simultaneous estimation of a *varying* airlight-map and varying t-map, by treating the A-map as another layer, and the t-map as a mask. This results in higher-quality image dehazing. The effect of estimating a *uniform* vs. *varying* airlight is exemplified in Fig. 10.

In dehazing, $Loss_{Reg}$ forces the mask $t(x)$ to be smooth

	He [23]	Meng [28]	Fattal [18]	Cai [10]	Ancuti [3]	Berman [8]	Ren [30]	Bahat [6]	Ours
PSNR	16.586	17.444	15.640	16.208	16.855	16.610	19.071	18.640	18.815

Table 1: **Comparison of dehazing methods on O-Haze Dataset.**

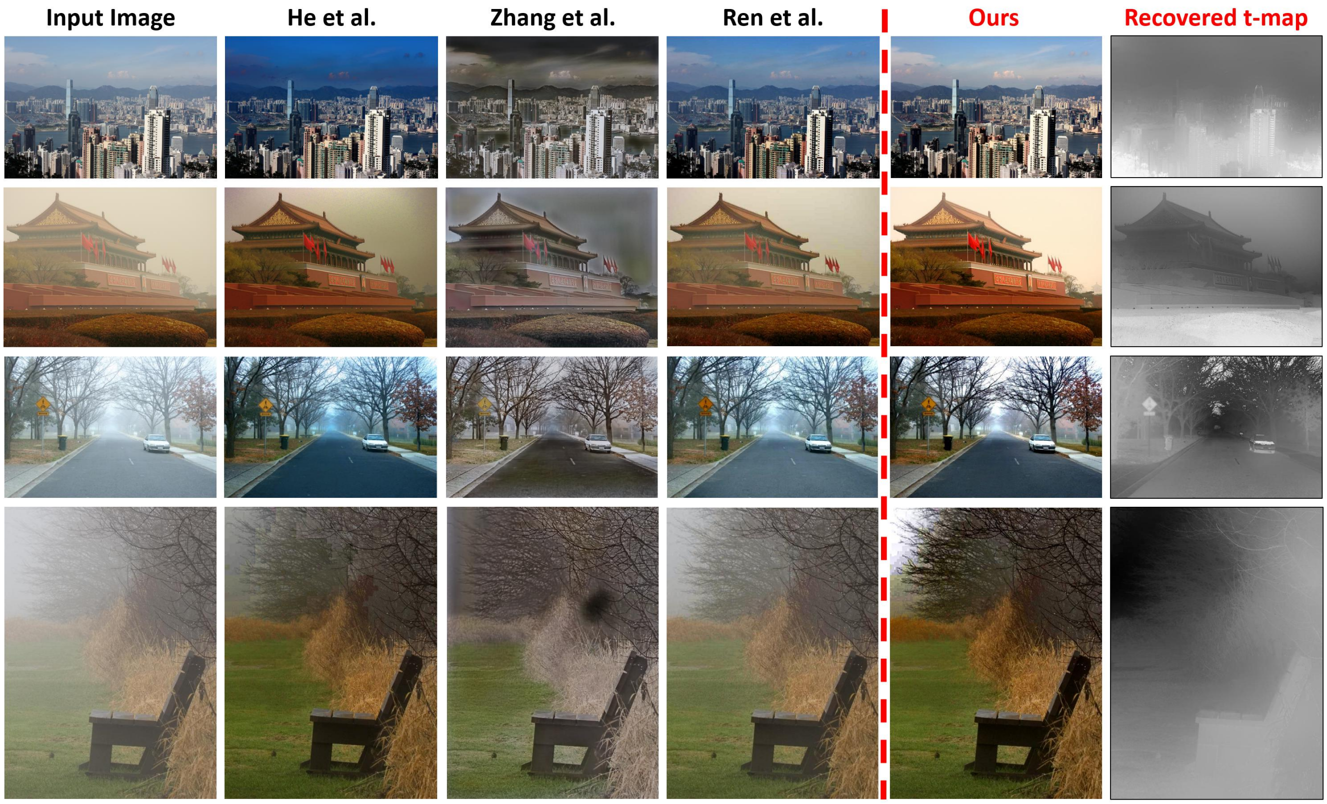

Figure 11: **Comparing Double-DIP's dehazing to *specialized* dehazing methods.** *(many more results in the project page)*

(by minimizing the norm of its Laplacian). The internal self-similarity of patches in a *hazy image* $I(x)$ is weaker than in its underlying *haze-free* image $J(x)$ [6]. This drives the first DIP to converge to a haze-free image. The A-map, however, is not a typical natural image. While it satisfies the strong internal self-similarity requirement, it tends to be much smoother than a natural image, and should not deviate much from a global airlight color. Hence, we apply an extra regularization loss on the airlight layer: $\|A(x) - \mathcal{A}\|_2$, where $\mathcal{A}$ is a single initial airlight color estimated from the hazy image I using one of the standard methods (we used the method of [6]). Although the deviations from the initial airlight $\mathcal{A}$ are subtle, they are quite crucial to the quality of the recovered haze-free image (see Fig. 10).

We evaluated our framework on the O-HAZE dataset [4] and compared it to unsupervised and self-supervised dehazing algorithms. Results are presented in Table 1. Numerically, on this dataset, we ranked second of all dehazing methods. However, visually, on images outside this dataset, our results seem to surpass all dehazing methods (see the project website). We further wanted to compare to the winning methods of NTIRE'2018 Dehazing Challenge [2], but only one of them had code available [39]. Our experiments show that while these methods obtained state-of-the-art results on the tiny test-set of the challenge (5 test images only!), they seem to severely overfit the challenge training-set. In fact, they perform very poorly on any hazy image outside this dataset (see Fig. 11 and the project website). A visual comparison to many more methods and on many more images is found in the project website.

6. CONCLUSION

"Double-DIP" is a unified framework for unsupervised layer decomposition, applicable for a wide variety of tasks. It needs no training examples other than the input image/video. Although general-purpose, in some tasks (e.g., dehazing) it achieves results comparable or even better than leading methods in the field. We believe that augmenting Double-DIP with semantic/perceptual cues, may lead to advancements also in *semantic* segmentation and in other high-level tasks. This is part of our future work.

References

[1] S. Alpert, M. Galun, R. Basri, and A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2007. 2

[2] C. Ancuti, C. O. Ancuti, and R. Timofte. Ntire 2018 challenge on image dehazing: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 891–901, 2018. 8

[3] C. Ancuti, C. O. Ancuti, C. D. Vleeschouwer, and A. C. Bovik. Night-time dehazing by fusion. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2256–2260, Sept 2016. 8

[4] C. O. Ancuti, C. Ancuti, R. Timofte, and C. D. Vleeschouwer. O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In *IEEE Conference on Computer Vision and Pattern Recognition, NTIRE Workshop*, NTIRE CVPR'18, 2018. 8

[5] S. Bagon, O. Boiman, and M. Irani. What is a good image segment? a unified approach to segment extraction. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, volume 5305 of *LNCS*, pages 30–44. Springer, 2008. 2, 3, 5

[6] Y. Bahat and M. Irani. Blind dehazing using internal patch recurrence. In *ICCP*, 2016. 2, 7, 8

[7] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *SIGGRAPH*, 2009. 2

[8] D. Berman, T. Treibitz, and S. Avidan. Non-local image dehazing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 7, 8

[9] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, volume 2, pages 60–65, 2005. 2

[10] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *CoRR*, abs/1601.07661, 2016. 7, 8

[11] T. S. Cho, S. Avidan, and W. T. Freeman. The patch transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. 2

[12] T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, New York, NY, USA, 2006. 2

[13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 2

[14] T. Dekel, M. Rubinstein, C. Liu, and W. T. Freeman. On the effectiveness of visible watermarks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6

[15] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, volume 2, pages 1033–1038, 1999. 2

[16] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. 2

[17] A. Faktor and M. Irani. Co-segmentation by composition. In *ICCV*, 2013. 2

[18] R. Fattal. Dehazing using color-lines. In *ACM Transaction on Graphics*, New York, NY, USA, 2014. ACM. 2, 7, 8

[19] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009. 2

[20] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(10):1915–1926, Oct. 2012. 4

[21] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5

[22] K. He, J. Sun, and X. Tang. Guided image filtering. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 1–14, Berlin, Heidelberg, 2010. Springer-Verlag. 4

[23] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2341–2353, Dec. 2011. 2, 7, 8

[24] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foregroundbackground segmentation using codebook model. *Real-Time Imaging*, 11(3):172 – 185, 2005. Special Issue on Video Object Processing. 2

[25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 4

[26] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(9):1647–1654, 2007. 2

[27] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 3

[28] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan. Efficient image dehazing with boundary constraint and contextual regularization. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 617–624, Washington, DC, USA, 2013. IEEE Computer Society. 7, 8

[29] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *ICCV*, 2009. 2

[30] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *European Conference on Computer Vision*, 2016. 2, 7, 8

[31] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, Aug. 2004. 2

[32] B. Sarel and M. Irani. Separating transparent layers through layer information exchange. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, pages 328–341, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 2

[33] O. Shahar, A. Faktor, and M. Irani. Space-time super-resolution from a single video. In *CVPR*, 2011. 4, 5

[34] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *IEEE Conference on Computer Vision and Pattern Recognition 2007 (CVPR'07)*, June 2007. 4, 5

[35] A. Shocher, N. Cohen, and M. Irani. "zero-shot" super-resolution using deep internal learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

[36] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *CVPR*, 2008. 2

[37] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *CVPR*, page 1246, 2000. 2

[38] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3, 4

[39] H. Zhang, V. Sindagi, and V. M. Patel. Multi-scale single image dehazing using perceptual pyramid deep network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 8

[40] X. Zhang, R. Ng, and Q. Chen. Single image reflection separation with perceptual losses. *CoRR*, abs/1806.05376, 2018. 4, 6

[41] M. Zontak and M. Irani. Internal statistics of a single natural image. *CVPR 2011*, pages 977–984, 2011. 2, 3