Adversarial Training for Solving Inverse Problems in Image Processing

Zhengxia Zou, Tianyang Shi, Zhenwei Shi, *Member, IEEE*, and Jieping Ye, *Fellow, IEEE*

***Abstract*— Inverse problems are a group of important mathematical problems that aim at estimating source data x and operation parameters z from inadequate observations y. In the image processing field, most recent deep learning-based methods simply deal with such problems under a pixel-wise regression framework (from y to x) while ignoring the physics behind. In this paper, we re-examine these problems under a different viewpoint and propose a novel framework for solving certain types of inverse problems in image processing. Instead of predicting x directly from y, we train a deep neural network to estimate the degradation parameters z under an adversarial training paradigm. We show that if the degradation behind satisfies some certain assumptions, the solution to the problem can be improved by introducing additional adversarial constraints to the parameter space and the training may not even require pair-wise supervision. In our experiment, we apply our method to a variety of real-world problems, including image denoising, image deraining, image shadow removal, non-uniform illumination correction, and underdetermined blind source separation of images or speech signals. The results on multiple tasks demonstrate the effectiveness of our method.**

***Index Terms*— Inverse problem, deep learning, generative adversarial networks, bidirectional mapping.**

I. Introduction

An INVERSE problem in science and mathematics is the process of inferring the causal factors from a set of inadequate measurements that produced them [1], [2]. Inverse problems are some of the most important mathematical problems in the image processing field because they can tell us about what we cannot directly observe, e.g., recovering clean images from noisy ones.

In image processing, many problems can be defined as the "inversion" of a forward operation. Consider a forward operation function $y = \mathcal{F}(x; z)$ where y represents an observed image (e.g., a noisy image), x represents the source image to be recovered (e.g. a clean image), and z represents the degradation parameters (e.g. the noise). In its inverse form, we aim to estimate the causal factors (x, z) that produce the observation y.

An inverse problem with both unknown source image x and unknown parameters z is essentially an ill-posed problem and is difficult to solve [3]. The observations are often noisy and contain incomplete information about the source images or operation parameters due to the physical limitations of the measurement devices. Consequently, solutions to inverse problems are not unique. In this case, since the observed images do not always contain enough information for the inversion, additional prior knowledge needs to be exploited. The priors can be derived from physical constraints, data distribution hypothesis [4], the underlying structure of the desired solution set [5], or by user interactions [6]. Some traditional methods for solving inverse problems include the least square approaches [2], total variation approaches [7], [8], the Bayesian approaches [1], and etc.

In recent years, deep learning has greatly promoted the research progress of the inverse problems in image processing [3], [9]–[12]. For some typical problems like image super-resolution, deblurring, and denoising, a recent popular solution is to deal with them under a pixel-wise prediction framework with pair-wise training supervision [3], [9], [10], [12]. In these methods, a learnable model, typically a Convolutional Neural Network (CNN), learns a mapping from the observed image y to the source image x by minimizing the "distance" between the predicted output and the corresponding ground truth. In this process, although the source image can be successful recovered from the degraded input, the degradation physics behind is simply ignored.

In this paper, we consider solving the inverse problem differently: instead of learning a mapping directly from the observation y to the source x, we learn to predict the operation parameters z behind so that the degradation can be revealed with a clear physical significance. We show that the recent success of adversarial training can help improve the previous solutions by integrating additional physical constraints into the training process. We also show that if an inverse problem to

Manuscript received May 10, 2020; revised September 23, 2020, November 28, 2020, and January 11, 2021; accepted January 12, 2021. Date of publication January 27, 2021; date of current version February 2, 2021. The work was supported in part by the National Key R&D Program of China under Grant 2019YFC1510905; in part by the National Natural Science Foundation of China under Grant 61671037; and in part by the Beijing Natural Science Foundation under Grant 4192034. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chun-Shien Lu. *(Corresponding author: Zhengxia Zou.)*

Zhengxia Zou is with the Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: zzhengxi@umich.edu).

Tianyang Shi is with Fuxi AI Lab, Netease, Hangzhou 310052, China.

Zhenwei Shi is with the Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China, also with the Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China, and also with the State Key Laboratory of Virtual Reality Technology and Systems, School of Astronautics, Beihang University, Beijing 100191, China.

Jieping Ye is with the Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI 48109 USA, and also with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA.

Digital Object Identifier 10.1109/TIP.2021.3053398

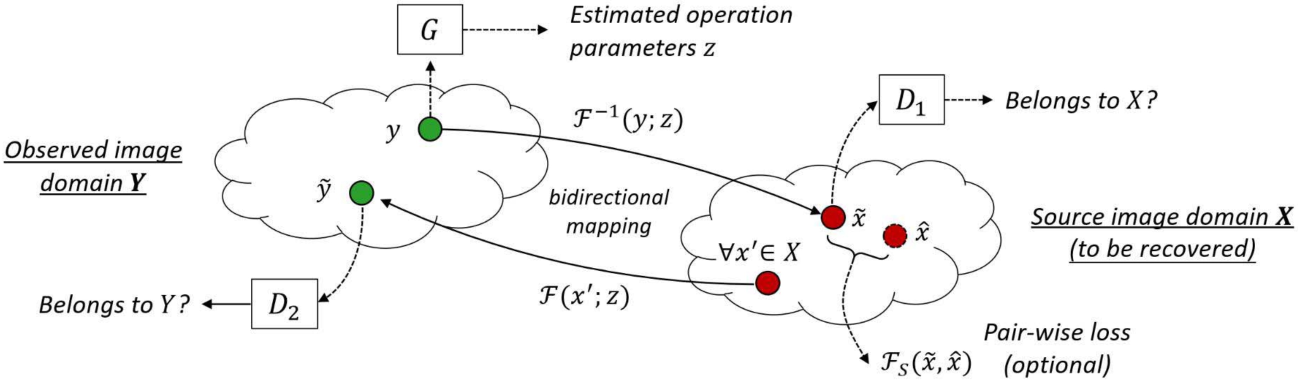

G
Estimated operation parameters z
D_1
Belongs to X?
$\mathcal{F}^{-1}(y;z)$
Observed image domain $\boldsymbol{Y}$
y
$\tilde{y}$
bidirectional mapping
$\tilde{x}$
$\hat{x}$
$\forall x' \in X$
Source image domain $\boldsymbol{X}$ (to be recovered)
Belongs to Y?
D_2
$\mathcal{F}(x';z)$
Pair-wise loss (optional)
$\mathcal{F}_S(\tilde{x},\hat{x})$

Fig. 1. An overview of our method. Our method consists of a generator G and two discriminators D_1 and D_2. The G aims to predict the operation parameters z given an observation image y. After recovering the source image by computing the inversion $\mathcal{F}^{-1}(y, z)$, in addition to discriminating whether the recovered image $\tilde{x}$ belongs to the source domain X, we also impose the estimated parameter z to any other source images x' and force the degraded images $\tilde{y}$ belongs to the observation domain Y.

be solved satisfies an “independent and universal operation” assumption, the problem can be even solved in a weakly supervised manner, i.e., without requiring any pair-wise supervision during the training process.

The key to our method is a group of adversarial constraints named “bidirectional mapping constraints” - if an observed image y can be inversely mapped to its source x by using the parameter z, then, when we impose the same z on any randomly selected image in the source domain, we also expect the newly generated image to be projected to the observation domain. Fig. 1 shows an overview of the proposed method. In this paper, we assume the operation $\mathcal{F}$ has both explicit forward mapping and inverse mapping functions but with unknown parameters z. For example, in image denoising, we assume that the noise is additive but the values are unknown. The bidirectional mapping provides essential self-guidance through the adversarial training and serves as auxiliary constraints to the solutions. Our method can be applied to a variety of real-world inverse problems in image processing, including image deraining, underdetermined blind image separation, image denoising, and non-uniform illumination correction.

II. Related Works

A. Generative Adversarial Networks (GANs)

GANs [13] have received great attention in the machine learning and computer vision field in recent years. A typical GAN consists of two neural networks: a generator G and a discriminator D, where the former learns a mapping from a latent space to a particular data distribution of interest, while the latter aims to discriminate between instances from the true data distribution and those generated. The key to GANs' success is the idea of an adversarial training framework under which the generator G and the discriminator D will contest with each other in a minimax two-player game and forces the generated data to be indistinguishable from real ones. GAN has achieved impressive results in a variety of tasks, including image generation [14], [15], image style transfer [16], [17], and domain adaptation [18], [19].

B. GANs for Inverse Problems in Image Processing

More recently, GANs have also been applied to solving inverse problems such as image deblurring [12], [20], image denoising [21], and image super-resolution [9], [22], [23]. In these problems, in addition to the standard supervised loss (e.g., l_1 or l_2 loss) used in the training process, adversarial losses are introduced to provide constraints on the model space so that to improve the visual quality of the recovered images. These methods typically require pair-wise image samples for training (i.e., a degraded input image and its ground truth) and may even require the ground truth of the operation parameters z to be known [20]. Although some recent improvements such as the cycle consistent loss [16] breaks such limitations, the degradation factors behind are still rarely considered [9], [12], [20]. Different from those previous GAN based approaches that directly learn a mapping between the source and the observation domain, our method learns the operation parameters behind and explains the degradation physics more clearly.

It is worth noting that the “blind super-resolution” [24] and the “depth-guided internal degradation learning” [25] are two recent methods that are similar to ours. These two methods both reveal the degradation (implicitly or explicitly) by using adversarial training. Besides, the latter one [25] also introduces bidirectional mapping and applies adversarial losses in both directions. The difference between these methods and ours is that the former ones are specifically designed for the image super-resolution tasks while our method aims to solve more general inverse problems in image processing. In addition, we also discuss under what conditions our method can be applied to an inverse problem and under what conditions the problem can be solved without pair-wise supervision. This part of the work has not been studied in previous papers.

The rest of this paper is organized as follows. In Section III, we give a detailed description of the proposed method, including the motivations, objective functions, and some discussions on the scope of applications. In Section IV, we describe the implementation details of our method. In Section V, we give a detailed description of our tasks, datasets, evaluation metrics, and experimental results. The conclusions are drawn in Section VI. Our code is available at https://github.com/jiupinjia/GANs-for-Inverse-Problems.

III. Methodology

The goal of our method is to train a generative network $G(\cdot)$ that maps any observed image y to its degradation parameters: $z = G(y)$. We assume the degradation function $\mathcal{F}$ has an explicit mathematical form but with unknown parameters. With the estimated z, the source image can be easily recovered based on its inversion: $\tilde{x} = \mathcal{F}^{-1}(y; z)$.

A. Objective Functions

The proposed method involves training multiple neural networks: a generator G and two discriminators D_1 and D_2, as shown in Fig. 1. Suppose X represents the source image domain and Y represents the observed image domain, given an observed image $y \in Y$, instead of learning a mapping directly from Y to X as suggested by conventional GAN-based methods [16], [17], we learn to predict its degradation parameters and then recover the source image. Clearly, if there are pair-wise training images available, this can be simply achieved by training the G to enforce the recovered image $\tilde{x} = \mathcal{F}^{-1}(y, G(y))$ and their ground truth $\hat{x}$ as close as possible, e.g. with pixel-wise l_2 loss:

$$\mathcal{L}_S(G) = \mathbb{E}_{y\sim p_Y(y)}\{\|\mathcal{F}^{-1}(y, G(y)) - \hat{x}\|_2^2\}, \tag{1}$$

In addition to the above pair-wise supervised loss, we introduce the following adversarial constraint:

$$\forall y \in Y, \quad \tilde{x} = \mathcal{F}^{-1}(y, G(y)) \in X, \tag{2}$$

and train the discriminator D_1 to distinguish between the recovered image $\tilde{x}$ and a real one x. Meanwhile, we train the G to make $\tilde{x}$ indistinguishable. The objective function can be written as follows:

$$\begin{aligned}\mathcal{L}_{\mathcal{F}^{-1}}(G, D_1) = {}& \mathbb{E}_{x\sim p_X(x)}\{\log D_1(x)\} \\ &+ \mathbb{E}_{y\sim p_Y(y)}\{\log(1 - D_1(\mathcal{F}^{-1}(y, G(y))))\},\end{aligned} \tag{3}$$

where $p_X(x)$ and $p_Y(y)$ are the probabilistic distributions of the source images and the observed ones.

We further assume the function $\mathcal{F}$ has an explicit form for both of its forward and inverse mapping with unknown parameters z. Under such assumptions, we also train the G to meet the following adversarial condition:

$$\forall x' \in X, \quad \tilde{y} = \mathcal{F}(x'; z) \in Y, \tag{4}$$

which means, if we impose the estimated parameter z to any images $x' \in X$, we also enforce the newly generated images to be mapped from X to Y. We refer to the adversarial conditions (2) and (4) as "**bidirectional mapping**" constraints. To meet the above conditions, we introduce another discriminator D_2 to distinguish between the transformed images and the real ones, and then train the G to make the $\tilde{y}$ indistinguishable. The objective function on this condition can be written as follows:

$$\begin{aligned}\mathcal{L}_{\mathcal{F}}(G, D_2) = {}& \mathbb{E}_{y\sim p_Y(y)}\{\log D_2(y)\} \\ &+ \mathbb{E}_{x'\sim p_X(x); y\sim p_Y(y)}\{\log(1 - D_2(\mathcal{F}(x', G(y))))\}.\end{aligned} \tag{5}$$

It is worth noting that the input y and $\tilde{y}$ here do not have to be close to each other. In models like Cycle-GAN [16], constraints like l-2 or l-1 distances are used to ensure $\tilde{y}$ is close to y. However, in our method, there are no such constraints. We only require the back-transformed images $\tilde{y}$ belong to the domain Y and do not make any constraints on their concrete values.

The final objective function is defined as follows:

$$\mathcal{L}(G, D_1, D_2) = \mathcal{L}_{\mathcal{F}^{-1}}(G, D_1) + \mathcal{L}_{\mathcal{F}}(G, D_2) + \beta\mathcal{L}_S(G), \tag{6}$$

where $\beta \geq 0$ controls the balance between the pixel-wise loss and adversarial losses. When β is set to zero, no pair-wise loss is provided. The training of our model can be considered as a minimax optimization process where G is trained to minimize its objective while D_1 and D_2 are trained to maximize it:

$$G^\star, D_1^\star, D_2^\star = \arg\min_G \max_{D_1, D_2} \mathcal{L}(G, D_1, D_2). \tag{7}$$

B. Discussion

In this subsection, we will discuss under which conditions our method can be used to solve an inverse problem and under which conditions the problem can be solved even without pair-wise supervision.

1) When can we apply the "bidirectional mapping" constraints? In the following, we show that if the inverse problem to be solved satisfies certain conditions, the bidirectional mapping can be introduced to solving an inverse problem.

Definition 1 (Independent and Universal Operation): Suppose X represents the source domain, Y represents the observation domain ($X \cap Y = \varnothing$). $x \in X$ and $y \in Y$ are their training samples. $z \in Z$ is random variable as operation parameters. We define $y = \mathcal{F}(x; z)$ as an "independent and universal operation", if it meets the following assumptions:

$$z \text{ is independent of } x. \tag{8a}$$

$$\forall y \in Y, \quad \exists z : \mathcal{F}^{-1}(y; z) \in X. \tag{8b}$$

$$\forall x \in X, \quad \forall z \in Z : \mathcal{F}(x; z) \in Y. \tag{8c}$$

In the above conditions, (8a) defines the variables z and x should be independent of each other. (8b) and (8c) define that the estimated parameter z not only can map y back to X but also can be applied to any $x \in X$ and maps them to Y. Many real-world tasks satisfy the above assumptions. Here we give two examples:

- *Image denoising*: the noise z is in most cases independent of the image x, and any estimated noise component $z = G(y) \in Z$ can be also imposed on any clean image $x \in X$ so that the clean image can be mapped to a noisy one.
- *Non-uniform illumination correction*: the lighting condition z is usually independent of the objects x in the image. We, therefore, can cast the estimated light parameters to any object and produce uneven lighting effects.

We will discuss the above examples in detail in our experiments. There are also some other tasks that do not satisfy the above assumptions, for example:

- *Haze removal*: the haze density z (the airlight and attenuation) is related to the depth of an image x. In other words, z and x are not independent of each other and thus the assumptions do not hold.

- *JPEG recompression/artifact removal*: the compression history is related to the image content and thus cannot be directly applied to other uncompressed images.
- *Image super-resolution*: the degradation (e.g., z-times down-sampling) is independent with the input images (8a satisfied). However, given a low-resolution image y, $\mathcal{F}^{-1}$ does not have an explicit formation and thus x cannot be explicitly recovered from y and z ((8b) and (8c) are not satisfied).

Notice that if the problem to be solved does not meet the statements (8a)-(8c) (like the negative cases we listed above), our method cannot be applied to such problems. This is a limitation of our method.

2) Is the pair-wise supervision necessary for the solution? In real-world applications, sometimes it is expensive and even impossible to obtain the training image pairs. Clearly, in this case, pixel-wise supervision is not available during the training and the problem becomes even more difficult to solve. So, a natural question arises: can an inverse problem be solved free from pair-wise training supervision? Previous research on image translation [16] shows that single discrimination constrain (2) on the recovered data cannot guarantee the correctness of the solution. This is because if the generator has a large enough capacity, the input image y can be mapped to any random permutation in the domain X if there are no one-to-one mapping instructions.

In our experiments, we found that by using the bidirectional mapping and adversarial training objectives, the network G can be well trained even there is no pair-wise ground truth available. Fig. 2 gives a simple explanation of why we need bidirectional mapping constraints and how they work when there is no pair-wise training data available. In this example, suppose we have a noisy image y and we aim at estimating the noise layer and then recovering the clean image by substituting the noise from the input. The noise layer z is predicted by the generator G. Notice that even if the recovered image $\tilde{x}$ was successfully mapped to the clean image domain Y ((2) holds), we still cannot guarantee the correctness of the prediction if pixel-wise supervision is not available. This is because without using the constraint (4), the noisy image can be mapped to any location in the clean image domain. In Fig. 2 (a), we give an example of such a condition where a noisy digit "0" is wrongly mapped to a clean digit "4". In Fig. 2 (b), we show that to ensure the correctness of the estimated causal factors, both of the conditions (2) and (4) should hold at the same time.

C. Toy Example

Here we give another toy example to further explain how our method works without paired training data. Suppose we have two groups of 2D data points randomly sampled from two distributions: $x \sim p_X(x)$ and $y \sim p_Y(y)$, where $x \in \mathbb{R}^{2\times 1}$, and $y \in \mathbb{R}^{2\times 1}$. We define the relationship between the two groups by an Affine transformation:

$$\mathcal{F}(x; z) = Mx + b = y, \tag{9}$$

where $M \in \mathbb{R}^{2\times 2}$ and $b \in \mathbb{R}^{2\times 1}$ are a set of pre-defined transformation parameters. In this example, we aim to predict the transformation between the two sets of points and we assume that we do not know the correspondence between the data points.

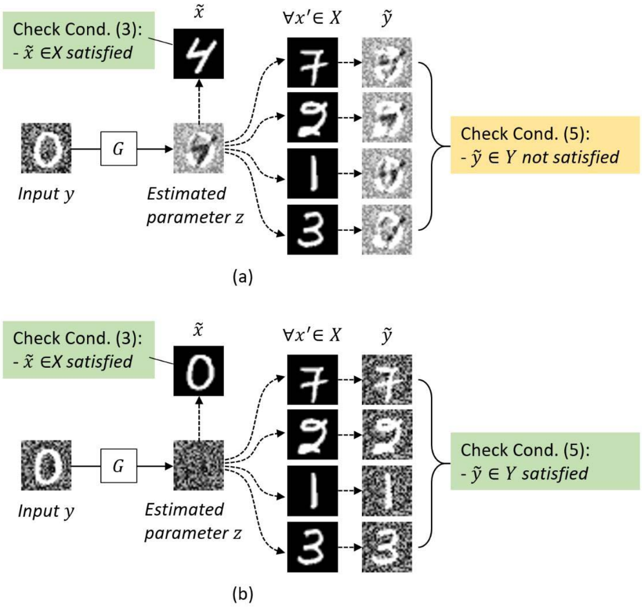


Fig. 2. A simple illustration of why we need both of the conditions (2) and (4) hold when there is no pair-wise training data available. Suppose we have a noisy image y and we aim at estimating its noise layer with a generator G. Consider the following two scenarios. (a) If we only have the condition (2) holds, a noisy digit "0" can be likely to be mapped to any clean digit (say, "4" in this example) if the G has large enough capacity. Clearly, in this case, we cannot guarantee the correctness of the recovered image $\tilde{x}$. (b) To ensure correctness of the estimated noise component z, both conditions (2) and (4) should hold at the same time.

Given a data point y, on one hand, we aim to inversely map y to the distribution X by using the estimated parameters $(\tilde{M}, \tilde{b})$:

$$\tilde{M}^{-1}(y - \tilde{b}) \rightarrow X, \tag{10}$$

on the other hand, we aim to generate new data points by imposing $\tilde{M}$ and $\tilde{b}$ to any other data points in $x' \in X$, and enforce the new data belongs to Y:

$$\tilde{M}x' + \tilde{b} \rightarrow Y. \tag{11}$$

Fig. 3 shows the mapping results of our method during the training process. We test on three groups of 2D data: 1) Gaussian data, 2) banana data, and 3) circle data. In this example, we simply build a 3-layer multi-layer perceptron (MLP) as our generator and build another two 2-layer MLPs as our discriminators. Although we do not apply any additional element-wise loss on the Affine parameters, our method still learns good transformation consistency among different data points.

IV. Implementation Details

In this section, we will describe the implementation details on the network architectures and the training details.

A. Configurations of the Networks

We build three CNNs as our generator G and discriminators D_1 and D_2. Our generator consists of a 12-layer convolutional

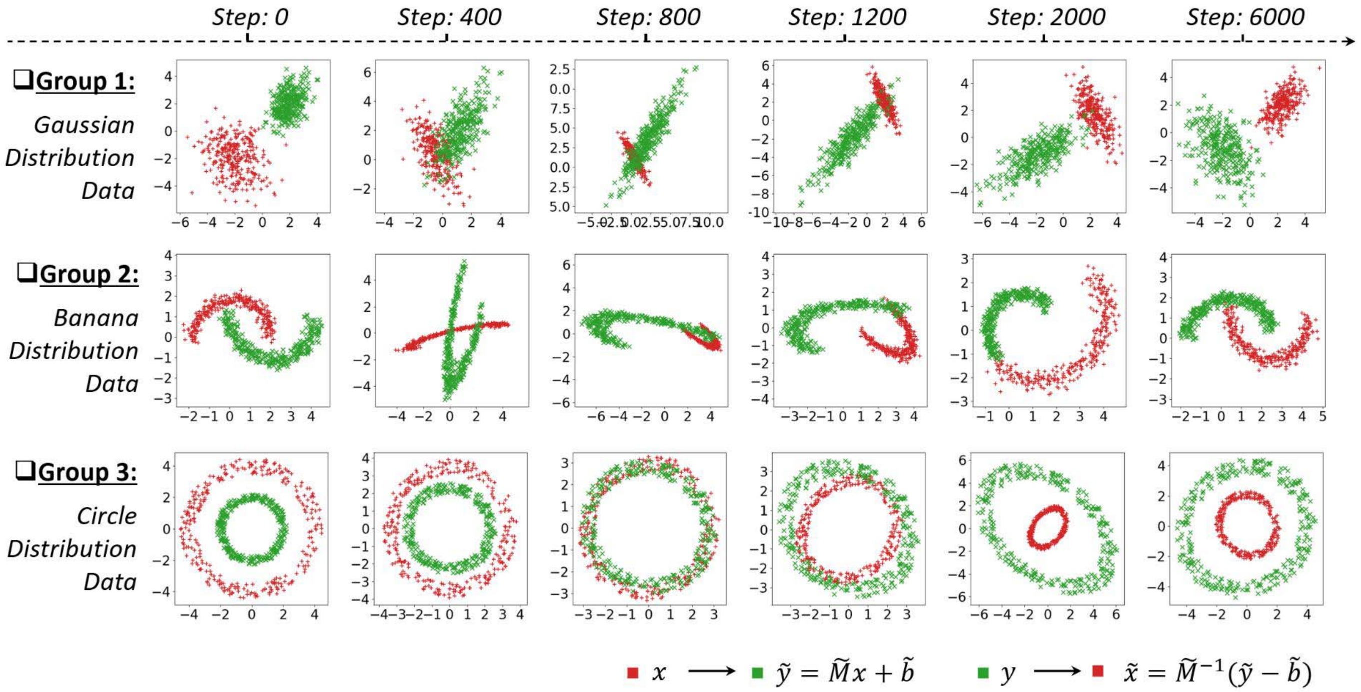


Fig. 3. An illustration of the training process of our method on three groups of 2D data: 1) Gaussian data, 2) banana data, and 3) circle data. Our method aims to learn an Affine transformation for each data point so that to map the green cluster to the red and inversely map the red one to the green.

TABLE I
DETAILED CONFIGURATIONS OF OUR GENERATOR (G) AND DISCRIMINATORS (D_1, D_2)

	Layer	Input	Stride	Filters
Generator	Conv(2)_ConvPool_1	image	2	4x4 / 64
	Conv(2)_ConvPool_2	Conv(2)_ConvPool_1	2	4x4 / 64
	Conv(2)_ConvPool_3	Conv(2)_ConvPool_2	2	4x4 / 64
	Conv(2)_ConvPool_4	Conv(2)_ConvPool_3	2	4x4 / 64
	Conv(2)_Up_1	Conv(2)_ConvPool_4	1/2	4x4 / 64
	Conv(2)_Up_2	Conv(2)_Up_1 + Conv(2)_ConvPool_3	1/2	4x4 / 64
	Conv(2)_Up_3	Conv(2)_Up_2 + Conv(2)_ConvPool_2	1/2	4x4 / 64
	Conv(2)_Up_4	Conv(2)_Up_3 + Conv(2)_ConvPool_1	1/2	4x4 / 64
	Conv(2)_9	Conv(2)_Up_4	1	4x4 / out_dims
Discriminator	ConvPool_1	image	2	4x4 / 32
	ConvPool_2	ConvPool_1	2	4x4 / 64
	ConvPool_3	ConvPool_2	2	4x4 / 128
	ConvPool_4	ConvPool_3	2	4x4 / 256
	FC_1	ConvPool_4	-	- / 1

encoder and a 12-layer convolutional decoder. We add skip connections between all channels at layer i and layer $n - i$, by following the general configuration of the "U-Net" [26] for building both of the high-level semantic features and low-level details. As suggested by Odena *et al.* [27], we use the bilinear interpolation to up-sample of the feature maps in our decoder instead of using the fractional-strided convolution (deconv) [28], which helps reduce the checkerboard artifacts. Our discriminators consist of 4 convolutional layers and one fully-connected layer. We use the Leaky ReLU [29] activation instead of the standard ReLU activation in our discriminators as suggested by Radford *et al.* [15]. For both our generator and our discriminators, we use strided convolution instead of pooling and use Batch-normalization after all convolution layers, except for the output layers. Table I shows the detailed configurations of the three networks. In the column "Filters", "$n \times n/m$" denotes (m) convolutional filters with the spatial size of ($n \times n$). In the column "Input", "Conv(2)_ConvPool" denotes two "stacked convolution layers" followed by a strided convolution layer. "Up" denotes an up-sampling layer with bilinear interpolation. "+" denotes the element-wise summation of two feature maps. "FC" denotes a fully-connected layer. Our generator G has 1.380M parameters. Our discriminator D_i ($i = \{1, 2\}$) has 3.459M parameters. Given an input image with a spatial size of 128×128 pixels, the computational cost for a single forward propagation in G and D_i

would be 3.619 Giga-FLOating Point operations (GFLOPs) and 2.832 GFLOPs, respectively.

Although the inference stage of our method involves both predicting z based on y and recovering x based on z, the amount of calculation using z to recover x is usually negligible. If we take denoising as an example, the latter one only needs to compute $x = y - z$. Given an image of 128×128 pixels, our method runs at 83.5 fps (frames per second) at the inference stage on an NVIDIA 1050Ti GPU. As a comparison, the CycleGAN or Pixel2pixel run at 62.6 fps at the same resolution (with the same backbone UNet-128).

B. Training Details

As suggested by Goodfellow *et al.* [13], instead of training G to directly minimize $\log(1 - D_i(\cdot))$, $i = 1, 2$, in practice, we train G to maximize $\log D(\cdot)$. This is because in early stage of learning, $\log(1 - D(\cdot))$ tends to saturate. This revision on objective provides much stronger gradients early in learning.

We use the Adam optimizer for training, with batch-size = 8. We use Xavier's initialization for all our networks. The learning rates are set to 10^{-5} for G and 10^{-6} for D_1 and D_2. To improve the training stability, when the discriminative loss is smaller than a predefined threshold, say, 0.5, we freeze the discriminators and update G several times until the loss becomes higher than the threshold. For the "MNIST+MNIST separation" experiment in the section V-C, we stop the training at 120,000 generator's iteration steps. For other experiments, we stop the training at 10,000 generator's iteration steps.

V. Experiments and Analysis

We apply our method to several image processing tasks to verify its effectiveness. We use the same set of network configurations shown in table I in all our experiments. In each task, the dimension of z is consistent with our input image, i.e., 256×256x3 pixels for image deraining and shadow removal; 64×64x3 for superimposed image separation and image denoising; and 128×128x3 for breaking image/audio CAPTCHAs and non-uniform illumination correction.

A. Image Deraining

The goal of image deraining is to recover the clean image from a degraded one captured in rainy days [11], [30], [30], [31]. When we take a picture through the rain, the image y we obtained can be viewed as a mixture of two layers [32]: a line-shape rain-streak layer z and a clean background image x:

$$\begin{aligned} y &= \mathcal{F}_{rain}(x; z) = x + z, \\ x &= \mathcal{F}_{rain}^{-1}(y; z) = y - z. \end{aligned} \tag{12}$$

In the above equations, we consider the background image x as the source image, and the rain-streak layer z as the degradation parameters. X represents the clean image domain and Y represents the rainy image domain.

We conduct experiments on two public deraining datasets named Rain100H [33] and Rain800 [11]. Rain100H is a synthetic dataset that consists of 1,900 rain images and corresponding ground truth. Rain800 consists of two subsets - a synthetic subset that consists of 800 image pairs (rain image + ground truth), and a real-world subset that consists of 52 rain images that are taken on rainy days (without ground truth).

In our deraining experiment, we set $\beta = 1000$. We compare with several recent image deraining methods, including "Gaussian Mixture Models (GMM-Deraining)" [31], "Adversarial Image Deraining (Adv-Deraining)" [11], "JOint Rain DEtection and Removal (JORDER)" [33], "Squeeze-and-Excitation Context Aggregation for single image deraining (SCAN)" [35], and "Density-aware Image De-raining with a Multistream Dense Network (DID-MDN)" [30]. Among these methods, the GMM-Deraining [31] formulates deraining as a layer decomposition problem by using layer sparsity priors. The JORDER [33], SCAN [35], and DID-MDN [30] formulate deraining as a pixel-wise regression problem by training a CNN between the predicted output and the clean background (ground truth). The Adv-Deraining [11] introduces adversarial losses to improve the visual quality of the deraining output. As the deraining can be essentially considered as an image-to-image translation task, we also compare with two well-known image translation methods, i.e., Pixel-to-Pixel [17] and Cycle-GAN [16].

Fig. 4 shows the comparison results of different methods on the Rain800 dataset [11]. We observe that the CycleGAN may introduce unexpected "color shift" and the Pixel-to-Pixel may suffer from undesired compression artifacts. Table II shows the evaluation results on the Rain100H [33] and the synthetic subset of Rain800 [11]. We use two classical evaluation metrics: the Peak Signal-to-Noise Ratio (PSNR), and the Structural Similarity (SSIM) index [34] to evaluate the deraining performance. These metrics are computed by comparing the deraining results and the ground truth references. We can see our method is among the best entries of all deraining methods in terms of both PSNR and SSIM.

We also evaluate our method on the real-world image subset of the Rain800 dataset [11]. As there is no ground truth reference for these images, we introduce two non-reference evaluation metrics, i.e., the Mode Score (MS) [36] and Fréchet Inception Distance (FID) [37], to evaluate the image quality of the deraining output. Note that these two metrics have been recently used in adversarial image synthesis to better evaluate the visual quality, and may not require pair-wise ground truth for the assessment. We randomly divide the deraining results and the clean training set into several groups and then compute the similarity between the images in every two groups. The mean score and the standard deviation are recorded. Since previous deraining literature did not report their MS and FID accuracy, we only compare our method with the two image-to-image translation methods: Pixel-to-Pixel [17] and the Cycle-GAN [16]. Table III shows the comparison results on the real-world rain images. We can see that our method achieves the best entries in both settings.

B. Image Shadow Removal

We test our method on a real-world inverse problem - "shadow removal", which aims at detecting the shadow region and recovering the image pixels under the shadow. Suppose y represents an input image with shadow regions, and x

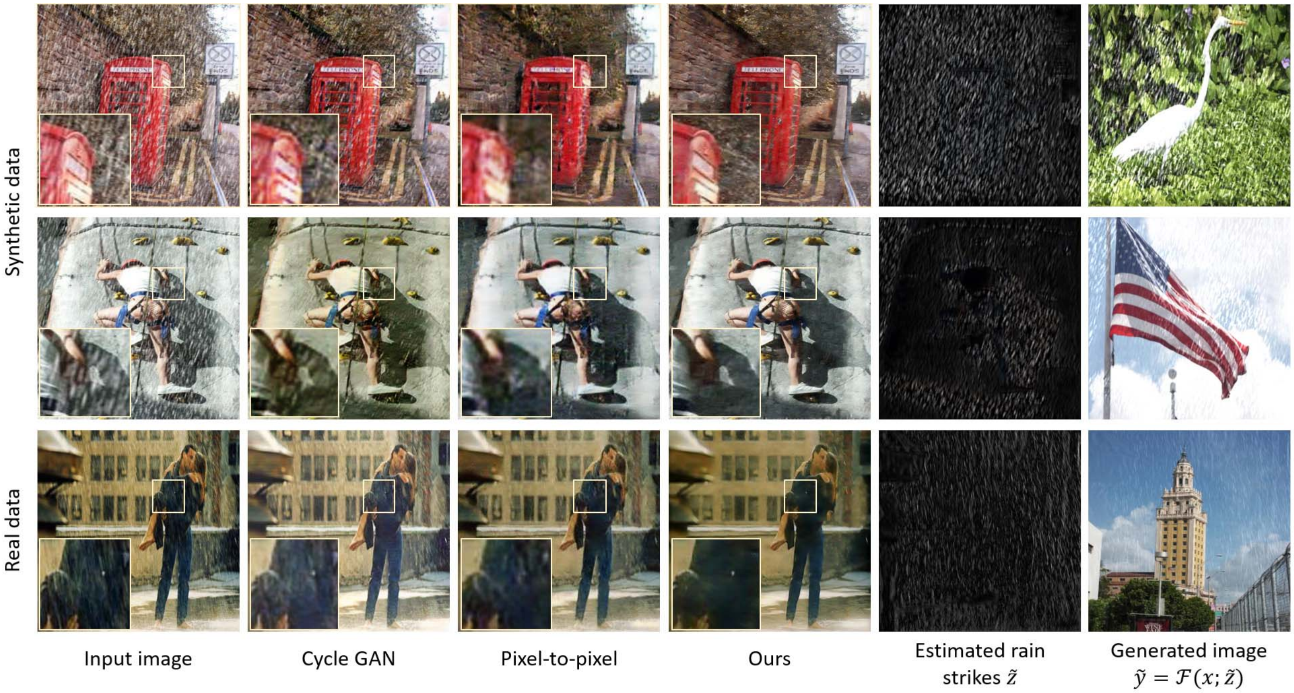

Synthetic data
Real data
Input image
Cycle GAN
Pixel-to-pixel
Ours
Estimated rain strikes $\tilde{z}$
Generated image $\tilde{y} = \mathcal{F}(x; \tilde{z})$

Fig. 4. Some deraining examples of different methods: CycleGAN [16], Pixel-to-Pixel [17], and ours. An advantage of our method is that the rain-streak map can be well-estimated. As a comparison, other methods simply ignore this part of the output.

TABLE II

RESULTS OF DIFFERENT METHODS FOR DERAINING ON TWO SYNTHETIC DERAINING DATASETS RAIN100H [33] AND RAIN800 (SYN) [11]. FOR BOTH PSNR AND SSIM [34], HIGHER SCORES INDICATE BETTER

	Rain100H [33]		Rain800 (Syn) [11]	
Method	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
GMM-Derain [31]	15.05	0.425	22.27	0.741
Adv-Derain [11]	—	—	22.73	0.813
JORDER [33]	22.15	0.674	22.24	0.776
JORDER-R [33]	23.72	0.749	22.29	0.792
SCAN [35]	23.56	0.746	23.45	0.811
DID-MDN [30]	25.00	0.754	—	—
Our method	**25.97**	**0.780**	**24.38**	**0.819**

TABLE III

RESULTS OF DIFFERENT METHODS ON A REAL-WORLD DERAINING DATASET RAIN800 (REAL) [11]. FOR MS [36], HIGHER SCORES INDICATE BETTER. FOR FID [37], LOWER SCORES INDICATE BETTER

	Rain800 (Real) [11]	
Method	MS ↑	FID ↓
Pixel-to-Pixel [17]	1.263 ± 0.064	0.118 ± 0.027
Cycle-GAN [16]	**1.289** ± 0.073	0.145 ± 0.036
Our method	1.283 ± 0.068	**0.111** ± 0.022

represents its shadow-free source image. z is an illumination component ($0 \leq z \leq 1$), where the pixels within shadow area have a low z value and those out of shadow area have a high z value (ideally $z = 1$ for non-shadow pixels). The operation functions can be written as follows:

$$\begin{aligned} y &= \mathcal{F}_{shadow}(x; z) = zx, \\ x &= \mathcal{F}^{-1}_{shadow}(y; z) = y/z. \end{aligned} \tag{13}$$

In this experiment, we test on two real-world shadow removal datasets - "Dataset with Image Shadow Triplets (ISTD)" [38] and "Shadow Removal Dataset (SRD)" [39]. The two datasets consist of 1,870 and 3,088 shadow and shadow-free image pairs respectively. All the images and ground truth are captured by using cameras in real-world environments. Since their ground truth images are available, we use pairwise the loss during the training and set $\beta = 1000$. We compare the proposed method with some recent shadow removal methods, including "DeshadowNet" [39], "Direction-aware Spatial Context features for shadow detection and removal (DSC)" [41], "STacked Conditional Generative Adversarial Networks for shadow detection and removal (ST-CGAN)" [38], "Attentive Recurrent Generative Adversarial Network (ARGAN)" [42], and "Deep Adversarial Decomposition (DeepDecomp)" [43]. Among these methods, ST-CGAN and ARGAN are two methods that also utilize adversarial training. DeepDecomp is a method previously proposed by our research group that focuses on superimposed image separation but are designed under a different framework compared to this paper. Table IV shows the quantitative results of different shadow removal methods. We follow the evaluation metric introduced by Guo *et al.* [40] (lower is better). Note that we did not list the results of ST-CGAN and ARGAN on SRD because the authors did not report their accuracy on this dataset and the code is not publicly available. Fig. 5 shows a group of shadow removal results of our method and several other methods on the ISTD dataset [38].

C. Underdetermined BSS

Blind Source Separation (BSS) [44]–[48] aims at separating source signals/images from a set of mixed ones. The research

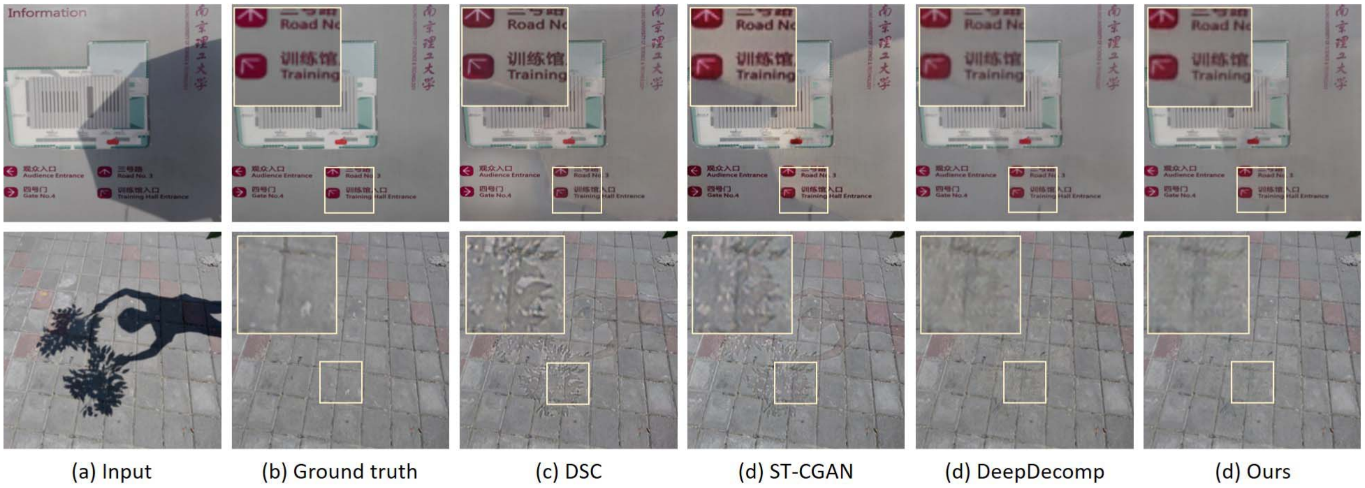

(a) Input (b) Ground truth (c) DSC (d) ST-CGAN (d) DeepDecomp (d) Ours

Fig. 5. A visual comparison of different shadow removal methods on the ISTD dataset [38]: DSC [41], ST-CGAN [38], DeepDecomp [43], and ours.

TABLE IV

QUANTITATIVE SCORES OF DIFFERENT SHADOW REMOVAL METHODS ON ISTD DATASET [38] AND SRD DATASET [39]. WE FOLLOW THE EVALUATION METRIC INTRODUCED BY GUO *et al.* [40] (LOWER IS BETTER)

	Shadow Removal Dataset	
Method	ISTD [39]	SRD [44]
DeshadowNet [44]	7.830	6.640
DSC [40]	7.100	6.210
ST-CGAN [39]	7.470	-
ARGAN [46]	6.680	-
DeepDecomp [41]	**6.566**	**5.823**
Our method	6.610	5.950

of this topic can be traced back to the 1990s [49], where the Independent Component Analysis (ICA) [44] was a representative of the methods at the time. Previous BSS methods typically requires multiple mixed inputs [44], [46]–[48] or additional user interactions [6]. Since multiple mixed inputs or user interactions are not always available in practice, the underdetermined BSS, e.g., to separate mixed signals from a single measurement, began to attract much research attention [4], [50]–[53]. In this experiment, we choose two applications, 1) separating superimposed images, 2) cracking image/audio captchas, and investigate whether these problems can be well-solved w/ or w/o the help the pair-wise supervision.

1) Separating Superimposed Images: The separation of superimposed images [52], [53] has long been a challenging task - the difficulty lies not only in the absence of the mixture function but also in the lack of constraints on the output space.

Suppose we have a superimposed image y with two unknown image layers x_1 and x_2, if we consider the operation parameters as one of the two image layers, i.e., $x = x_1$ and $z = x_2$, the operation functions can be written as follows:

$$\begin{aligned} y &= \mathcal{F}_{ubss}(x; z) = \alpha_1\ x_1 + \alpha_2\ x_2, \\ x_1 &= \mathcal{F}_{ubss}^{-1}(y; z) = (y - \alpha_2 x_2)/\alpha_1. \end{aligned} \tag{14}$$

where $0 < \alpha_i < 1$, $i = 1, 2$ are the linear mixing coefficients.

We train our model on the MNIST [54] and CIFAR-10 [55] datasets by randomly mixing two images from these datasets. To simplify the problem, in our experiment, we set $\alpha_1 = \alpha_2 = 0.5$. In this way, the coefficients are eliminated and the causal factors turn out to be the two image layers: $x = x_1$, $z = x_2$. Note that the problem is still very changing even we ignore the linear coefficients. This is because when we recover both x_1 and x_2 from a single observation y and there could be still an infinite number of possible solutions.

Notice that in the "MNIST+MNIST separation" task, when we train our model with the help of pair-wise loss, we compute the l_2 losses according to both ground truth components $\hat{x}_1$ and $\hat{x}_2$. The loss is defined as follows:

$$\begin{aligned} \mathcal{L}_S(G) = \mathbb{E}_{y \sim p_Y(y)}\{\min(&\|\mathcal{F}^{-1}(y, G(y)) - \hat{x}_1\|_2^2, \\ &\|\mathcal{F}^{-1}(y, G(y)) - \hat{x}_2\|_2^2)\}. \end{aligned} \tag{15}$$

The above expression is slightly different from the original l_2 loss we introduced in our main paper (Eq. 1). This is because if we compute the loss based on a single ground truth reference, it will cause ambiguity during the training as the two ground truth components are interchangeable.

We compare our method with other two popular methods for single mixed image separation: the Double-DIP [4] and Levin's method [6]. Notice that both of the two methods are unsupervised methods. The Double-DIP is an unsupervised deep learning based framework that exploits the image priors introducing an "internal self-similarity" assumption [4], where they assume that the distribution of small patches within each separate layer should be "simpler" (more uniform) than in the original mixed one. Levin's method is designed based on image statistics and requires additional user-interactions. Fig. 6 shows the separating results of different methods. We show that our method nicely separates the image from the mixed ones even without the help of pair-wise training supervision.

We also compare the separation results of our method on taking or without taking consideration of the bidirectional mapping during the training, as shown in Fig. 7. We show that without the bidirectional mapping, the generator may produce

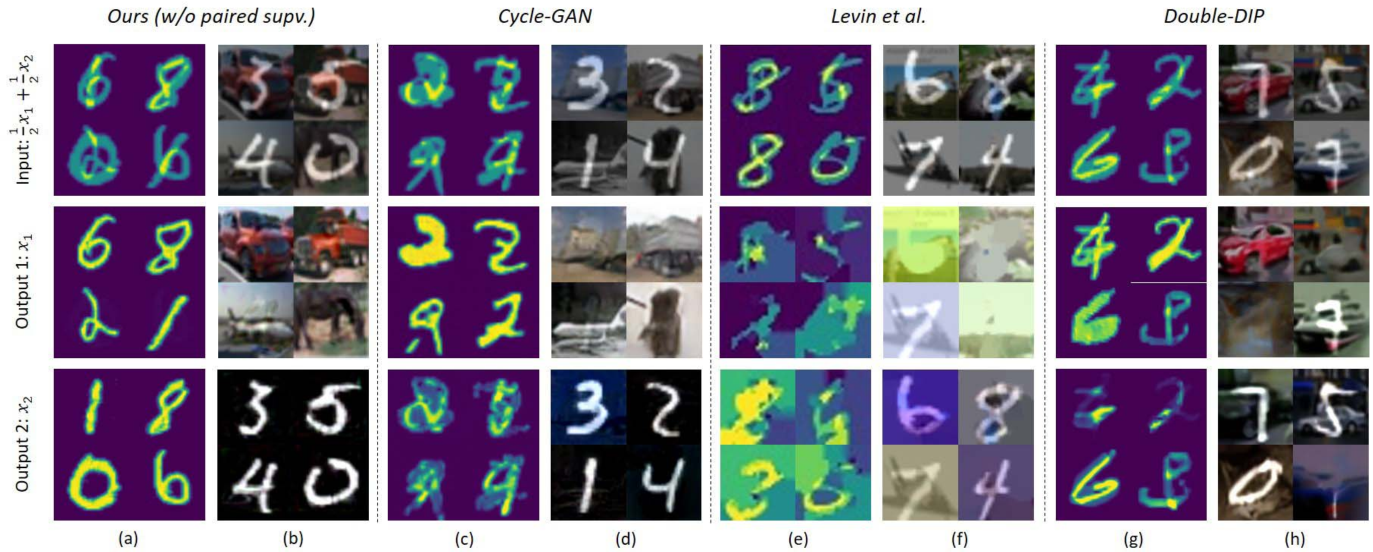


Fig. 6. Some examples of the superimposed image separation on MNIST [54] and CIFAR-10 [55] datasets. The columns are the separation results of different methods: (a-b) our method, (c-d) Cycle-GAN [16], (e-f) Levin *et al.* [6], and (g-h) Double-DIP [4]. For a fair comparison, we do not use pair-wise training supervision in our method. The images are fair random selected, not cherry-picked.

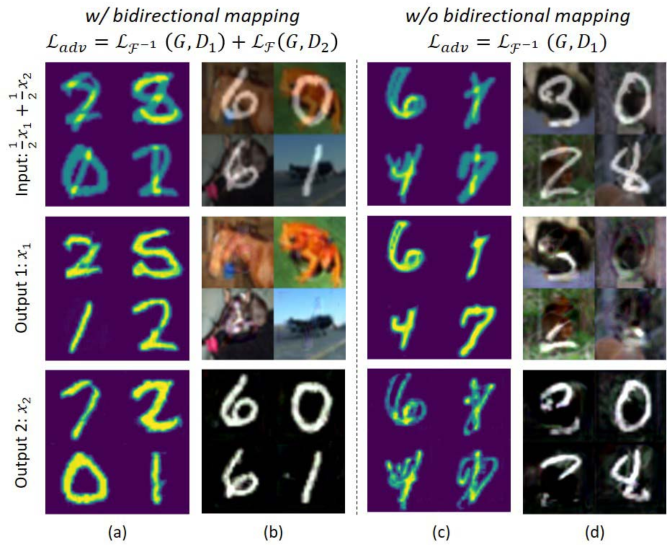

w/ bidirectional mapping
$\mathcal{L}_{adv} = \mathcal{L}_{\mathcal{F}^{-1}}(G, D_1) + \mathcal{L}_{\mathcal{F}}(G, D_2)$
w/o bidirectional mapping
$\mathcal{L}_{adv} = \mathcal{L}_{\mathcal{F}^{-1}}(G, D_1)$
Input: $\frac{1}{2}x_1 + \frac{1}{2}x_2$
Output 1: x_1
Output 2: x_2
(a)
(b)
(c)
(d)

Fig. 7. Superimposed image separation results in two groups of images: MNIST+MNIST, MNIST+CIFAR10. The columns (a)-(b) and (c)-(d) show the results of our method w/ and w/o the help of bidirectional mapping constraints. We observe the broken and nonsensical characters in the second output component of (c)-(d). No pair-wise supervision is provided during the training.

some broken and nonsensical image components in its outputs (see the 3rd row of the column (c-d)). This is because, in this case, we do not make additional constraints on these output components.

Table V shows the comparison results of different methods. To make a fair comparison, we train two variants of our model, one with pair-wise supervision ("w/ Supvs"), and another one without pair-wise supervision ("w/o Supvs"). For CycleGAN and Pix2pix, we set their input image size to 64×64 pixels and 128 × 128 pixels (the minimum supported input size) while keeping other default settings unchanged. Our method achieves the best results in both experimental settings ("w/ Supvs" and "w/o Supvs").

TABLE V

Comparison Results of Separating Superimposed Images: MNIST+MNIST, MNIST+CIFAR10. To Make a Fair Comparison, We Train Two Variants of Our Model, One With Pair-Wise Supervision ("w/ Supvs"), and Another One Without Pair-Wise Supervision ("w/o Supvs"). For PSNR (dB) and SSIM [34], Higher Scores Indicate Better

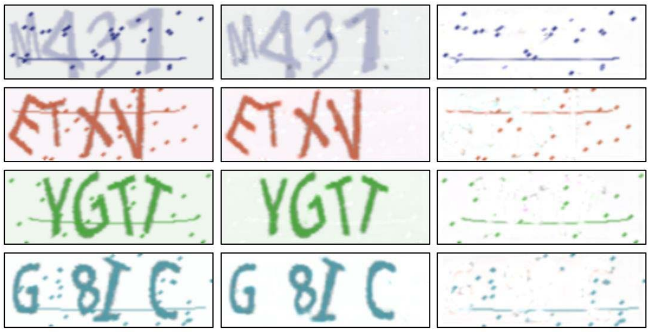
	MNIST+MNIST		MNIST+CIFAR	
Metric	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Levin *et al.* [6]	8.5940	0.1635	9.5090	0.2114
Double-DIP [4]	13.842	0.4206	12.368	0.2748
CycleGAN [16]	20.291	0.7712	20.460	0.7603
Ours (w/o Supvs)	**21.736**	**0.8539**	**24.039**	**0.8028**
Pix2pix [17]	18.551	0.7399	23.805	0.8731
Ours (w/ Supvs)	**22.729**	**0.8753**	**32.821**	**0.9461**

2) Cracking Image/Audio CAPTCHAs: Our method can be also used to crack CAPTCHAs. In this experiment, we first generate 3 million image CAPTCHAs as well as their labels by using the package "captcha 0.3" [56] and train a CAPTCHA solver [57] to recognize these characters. Then, we generate another two groups of image CAPTCHAs, one group of clear images, and another group with random stripes and spots. We train our generator using these images to separate their noise components and then record the recognition rate of the solver [57] before and after separation. Fig. 8 shows our image-noise separation results. The column "Captcha" in Table VI shows the recognition accuracy (including the "word-level" accuracy ACC_{word} and "character-level" accuracy ACC_{char}) under different experimental configurations. We observe a noticeable improvement of the recognition rate when we integrate the bidirectional mapping regardless of the

TABLE VI

CONTROLLED EXPERIMENTS ON THE INFLUENCE OF EACH COMPONENT OF OUR METHOD, INCLUDING THE ADVERSARIAL TRAINING ("ADV-TRAINING"), BIDIRECTIONAL MAPPING ("BIDIRECT-MAP"), AND PAIR-WISE SUPERVISED TRAINING ("PAIRED-SUPVS"). ACCURACY IS COMPUTED ON THREE DIFFERENT TASKS: CRACKING IMAGE CAPTCHAS, DENOISING (ON MNIST) AND NON-UNIFORM ILLUMINATION CORRECTION (ON CELEBA). FOR ALL EVALUATION CRITERIA, A HIGHER SCORE INDICATES BETTER

	Ablations			Captcha		Denoising	Illum-corr
Paired-Supvs	Adv-Training	Bidirect-Map	Adv-Objectives	ACC (word) ↑	ACC (char) ↑	PSNR ↑	SSIM ↑
✓	✓	✓	GAN [13]	91.4%	97.3%	35.29	0.949
✓	✓	✓	WGAN [59]	91.4%	97.3%	35.30	0.951
✓	✓	✓	LSGAN [60]	**91.6%**	**97.5%**	**35.37**	**0.960**
×	×	×	—	18.1%	49.9%	—	—
×	✓	×	GAN [13]	85.0%	95.5%	31.64	0.796
✓	✓	×	GAN [13]	**90.6%**	96.8%	**35.21**	**0.930**
×	✓	✓	GAN [13]	86.5%	95.6%	34.37	0.822
×	✓	✓	WGAN [59]	88.5%	96.9%	34.87	0.826
×	✓	✓	LSGAN [60]	89.2%	**97.1%**	34.94	0.831



(a) Input: y (b) Output 1: x (c) Output 2: z

Fig. 8. Some examples of undetermined BSS for image CAPTCHAs with our method (without pair-wise supervision).

presence of pair-wise training supervision. A detailed discussion will be given in our controlled experiments (section V-E).

We also test our method for speech data. In this experiment, we use "captcha 0.3" [56] to generate 8,000 audio fragments, where each of them is about 2.5 seconds long (8kHz) and consists of 4 random pieces of the human voice corrupted by both dial-up noise and Gaussian white noise. We compute the spectral "image" of each speech fragment by using a 256-point Short-Time Fourier Transform (STFT) and then train our method to separate human voices from background noise. The voice separation problem can thus be easily implemented under our image separation framework. When predicting the noise components, we retain its phase map and perform separation based on its amplitude map. After removing the noise amplitude from the input, we impose the phase spectrum on the refined amplitude map and perform inverse STFT to restore the clean speech signal. Fig. 8 shows an example of separation result. The separation is trained without using pairwise ground truth data. We can see from both spectra and signal waveform that our method can clearly separate the speech signal from noise interference.

D. Image Denoising and Illumination Correction

The image denoising and the correction of non-uniform illumination are both fundamental tasks in image processing. Our method can be applied to these tasks.

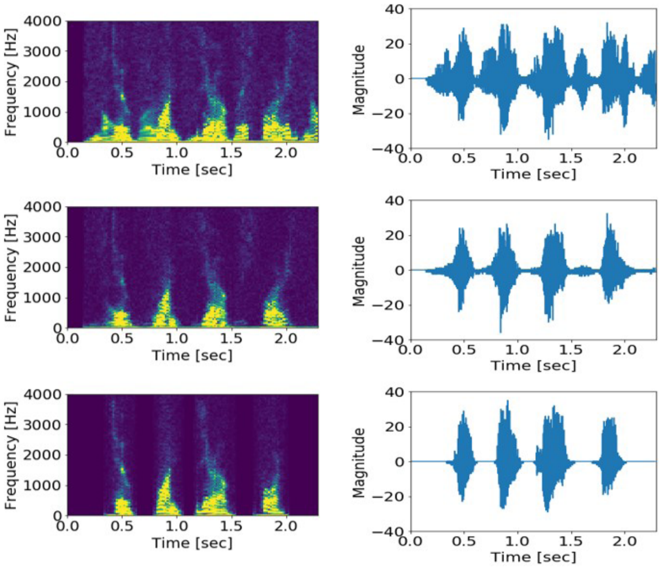


Fig. 9. An example of undetermined BSS for audio CAPTCHAs with our method (without pair-wise supervision). Each audio fragments is about 2.5s long (8kHz), consisting of a piece of the human voice and background noise. The first row shows the input noisy speech signal y. The second row shows the recovered clean speech signal x. The third row shows the ground truth.

1) Image Denoising: An image y with additive noises can be modeled as the mixture of a clean image x and a random noise component z, where z is independent of x. The degradation thus has the same form in Eq. (12). To verify our method on image denoising, we experiment on two datasets: MNIST [54] and VGG-Flowers [60]. Given a clean input image of the two datasets, we add strong Gaussian white noise to the image (PSNR < 0.01 dB for MNIST images, PSNR < 10 dB for VGG-Flowers images) and then train our generator to predict the noise components and the clean input. Fig. 10 and Fig. 11 show some denoising results of our method on the images from the above two datasets. Although the images are seriously corrupted by the noise, our method still nicely recovers them even no pair-wise supervision is used during the training. It is worth noting that although the noisy images in this experiment were not captured in the real-world environment, we believe it can still reflect the capability of

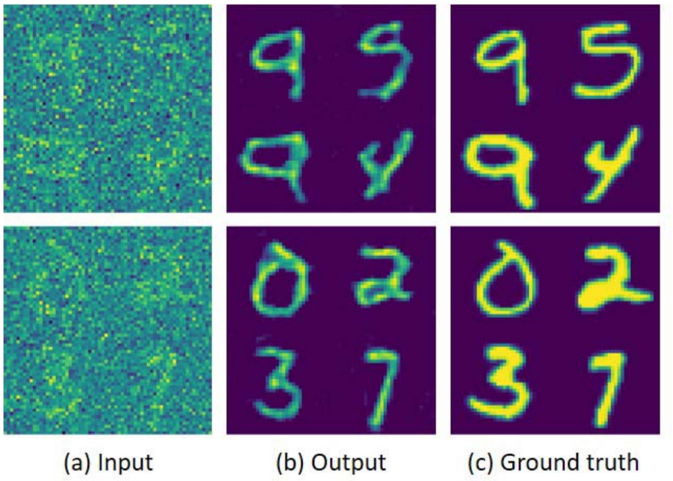


Fig. 10. Some denoising examples of MNIST [54] images by using our method. Despite the strong noise (PSNR < 0.01 dB), the images are still nicely recovered (without pair-wise supervision).

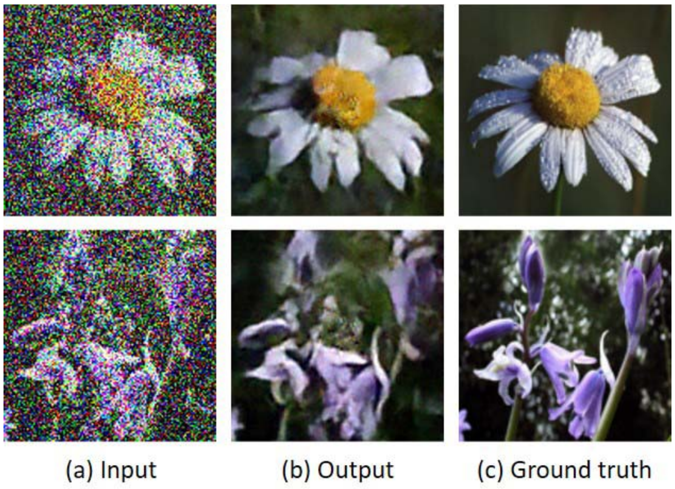


Fig. 11. Denoising results on VGG-Flowers [60] dataset by using our method (input PSNR < 10 dB, without pair-wise supervision).

our method on image restoration tasks, particularly for the input with heavy degradations. We design this experiment mainly to verify whether our method can handle heavy image degradations without the help of pair-wise supervision.

2) Non-Uniform Illumination Correction: According to the physical imaging model [61], [62], the image y of an object in a lighting environment can be formulated as the production of two individual components, 1) the illumination component z, which is determined by the lighting condition, and 2) reflectance component x, which is determined by the surface material of an object. Given an image y, we consider the correction of the non-uniform illumination as the separation of z from x ($z \neq 0$):

$$\begin{aligned} y &= \mathcal{F}_{illum}(x; z) = zx, \\ x &= \mathcal{F}^{-1}_{illum}(y; z) = y/z. \end{aligned} \tag{16}$$

The above functions have the similar formations as those we used in shadow removal. In this experiment, we train our model on CelebA dataset [63] by synthesizing a set of non-uniform illuminated face images. Fig. 12 shows the correction results and the estimated illumination components by using our method.

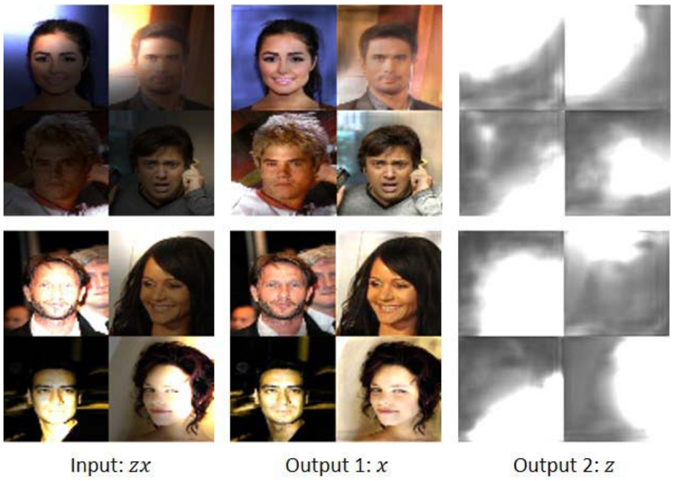


Fig. 12. Some example results of non-uniform illumination correction on CelebA dataset [63] by using our method (without pair-wise supervision).

E. Controlled Experiments

To further verify the importance of each component of the proposed method, ablation analyses are conducted. Evaluations are made on the above mentioned three different tasks: cracking CAPTCHAs, denoising, and non-uniform illumination correction. The full implementations of our model are first evaluated, then we gradually remove the following terms from the loss function during the training, including:

1) Adversarial training: the adversarial term $\mathcal{L}_{\mathcal{F}^{-1}}(G, D_1)$ in Eq. 3. This term has been commonly applied to previous image translation tasks [11], [17] to improve the visual quality of the generated samples.
2) Bidirectional mapping: the adversarial term $\mathcal{L}_{\mathcal{F}}(G, D_2)$ in Eq. 5, together with the Eq. 3, which form the core of this paper.
3) Pair-wise supervised training: $\mathcal{L}_S(G)$ in Eq. 1. We remove this loss term to investigate whether our model can be well trained using the bidirectional mapping without the help of pair-wise supervision.

Table VI shows the results of the above controlled experiment. For the image captcha recognition task, the "word accuracy" and "character accuracy" are recorded. For the image denoising and image illumination correction tasks, the PSNR and SSIM are recorded. We observe a consistent improvement when we integrate the bidirectional mapping constraints to our objective regardless of the presence of pair-wise training supervision. This indicates that the bidirectional mapping provides a self-guidance on the model through adversarial training.

In addition to the vanilla GAN proposed by Goodfellow *et al.* [13], we also consider the recent two modifications of GANs, i.e. Wasserstein GAN (WGAN) [58] and Least Squares GAN (LSGAN) [59], and compare the performance of our framework under different choices of adversarial losses. In these cases, the activation functions at the output layer of the two discriminators are removed so that to produce logits rather than probabilities. We use the RMSProp optimizer for the WGAN-based objectives and use the Adam optimizer for the LSGAN-based objectives. The results in Table VI suggest that using different GAN objectives will result in slightly different performance but the overall accuracy is stable.

VI. Conclusion and Future Work

We propose a novel adversarial training framework for solving inverse problems in image processing. The proposed framework inherently incorporates the physics behind an inverse problem and considers the problem as a prediction of both the unknown source image and operation parameters. By introducing the "independent and universal operation" assumption and the "bidirectional mapping" constraints, a group of the inverse problems in image processing can be nicely solved even without the help of pair-wise training supervision. We also discussed the conditions under which our method can be applied to an inverse problem. The experimental results on a variety of tasks demonstrate the effectiveness of our method. In our future work, we will focus on the inverse problems with multiple (more than two) unknown factors, e.g., superimposed image separation with over two unknown image layers and image reconstruction with multiple degradations. We will also focus on nonlinear degradations, which is more common in real-world inverse problems.

References

[1] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, vol. 89. Philadelphia, PA, USA: SIAM, 2005.

[2] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems*. Amsterdam, The Netherlands: Elsevier, 2018.

[3] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4509–4522, Sep. 2017.

[4] Y. Gandelsman, A. Shocher, and M. Irani, "Double-dip': Unsupervised image decomposition via coupled deep-image-priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, p. 2.

[5] R. Neelamani, "Inverse problems in image processing," Ph.D. dissertation, Dept. Elect. Comput. Eng., Rice Univ., Houston, TX, USA, 2004.

[6] A. Levin and Y. Weiss, "User assisted separation of reflections from a single image using a sparsity prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1647–1654, Sep. 2007.

[7] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, Nov. 1992.

[8] M. Nikolova, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, nos. 1–2, pp. 89–97, Jan. 2004.

[9] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.

[10] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[11] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," 2017, *arXiv:1701.05957*. [Online]. Available: http://arxiv.org/abs/1701.05957

[12] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3883–3891.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[14] E. L. Denton *et al.*, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1486–1494.

[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[16] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[18] J. Hoffman *et al.*, "CyCADA: Cycle-consistent adversarial domain adaptation," 2017, *arXiv:1711.03213*. [Online]. Available: http://arxiv.org/abs/1711.03213

[19] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, "Fully convolutional adaptation networks for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6810–6818.

[20] J. Pan *et al.*, "Physics-based generative adversarial models for image restoration and beyond," 2018, *arXiv:1808.00605*. [Online]. Available: http://arxiv.org/abs/1808.00605

[21] N. Divakar and R. V. Babu, "Image denoising via CNNs: An adversarial approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 80–87.

[22] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018.

[23] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 701–710.

[24] S. Bell-Kligler, A. Shocher, and M. Irani, "Blind super-resolution kernel estimation using an internal-GAN," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 284–293.

[25] X. Cheng, Z. Fu, and J. Yang, "Zero-shot image super-resolution with depth guided internal degradation learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 265–280.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[27] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, Oct. 2016.

[28] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833.

[29] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, vol. 30, no. 1, p. 3.

[30] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 695–704.

[31] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2736–2744.

[32] S. Li *et al.*, "Single image deraining: A comprehensive benchmark analysis," 2019, *arXiv:1903.08558*. [Online]. Available: http://arxiv.org/abs/1903.08558

[33] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1377–1393, Jun. 2020.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[35] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 254–269.

[36] Q. Xu *et al.*, "An empirical study on evaluation metrics of generative adversarial networks," 2018, *arXiv:1806.07755*. [Online]. Available: http://arxiv.org/abs/1806.07755

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.

[38] J. Wang, X. Li, and J. Yang, "Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1788–1797.

[39] L. Qu, J. Tian, S. He, Y. Tang, and R. W. H. Lau, "DeshadowNet: A multi-context embedding deep network for shadow removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4067–4075.

[40] R. Guo, Q. Dai, and D. Hoiem, "Paired regions for shadow detection and removal," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2956–2967, Dec. 2013.

[41] X. Hu, L. Zhu, C.-W. Fu, J. Qin, and P.-A. Heng, "Direction-aware spatial context features for shadow detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7454–7462.

[42] B. Ding, C. Long, L. Zhang, and C. Xiao, "ARGAN: Attentive recurrent generative adversarial network for shadow detection and removal," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10213–10222.

[43] Z. Zou, S. Lei, T. Shi, Z. Shi, and J. Ye, "Deep adversarial decomposition: A unified framework for separating superimposed images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12806–12816.

[44] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Netw.*, vol. 13, nos. 4–5, pp. 411–430, Jun. 2000.

[45] A. Cichocki and S.-I. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, vol. 1. Hoboken, NJ, USA: Wiley, 2002.

[46] K. Gai, Z. Shi, and C. Zhang, "Blind separation of superimposed moving images using image statistics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 19–32, Jan. 2012.

[47] K. Gai, Z. Shi, and C. Zhang, "Blindly separating mixtures of multiple layers with spatial shifts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[48] K. Gai, Z. Shi, and C. Zhang, "Blind separation of superimposed images with unknown motions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1881–1888.

[49] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Comput.*, vol. 9, no. 7, pp. 1483–1492, Oct. 1997.

[50] A. Taleb and C. Jutten, "On underdetermined source separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 3, Mar. 1999, pp. 1445–1448.

[51] Y. Li, S. Amari, A. Cichocki, D. W. C. Ho, and S. Xie, "Underdetermined blind source separation based on sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 2, pp. 423–437, Feb. 2006.

[52] A. Levin, A. Zomet, and Y. Weiss, "Learning to perceive transparency from the statistics of natural scenes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1271–1278.

[53] A. Levin, A. Zomet, and Y. Weiss, "Separating reflections from a single image using local features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2004, p. 1.

[54] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[55] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.

[56] H. Yang. (2018). *Captcha*. [Online]. Available: https://github.com/lepture/captcha

[57] P. Yang. (2017). *Breaking Captcha*. [Online]. Available: https://github.com/ypwhs/captcha_break

[58] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[59] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2794–2802.

[60] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1447–1454.

[61] E. H. Land, "The retinex theory of color vision," *Sci. Amer.*, vol. 237, no. 6, pp. 108–129, 1977.

[62] D. J. Jobson, Z. Rahman, and G. A. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 965–976, Jul. 1997.

[63] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.

Zhengxia Zou received the B.S. and Ph.D. degrees from the Image Processing Center, School of Astronautics, Beihang University, in 2013 and 2018, respectively. He is currently working with the University of Michigan, Ann Arbor, MI, USA, as a Postdoctoral Research Fellow. His research interests include computer vision and the related applications in remote sensing, self-driving vehicles, and video games. He serves as a Senior Program Committee Member/Reviewer for a number of top conferences and top journals, including the NeurIPS, CVPR, AAAI, IEEE TIP, IEEE SPM, and IEEE TGRS.

Tianyang Shi received the B.S. and M.S. degrees from the School of Astronautics, Beihang University, in 2016 and 2019, respectively. He is currently a Researcher with Netease Fuxi AI Lab. His research interests include image processing, deep learning, and their application in games. He serves as a Reviewer for several top conferences, including ICCV, CVPR, AAAI, among others.

Zhenwei Shi (Member, IEEE) received the Ph.D. degree in mathematics from the Dalian University of Technology, Dalian, China, in 2005. He was a Postdoctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China, from 2005 to 2007. He was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA, from 2013 to 2014. He is currently a Professor and the Dean of the Image Processing Center, School of Astronautics, Beihang University, Beijing. He has authored/coauthored over 100 scientific articles in refereed journals and proceedings, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE Conference on Computer Vision and Pattern Recognition, and IEEE International Conference on Computer Vision. His research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning. He serves as an Associate Editor for *Infrared Physics and Technology* and an Editorial Advisory Board Member for the *ISPRS Journal of Photogrammetry and Remote Sensing*.

Jieping Ye (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Minnesota, Twin Cities, MN, USA, in 2005. He is currently a Professor with the University of Michigan, Ann Arbor, MI, USA. His research interests include big data, machine learning, and data mining, with applications in transportation and biomedicine. He was a recipient of the NSF CAREER Award in 2010. His papers have been selected for the Outstanding Student Paper at ICML in 2004, the KDD Best Research Paper Runner Up in 2013, and the KDD Best Student Paper Award in 2014. He has served as a Senior Program Committee/Area Chair/Program Committee Vice Chair for many conferences, including NeurIPS, ICML, KDD, IJCAI, ICDM, and SDM. He serves as an Associate Editor for *Data Mining and Knowledge Discovery* (since 2014), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (since 2014), and IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (2013–2018).