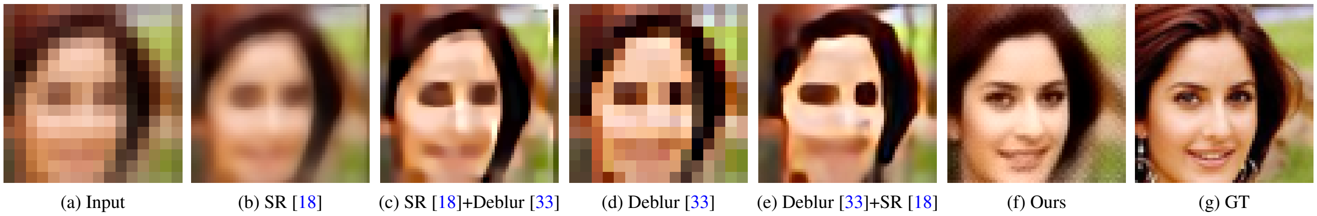
Learning to Super-Resolve Blurry Face and Text Images

Xiangyu Xu[1,2,3] Deqing Sun[3,4] Jinshan Pan[5] Yujin Zhang[1]
Hanspeter Pfister[3] Ming-Hsuan Yang[2]
[1]Tsinghua University [2]University of California, Merced [3]Harvard University
[4]Nvidia [5]Nanjing University of Science & Technology
https://sites.google.com/view/xiangyuxu/deblursr_iccv17



(a) Input (b) SR [18] (c) SR [18]+Deblur [33] (d) Deblur [33] (e) Deblur [33]+SR [18] (f) Ours (g) GT

Figure 1. Low-resolution blurry images (a) are challenging for the state-of-the-art super-resolution and deblurring methods ((b) and (d)). Sequentially applying super-resolution and deblurring methods further exacerbates the artifacts ((c) and (e)). Our method (f) learns to reconstruct realistic results with clear structures and fine details. The low-resolution images ((a) and (d)) are resized for visualization.

Abstract

We present an algorithm to directly restore a clear high-resolution image from a blurry low-resolution input. This problem is highly ill-posed and the basic assumptions for existing super-resolution methods (requiring clear input) and deblurring methods (requiring high-resolution input) no longer hold. We focus on face and text images and adopt a generative adversarial network (GAN) to learn a category-specific prior to solve this problem. However, the basic GAN formulation does not generate realistic high-resolution images. In this work, we introduce novel training losses that help recover fine details. We also present a multi-class GAN that can process multi-class image restoration tasks, i.e., face and text images, using a single generator network. Extensive experiments demonstrate that our method performs favorably against the state-of-the-art methods on both synthetic and real-world images at a lower computational cost.

1. Introduction

We address the problem of jointly super-resolving and deblurring low-resolution blurry images. Such images often arise when the objects of interest are far away from cameras and under fast motion, *e.g.* in surveillance and sports videos. Reconstructing high-resolution clear images from the degraded input not only generates visually pleasing images but also helps other vision tasks, such as recognition [51].

This problem is highly ill-posed and causes significant challenges for state-of-the-art super-resolution and deblurring methods by breaking the basic assumptions on the input. On one hand, super-resolution methods usually assume the blur kernel is known or of simple form, such as Gaussian. When the low-resolution input undergoes complex motion blur, existing super-resolution methods often generate results with large structural distortions, as shown in Figure 1(b). On the other hand, blind deblurring methods often assume that the input is of high resolution and contains salient edges that can be extracted to recover the unknown blur kernel. When the input lacks clear details, the recovered blur kernel and image are not accurate (Figure 1(d)). If we apply super-resolution and deblurring methods sequentially, the artifacts are exacerbated, as shown in Figures 1(c) and (e). A successful solution to this problem should simultaneously deblur and super-resolve the low-quality input.

Toward this end, we propose to focus on two important classes of images, *i.e.*, faces and text, and learn strong category-specific priors to solve this problem. Specifically, we adopt a generative adversarial network (GAN) [13], which consists of generator and discriminator sub-networks that compete with each other. We find that the discriminative network is trained to distinguish fake and real images, thereby learning a semantic prior. This is in sharp contrast to empirical priors [9, 20, 23, 32, 33, 45] that are developed using the statistics of natural images. These statistical priors become less discriminative when the structures of the degraded images are similar to those of the clear images.

Although the basic GAN formulation is effective at capturing semantic information, the recovered images usually contain content and structure errors. To address this issue, we introduce a novel feature matching loss that enforces the output of the generative network to have similar intermediate feature representations with the ground truth training data. Our feature matching loss helps recover realistic details. Furthermore, we develop a multi-class GAN formulation that can learn to super-resolve blurry face and text images using one single generator network.

In this paper, we make the following contributions. First, we propose a new method to simultaneously reconstruct a clear high-resolution image from a blurry low-resolution input. Second, we develop a discriminative image prior based on GAN that semantically favors clear high-resolution images over blurry low-resolution ones. Furthermore, we present a new feature matching method to further retain both the fidelity and sharpness of the reconstructed high-resolution images. Finally, we design a multi-class GAN method that handles both text and face images using one single generator network. We demonstrate that our method performs favorably against the state-of-the-art super-resolution and deblurring methods on both synthetic and real face and text images.

2. Related Work

Image deblurring. Most existing deblurring methods rely heavily on prior models to solve the ill-posed problem. A widely-used prior assumes that gradients of natural images have a heavy-tailed distribution [9, 23, 38]. However, Levin *et al.* [24] show that these priors tend to favor blurry images over the original ones when the blur kernel and clear image are jointly solved using the maximum a posterior (MAP) framework. Therefore, heuristic edge selection steps are often adopted [5, 44] for MAP estimation.

Several recent methods introduce new image priors that favor clear images over blurred ones in the MAP framework [20, 45, 30, 33, 46]. These methods either explicitly or implicitly recover salient edges to estimate the blur kernel, which is complex and time-consuming. More importantly, existing methods do not perform well when low-resolution blurry images do not contain salient edges.

Deep learning achieves promising performance on many applications [21, 52, 1, 14]. Recently, neural networks have also been used for blind image deblurring [39, 3, 37]. However, these deblurring methods still involve explicit kernel estimation. If the estimated kernels are inaccurate, the deblurred images often have significant ringing artifacts. Hradiš *et al.* [15] develop a deep convolutional neural network (CNN) model for text image reconstruction, which does not involve blur kernel estimation. However, their network has been designed for deblurring and cannot be easily extended for joint super-resolution and deblurring.

Super-resolution. Existing super-resolution methods can be broadly categorized as exemplar-based [10, 4, 49, 41, 47] or regression-based [48, 40]. One typical exemplar-based method uses sparse coding [49], which tends to introduce unrealistic details in the reconstructed images. Regression-based methods typically learn the priors from patches [48, 40]. However, the reconstructed results may be over-smoothed. Recently, CNNs have also been applied to super-resolution [7, 43, 8, 18] and obtain promising results when the downsampling kernel is known.

Joint super-resolution and deblurring. This problem has received considerably less attention in the literature although real-world images are often low-resolution with significant blur. Michaeli and Irani [29] propose a blind super-resolution framework that can simultaneously estimate the downsampling blur kernels. Liu and Sun [25] develop a video super-resolution method that jointly estimates the high-resolution image, blur kernel, noise level, and motion. However, these methods do not perform well on low-resolution images with complex motion blurs. Blur kernel estimation becomes extremely challenging and small errors in the estimated kernels are exacerbated by super-resolution. By focusing on images of a certain class (*i.e.*, faces and text) and learning category-specific priors, we can bypass the kernel estimation and obtain superior results.

Generative adversarial networks. Goodfellow *et al.* [13] introduce the GAN framework for training generative models that can generate realistic-looking images from random noise. GANs simultaneously train generator and discriminator sub-networks that compete with each other, making the training process quite challenging. Radford *et al.* [34] use CNNs for both the generator and discriminator to facilitate training. Because strong image priors can be learned, GANs have been applied to image enhancement tasks such as face hallucination [50] and super-resolution [22]. In this work, we extend GANs to the more challenging task of super-resolving low-resolution, severely blurred face and text images.

3. Proposed Algorithm

We first review the basic formulation of GAN, and then introduce the proposed algorithm.

3.1. Overview of GAN

The GAN learns a generative model via an adversarial process. It simultaneously trains a generator network, G, and a discriminator network, D. The training process alternates optimizing the generator and discriminator, which compete with each other. Given D, the generator learns to

Table 1. Architecture of the generator and discriminator. “conv” denotes a convolutional layer, “fc” denotes a fully connected layer, “uconv” denotes a fractionally-strided convolutional layer, and 2× denotes upsampling by a factor of 2.

	Generator												Discriminator				
Layer	uconv	conv	uconv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	fc
Kernel Number	64	64	64	64	64	64	64	64	64	64	64	3	64	64	64	64	1
Kernel Size	6	5	6	5	5	5	5	5	5	5	5	5	4	4	4	4	-
Stride	2×	1	2×	1	1	1	1	1	1	1	1	1	2	2	2	2	-

generate samples that can fool the discriminator; given G, the discriminator learns to distinguish real data and samples from the generator. Mathematically, the loss function is:

$$\max_{\theta} \min_{\omega} \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D_\theta(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_\theta(G_\omega(z)))], \quad (1)$$

where z is random noise; x denotes the real data; ω and θ are parameters of G and D respectively. The discriminator is trained to assign a large probability to real data (first term) but a small one to generated samples by the generator.

The discriminator can be regarded as a semantic prior that can classify clear images (data) from blurry images (samples). Note that the priors used in the MAP-based image deblurring methods, such as dark channel [33], text image [32], and normalized sparsity [20], all exploit some hand-crafted features to distinguish clear images from blurry ones. This observation motivates us to use GAN to learn the discriminator and the features using the following model.

3.2. Network Architecture

Our generator takes low-resolution blurry images as inputs, instead of random noise, and generates high-resolution clear images. The discriminator distinguishes images synthesized by the generator from ground truth clear images.

Generator network. As shown in Table 1, we use a deep CNN architecture that has been shown effective for image deblurring by Hradiš *et al*. [15]. In contrast to their network, our generator contains upsampling layers, *i.e*., uconv in Table 1. These two upsampling layers are fractionally-strided convolutional layers [34], which are also called deconvolution layers. Each deconvolution layer consists of learned kernels that perform jointly to upsample images better than a single bicubic kernel [8]. Our generator first upsamples low-resolution blurry images by the deconvolution layers and then performs convolutions to generate clear images. Similar to the method by Radford *et al*. [34], we use batch normalization [16] and Rectified Linear Unit (ReLU) activations after each layer. The exception is the last layer, which is followed by a hyper-tangent function.

Discriminator network. Our discriminator is a 5-layer CNN network, as shown in Table 1. The input is an image and the output is the probability of the input being a clear image. We use LeakyReLU [27] as the activation function, except for the last layer which uses a sigmoid function [34]. We also use batch normalization [16] after each convolution layer except for the first one.

3.3. Loss Function

A straightforward way for training is to use the original GAN formulation in (1). Let $\{x^i, i = 1, 2, ..., N\}$ denote the high-resolution clear images, and $\{y^i, i = 1, 2, ..., N\}$ represent the corresponding low-resolution blurry images. The training loss for the generator is

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^{N} \log(1 - D_\theta(G_\omega(y^i))). \quad (2)$$

The generated images based on this training loss appear realistic at first glance, *e.g*., the face image in Figure 2(b). However, upon close inspection the generated images are of low quality, especially around the face contours and eyes. As these details are not important features for the discriminator, the generator can still fool the discriminator when making mistakes in these regions. To encourage the generator to construct high-quality results, we propose adding the following terms to the loss function.

Pixel-wise loss. A natural solution is to enforce the output of the generator to be close to the ground truth,

$$L_c(\omega) = \frac{1}{N} \sum_{i=1}^{N} \|G_\omega(y^i) - x^i\|^2, \quad (3)$$

which penalizes the difference in pixel values between the generated output and ground truth. The loss function in (3) leads to visually more pleasing images, as shown in Figure 2(c). However, the restored images are less sharp.

We can combine semantic (2) and pixel-wise (3) losses

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^{N} \|G_\omega(y^i) - x^i\|^2 + \lambda \log(1 - D_\theta(G_\omega(y^i))), \quad (4)$$

where the scalar λ is a trade-off weight. The restored images look more realistic but still contain some artifacts in smooth regions (Figure 2(d)). In addition, the restored images have lower PSNR values than those using only the pixel-wise loss (3).

Feature matching. To achieve more realistic results, we adopt a feature matching loss term [35], defined as

$$\frac{1}{N} \sum_{i=1}^{N} \|\phi_\theta^l(G_\omega(y^i)) - \phi_\theta^l(x^i)\|^2, \quad (5)$$

where $\phi_\theta^l(x)$ represents the feature response to input x at the l-th layer of the discriminator. This term forces the restored images and the real images to have similar feature responses at the intermediate layers of the discriminator network. These features tend to capture the structural information of the images. Different from the perceptual loss in [17] which uses the features of a fixed VGG network, our features are dynamically extracted from the discriminator network, which are most discriminative of real data versus generated data of specific class. Thus, with the help of the feature matching term, the reconstructed results will have more realistic features.

Based on above considerations, we incorporate the pixel-wise loss (3) and feature matching loss (5) into the original GAN formulation (1). The generator and discriminator can be trained by

$$\max_{\theta}\min_{\omega}\frac{1}{N}\sum_{i=1}^{N}\|G_{\omega}(y^i)-x^i\|^2+\lambda_1\|\phi_{\theta}^{l}(G_{\omega}(y^i)) \quad (6)$$
$$-\phi_{\theta}^{l}(x^i)\|^2+\lambda_2(\log D_{\theta}(x^i)+\log(1-D_{\theta}(G_{\omega}(y^i)))),$$

where λ_1 and λ_2 are trade-off weights.

Directly optimizing (6) with respect to θ for updating D makes θ diverge to infinity rapidly, as a large θ always makes the second term $\|\phi_{\theta}^{l}(G_{\omega}(y^i))-\phi_{\theta}^{l}(x^i)\|^2$ larger than a small θ. Instead of updating D to increase the absolute distance between a generated pair (real, generated), we want to make sure the distance between a generated pair is relatively larger than that between a real pair (real, real). Therefore, we modify the loss function of D and optimize G and D by

$$\min_{\omega}\frac{1}{N}\sum_{i=1}^{N}\|G_{\omega}(y^i)-x^i\|^2+\lambda_1\|\phi_{\theta}^{l}(G_{\omega}(y^i))-\phi_{\theta}^{l}(x^i)\|^2$$
$$+\lambda_2\log(1-D_{\theta}(G_{\omega}(y^i))), \quad (7)$$

and

$$\min_{\theta}\frac{1}{N}\sum_{i=1}^{N}-(\log D_{\theta}(x^i)+\log(1-D_{\theta}(G_{\omega}(y^i))))+ \quad (8)$$
$$\lambda_3[\|\phi_{\theta}^{l}(\hat{x}^i)-\phi_{\theta}^{l}(x^i)\|^2-\|\phi_{\theta}^{l}(G_{\omega}(y^i))-\phi_{\theta}^{l}(x^i)\|^2+\alpha]_{+},$$

where α is a margin that is enforced between real and generated pairs and $[\cdot]_{+}$ is the ReLU function. The loss function for G in (7) is composed of (3), (5) and (2), which enforce the output of the generator to be similar to the real data on pixel, structure, and semantic levels, respectively. The loss function for D in (8) introduces the triplet loss [36] into the standard formulation of GAN to ensure that a real sample x is closer to another real sample $\hat{x}$ than the generated one $G_{\omega}(y)$. By introducing the triplet loss, the trivial solution of θ in (6) is naturally avoided since increasing θ will enlarge both the distances between real and generated pairs. Note that the layer l for updating G in (7) and D in (8) can be different. By default, we use the second convolutional layer of D in (7) which maintains the main structure features of the input while using the third layer in (8) which better represents higher level semantic embeddings. As shown in Figure 2(e), this new loss function leads to visually sharp results with higher image quality. Further detailed analysis of the different loss functions is presented in Section 5.

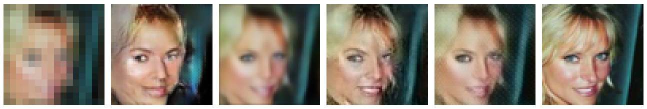

(a) Input (b) Loss (2) (c) Loss (3) (d) Loss (4) (e) Loss (7) (f) GT

Figure 2. Effect of different loss functions. The low-resolution input is resized for visualization. The feature matching loss leads to more realistic images with competitive PSNR. PSNR (dB) values are respectively (b)18.68, (c) 24.31, (d)22.65 and (e) 24.16.

3.4. Multi-Class GAN

The original GAN formulation is designed for images of a single class (SCGAN). Each application or image category requires a new network. It is therefore desirable to train a single network model for multiple categories. To this end, we develop a multi-class GAN (MCGAN) using a single model. Our MCGAN has one generator but K discriminators $\{D_{\theta_j}, j=1,2,...,K\}$. These discriminators are trained to classify real and generated images for each of the K different classes, *e.g.*, text and face images.

Let $D_{\theta_j}(x)$ denotes the probability of x being classified as a real image in the j-th class. The loss functions in equations (2) and (5) respectively become

$$L_{p,M}(\omega)=\frac{1}{N}\sum_{i=1}^{N}\log(1-\sum_{j=1}^{K}D_{\theta_j}(G_{\omega}(y^i))\mathbb{1}(y^i\in C_j)), \quad (9)$$

$$L_{f,M}^{l}(\omega)=\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\|\phi_{\theta_j}^{l}(G_{\omega}(y^i))-\phi_{\theta_j}^{l}(x^i)\|^2\mathbb{1}(y^i\in C_j), \quad (10)$$

where $\phi_{\theta_j}^{l}(x)$ denotes the feature map at the l-th layer of the discriminator D_{θ_j}, and C_j denotes the j-th image class. The indicator function $\mathbb{1}(x)$ is 1 if the expression x is true, and 0 otherwise.

The training process for MCGAN alternates between updating the generator and the discriminator, where the training loss for the generator is

$$\min_{\omega} L_c(\omega)+\lambda_1 L_{f,M}^{l}(\omega)+\lambda_2 L_{p,M}(\omega). \quad (11)$$

Given a fixed generator, the discriminators $\{D_{\theta_K}\}$ are updated simultaneously by (8). After training, the learned generator can be used to restore images in any of the K classes.

4. Experimental Results

We evaluate the proposed method on both text and face images. Since there is no prior work designed for such input data, we compare our method with the state-of-the-art super-resolution and deblurring methods. We show the main results in this section and present more evaluations in the supplementary material.

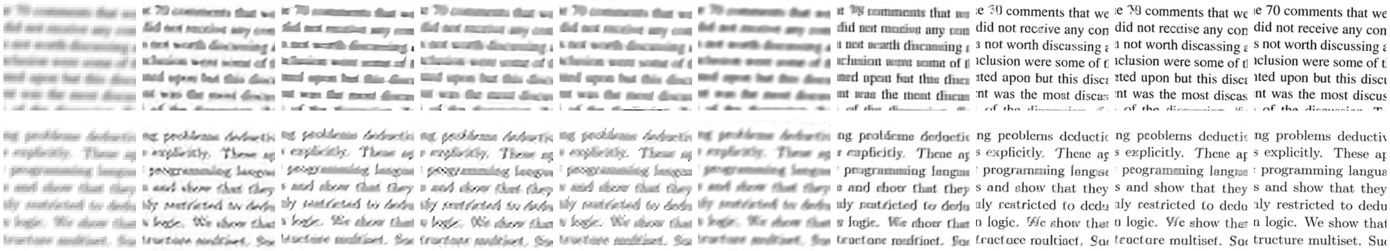

(a) Input (b) [45]+[43] (c) [33]+[18] (d) [43]+[32] (e) [18]+[33] (f) [29] (g) Fine-tune (h) MCGAN (i) SCGAN (j) GT

Figure 3. Results on text images. (a) The low-resolution input images are resized for visualization. (b)-(f) sequentially applying super-resolution and deblurring methods. (g) obtained by combining [18] and [15] and fine-tuning on the text image training dataset. MCGAN (h) and SCGAN (i) generate text images with much clearer characters.

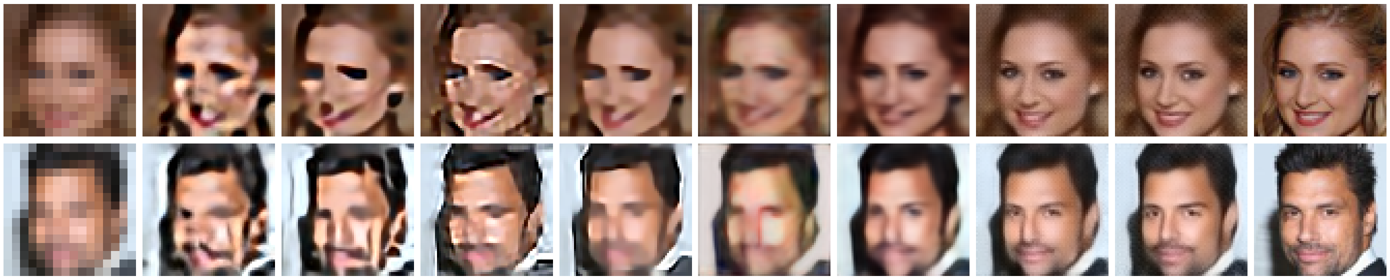

(a) Input (b) [32]+[43] (c) [33]+[18] (d) [43]+[45] (e) [18]+[33] (f) [43]+[15] (g) Fine-tune (h) MCGAN (i) SCGAN (j) GT

Figure 4. Results on face images. (a) The low-resolution input images are resized for visualization. (b)-(f) sequentially applying super-resolution and deblurring methods. (g) obtained by combining [18] and [15] and fine-tuning on the face image training dataset. MCGAN (h) and SCGAN (i) generate face images with fewer artifacts.

Datasets. For text images, we use the training dataset of Hradiš *et al*. [15], which consists of images with both defocus blur generated by anti-aliased disc and motion blur generated by random walk. We randomly crop one million 64×64 blurred image patches from the dataset and down-sample the patches using bicubic interpolation by a factor of 4. For face images, we randomly collect clear face images from the CelebA training dataset [26]. We obtain one million degraded face images by convolving the clear faces with the blur kernels from Hradiš *et al*. [15] and downsampling them by a factor of 4. We also add Gaussian noise to the blurred patches, with the standard deviation uniformly sampled from $[0, 7/255]$. We train the SCGAN models on the text and face datasets separately, and the MCGAN model using both datasets.

To evaluate text image restoration, we use the test set of Hradiš *et al*. [15], which has 100 blurry images. To evaluate face image restoration, we randomly sample 100 images from the CelebA test dataset and convolve them with blur kernels generated by Hradiš *et al*. [15]. Both test sets are downsampled using bicubic interpolation. In addition to these synthetic data, we also capture real text and face images obtained by camera shake or downloaded from the Internet.

Parameter settings. We set the trade-off weights in equation (7) and (8) to be $\lambda_1 = 1$, $\lambda_2 = 10^{-3}$, and $\lambda_3 = 0.1$, and set the margin $\alpha = 1$. Similar to Radford *et al*. [34] we train the models using the Adam optimizer [19] with momentum terms $\beta_1 = 0.5$, $\beta_2 = 0.999$, and a learning rate $lr = 0.0002$. The batch size is 16. Similar to Glorot and Bengio [12], the weights of filters in each layer are initialized using a Gaussian distribution with zero mean and variance of $2/n_{in}$, where n_{in} is the size of the respective convolutional filter. The slope of the LeakyReLU is 0.2. Similar to Goodfellow *et al*. [13], in practice we train G to maximize $\log(D_\theta(G_\omega(y)))$ which provides more sufficient gradients and leads to more stable solution than minimizing $\log(1 - D_\theta(G_\omega(y)))$ in (2). To evaluate real text images, we pre-process the input by gamma correction and contrast transformation to decrease the effect of illumination.

Splitting batches in training the discriminator. Wang and Gupta [42] observe that batch normalization in D causes convergence issues in training GAN. We find that this issue can be resolved by splitting the batches into real and generated ones when training D, as shown in Figure 5. Due to the page limit, we present our analysis and the details of our proposed solution in the supplemental material.

Baseline methods.

We compare our method with all possible combinations of state-of-the-art deblurring [45, 15, 32, 33] and super-resolution [8, 43, 18] methods. Since [15] is specifically designed for text images, we fine-tune the model on face

Table 2. Quantitative comparison with state-of-the-art methods on the text dataset. ”fine-tune” represents the model obtained by combining [18] and [15] and fine-tuning on the text training data.

Methods	[45]+[43]	[45]+[8]	[45]+[18]	[32]+[43]	[32]+[8]	[32]+[18]	[33]+[43]	[33]+[8]	[33]+[18]
PSNR (dB)	14.58	14.41	14.21	14.13	13.90	13.56	14.46	14.22	13.87
SSIM	0.5775	0.5774	0.6002	0.5534	0.5504	0.5603	0.5742	0.5700	0.5783
Methods	[15]+[43]	[15]+[8]	[15]+[18]	[43]+[45]	[8]+[45]	[18]+[45]	[43]+[32]	[8]+[32]	[18]+[32]
PSNR (dB)	13.85	13.66	13.64	15.49	15.49	15.57	15.39	15.35	15.27
SSIM	0.4895	0.4737	0.4766	0.6341	0.6205	0.6584	0.6408	0.6499	0.6512
Methods	[43]+[33]	[8]+[33]	[18]+[33]	[43]+[15]	[8]+[15]	[29]	Fine-tune	MCGAN	SCGAN
PSNR (dB)	15.44	15.54	15.54	16.40	16.45	14.43	17.84	**20.12**	**20.65**
SSIM	0.6396	0.6545	0.6651	0.7171	0.7233	0.5367	0.8142	**0.8970**	**0.9069**

Table 3. Quantitative comparison with state-of-the-art methods on the face dataset. ”fine-tune” represents the model obtained by combining [18] and [15] and fine-tuning on the face training data. The results of [29] are omitted (-) since it is problematic to run this algorithm on the face dataset with small image sizes.

Methods	[45]+[43]	[45]+[8]	[45]+[18]	[32]+[43]	[32]+[8]	[32]+[18]	[33]+[43]	[33]+[8]	[33]+[18]
PSNR (dB)	14.97	14.29	13.62	16.43	15.75	15.22	17.19	16.58	16.16
SSIM	0.3488	0.3240	0.3046	0.4168	0.3870	0.3723	0.4487	0.4218	0.4140
Methods	[15]+[43]	[15]+[8]	[15]+[18]	[43]+[45]	[8]+[45]	[18]+[45]	[43]+[32]	[8]+[32]	[18]+[32]
PSNR (dB)	18.01	17.97	17.91	18.38	18.02	18.12	17.24	16.66	16.79
SSIM	0.4399	0.4375	0.4348	0.5060	0.4708	0.5216	0.4677	0.4314	0.4592
Methods	[43]+[33]	[8]+[33]	[18]+[33]	[43]+[15]	[8]+[15]	[29]	Fine-tune	MCGAN	SCGAN
PSNR (dB)	21.00	20.77	20.61	22.17	22.04	-	22.60	**23.95**	**24.57**
SSIM	0.6201	0.6138	0.6214	0.6453	0.6453	-	0.7137	**0.7479**	**0.7656**

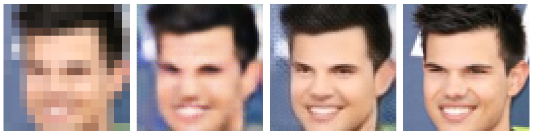

(a) Input (b) w/o (c) w (d) GT

Figure 5. Effect of splitting batches. (b) batch normalization without splitting has convergence issues and does not perform properly; (c) splitting batches leads to more clear results.

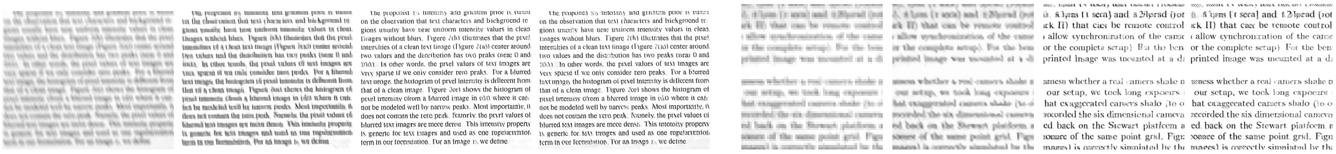
images. In addition, we combine and fine-tune the models in [18] and [15] with both the face and text images in an end-to-end manner. We also compare our method with the blind super-resolution algorithm [29][1].

Results on synthetic datasets. We quantitatively evaluate our method using the text and face image datasets described above. Tables 2 and 3 show that the proposed algorithm performs well in terms of PSNR and structural similarity (SSIM). Note that the MCGAN algorithm performs only slightly worse than the SCGAN method, suggesting the feasibility of using a single network for different image categories. Figures 3 and 4 show some restored images on the text and face datasets. Baseline methods based on straightforward combination of state-of-the-art super-resolution and deblurring schemes do not generate clear images from low-resolution blurry inputs. To analyze the reasons, we show intermediate results in Figure 1. As no salient edges can be extracted from the low-resolution blurry input, it is difficult to estimate blur kernels accurately. The deblurred images contain artifacts (Figure 1(d)), which are exacerbated by the following super-resolution method (Figure 1(e)). Directly applying super-resolution methods to the blurred low-resolution images does not generate reliable results either (Figure 1(b)), as most super-resolution algorithms are developed based on parametric kernels that cannot account for complex motion blurs. Similarly, the artifacts caused by super-resolution are exacerbated by the deblurring methods, as shown in Figure 1(c).

Figure 3(f) shows that the blind super-resolution method [29] does not generate text images well. This method cannot accurately estimate complex motion kernels, and small errors in the estimated kernels are exacerbated by super-resolution. In contrast, our method obtains plausible results without the kernel estimation step. Moreover, different from most methods based on GAN [13, 34, 6] that generate images from random noise, the input of our network is degraded images which contain substantial information for reconstruction. Thus the proposed GAN can restore image details of specific object classes from low-resolution blurry inputs. Note that the fine-tuned model of [18] and [15] is a deep CNN model with 35 layers that has been trained with a mean squared error (MSE) loss function. Since super-resolution based on MSE usually generates over-smoothed results and training a deep network is likely to result in a local minimal solution, this method with the fine-tuned model does not perform well as shown in Figure 3(g) and 4(g).

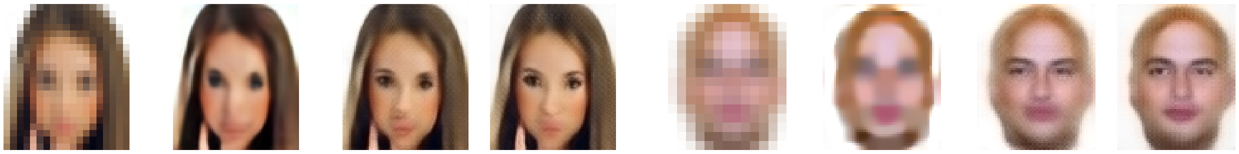
Subjective study. We conduct subjective user study on image quality, which uses 20 images randomly selected from the text and face test datasets. Each low-quality input is restored by 4 different methods: bicubic, [18]+[15] (fine-tuned), MCGAN, and SCGAN. 21 subjects are asked to assign an integer score from 1 (poor) to 5 (excellent) to the reconstructed images by each method, with the original high-resolution image as a reference. The average scores for the 4 methods are respectively 1.14, 2.16, 3.49, and 4.04 on

[1] The results have been kindly provided by the authors using [29] for kernel estimation and [11] for super-resolution.



Input [18] +[15] (fine-tuned) MCGAN SCGAN Input SR[18]+Deblur[33] MCGAN SCGAN

Figure 6. Results on real text images. Our method generates images with much clearer characters.



Input [18] +[15] (fine-tuned) MCGAN SCGAN Input SR[18]+Deblur[33] MCGAN SCGAN

Figure 7. Results on real face images. Our method generates more realistic-looking faces, especially around the eye and mouth regions.

face images, and 1.10, 2.31, 3.49, and 3.68 on text ones, suggesting that the proposed methods can produce results with high perceptual image quality. Due to the page limit, more study on face and text recognition is presented in the supplemental material.

Results on real images. Our method generates visually better results with clearer characters and more realistic faces than other methods, as shown in Figure 6 and 7.

5. Analysis and Discussion

Priors learned by the proposed method. The success of a blind image deblurring method usually depends on a good image prior that favors clear images over blurred ones. Instead of using hand-crafted features, our method learns a discriminator that can distinguish clear and blurred images. To analyze this property, we apply horizontal blurs with size 2 to 10 pixels to clear images from the CelebA dataset [26] and compute the average energies of the blurred images. Figure 8 shows that the learned prior achieves similar effects as the empirical dark channel prior [33], where both priors give higher energies to blurred images.

In addition, the discriminator network also learns to distinguish clear images from generally degraded ones, *e.g.*, images with severe ringing artifacts. As a result, the generator network needs to generate more realistic images with fewer ringing artifacts to fool the discriminator. We compute the values of the learned prior, the normalized sparsity prior [20], and the dark channel prior [33] on clear images and images with severe non-blur artifacts. We use the reconstructed results by the combinations of existing methods [32, 33, 43, 18] as the images with artifacts (examples shown in Figure 4). Table 4 shows the learned prior favors clear images over images with artifacts, while the normalized sparsity prior and the dark channel prior do not.

Effectiveness of the feature matching loss. To understand the effect of each loss term proposed in Section 3.3, we define a dark channel ratio (DCR) to measure sharpness,

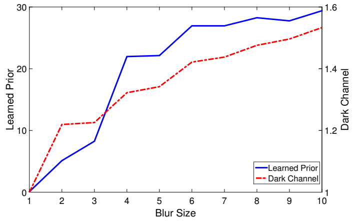


Figure 8. The learned discriminator favors clear images over blurred ones, similar to the empirical dark channel prior. We blur clear images from CelebA [26] using horizontal motion blur kernels ranging from 2 to 10 pixels and evaluate the learned discriminator $-\sum \log D(I_i)$. The left y-axis represents the energy of the learned prior. The right y-axis shows the energy of the dark channel prior on blurry images relative to that on clear images.

Table 4. The learned prior favors clear images over images with artifacts and gives lower energy values to clear images (the first row), while the empirical priors give higher energy values to clear images (the second and third rows). Artifact1 and Artifact2 represent the images with artifacts generated by [33]+[18] and [32]+[43], respectively. The energies of the empirical priors on images with artifacts are relative to those on clear images.

Priors	Clear	Artifact1	Artifact2
Learned Prior	0.0832	4.3659	10.0623
Normalized Sparsity Prior [20]	1	0.7277	0.8513
Dark Channel Prior [33]	1	0.3947	0.3759

$$\mathrm{DCR}(x, x_{gt}) = \frac{f_L(\varphi(x))}{f_L(\varphi(x_{gt})) + \varepsilon}, \tag{12}$$

where x is the input image, x_{gt} is its corresponding ground truth image, $\varphi(x)$ represents the dark channel of x, $\varepsilon = 10^{-8}$ is used to avoid division by zero. $f_L(z) = \sum_{i,j} \mathbb{1}(z_{ij} < t_h)$ approximates the ℓ_0 norm, where z_{ij} denotes the pixel in image z, and t_h is the threshold, which we set to be 0.1. As demonstrated in [33], the dark channel of a clear image is sparser than that of a blurred image. Thus, smaller DCR values indicate sharper results.

As shown in Table 5, the result with (3) has the highest PSNR value but is over-smoothed (Figure 2(c)). The result with (4) has the lowest DCR value but has corrupted structures (Figure 2(b)). Results using the feature matching loss

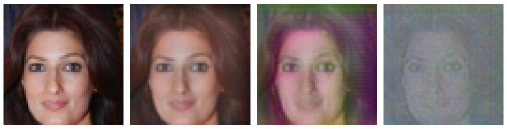

(a) Input (b) 1st (c) 2nd (d) 3rd

Figure 9. Visualization of the features from different layers of discriminator D using [28]. (a) is the original image. (b), (c), and (d) are the reconstructed results from the first, second, and third convolution layer of D, respectively.

Table 5. Effect of loss terms. "pixel" and "basic" denote models trained using the pixel-wise (3) and the basic GAN (4) losses. "GnDm" denotes the model trained using the feature loss (5) with $l = n$ to update G by (7) and $l = m$ to update D by (8) respectively.

Methods	pixel	basic	G2D3	G3D3	G2D2
PSNR (dB)	25.12	22.82	24.57	23.17	22.95
DCR	1.1331	0.9606	1.0687	1.0477	0.9962

Table 6. Average running time (in seconds) of different methods.

Image resolution	[18]+[33]	[33]+[18]	[18]+[15]	Ours
16×16	20.1621	1.7717	0.2596	**0.0080**
50×50	116.5380	4.8499	0.4396	**0.1278**

(GnDm) have competitive PSNR and DCR values, suggesting that the feature matching term is effective at achieving a compromise between fidelity and sharpness. Note that G2D2 leads to worse results than G2D3. This is because the triplet loss introduced in (8) is more effective with higher level features which represent semantic embeddings of real and generated samples.

To better understand features at different layers, we visualize the feature maps of the discriminator network using [28]. Figure 9 shows that shallow layers retain most of the original information while deep layers only retain the basic structures. Therefore, the features from deeper layers tend to guide the generator to generate more semantically realistic results, while the features from shallower layers put emphasis on the pixel-wise similarity with the real images. All features help improve the results as shown in Table 5.

Running time. Our method restores images by a feed-forward network and is therefore more efficient than other state-of-the-art methods. Table 6 summarizes the running time of representative methods on the same PC with an Intel i7 CPU, GTX Titan GPU, and 32GB memory. Our method is 30+ times faster than methods based on empirical priors and 3+ times faster than the deep network ([18]+[15]).

Limitations. Although visually realistic, the reconstructed faces may contain checkerboard artifacts [31]. To analyze their cause, we initialize the generator with random weights, as shown in Figure 10(a) and (b). Using the deconvolutional layer already results in some artifacts for a randomly initialized generator and is likely to be the cause. However, using bicubic interpolation decreases the average PSNR from 24.57 dB to 23.45 dB on the synthetic face dataset. Figure 10(c) and (d) show one example. Future work will address this issue using techniques proposed in [31].

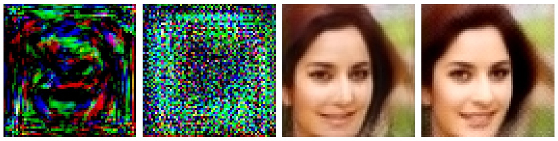

(a) Bicubic (b) Deconv (c) Bicubic/22.00 (d) Deconv/22.78

Figure 10. Analyzing the checkerboard artifacts. (a) and (b) are the output of randomly initialized generators with bicubic and deconvolution layers respectively. Using bicubic interpolation for upsampling (c) reduces the artifacts but has lower PSNR than deconvolutional (d). See Figure 1 for the input and ground truth.

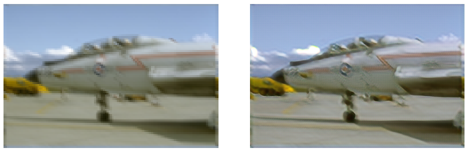

(a) Input (b) Our result

Figure 11. A failure example. The model is trained by using the aforementioned method on the BSDS500 dataset [2].

Furthermore, we note that the generator of GAN learns to model the distribution of real images with guidance from the discriminator. When trained on multi-class images, the proposed model is designed to approximate the mixture distribution of the multi-class images. When this mixture distribution becomes too complex, it is difficult to learn a unified model for the diversity of all image classes. Thus our method is less effective for generic images (Figure 11). Our observation is consistent with the findings for generative models that it is more difficult to generate realistic samples for generic images [35].

6. Conclusions

Our algorithm reconstructs high-resolution clear images from low-resolution blurry inputs. This problem is highly ill-posed and breaks the underlying assumptions of existing super-resolution and deblurring methods. By focusing on images of certain categories (i.e. face and text), we learn strong priors using the GAN framework with new loss functions and obtain promising results. Our method performs favorably against state-of-the-art methods on both synthetic and real-world datasets. In addition, our approach is more efficient since the image restoration process involves only a feedforward network.

Acknowledgement

This work is supported in part by the NSF CAREER Grant #1149783, gifts from Adobe and Nvidia. X. Xu is supported in part by a scholarship from China Scholarship Council. Hanspeter Pfister is supported in part by NSF grants IIS-1447344 and IIS-1607800, and IARPA via DoI/IBC contract D16PC00002.

References

[1] A. Almahairi, N. Ballas, T. Cooijmans, Y. Zheng, H. Larochelle, and A. Courville. Dynamic capacity networks. In *ICML*, 2016. 2

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011. 8

[3] A. Chakrabarti. A neural approach to blind motion deblurring. *CoRR*, abs/1603.04771, 2016. 2

[4] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *CVPR*, 2004. 2

[5] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graph. (SIGGRAPH)*, 28(5):145, 2009. 2

[6] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015. 6

[7] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 2

[8] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 2, 3, 5, 6

[9] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph. (SIGGRAPH)*, 25(3):787–794, 2006. 1, 2

[10] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer graphics and Applications*, 22, 2002. 2

[11] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009. 6

[12] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 5

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 2, 5, 6

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[15] M. Hradiš, J. Kotera, P. Zemcík, and F. Šroubek. Convolutional neural networks for direct text deblurring. In *BMVC*, 2015. 2, 3, 5, 6, 7, 8

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. 3

[17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. 4

[18] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 1, 2, 5, 6, 7, 8

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[20] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, 2011. 1, 2, 3, 7

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[22] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. 2

[23] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph. (SIGGRAPH)*, 26(3):70, 2007. 1, 2

[24] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009. 2

[25] C. Liu and D. Sun. A bayesian approach to adaptive video super resolution. In *CVPR*, 2011. 2

[26] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 5, 7

[27] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013. 3

[28] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015. 8

[29] T. Michaeli and M. Irani. Nonparametric blind super-resolution. In *ICCV*, 2013. 2, 5, 6

[30] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *ECCV*, 2014. 2

[31] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. http://distill.pub/2016/deconv-checkerboard/, 2016. 8

[32] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring text images via L0-regularized intensity and gradient prior. In *CVPR*, 2014. 1, 3, 5, 6, 7

[33] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *CVPR*, 2016. 1, 2, 3, 5, 6, 7, 8

[34] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. 2, 3, 5, 6

[35] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. 3, 8

[36] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 4

[37] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Learning to deblur. *CoRR*, abs/1406.7444, 2014. 2

[38] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graph. (SIGGRAPH)*, 27(3):73, 2008. 2

[39] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, 2015. 2

[40] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, 2014. 2

[41] S. Wang, L. Zhang, Y. Liang, and Q. Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *CVPR*, 2012. 2

[42] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016. 5

[43] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *ICCV*, 2015. 2, 5, 6, 7

[44] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, 2010. 2

[45] L. Xu, S. Zheng, and J. Jia. Unnatural L0 sparse representation for natural image deblurring. In *CVPR*, 2013. 1, 2, 5, 6

[46] Y. Yan, W. Ren, Y. Guo, R. Wang, and X. Cao. Image deblurring via extreme channels prior. In *CVPR*, 2017. 2

[47] C.-Y. Yang, S. Liu, and M.-H. Yang. Structured face hallucination. In *CVPR*, 2013. 2

[48] C.-Y. Yang and M.-H. Yang. Fast direct super-resolution by simple functions. In *CVPR*, 2013. 2

[49] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *TIP*, 19(11):2861–2873, 2010. 2

[50] X. Yu and F. Porikli. Ultra-resolving face images by discriminative generative networks. In *ECCV*, 2016. 2

[51] H. Zhang, J. Yang, Y. Zhang, N. M. Nasrabadi, and T. S. Huang. Close the loop: Joint blind image restoration and recognition with sparse representation prior. In *ICCV*, 2011. 1

[52] Y. Zheng, Y.-J. Zhang, and H. Larochelle. A deep and autoregressive approach for topic modeling of multimodal data. *PAMI*, 38, 2016. 2