

# Deep Context-Aware Descreening and Rescreening of Halftone Images

TAE-HOON KIM, Intel Corporation  
SANG IL PARK, Sejong University, Republic of Korea

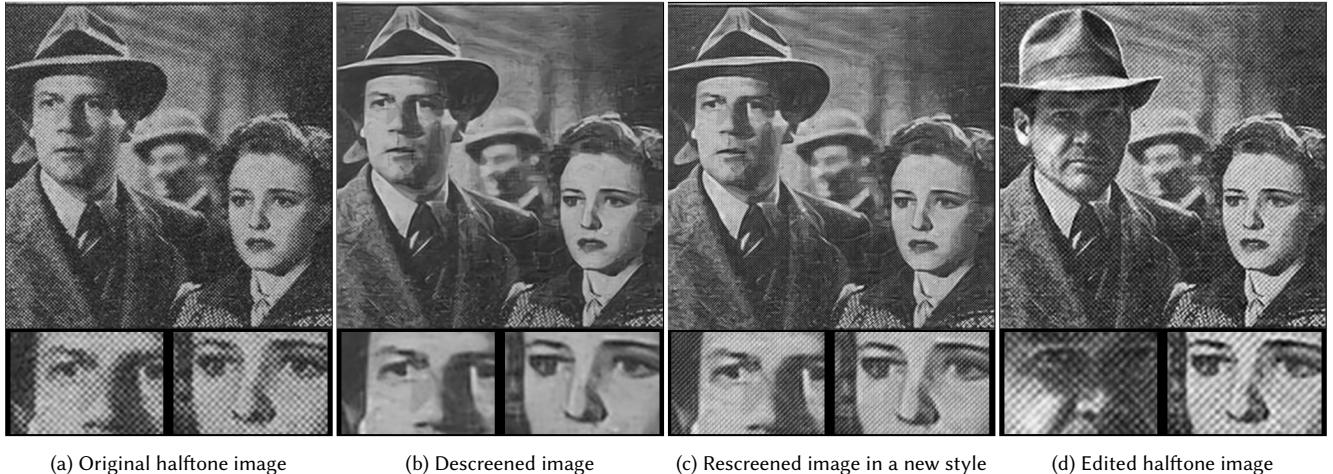


Fig. 1. Descreening and rescreening a halftone image scanned from an old magazine (1940, September 23), *Movie and Radio Guide*. The images below are magnified twice: (a) given original halftone image; (b) descreened image by removing halftone patterns and synthesizing fine details; (c) rescreening the image by applying different halftone styles; (d) editing the descreened image by replacing Joel McCrea's head with Harrison Ford's and applying the original halftone style. Harrison Ford's image adapted from *Indiana Jones and the Last Crusade* (1989) ©Lucasfilm, Ltd.

A fully automatic method for descreening halftone images is presented based on convolutional neural networks with end-to-end learning. Incorporating context level information, the proposed method not only removes halftone artifacts but also synthesizes the fine details lost during halftone. The method consists of two main stages. In the first stage, intrinsic features of the scene are extracted, the low-frequency reconstruction of the image is estimated, and halftone patterns are removed. For the intrinsic features, the edges and object-categories are estimated and fed to the next stage as strong visual and contextual cues. In the second stage, fine details are synthesized on top of the low-frequency output based on an adversarial generative model. In addition, the novel problem of rescreening is addressed, where a natural input image is halftoned so as to be similar to a separately given reference halftone image. To this end, a two-stage convolutional neural network is also presented. Both networks are trained with millions of before-and-after example image pairs of various halftone styles. Qualitative and quantitative evaluations are provided, which demonstrates the effectiveness of the proposed methods.

CCS Concepts: • Computing methodologies → Computational photography;

Authors contributed equally to the work.

Authors' addresses: Tae-hoon Kim, Intel Corporation, thnkim@gmail.com; Sang Il Park, Sejong University, Republic of Korea, sipark@sejong.ac.kr.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

0730-0301/2018/8-ART48 \$15.00

<https://doi.org/10.1145/3197517.3201377>

Additional Key Words and Phrases: computational photography, image processing, halftone, dithering, descreening, image reconstruction, editing

## ACM Reference Format:

Tae-hoon Kim and Sang Il Park. 2018. Deep Context-Aware Descreening and Rescreening of Halftone Images. *ACM Trans. Graph.* 37, 4, Article 48 (August 2018), 12 pages. <https://doi.org/10.1145/3197517.3201377>

## 1 INTRODUCTION

Since the invention of photography, important personal moments have been recorded by taking photos, and photos of historical importance have been published in newspapers, magazines, and books. These printed photos published several decades ago in mass media can still be easily found in public archives despite the fact that their original films might have been lost. Although invented in the late 19th century, photographic printing techniques have not been fundamentally changed, whereby continuous tone images are simulated using dots of a limited number of colors but of varying size and spacing. These techniques are called *Halftone*. However, owing to the physical limitations of the medium (paper) and the process itself, quality degradation during the halftone process is inevitable; thus, older printed photos exhibit visible dots of non-ignorable size. Therefore, restoring halftone images into high quality images using modern digital technologies is as important as colorizing grayscale photos from the perspective of giving a new breath to old photos.

A typical degradation observed in halftone images is the undesirable regular dot patterns that do not exist in the original. This process also results in sudden tone changes, owing to its limited capacity of tone variations, and in loss of the original details. *Image*

*descreening*, on the other hand, is a process for restoring the original from a halftone image. However, effectively removing the dot patterns, reconstructing natural tone changes, and supplementing details are non-trivial tasks. Moreover, a number of different choices of styles and resolutions during the halftone process render the problem even more difficult.

In this paper, a fully automatic, integrated descreening method for halftone images is presented based on end-to-end learning of a convolutional neural network. The proposed method is different from previous methods in the handling of the artifacts: rather than explicitly handling each artifact, a natural-looking image resembling the given input image as much as possible is synthesized as a whole. During this process, the natural tone changes are recovered, and the detailed textures are restored according to the image context. It also effectively handles the so-called *Moiré* artifacts commonly observed in scanned images. Furthermore, the method runs blindly without requiring any extra input information regarding the halftone patterns.

Additionally, novel applications of editing halftone images are presented based on *rescreening*. Inspired by the movie *Forrest Gump* (1994), where historical old newsreels were realistically edited by replacing real persons with actors, the proposed method supplements editing old photos with object removal or insertion while maintaining their halftone appearance, which is a challenging and time-consuming task when typical image editing tools are used. Moreover, the application of transferring the halftone styles from an image to another is presented, so that a photo may appear as if it was printed in a different media or era.

The contributions are as follows: (1) A comprehensive method for descreening is presented based on deep learning. To this end, a novel two-stage model is proposed, where intrinsic features and a low-frequency reconstruction of the image are first estimated, and then fine details are synthesized on top of the low-frequency output. (2) Novel utilization of edge information is proposed for the guidance of a conditional generative model and its objective function. (3) New image editing applications are introduced based on rescreening, and their practicality is demonstrated. (4) A locally space-variant convolutional neural network is proposed for synthesizing the regularity of the halftone patterns.

## 2 BACKGROUND AND RELATED WORK

Halftone is a process for generating perceptually continuous tone changes with a limited number of colors, such as the bi-level case (black and white), which has been a central and active research topic in the printing industry for several decades. To create the perception of tone change, the process usually controls the density of the single-colored dots in a unit area by adjusting their size and spacing. In classical halftone, a physical *screen* has been used to generate a cross-lined effect resulting in a regular grid of dots of varying size. In color halftone, multiple screens of different colors are used. The resolution of the halftone is limited by the types of the printing medium as well as the printing technology. For example, softer paper allows applying a finer grid. Typically, newspaper quality is approximately 85 lines per inch, which suffers from visible dots and loss of detail. Moreover, artificial regular patterns may noticeably

appear in halftone images, which usually worsens in color images because multiple fine screens interfere with each other resulting in undesired large patterns called *Moiré patterns*. A more in-depth discussion of this process is beyond the scope of the paper; readers can refer to introductory books, such as [Kang 1999], for a detailed explanation.

Image descreening is the inverse process for recovering the original by removing the artifacts arising from the halftone process. It has attracted considerable attention from the scanner industry because *Moiré* artifacts are usually more severe when they are scanned and reprinted [Siddiqui and Bouman 2007]. Owing to the inevitable information loss during the halftone process, image descreening is an ill-posed problem, and various approaches have been proposed. Although a simple low-pass filter based method can be applied for the purpose, it suffers from edge softening [Wong 1995]. Thus, most of the work has been focused on developing an edge-preserving filter while suppressing the visible dot patterns. Given prior knowledge of a specific halftone pattern, customized filters have been presented such as [Kim et al. 1995; Wong 1995]. Some exploit wavelet transforms for adjusting the image in the frequency domain [Luo et al. 1998; Xiong et al. 1999]. To cope with multiple halftone styles simultaneously, train-based classifiers for halftone patterns were presented in [Siddiqui and Bouman 2007]. Shou et al. proposed using neural networks for screening pattern classification [Shou and Lin 2010]. Our method is different from aforementioned methods in that not only halftone artifacts are removed but also lost details are recovered. Hou and Qiu presented an inverse halftone method using convolutional neural networks [Hou and Qiu 2017]. However, it is primarily concerned with dithering among several halftone styles.

Recently, in the graphics community, Kopf and Lischinski presented a descreening method specialized for halftone color comics, whereby the grid parameters are estimated by optimization [Kopf and Lischinski 2012]. Later, structure preserving filters were developed and applied to similar comics images to demonstrate the ability of descreening [Cho et al. 2014; Karacan et al. 2013]. Using deep neural networks, Li et al. also presented a method for removing structural lines in manga [Li et al. 2017]. However, these methods were not intended to be used for descreening general halftone images.

Our method is closely related to recent conditional generative models based on deep neural networks. Conditional generative adversarial networks (GANs) [Goodfellow et al. 2014] have been widely used to learn mappings from input to output images in various applications, such as super-resolution [Ledig et al. 2017], inpainting [Pathak et al. 2016; Van Den Oord et al. 2016], and transfer between styles and between domains [Isola et al. 2017; Zhu et al. 2017a,b]. Among them, the image-to-image translation so called pix2pix [Isola et al. 2017] provides a general framework for the majority of recent GAN-based applications. Our method is also inspired by their work. However, the focus is more on providing a tailored, practical solution for the descreening problem. By incorporating different types of networks and designing sophisticated objective functions, the problems of undesired dot patterns, discrete tone changes, and lost details are explicitly addressed.

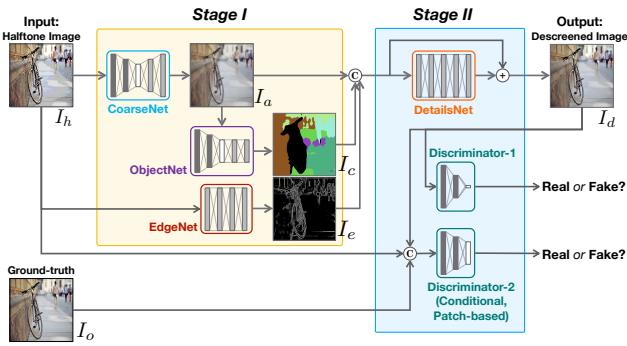


Fig. 2. Overview of the two-stage DescreenNet.

### 3 DESCREENING

Halftone transforms an input image  $I_o \in [-1, 1]^{H \times W \times C}$  into  $I_h = \mathcal{H}(I_o)$ , where  $I_o$  is an original image of height  $H$ , width  $W$ , and color channels  $C$ ;  $I_h$  is a halftone image of  $I_o$ , and  $\mathcal{H}$  is a halftone function. Descreening  $\mathcal{D}$  is an inverse function of  $\mathcal{H}$  and generates an image  $I_d$  from  $I_h$  such that  $I_d = \mathcal{D}(I_h) \approx I_o$ . This problem is ill-posed owing to information loss during halftone.

We propose a learning-based approach for approximating the function  $\mathcal{D}$  so that  $I_d$  is estimated to be as close to  $I_o$  as possible. By exploiting deep convolutional neural networks, we factorize  $\mathcal{D}$  into two stages of consecutive subproblems  $\mathcal{D}_1$  and  $\mathcal{D}_2$ . In the earlier stage  $\mathcal{D}_1$ , we extract intrinsic features of the scene, such as its overall *shapes, colors and tones, edges and contours*, and *contextual information* about objects in the scene. This low-frequency scene reconstruction and high-level scene features are fed into the following stage  $\mathcal{D}_2$ , in which lost details, such as fine textures, are augmented to complete the reconstruction. The two-stage descreening function is given as follows:

$$\begin{aligned} I_d &= \mathcal{D}(I_h) \\ &= \mathcal{D}_2(\mathcal{D}_1(I_h)). \end{aligned} \quad (1)$$

In the following subsections, we describe the detailed building blocks comprising the networks, the objective functions, and the training method, respectively.

#### 3.1 Architecture

The first stage  $\mathcal{D}_1$  is further divided into three sub-networks including CoarseNet which roughly reconstructs colors, tones, and shapes, ObjectNet which creates an object-level context abstraction, and EdgeNet which extracts edge information. These multiple outputs are concatenated and fed into the second stage  $\mathcal{D}_2$  to obtain the final resulting image. The architectures of the sub-networks are summarized in Table 1.

**3.1.1 CoarseNet for low-frequency scene reconstruction.** This is the primary sub-network in  $\mathcal{D}_1$ , and we adopt an autoencoder with U-net structure consisting of a convolutional encoder and a decoder with skip-connections [Ronneberger et al. 2015]. Its skip connections between paired layers in the encoder and the decoder enable the conveyance of more information across the layers and facilitate

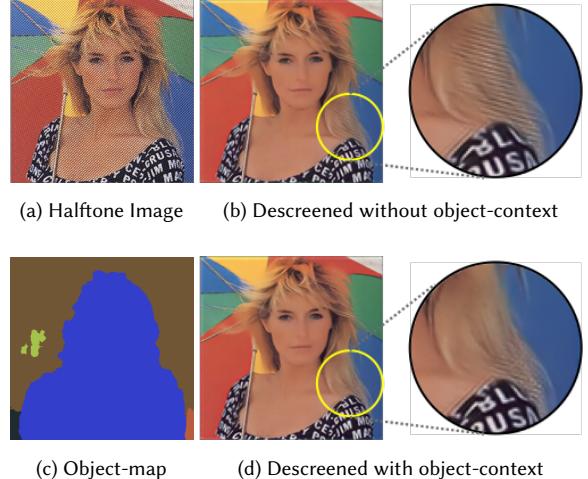


Fig. 3. The effectiveness of the object-level contextual information for the detail synthesis. Input image downloaded from [http://www.descreen.net/eng/soft/descreen/descreen\\_examples.htm](http://www.descreen.net/eng/soft/descreen/descreen_examples.htm) in January 2018.

the generation of consistent results [Isola et al. 2017]. Through the encoder, the input halftone image  $I_h$  is transformed into a code (tensor) of low spatial resolution with a large number of channels representing abstract information of the scene. On the other hand, the decoder expands the code by progressively increasing the spatial resolution and reduces the channel-depth, which finally yields an intermediate resulting image  $I_a$  roughly recovering the overall structure, colors, and tones as well as removing the halftone patterns.

**3.1.2 ObjectNet for object-level context abstraction.** Contextual features at the object-level can provide important prior knowledge for the stage of detail synthesis. For instance, as shown in Figure 3, reliable details around the hair region are not easy to synthesize resulting in contextually erroneous results, such as putting hair in the wrong direction, as shown in Figure 3(b). The object-level features can resolve these issues by indicating the pixels belonging to the hair region. Then, a large receptive field of our model can correctly reconstruct the details (see Figure 3(d)). To assign a proper object label to each pixel, we adopt the scene segmentation method of [Zhou et al. 2017b]. Given the image  $I_a$  from CoarseNet as an input, we directly employ a pre-trained model from [Zhou et al. 2017b] based on the ADE20K dataset with 150 object categories. However, as post-processing, similar categories, such as a river and the sea, are merged, and then the most frequent 25 distinct object categories are chosen, thus generating an object-category map  $I_c$  of 25-dimensional one-hot vector per pixel. The 25 object categories are summarized in Table 2. Figure 3(c) shows an example of  $I_c$ .

**3.1.3 EdgeNet for edge estimation.** Edges and contours are significant visual cues for image reconstruction and have been widely used as a primary user-given constraint for image synthesis [Isola et al. 2017]. To enhance the quality of the detail reconstruction, we propose to detect edge information and to form an edge-map  $I_e$  (see Figure 4). While the state-of-the-art methods for similar purposes

Table 1. Model architectures. The columns represent layer, kernel-size, stride, padding, and output channels. The symbols, C, B, R, L, and E denote a convolution layer, batch normalization, ReLU, LeakyReLU ( $\alpha = 0.2$ ), and ELU ( $\alpha = 1.0$ ), respectively. In (a), layers above and below the dashed line correspond to the encoder and decoder, respectively. In (c), a sequence of layers partitioned by the dotted line is a residual block. The final layers of (b) EdgeNet and (c) DetailsNet are activated by Sigmoid and Tanh functions, respectively.

(a) CoarseNet				(b) EdgeNet				(c) DetailsNet				(d) Discriminator-1				(e) Discriminator-2									
Layer	Krn	Str	Pad	Out	Layer	Krn	Str	Pad	Out	Layer	Krn	Str	Pad	Out	Layer	Krn	Str	Pad	Out	Layer	Krn	Str	Pad	Out	
CL	4	2	1	64	CBR	5	1	2	32	CBL	3	1	1	64	CL	4	2	1	128	CL	5	2	0	128	
CBL	4	2	1	128	CBR	5	1	2	32	CBL	4	2	1	256	CBL	5	2	0	256	CBL	5	2	0	512	
CBL	4	2	1	256	CBR	5	1	2	32	CBL	4	2	1	512	CBL	5	2	0	1024	CBL	5	2	0	1024	
CBL	4	2	1	512	CBR	5	1	2	32	CBL	4	2	1	2048	CBL	5	2	0	2048	CBL	5	2	0	2048	
CL	4	2	1	512	CBR	5	1	2	32	C	3	1	1	64	CL	3	1	1	64	C	4	1	0	1	
CE	4	2	1	512	C	3	1	1	1	CBL	3	1	1	64	CL	3	1	1	64	C	3	1	1	3	
CE	4	2	1	256						CBL	3	1	1	64	CL	3	1	1	64						
CE	4	2	1	128						C	3	1	1	3											
CE	4	2	1	64																					
CE	4	2	1	64																					
CE	3	1	1	64																					
C	3	1	1	3																					

Table 2. Top 25 categories selected as similar categories are merged, so that the coverage of the selected 25 categories is 84.29% of the original 150 categories in [Zhou et al. 2017b]. Other categories are ignored.

No.	Category	Ratio	No.	Category	Ratio	No.	Category	Ratio
1	wall	15.9%	10	bed	2.5%	19	signboard	0.9%
2	building	11.6%	11	cabinet	2.2%	20	sofa	0.6%
3	sky	8.7%	12	ground	2.1%	21	shelf	0.6%
4	floor	6.2%	13	person	1.6%	22	stair	0.4%
5	road	5.9%	14	mountain	1.5%	23	lamp	0.4%
6	tree	4.9%	15	sea, river	1.5%	24	toilet	0.4%
7	ceiling	4.5%	16	vehicle	1.4%	25	PC, TV	0.3%
8	table	3.8%	17	door	1.3%			
9	flower	3.0%	18	curtain	1.1%			
			<b>total</b>			<b>84.2%</b>		

primarily focus on detecting significant contours around objects from conventional images [Bertasius et al. 2015; Shen et al. 2015; Xie and Tu 2015], our objective is to perform a pixel-wise dense edge detection on halftone images. However, halftone patterns render the application of conventional edge-detection algorithms difficult. Moreover, end-to-end learning requires differentiable operations for the inference of an edge-map and for the measurement of the error between the edge-map and its corresponding ground truth. Considering these issues, a convolutional neural network called EdgeNet is designed that, given an input halftone image, estimates each pixel’s conditional likelihood of belonging to an edge. The network consists of a sequence of convolutional layers and activations, as summarized in Table 1.

**3.1.4 Fusion of features.** The three outputs from  $\mathcal{D}_1$ , which are  $I_a$ ,  $I_c$ , and  $I_e$ , are concatenated along the channel-axis into a single feature tensor to be fed into the next stage  $\mathcal{D}_2$  together with the input halftone  $I_h$ , that is,

$$I_f(x, y) = \begin{bmatrix} I_h(x, y) \\ I_a(x, y) \\ I_c(x, y) \\ I_e(x, y) \end{bmatrix}, \quad (2)$$

where  $I_f(x, y) \in \mathbb{R}^{32}$  is the fused feature at position  $(x, y)$ . The dimensions of the composing elements are  $I_h(x, y) \in \mathbb{R}^3$ ,  $I_a(x, y) \in \mathbb{R}^3$ ,  $I_c(x, y) \in \mathbb{R}^{25}$  and  $I_e(x, y) \in \mathbb{R}$ , respectively.

**3.1.5 DetailsNet for adding details.** In the second stage  $\mathcal{D}_2$  with given fused features  $I_f$ , we augment the details onto the resulting

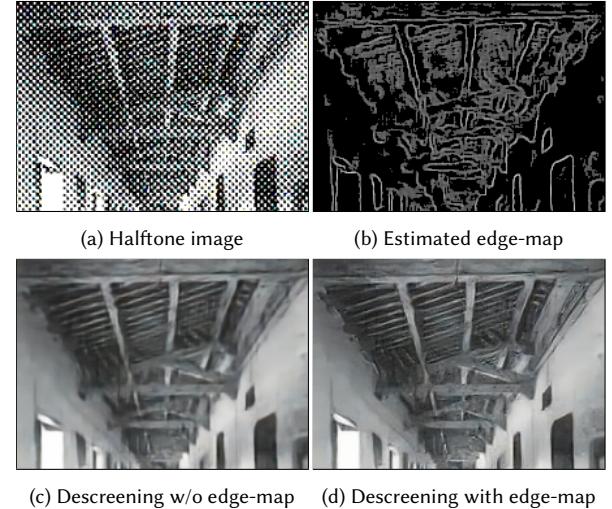


Fig. 4. Comparing descreening without and with the edge-map. The edge-map enhances the sharpness around significant lines, even though it is not pixel-wise accurate. The input image adapted from *Places365* dataset.

image of the CoarseNet  $I_a$  by a convolutional neural network. It is called DetailsNet, and its design is inspired by the residual network [He et al. 2016], that is,

$$\begin{aligned} I_d &= \mathcal{D}_2(\mathcal{D}_1(I_h)) \\ &= I_a + \mathcal{F}(I_f), \end{aligned} \quad (3)$$

where  $\mathcal{F}$  is the residual function represented by DetailsNet and  $I_d$  is the final descreened image. We optimize the parameters of  $\mathcal{F}$  by adopting the framework of generative adversarial networks (GANs) [Goodfellow et al. 2014] because it is applicable to automatically learning an appropriate generative model for resembling ground-truth training images and is suitable for synthesizing high-frequency details [Isola et al. 2017]. GANs are mainly characterized by their use of discriminators for determining whether a given image is real or fake during the training of the generator. Here, we propose the simultaneous use of two discriminators: one is by examining the entire image; the other is by examining local regions patch

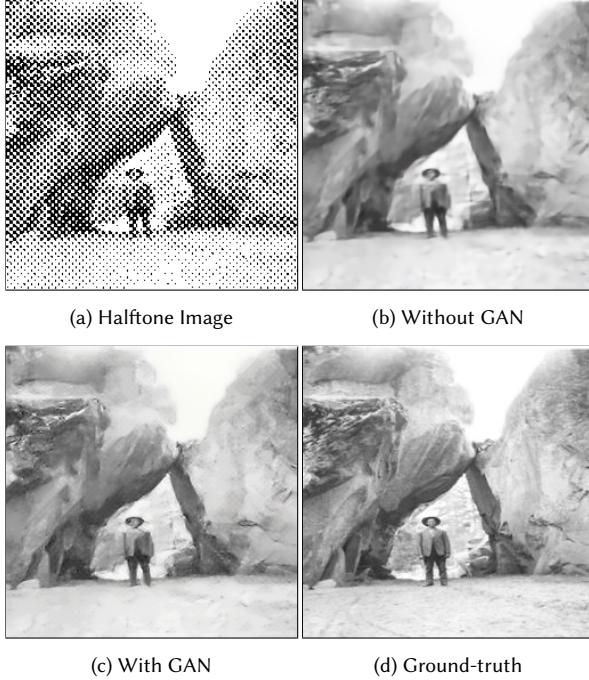


Fig. 5. Descreening without and with adversarial training. Adversarial training helps generate the texture details similar to the ground truth. The input image adapted from *Places365* dataset.

by patch. In the latter case, PatchGAN [Isola et al. 2017] with  $94 \times 94$  overlapping image patches is adopted (see Table 1 for details).

### 3.2 Objective Functions

We propose to use various objective functions for training the networks. The loss functions that should be minimized for training each network are first explained in this section. The effectiveness of these functions is discussed in light of various experiments in Section 5.1. The explanation regarding the objective functions of ObjectNet is omitted because the pre-trained model with its parameter values is directly taken from [Zhou et al. 2017b].

**3.2.1 Loss function for CoarseNet.** The objective of this network is to generate output image resembling the given ground truth as much as possible and to remove halftone artifacts. This is realized into the weighted sum of two loss terms: one is for measuring the difference between two images, and the other is for measuring the remaining artificial dot patterns using global texture statistics.

The difference between two images can be intuitively measured by summing up  $L_1$  or  $L_2$  (MSE) pixel-wise differences [Iizuka et al. 2016; Sajjadi et al. 2017]. We choose to use  $L_1$  loss, which is known to generate a less blurry image compared to  $L_2$  [Isola et al. 2017; Zhu et al. 2017a]. The loss function is applied to train CoarseNet as follows:

$$\mathcal{L}_P(I_o, I_a) = \|I_o - I_a\|_1. \quad (4)$$

With the  $L_1$  loss only, the decoder may leave visible halftone patterns unchanged. We address this by adopting the global level

texture loss function motivated by neural style transfer [Gatys et al. 2016, 2017], where the loss is based on the VGG network [Simonyan and Zisserman 2014], which was originally developed for image recognition. A simpler VGG-16 network with 13 convolutional and 5 pooling layers was chosen because it exhibits sufficient expressiveness for the present purposes. The global texture loss function is defined as follows:

$$\mathcal{L}_T(I_o, I_a) = \sum_l w_l \|G_l(I_a) - G_l(I_o)\|_2^2, \quad (5)$$

where  $w_l$  is a normalization term for layer  $l$  of the VGG network, and  $G_l(\cdot)$  is the Gram Matrix of the feature maps in layer  $l$ . This global texture loss induces the generated output,  $I_a$ , to have similar texture statistics as the ground-truth,  $I_o$ , so that the regular halftone artifacts can be removed. The final loss function  $\mathcal{L}_1$  is

$$\mathcal{L}_1 = w_1 \mathcal{L}_P(I_o, I_a) + w_2 \mathcal{L}_T(I_o, I_a), \quad (6)$$

where the weight values,  $w_1$  and  $w_2$ , were set to 50 and 1, respectively, in the experiments.

**3.2.2 Edge-map matching loss for EdgeNet.** The edge-map  $I_e(x, y)$  is a binary image, where the value at each pixel is either 1 when it belongs to an edge or 0 otherwise. EdgeNet is trained to estimate edge-maps from halftone input images to be as close as possible to the ground-truth edge-maps acquired by applying the Canny edge detector [Canny 1986] to the ground-truth images. It should be noted that more sophisticated edge detectors can be applied here; however, the simplest detector is used, so that as many edges as possible may be found for reconstructing small details. The ground-truth edge-map is denoted by  $I_e^{\text{gt}}$ , and the cross entropy loss is formulated as follows:

$$\begin{aligned} \mathcal{L}_E(I_e) = & -\frac{1}{N} \sum_{x, y} (I_e^{\text{gt}}(x, y) \log(I_e(x, y)) \\ & + (1 - I_e^{\text{gt}}(x, y)) \log(1 - I_e(x, y))), \end{aligned} \quad (7)$$

where  $N$  is the number of pixels in the edge-map.

**3.2.3 Loss function for DetailsNet.** The objective of this network is to generate details for the final output image and to clean up the remaining artifacts in the low frequency reconstruction. The loss function is formulated as the weighted sum of the three content losses and the adversarial loss, that is,

$$\mathcal{L}_2 = w_1 \mathcal{L}_P(I_o, I_d) + w_2 \mathcal{L}_E(I_e^d) + w_3 \mathcal{L}_{\text{local}}(I_o, I_d) + w_4 \mathcal{L}_G, \quad (8)$$

where  $I_e^d$  is the edge-map obtained by feeding  $I_d$  to EdgeNet, and  $\mathcal{L}_{\text{local}}$  is the local patch-wise texture matching loss [Sajjadi et al. 2017] that forces the generated image to have locally similar texture statistics to the ground truth. We use a patch size of  $16 \times 16$  that we empirically found to be sufficient for creating realistic textures.  $\mathcal{L}_G$  is the adversarial loss, which is described in the next paragraph. We set the weights as  $w_1 = 100$ ,  $w_2 = 0.1$ ,  $w_3 = 0.5$ , and  $w_4 = 1$ .

**3.2.4 Adversarial training.** Using  $L_1$  or  $L_2$  loss alone is known to generate a blurry result because the loss functions tend to induce the network to reduce the average distance to multiple solutions. By contrast, conditional generative adversarial networks (GANs) [Goodfellow et al. 2014] have proven capable of producing realistic natural

images with fine details. GANs usually consists of two distinct networks, a generator and a discriminator, and they are alternatively trained to compete with each other

In our case, DetailsNet denoted by  $\mathcal{F}$  is the generator of a residual image given with the fused feature map  $I_f$ . Additionally, following a conventional GAN training practice, we supplement a random noise map  $\mathbf{z}$  to the input parameter for introducing stochasticity, that is,  $\mathcal{F}(I_f, \mathbf{z})$ . For the discriminator, we provide two discriminators: The first discriminator network  $D_1$  is trained to distinguish between the ground truth residual ( $I_o - I_a$ ) and the generated  $\mathcal{F}(I_f, \mathbf{z})$ , whereas the second discriminator network  $D_2$  incorporates the input halftone image  $I_h$  and the object-category map  $I_c$  as input parameters and verifies whether the generated details are appropriate according to the given halftone image as well as its object-level context (see Figure 2).

Although the original adversarial training uses cross entropy loss, we adopt Least Squares GAN [Mao et al. 2017], where the objective functions are  $L_2$  losses between the output of the discriminator for the real and generated images. The training procedure leads to a minimax game, in which the generator is trained to fool the discriminators by misleading their decision to the generated image to be real, whereas the discriminator  $D_1$  and  $D_2$  are trained to make the right decision. Figure 5 shows that our adversarial training can synthesize the high-frequency details well.

### 3.3 Dataset and Training Procedure

For training the network, we use Places-365 dataset [Zhou et al. 2017a] which consists of 2,168,461 images in total (1,803,460 for training, 36,501 for validation, and 328,500 for testing). The dataset has 365 classes of scenes such as an art gallery, a cafeteria, a sky, etc. We first scale down the training images into  $256 \times 256$  pixels for training efficiency. To provide the corresponding halftone image for each image, we made a separate program, in which the subtractive color process is simulated with the CMYK color model. For variety in halftone styles, the dot shapes (round or square), screen angles, and screen resolutions are randomly selected for each example. We also include the dithering based halftone. Eventually, 12,624,220 training image pairs of  $(I_h, I_o)$  are generated (see Figure 6). Given the dataset, we randomly select 500,000 training image-pairs sorted in random order for each epoch. For robustness, a random transformation is applied to each pair including random cropping to  $224 \times 224$ , rotation by  $\theta \in [-30, 30]$  degrees, scaling by  $s \in [0.8, 1.2]$ , horizontal flipping with 50% probability, and adding Gaussian noise  $\mathcal{N}(0, 0.1)$ . In adversarial training, we additionally use distilled hoax examples generated by randomly blending halftone images with their ground truth for a stricter judgment of "fake" to the image failed in removing halftone patterns to the end.

Given the training dataset, we optimize the model parameters by minimizing the loss functions. The filter weights are initialized following [He et al. 2015] and biases are initially set to 0. We use the ADAM optimizer [Kingma and Ba 2014] with an initial learning rate of 0.0001, which is multiplied by 0.9 for every epoch. As it is not simple to perform such an end-to-end joint training of multiple networks using GAN at once, ObjectNet and the EdgeNet are first pre-trained. Subsequently, the other sub-networks join the training,

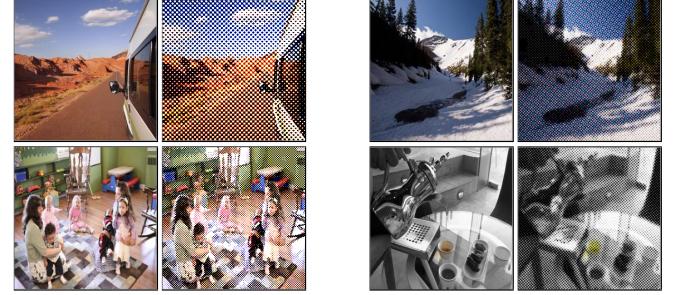


Fig. 6. Excerpts from the training image pairs generated by randomly varying halftone parameters. The input images adapted from *Places365* dataset.

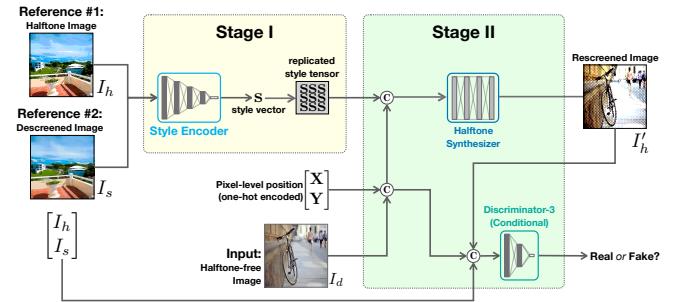


Fig. 7. Overview of our two-stage RescreenNet.

Table 3. Rescreen model architectures. See Table 1 for the notations. In (a), layers above and below the dashed line correspond to the halftone-style encoder for the first stage, and halftone synthesizer for the second stage, respectively.

(a) RescreenNet				(b) Discriminator-3					
Layer	Krn	Str	Pad	Out	Layer	Krn	Str	Pad	Out
CBR	4	2	1	8	CL	4	2	0	32
CBR	4	2	1	16	CBL	4	2	0	64
CBR	4	2	1	32	CBL	4	2	0	128
CBR	4	2	1	32	CBL	4	2	0	256
CB	4	2	1	32	CBL	4	2	0	512
<hr/>									
CBR	5	1	2	32					
CBR	5	1	2	32					
CBR	5	1	2	32					
C	3	1	1	3					

so that CoarseNet is trained for five epochs. Finally, CoarseNet the DetailsNet, and the EdgeNet are jointly trained. After epoch 10, we activate discriminators in stage  $\mathcal{D}_2$  for fine-tuning DetailsNet whereas the other sub-networks are fixed.

The training was performed with PyTorch [Paszke et al. 2017] on two nVidia® GeForce® GTX 1080 Ti and required one week. It should be noted that although the training set has a fixed resolution for simplicity, the trained networks can handle input images of arbitrary resolutions.

## 4 RESCREENING

We address the novel problem of *Rescreening*, where a natural input image is halftoned so as to be similar to a separately given input halftone image. Given a natural input image  $I_d$ , a reference halftone image  $I_h$ , and its descreened image  $I_s$ , the rescreen function  $\mathcal{R}(\cdot)$

synthesizes a new halftone image  $I'_h$  of  $I_d$ . Inspired by the well-known image analogy framework [Hertzmann et al. 2001] of  $I_s :: I_h = I_d :: I'_h$ , we define the function as follows:

$$I'_h = \mathcal{R}(I_d, I_s, I_h), \quad (9)$$

where we model the function  $\mathcal{R}$  as a two-stage network called RescreenNet. In the first stage, a global feature vector describing the intrinsic halftone style is extracted from the pair of  $I_s$  and  $I_h$ . During the second stage, the new halftone image  $I'_h$  is synthesized using a generative adversarial network. The overall network structure is shown in Figure 7.

For the first stage, we adopt a convolutional encoder to extract the global feature vector. To output a fixed size vector, the input image pair is cropped (rather than rescaled) to a resolution of  $64 \times 64$  around its center to retain the original halftone pattern. The input tensor of the network consists of the pair of the RGB values from  $I_s$  and  $I_h$ . During this stage, the spatial dimension of the tensor gradually decreases, whereas the channel depth increases resulting in a  $1 \times 1 \times 32$  dimensional tensor. The detailed layer architecture is given in Table 3.

In the second stage, we convert the input image  $I_d$  to its halftone image  $I'_h$  by adopting a conditional generative adversarial network. However, we observe that a convolutional method does not properly create position-dependent properties, such as regular patterns, because it is intended to be translation invariant. To overcome this, we propose to embed the position information to the input tensor. To deliver distinct, discrete positional information, we apply the modulo operation to each  $x$  and  $y$  value by a certain denominator, and encode them into one-hot vectors. We choose 16 as the denominator for all of our experiments because it is sufficiently large for covering the repetitive patterns of our halftone examples. In summary, the input tensor for each pixel has 67-channel depth in total, consisting of the RGB values from  $I_d$ , the 16-dimensional one-hot vector for each  $x$  and  $y$ , and the 32-dimensional global feature vector from the first stage which is duplicated for every pixel.

We train the network with objective functions of  $L_1$  loss and the global texture matching loss. The  $L_1$  loss is essential for pixel-wise matching between the generated new halftone image and the desired target. The global texture matching loss causes the generated halftone to have similar patterns to the reference  $I_h$  overall. For adversarial training, the conditional discriminator  $D_3(\cdot)$  is used (see Table 3 for details). Using the dataset of descreening, we pick two pairs of random  $64 \times 64$  patches from each sample and use them as the source  $\{I_s, I_h\}$  and the target  $\{I_d, I'_h\}$ , respectively.

## 5 EXPERIMENTAL RESULTS

### 5.1 Descreening

The effect of descreening on various synthetic halftone images is first shown in Figure 8. All test images are from the test sets and have never been used in the training. The process is fully automatic, successfully reconstructs the resulting images with fine details, and removes the halftone patterns.

The quantitative evaluation results are summarized in Table 4, where the PSNR and SSIM [Lim et al. 2017] of the generated outputs are shown for varying the network configurations. The halftone

Table 4. Quantitative evaluation (PSNR(dB)/SSIM). Output images from DetailsNet with edge-map and object-category-map show the best quantitative results. The DetailsNet minimal notes the DetailsNet without object-category-map or edge-map, that is,  $I_c$  and  $I_e$  are zero tensors. The Row 5 corresponds to the full stage-II. The bottom three rows correspond to [Ledig et al. 2017], [Lim et al. 2017], and [Isola et al. 2017], respectively.

PSNR/SSIM	Average	Min.	Max.	Stddev
1. Halftone	16.13/0.4241	6.27/0.0268	26.36/0.9440	6.63/0.2614
2. CoarseNet	27.76/0.8792	16.80/0.4803	39.74/0.9939	3.98/0.0796
3. DetailsNet min.	27.80/0.8833	16.81/0.4977	38.16/0.9939	3.68/0.0751
4. 3+EdgeNet	27.89/0.8863	16.88/0.5058	38.46/0.9944	3.69/0.0736
5. 4+ObjectNet	<b>27.93/0.8867</b>	16.89/0.5030	38.68/0.9945	3.72/0.0739
SRGAN (4x)	16.38/0.4430	8.00/0.0274	25.63/0.9310	4.24/0.2002
EDSR (4x)	20.65/0.6449	6.63/0.0101	32.15/0.9794	5.34/0.2409
Pix2Pix	23.43/0.7588	15.62/0.3696	30.06/0.9777	2.66/0.0942

images exhibit low PSNR and SSIM in average with large variance owing to the large variety of halftone patterns in the test set. By contrast, the output of the proposed models exhibit small variance in both PSNR and SSIM. This also verifies that the use of EdgeNet and ObjectNet decreases errors.

The effectiveness of using the object-map is qualitatively demonstrated in Figure 9. Comparing with the result without the object-map (Figure 9(b)), it is seen that the object-map allows synthesizing high-frequency details (Figure 9(c)). The effectiveness of the edge-map is demonstrated in Figure 10. The halftone image in this example has severe Moiré artifacts, which cause ambiguity in the descreening process and lead to undesired zigzag patterns (Figure 10(c)). However, EdgeNet guides DetailsNet well to successfully synthesize vertical bars as shown in Figure 10(d).

In real-life examples, we apply our model to scanned newspapers printed in different eras. The scanned images of older newspapers usually exhibit additional visual artifacts, such as scratches, inconsistent paper textures, and spatial distortions, as shown in Figure 11, which may cause difficulty in descreening. However, the method still yields convincing reconstructions. Figure 12 is the example of a recent newspaper with interesting fine halftone patterns. The descreened result clearly shows the reconstructed fine details of clothes and regions around the edges of windows. Figure 13 is an example image of clothes from a scanned magazine. This is particularly challenging because various fabric textures cause confusion with the halftone artifacts. Although some failures are observed in the resulting image, such as those shown in the bottom left of the blue box, the method retains the textures of the clothes and mostly removes visible halftone patterns.

Finally, the method is applied to a variety of halftone images collected from the web as well as artwork and comics, as shown in Figure 20. Although we do not train for such examples, the method can still yield decent results, comparable to those of specialized methods such as [Cho et al. 2014; Kopf and Lischinski 2012].

### 5.2 Comparison with State-of-the-art methods

We first compare our results with those obtained by Pix2Pix [Isola et al. 2017] for image translation. Their network was trained (using their original PyTorch implementation with its default configuration) on the same training dataset as the present. As shown in

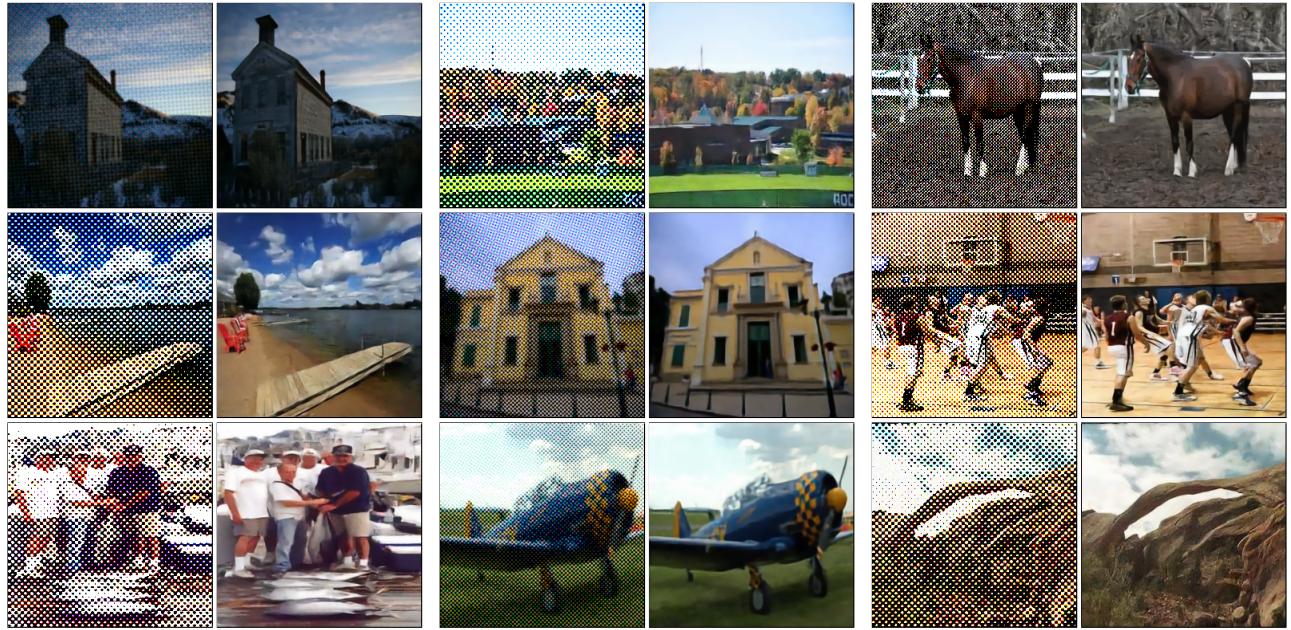


Fig. 8. Results of our approach on new unseen images sampled from Places dataset. Halftones are automatically generated with random parameters. The *left* and *right* of each pair is halftone and descreened, respectively. The results have been produced fully automatically without any user intervention. The input images adapted from *Places365* dataset.

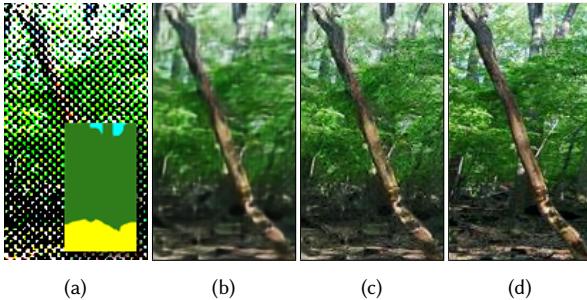


Fig. 9. Comparing descreening without and with object-map: (a) input halftone image and its object-map (cyan, green, and yellow colored regions correspond to sky, tree, and ground, respectively); (b) descreened image without the object map; (c) descreened image with the object map; (d) ground-truth. The input image adapted from *Places365* dataset.

Figure 14, the proposed method can handle halftone patterns more effectively. We further compare with other related state-of-the-arts methods. Given an input image (Figure 15(a)), we apply different methods. First, Figure 15(b) shows our result. Figure 15(c) is a synthesized result from Pix2Pix [Isola et al. 2017]. Figure 15(d) is obtained by applying the bilateral texture filter [Cho et al. 2014], which is one of the most recent related work in the graphics community. Although halftone patterns are removed, the result appears lumpy and blurry. An indirect but intuitive method for descreening may be scaling down the input image followed by applying super resolution to restore its original resolution, owing to recent significant advances in the area. To hide the visible halftone patterns of the input

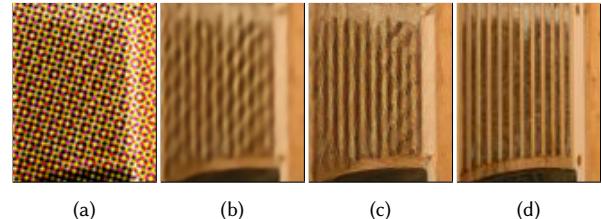


Fig. 10. Comparing descreening without and with edge-map: (a) input halftone image; (b) descreened image without edge-map; (c) descreened image with edge-map; (d) ground-truth. The simple EdgeNet estimates an edge-map from the challenging halftone image and enhances the sharpness and smoothness of significant lines. These figures zoom-in on the low resolution images of  $80 \times 100$  pixels cropped from a  $256 \times 256$  image in the *Places365* dataset.

image, at least 4x scale-down was necessary. Figure 15(e) and (f) show the results of the state-of-the-arts methods, which are sharper but undesirably noisy. Table 4 further shows the quantitative comparison, which demonstrates the proposed method outperforms the other methods.

### 5.3 Rescreening

We first show the effectiveness of our positional data embedding. As shown in Figure 16, without the embedding, the convolutional network does not effectively reproduce the position-dependent patterns. By contrast, the proposed method reproduces the original halftone patterns well.

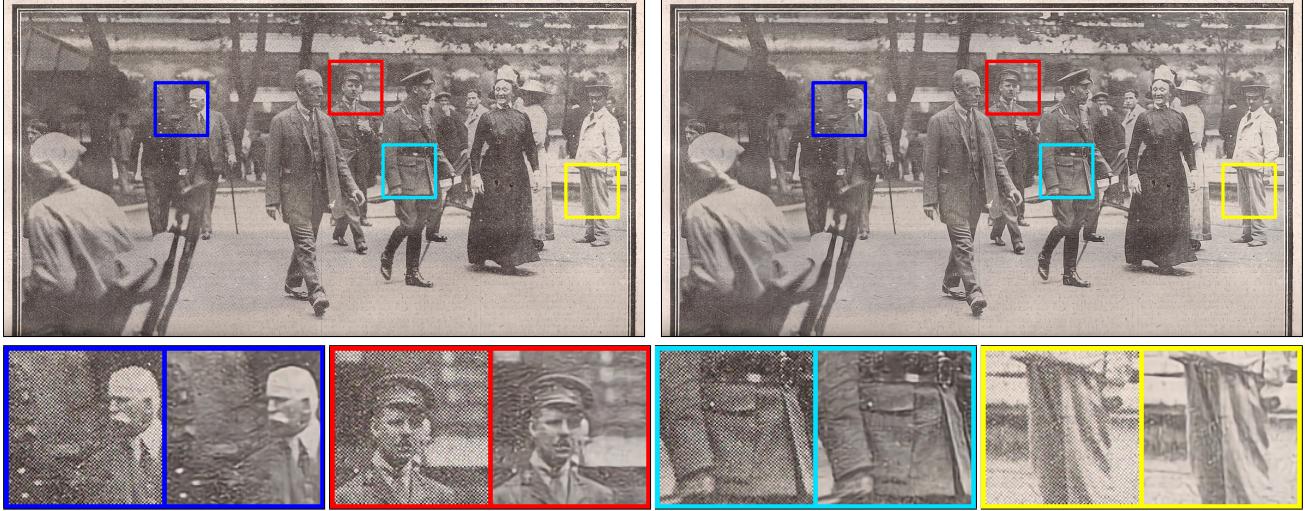


Fig. 11. Descreening of a scanned old newspaper (1917, June 14), *The Daily Mirror*. It should be noted that this example is challenging owing to long-term archiving and additional quality degradation during scanning. The image has a width of 2048 pixels and is best viewed at maximum zoom-in. The *left* and *right* of each pair is halftone and descreened, respectively.

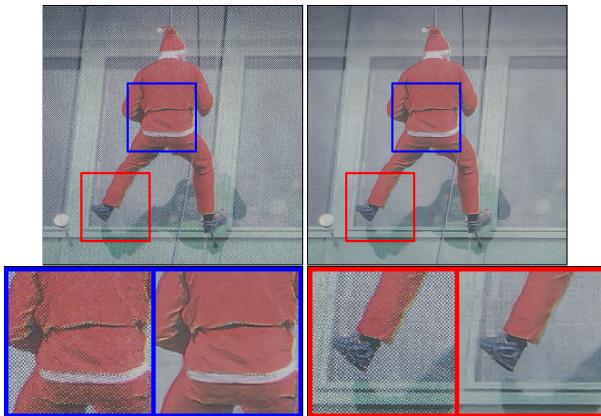


Fig. 12. Descreening of a scanned recent newspaper (2017, December 22), *The Korea Economic Daily*. The scanning process adds visible Moiré artifacts on top of the halftone, which is successfully cleaned up by our method.

Two applications using RescreenNet are now presented. The first application is to transfer halftone style from one image to another. Given two halftone images, this is achieved by descreening both inputs and applying RescreenNet to one of the descreened images referencing the pair of halftone and descreened images of the other, as shown in Figure 1(c) and Figure 17.

In the second application, a given halftone image is edited by inserting or deleting objects. To this end, the input image is first descreened, edited using a conventional image tool, and then rescreened for restoring its original halftone style. Figure 1(d) and Figure 19 show the examples.

Table 5. Computation time elapsed per image. For images from  $1024 \times 1024$ , we measured the performance on CPU only due to the memory issue.

Image Size	Batch-size	CPU (sec.)	GPU (sec.)
$256 \times 256$	10	3.29	0.008
$256 \times 256$	1	3.94	0.032
$512 \times 512$	1	14.39	0.070
$1024 \times 1024$	1	55.88	N/A
$2048 \times 2048$	1	226.2	N/A

#### 5.4 Computation Time

Table 5 summarizes the measured computation time for processing images of different sizes. Intel® Xeon® E5-2686 v4 @ 2.30GHz with 128GB RAM and nVidia® GeForce® GTX 1080 Ti with 12GB RAM were used. The computation time for scene segmentation [Zhou et al. 2017b] of ObjectNet is excluded, which is 0.67 second per image regardless of the resolution owing to its internal scale normalization.

## 6 DISCUSSION

We present novel architectures for descreening and rescreening of halftone images. Based on deep neural networks, two-stage generative models are proposed for both problems, whereby the overall features are constructed in the first stage and the residual details are manipulated in the second stage. Compared to previous descreening methods, the proposed method yields more convincing results and effectively handles a variety of halftone styles by virtue of the expressiveness of the deep neural networks. The extension to rescreening allows editing halftone images easily, which would be difficult and time-consuming using conventional image editors.

As is the case with other learning-based methods, a limitation of the proposed method is its dependency on the training examples. Particularly, synthetic halftone examples were generated for training, which may be a potential weakness when actual, scanned

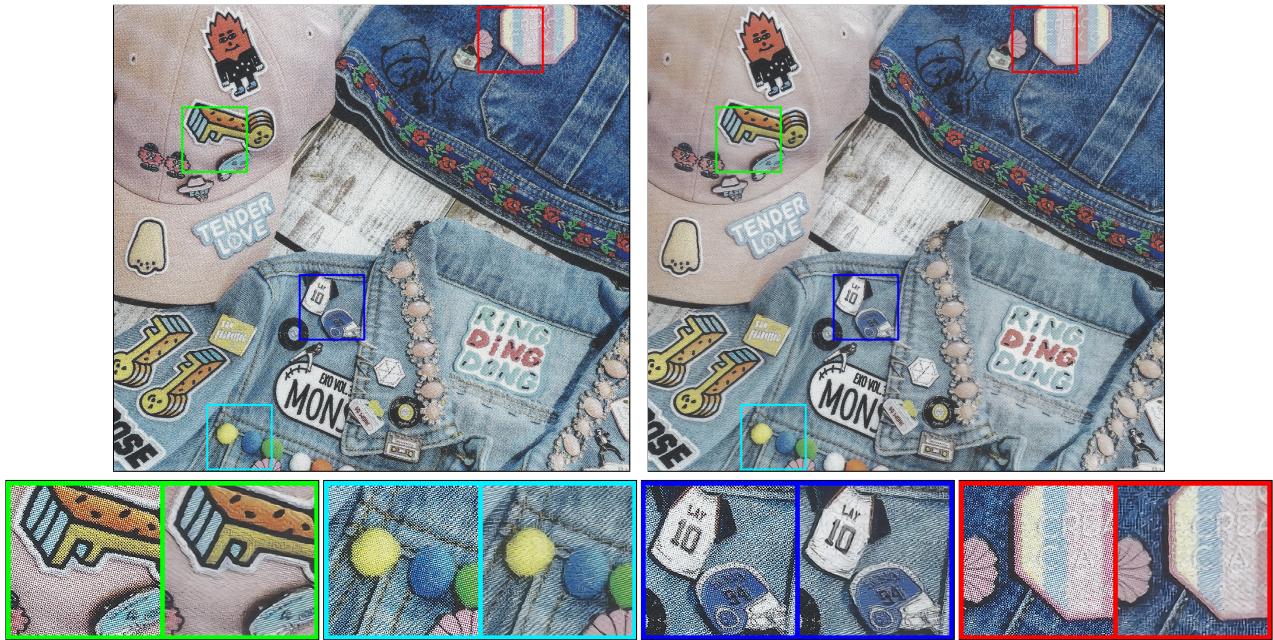


Fig. 13. Descreening of scanned magazines (2017, June), *Nespresso Magazine #28 Seoul*. This example shows the challenging combination of fabric textures and halftone with Moiré patterns. Our method removes the halftone artifacts successfully while retaining the fabric details. The *left* and *right* of each pair is halftone and descreened, respectively.

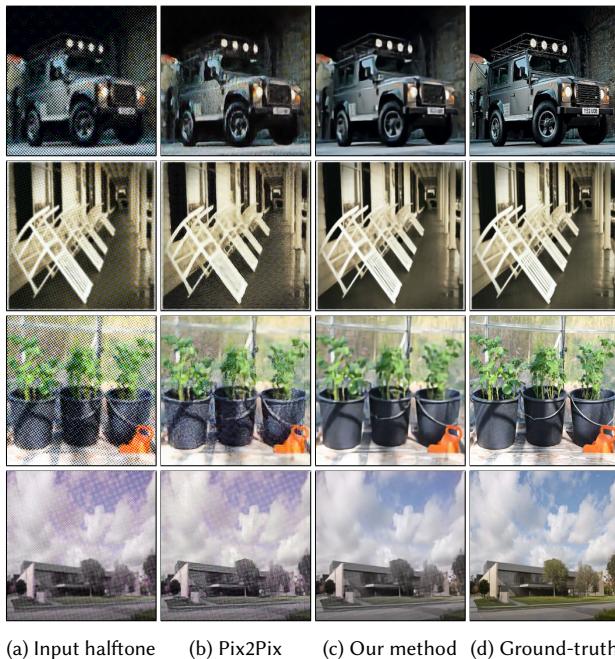


Fig. 14. Comparing our method with image translation [Isola et al. 2017]. The input images adapted from *Places365* dataset.

images in various conditions are handled. However, in the experiments, decent results (to a certain extent) were obtained when the

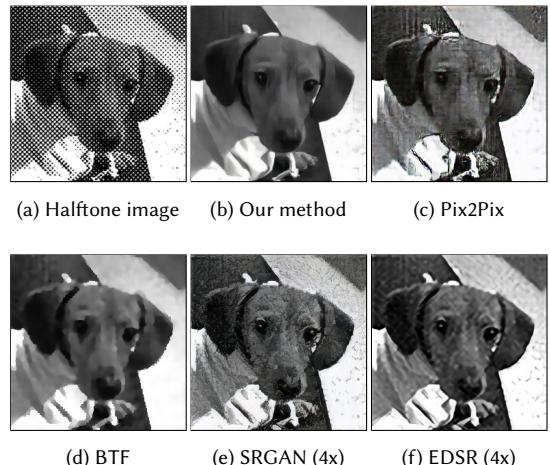
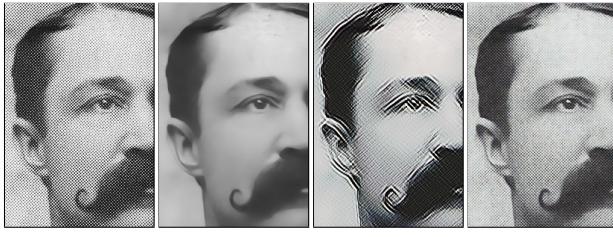


Fig. 15. Comparing (a) a halftone image, (b) our method, (c) image translation by Pix2Pix [Isola et al. 2017], (d) bilateral texture filtering [Cho et al. 2014] and (e)-(f) 4x scale-down followed by scale-up: (e) SRGAN [Ledig et al. 2017], and (f) EDSR [Lim et al. 2017]. Input image downloaded from [http://www.draplin.com/2007/12/halftone\\_histor.html](http://www.draplin.com/2007/12/halftone_histor.html).

proposed method was applied to various scanned images, as shown in Figure 11, 12, and 13. Nevertheless, failure cases were observed when the input images had apparently different characteristics. Figure 18 illustrates those examples. Figure 18(a) is the example of a large, severe moiré artifact that could not be entirely resolved because the training set did not include such an example. Figure 18(b)



(a) Input halftone (b) Descreened (c) w/o position (d) with position

Fig. 16. Effectiveness of the positional data embedding: (a) original halftone image; (b) descreened result; (c) rescreened result back to its original style without positional data; and (d) rescreened result with positional data.

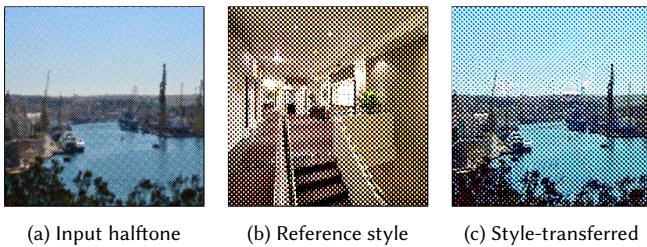


Fig. 17. Halftone style of the reference in (b) is applied to (a). The original halftone in (a) has been replaced to obtain a new halftone image in (c). The input images adapted from *Places365* dataset.

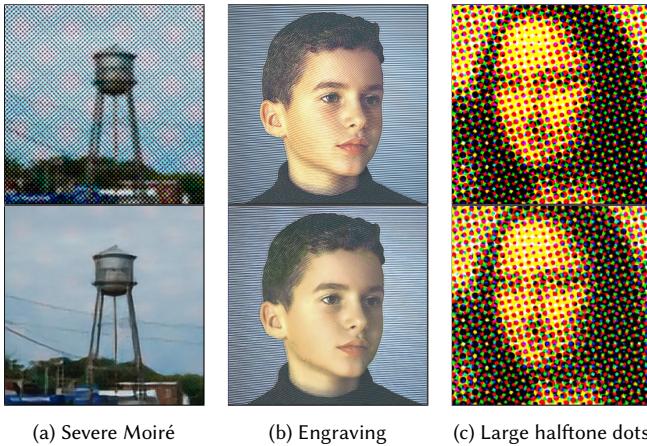


Fig. 18. Failure cases for unseen images whose patterns are outside the scope of the trained images. Input images taken from (a) *Places365* dataset, (b) [Ostromoukhov 1999], and (c) *Mona Lisa* (Leonardo da Vinci).

shows the interesting result of applying the method to an engraved image [Ostromoukhov 1999], where the fine patterns on the face and the hair are smoothed out, whereas the coarse background patterns remain unchanged. Because we don't have any engraving examples, this test shows the extent to which the current model can be tolerated. Figure 18(c) shows the case of considerably larger halftone dots than those in the examples. In that case, the model treats them as meaningful figures to be retained. With intensive

experiments, it is expected that most of those failure cases can be properly handled when more examples are added to the training set.

## ACKNOWLEDGMENTS

The authors would like to thank Prof. Jessica K. Hodgins for providing valuable comments and support. We also thank Hojin Cho and Prof. Seungyong Lee for providing the BTF result. This work (Grants No. C0531029) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2017.

## REFERENCES

- Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. 2015. DeepEdge: A Multi-Scale Bifurcated Deep Network for Top-Down Contour Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- John Canny. 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 6 (Nov 1986), 679–698.
- Hojin Cho, Hyunjoon Lee, Henry Kang, and Seungyong Lee. 2014. Bilateral Texture Filtering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014)* 33, 4 (2014), 128:1–128:8.
- Fernando de Goes, Katherine Breeden, Victor Ostromoukhov, and Mathieu Desbrun. 2012. Blue Noise Through Optimal Transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2012)* 31, 6 (2012), 171:1–171:11.
- Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2414–2423.
- Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. 2017. Controlling Perceptual Factors in Neural Style Transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3730–3738.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems* 27. 2672–2680.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *IEEE International Conference on Computer Vision (ICCV)*. 1026–1034.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. 2001. Image Analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. ACM, New York, NY, USA, 327–340. <https://doi.org/10.1145/383259.383295>
- Xianxu Hou and Guoping Qiu. 2017. Image Companding and Inverse Halftoning using Deep Convolutional Neural Networks. *CoRR* abs/1707.00116 (2017). arXiv:1707.00116 <http://arxiv.org/abs/1707.00116>
- Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2016. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2016)* 35, 4 (2016), 110:1–110:11.
- Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Henry R. Kang. 1999. *Digital Color Halftoning*. SPIE Press.
- Levent Karacan, Erkut Erdem, and Aykut Erdem. 2013. Structure-preserving Image Smoothing via Region Covariances. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)* 32, 6 (2013), 176:1–176:11.
- Yeong-Taeg Kim, Gonzalo R. Arce, and Nikolai Grabowski. 1995. Inverse Halftoning Using Binary Permutation Filters. *IEEE Transactions on Image Processing* 4, 9 (1995), 1296–1311.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).
- Johannes Kopf and Dani Lischinski. 2012. Digital Reconstruction of Halftoned Color Comics. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2012)* 31, 6 (2012).
- Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 105–114.

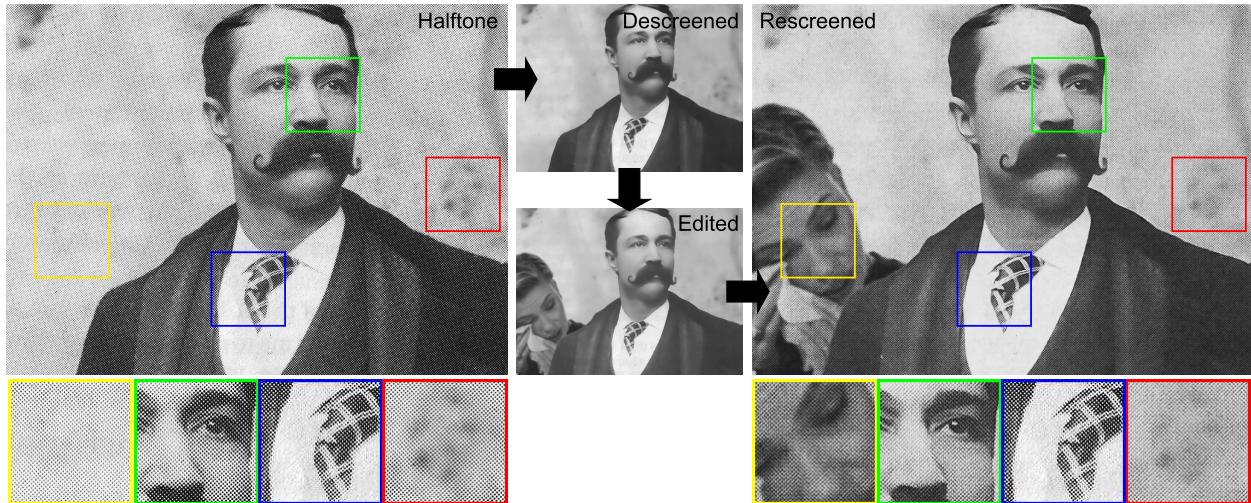


Fig. 19. Editing a halftone image of Frank Matty's portrait: First, the input halftone image is descreened. Then, Scarlett Johansson's photo is composed, and finally the edited image is rescreened into its original halftone style. The colored boxes are zoomed-in on the bottom rows.

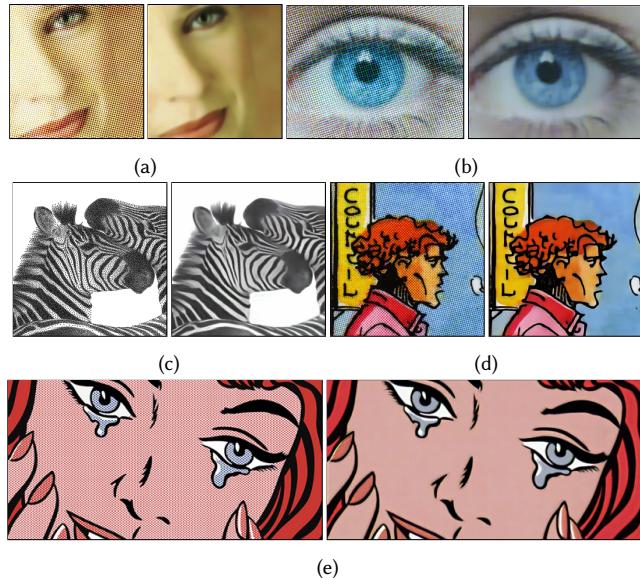


Fig. 20. Descreening of web-crawled photographs and artworks. The *left* and *right* of each pair is halftone and descreened, respectively. Input images from (a) <http://polygonpixelsandpaint.tumblr.com/post/1659606657>, (b) ©IFLScience.com, (c) [de Goes et al. 2012], (d) ©Dupuis, and (e) *Happy Tears* (Roy Lichtenstein), respectively.

- Chengze Li, Xuetong Liu, and Tien-Tsin Wong. 2017. Deep Extraction of Manga Structural Lines. *ACM Trans. Graph.* 36, 4, Article 117 (July 2017), 12 pages.  
 Bee Lim, Sanghyun Son, Heewon Kim, Seungyun Nah, and Kyoung M. Lee. 2017. Enhanced Deep Residual Networks for Single Image Super-Resolution. *2nd NTIRE: New Trends in Image Restoration and Enhancement workshop and challenge on image super-resolution in conjunction with CVPR* abs/1707.02921 (2017).  
 Jiebo Luo, Ricardo de Queiroz, and Zhigang Fan. 1998. A Robust Technique for Image Descreening Based on the Wavelet Transform. *IEEE Transactions on Signal Processing* 46, 4 (1998), 1179–1184.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*.

Victor Ostromoukhov. 1999. Digital Facial Engraving. In *SIGGRAPH 99*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. 2016. Context Encoders: Feature Learning by Inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

O. Ronneberger, P. Fischer, and T. Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI) (LNCS)*. Vol. 9351. Springer, 234–241. (available on arXiv:1505.04597 [cs.CV]).

M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch. 2017. EnhanceNet: Single Image Super-Resolution through Automated Texture Synthesis. In *IEEE International Conference on Computer Vision (ICCV)*. arXiv:1612.07919 (2017), 4491–4500.

Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhiqiang Zhang. 2015. DeepContour: A Deep Convolutional Feature Learned by Positive-Sharing Loss for Contour Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Yu-Wen Shou and Chin-Teng Lin. 2010. Hardware-Friendly Descreening. *IEEE Transactions on Circuits and Systems* 19, 3 (2010), 2287–2299.

Hasib Siddiqui and Charles A. Bouman. 2007. Training-Based Descreening. *IEEE Transactions on Image Processing* 16, 3 (2007), 789–802.

K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel Recurrent Neural Networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. 1747–1756.

Ping Wah Wong. 1995. Inverse Halftoning and Kernel Estimation for Error Diffusion. *IEEE Transactions on Image Processing* 4, 4 (1995), 486–498.

Saining Xie and Zhuowen Tu. 2015. Holistically-Nested Edge Detection. In *IEEE International Conference on Computer Vision (ICCV)*.

Zixiang Xiong, Michael T. Orchard, and Kannan Ramchandran. 1999. Inverse Halftoning Using Wavelets. *IEEE Transactions on Image Processing* 8, 10 (1999), 1479–1483.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017a. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017b. Scene Parsing through ADE20K Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017a. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv preprint arXiv:1703.10593* (2017).

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. 2017b. Toward Multimodal Image-to-Image Translation. In *Advances in Neural Information Processing Systems 30*.