https://doi.org/10.2352/EI.2023.35.15.COLOR-203

Descreening of Dispersed-dot Halftone Images with Generative Adversarial Networks

Baekdu Choi [1] ***and Jan P. Allebach*** [1]

[1] ***Electronic Imaging Systems Laboratory, School of Electronic and Computer Engineering, Purdue University, West Lafayette, IN 47907, US; choi504, allebach@purdue.edu***

Abstract

Halftoning a continuous-tone image inherently results in loss of information, which makes the inverse process, descreening, a challenging problem. Current state-of-the-art descreening algorithms have two issues: first, they mostly are PSNR-oriented reconstruction algorithms, which tend to generate piecewise smooth images that do not appear realistic due to their lack of texture. Furthermore, these algorithms are typically trained with halftone images generated from the Floyd-Steinberg error diffusion algorithm, which is not an optimal choice since the algorithm is known to generate visible artifacts in the halftone image. We address these issues by the following: first, we propose a GAN-based descreening algorithm that generate descreened images with abundant texture resulting in more realistic appearance. In the GAN generator, we use residual in residual dense blocks (RRDBs) which are known to perform well in image reconstruction tasks. For the discriminator, we use a U-net to process the image features in multiple resolutions. Next, we propose using the direct binary search (DBS) algorithm instead of Floyd-Steinberg error diffusion for generating the halftone images, since it is known to generate halftone images without visible artifacts. Both qualitative and quantitative comparisons show that our algorithm outperforms state-of-the-art descreening algorithms significantly.

Introduction

Digital halftoning is a process of converting an image into a binary image with the aim of preserving the image quality as much as possible. In the halftoning context, the input image is often called a continuous-tone image. The reverse process that converts the halftone image into a continuous-tone image is called inverse halftoning or descreening, which is the topic of interest in this paper. Since there are only a limited number of tone levels to be used for halftone images, digital halftoning inherently results in loss of information, which makes inverse halftoning rather tricky. Halftoning is most commonly used in printers to reproduce given image since they can only use a limited number of tones available for each printer-addressable pixel. However, halftoning and inverse halftoning also have other potential applications such as image retrieval [1], watermarking [2, 3], high dynamic range imaging [4], and so on.

Recent advances in applying deep neural networks (DNNs) to image processing problems have brought in significant interest in performing inverse halftoning using DNNs, such as in [5, 6, 7, 8, 9]. While the state-of-the-art inverse halftoning algorithms generate greatly improved results, the output images suffer from lack of texture because they are trained with PSNR-oriented objectives. In this paper, we address this issue by applying generative adversarial networks (GANs) [10] to inverse halftoning, which is known to overcome the limitations of DNNs trained with PSNR-oriented objectives [11].

Another issue with the state-of-the-art inverse halftoning algorithms is that they are generally trained using halftone images generated with the Floyd-Steinberg error diffusion [12] algorithm. While this algorithm generates halftone images with some desirable properties such as sharpened edges [13], the images tend to suffer from visible artifacts [14]. Instead of the Floyd-Steinberg error diffusion algorithm, we suggest generating the training dataset halftone images using the direct binary search (DBS) algorithm [15], which does not generate images with artifacts associated with the Floyd-Steinberg error diffusion algorithm.

This paper is organized as follows. First, we review the literature on digital halftoning and inverse halftoning. We also review the literature on the application of GANs for image restoration and enhancement since the topic is relevant to our paper. Next, we describe our inverse halftoning algorithm and its novelties. Lastly, we provide experimental results, including a state-of-the-art comparison and ablative study.

Our Contributions To summarize, our contributions are as follows: first, we propose a GAN-based descreening algoritm that is capable of generating descreened images with realistic texture. Next, we propose using the direct binary search (DBS) algorithm instead of the Floyd-Steinberg error diffusion algorithm for generating the training dataset, since the halftone images generated with Floyd-Steinberg error diffusion tend to include visible artifacts. We show both qualitative and quantitative comparisons that illustrate the benefits of our descreening algorithm.

Related Work

Digital halftoning

Digital halftoning algorithms can be categorized based on whether the dots (black pixels) tend to cluster together or be dispersed apart. The former type of halftoning algorithms is called clustered-dot halftoning, whereas the latter is called dispersed-dot halftoning. Clustered-dot halftoning is widely used in electrophotographic (laser) printers due to their placement of the toner particles being relatively instable, whereas for inkjet printers dispersed-dot halftoning is commonly used. For clustered-dot halftoning, screening [16] is typically used, where a predesigned threshold array is compared with the input image pixel-by-pixel for binarization. In this paper, we focus our interest on dispersed-dot halftoned images.

Among the dispersed-dot halftoning algorithms, the two types that are most related to our paper are error diffusion algo-

rithms such as Floyd-Steinberg error diffusion [12] and search-based algorithms such as direct binary search (DBS) [15]. Error diffusion algorithms scan through each pixel in the continuous-tone image in some scan order (e.g., raster scan), binarizing the pixel using some quantization scheme. After binarization, the error between the continuous-tone image pixel and its binarized result is diffused to nearby pixels, which then is reflected later when binarizing those pixels. While they are efficient, error diffusion algorithms often suffer from visible streaks of dots that do not match the contents of the continuous-tone image, called artifacts, and thus are rarely used in its native form.

On the other hand, search based algorithms optimize a cost function, typically the loss between the continuous-tone and halftone images after applying human visual system filters, e.g. that described in Näsänen's work [17]. Since the solution space is discrete, the optimization is nonconvex, so typically greedy heuristics and multiple iterations of the entire image are used until convergence for optimization. For instance, the DBS algorithm [15] starts from an initial halftone of the given continuous-tone image, and for each pixel in the image searches for a potential toggle or swap of the dots within a local neighborhood of the pixel that most reduces the cost. This is iterated multiple times over the entire image until there is no acceptable toggle or swap for the current halftone image. Because of their iterative nature, search based halftoning algorithms tend to have a relatively high cost of computation [18].

Recent advances in deep learning have also inspired research in its application for halftoning algorithms. Guo and Sankarasrinivasan [19] jointly learn to halftone and inverse halftone images using unsupervised training similar to the cycleGAN [20] framework. Xia *et al.* [21] notes that there are multiple optimal dot distributions for a given continuous-tone image, and leverages it by learning to encode the fine details of the continuous-tone image in the dot distribution of the halftone images. Compared to both these papers, which employ unsupervised training for halftoning, Choi and Allebach [22] train a GAN framework similar to [10] by using the halftone images generated with DBS [15] as the ground truth halftone images.

Inverse halftoning

Inverse halftoning algorithms taking digital halftones as inputs have been extensively researched. Traditional approaches to inverse halftoning include methods such as MAP estimation [25], LUT approaches [26, 27], wavelets [28], and so on. Recently, due to advances in the application of deep neural networks for image processing, several papers proposed using deep neural networks for inverse halftoning. Hou and Qiu [29] and Xiao *et al.* [5] use a U-net [30] and end-to-end learning for inverse halftoning. Xiao *et al.* [7] estimate the gradient map of the output continuous-tone image from the halftone image as an intermediate step of inverse halftoning, and use the estimated gradient map along with the halftone image for final inverse halftoning. Xia and Wong [6] perform a two-step estimation of the continuous-tone image, where for the second step the output from the first step is used alongside the halftone image as input. Both stages use networks designed with residual blocks [31]. Yuan *et al.* [8] combine the key ideas from [7] and [6], by estimating the gradient map to estimate the initial continuous-tone image and refining it using residual learning. Wicaksono *et al.* [9] learns to inverse halftone in the wavelet domain for the initial estimation, which is then refined in the image domain. All the inverse halftoning algorithms discussed use halftone images generated with Floyd-Steinberg error diffusion for training the networks.

While not as much as inverse halftoning on digital halftone images, research has also been performed with the goal of performing inverse halftoning on scanned prints of images. Gao *et al.* [32] perform inverse halftoning on scanned halftones by first converting them to a synthetic halftone that mimicks the look of digital halftone images, then performing inverse halftoning on them. Kim and Park [33] perform inverse halftoning in two stages as well, where an initial estimate of the continuous-tone image is generated, then using semantic segmentation and edge detection on the initial estimate, the refined final continuous-tone image is generated. They show inverse halftoned results on both digital and scanned halftone images. The two works are similar to our work in that GANs [10] are used for inverse halftoning. However, they are crucially different to our work in two aspects: first, we focus our work on restoring the images from digital halftone images instead of scanned prints, and second, the printed images used in the papers as inputs are the results of printing halftone images that are generated by clustered-dot screening algorithms.

Image restoration with deep neural networks

Another topic that is closely related to our work is applying deep neural networks for image restoration tasks such as image inpainting (e.g., in [35, 36]) and image super-resolution (e.g., in [37, 11]). Since image restoration tasks often aim to recover the lost information from the degradation, generative models, especially GANs, have been successfully applied. Ledig *et al.* [37] first applied GAN training [10] to image super-resolution tasks to greatly improve the perceptual quality of the output images. Wang *et al.* [11] improved the results further by carefully designing the generator architecture along with introducing two-step training, where first the generator network is trained with a PSNR-oriented objective, then is refined using GAN training. Lugmayr *et al.* [38] applied invertible neural networks [39] to image super-resolution, which are available to generate multiple outputs that correspond to the same low resolution input image. For image inpainting, Yu *et al.* [35] were able to succesfully apply GANs. In our work, we also apply GANs similar to those in image restoration literature for inverse halftoning since halftoned images also suffer from loss of information from the continuous-tone images.

Our Work

Network architectures

Our inverse halftoning algorithm is based on the GAN framework in [10]. The overall architecture of the generator is described in Figure 1. The key building block of the generator is the residual in residual dense blocks (RRDBs), which were originally proposed in [11]. The structure of a RRDB block is illustrated in Figure 2. As shown in Figure 1, the input halftone image is passed through two convolution layers with a leaky ReLU [24] activation in between them, then is downscaled by a factor of 2 using bilinear downscaling before being passed through 12 consecutive RRDBs, then is upscaled back to the original scale using pixel shuffling [23]. Lastly, a 1×1 convolution is used to generate the output image. We decided to perform the main computation using the downscaled features since we postulate that halftone images

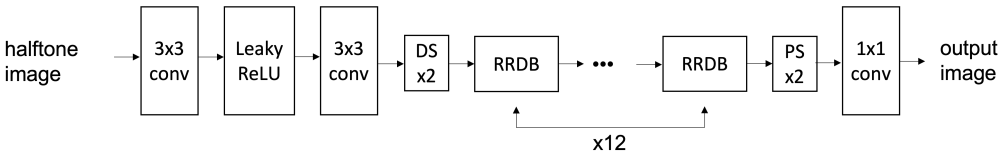

Figure 1. *The overall architecture of the generator in our GAN, which is used for inferencing to generate descreened images. In the figure, DS stands for bilinear downscaling, and PS stands for pixel shuffling [23].*

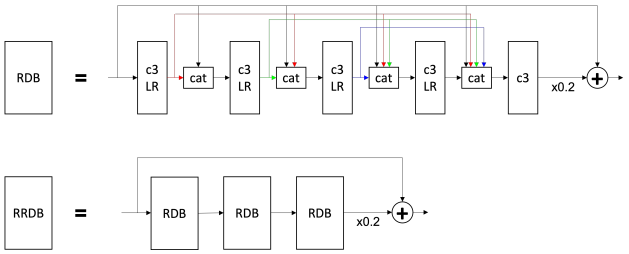

Figure 2. *The structure of a residual in residual dense block (RRDB) [11]. In the figure, c3 stands for a convolution kernel of size 3×3, LR stands for leaky ReLU [24] activation, and cat stands for concatenation of the tensors in the channel dimension.*

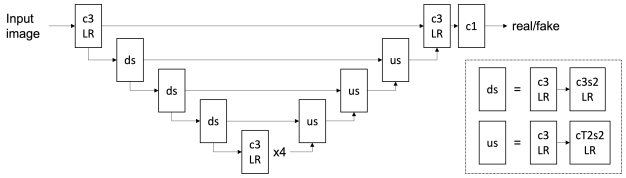

Figure 3. *The overall architecture of the discriminator in our GAN. In the figure, cisj stands for a convolution kernel of size $i \times i$ with stride j, cTisj stands for transposed convolution kernel of size $i \times i$ with stride j, and LR stands for leaky ReLU [24] activation. Blocks which have two inputs first concatenate the inputs in the channel dimension and then process them, whereas blocks with two output arrows simply mean that the same output features are passed onto two different blocks to be processed. All convolution layers in the discriminator are followed by spectral normalization [34].*

are oversampled; that is, the true dimension of the space where the halftone images lie is likely smaller than the true dimension of the space where the continuous-tone images with the same resolution lie.

For the discriminator, we use a U-net [30]. This is because a U-net discriminator has a higher model capacity compared to a typical VGG-style network [40] and can lead the generator to synthesize realistic texture locally [41]. Following [41], we also employ spectral normalization [34] in our U-net discriminator to stabilize the overall GAN training. The overall architecture of the discriminator is illustrated in Figure 3.

Training dataset

We propose using a training dataset of halftone and continuous-tone image pairs that are generated using the direct binary search (DBS) algorithm [15] instead of error diffusion [12], which is commonly used for generating training datasets in the literature. This is because halftone images generated with error diffusion tends to contain visible artifacts in the area corresponding to the smooth region of the continuous-tone image which to human observers is perceived as artificial texture [14], as is shown in Figure 4. On the other hand, halftone images generated with DBS do not show such artifacts. For training our GAN, we use the same dataset as in [22], the training split of the DIV2K dataset [42] halftoned using DBS [15] divided into 27,958 nonoverlapping 256×256 patches.

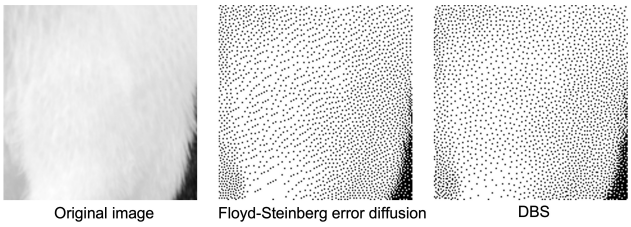

Figure 4. *Comparison of halftone images generated with Floyd-Steinberg error diffusion [12] versus direct binary search (DBS) [15]. In the halftone image generated with Floyd-Steinberg error diffusion, there are visible artifacts appearing similar to texture with a diagonal direction, which is not in the original image. The halftone image generated with DBS does not have such artifacts.*

Training details

We train the GAN network for descreening in two steps, inspired by the work in [11]. In the first step, we do not train the entire GAN, but only train the generator using PSNR-oriented loss terms. The loss terms used for the first step include the perceptual loss [43] $\mathscr{L}_p$ and the L_1 loss $\mathscr{L}_{GT}$ between the ground truth and the descreened images. For computing the perceptual loss, we use the pretrained VGG-19 network [40], and the features extracted from the last convolution layers of block 2 and 3 before activation are used. The overall weighting for the loss term of the first stage training is given by $\mathscr{L} = \mathscr{L}_{GT} + 0.01\mathscr{L}_p$.

For the second stage of training, we incorporate the entire GAN. For the adversarial loss, we use the hinge GAN loss [44] in its relativistic average form [45]. The overall weighting for the loss term of the second stage, for the generator update, is given by $\mathscr{L} = \mathscr{L}_{GT} + \mathscr{L}_p + 0.1\mathscr{L}^G_{adv}$. The relativistic average hinge GAN loss for the discriminator is given by

$$\mathscr{L}^D_{adv} = \mathbb{E}_{x_r \sim P_r}\Big[\max\big(0, 1 - \tilde{D}(x_r)\big)\Big] + \mathbb{E}_{x_f \sim q}\Big[\max\big(0, 1 + \tilde{D}(x_f)\big)\Big] \quad (1)$$

whereas the relativistic average hinge GAN loss for the generator is given by

$$\mathscr{L}_{adv}^{G} = \mathbb{E}_{x_f \sim q}\Big[\max\big(0, 1-\tilde{D}(x_f)\big)\Big] + \mathbb{E}_{x_r \sim P_r}\Big[\max\big(0, 1+\tilde{D}(x_r)\big)\Big] \quad (2)$$

$x_r \sim P_r$ refers to the samples from the training dataset and $x_f \sim q$ refers to the generated images. $\tilde{D}(x_r)$ and $\tilde{D}(x_f)$ are defined as

$$\tilde{D}(x_r) = C(x_r) - \mathbb{E}_{x_f \sim q}\big[C(x_f)\big] \quad (3)$$

and

$$\tilde{D}(x_f) = C(x_f) - \mathbb{E}_{x_r \sim P_r}\big[C(x_r)\big] \quad (4)$$

where $C(\cdot)$ denotes the output from the discriminator network.

Each stage in the training is performed for 20 epochs with a batch size of 4, which results in around 140K updates. For the first stage, we fix the learning rate to be 2×10^{-4}. For the second stage, the learning rate starts at 2×10^{-4} and after 10 epochs it is scaled down by 0.9 for each epoch. For the discriminator, the learning rate is scaled by 4.0. We use the Adam optimizer [46] with hyperparameters $(\beta_1, \beta_2) = (0.5, 0.999)$ for both stages.

Experiments

State-of-the-art comparison

For the state-of-the-art comparison, we choose the works described in the following paragraphs. Since we are using a dataset generated with the DBS algorithm, we also train the networks in the baselines using the training dataset we used for our work for a fair comparison. For the training procedures, we follow the training process described in the papers as closely as possible.

PRL [6]. PRL performs descreening in two steps. In the first step, a coarse reproduction of the continuous-tone image is generated from the input halftone image, using a variant of U-net [30] where the skip connections are replaced by residual connections [31]. Then, the coarse reproduction, along with the initial halftone image, is passed through the *detail enhancement* phase where the fine details of the continuous-tone image are refined via residual learning.

GGDCNN [7]. GGDCNN first estimates the x and y direction gradients of the continuous-tone image from the input halftone image. The gradients are defined using Sobel filters. Next, using both the estimated gradients and the input halftone image, the final estimated continuous-tone image is generated. While due to its gradient enhancing nature the output images appear sharper, the images generated with GGDCNN also tend to include unnatural artifacts.

GGRL [8]. GGRL combines the ideas from PRL and GGDCNN by using GGDCNN for the first phase of PRL that generates a coarse reproduction of the continuous-tone image. From the output of GGDCNN, the final output image is generated via residual learning similar to as in PRL.

SAIHN [9]. SAIHN performs initial estimation of the continuous-tone image in the stationary wavelet transform (SWT) domain. The input used is the SWT of the halftone image, and the estimation output is the SWT of the continuous-tone image. After the continuous-tone image is estimated in the SWT domain, the inverse SWT is performed to get the estimation in the image domain, which for refinement is then passed through a relatively small convolutional neural network.

Real-ESRGAN [41]. Since our algorithm is closely related to the ESRGAN [11] variants, we also compare the results to the pretrained Real-ESRGAN [41] network. For this, we first filter the image with an isotropic Gaussian filter with $\sigma = 1.25$, downscale the image using bilinear downscaling by a factor of 2, and then perform super-resolution on the images. σ was chosen to find the Gaussian filter that removes the dot structure from the halftone image while preserving the image details as much as possible, by searching for the value that gives the best value of the LPIPS [48] image quality metric after reconstruction using the pretrained Real-ESRGAN network.

For quantitative comparison, we report multiple image quality (IQ) metric values. Along with the commonly used metrics PSNR and SSIM [49], we report the following IQ metrics:

BRISQUE [50]. BRISQUE computes the image features by taking into account the statistics of the local luminance and the relations between the neighboring pixels in the image. These features are then used to compute an image quality score, using a support vector regression [51] module trained on a dataset of images and their corresponding quality scores.

LPIPS [48]. LPIPS is an extension to the works of Johnson *et al.* [43], where features of the two images to be compared are extracted using a pretrained classification network and then are compared to determine the similarity between the two images. A linear mapping from the feature differences of the two images computes the similarity score of the two images. Using the ground truth image as one of the images, LPIPS can be used as an image quality metric when ground truth exists, as in our case. We choose to use AlexNet [52] as the feature extractor for LPIPS in our paper for fair comparison, since we are using a perceptual loss term using the pretrained VGG-19 [40] in our training.

DISTS [53]. DISTS also uses a pretrained network to extract features from two images to be compared, but before fitting the features into a score they are converted into texture and structure features using a method similar to SSIM [49]. The SSIM-like features are then used to estimate the similarity score between the two images via a linear mapping. While DISTS uses a pretrained VGG-16 for feature extraction, they substitute the max-pooling layers of the VGG-16 network with weighted L_2 pooling and then retrain the network to avoid aliasing during downscaling of the features, so there is no potential issue of overfitting of the scores coming from using perceptual loss in the training.

Figure 5 includes the results generated by the different descreening algorithms to be compared. The descreened images generated with our algorithm contain realistic texture whereas algorithms trained with PSNR-oriented objectives generally are overly smooth. This observation is supported by the IQ metric values reported in Table 1. The IQ metrics in Table 1 are computed over the DIV2K validation set [42] and the Waterloo Exploration Database [47]. Note that for PSNR and SSIM, our algorithm does not appear to produce better results, but this is because PSNR and SSIM do not correlate well with image quality observed by human observers [38]. Indeed, we can see that for the other IQ metrics that fit better with perceived image qualities, our

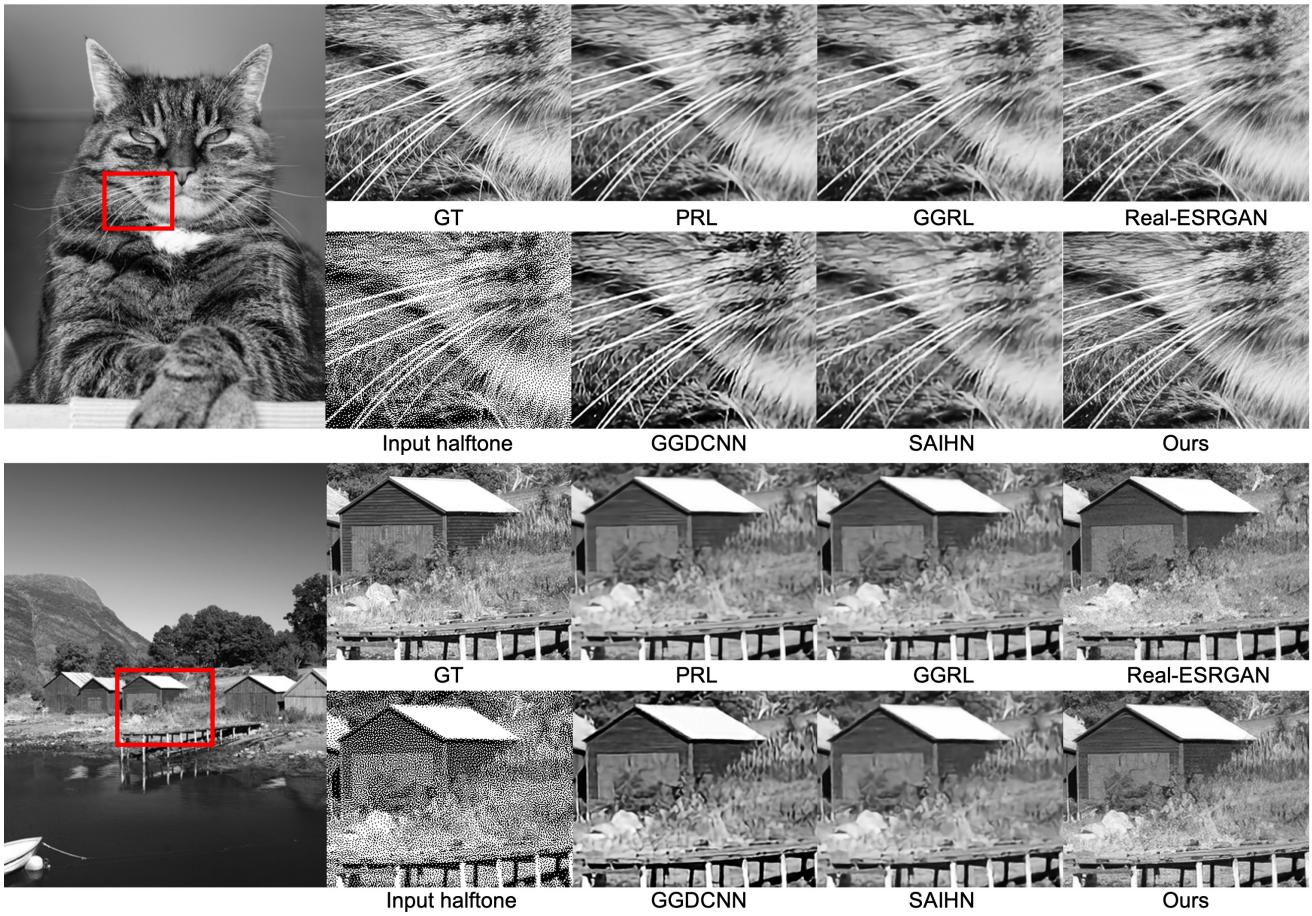

Figure 5. *State-of-the-art comparison. GT stands for ground truth and the halftone generated with DBS is the input used for all descreening algorithms. Note that our algorithm generates descreened images with abundant texture compared to the state-of-the-art descreening algorithms.*

Algorithm\IQ metric	PSNR (↑)	SSIM (↑)	BRISQUE (↓)	LPIPS (↓)	DISTS (↓)
GGDCNN	25.878/25.731	0.875/0.876	32.118/32.802	0.161/0.143	0.147/0.168
GGRL	30.804/30.505	0.925/0.926	37.628/37.136	0.153/0.129	0.144/0.160
PRL	**31.044**/**30.762**	**0.928**/**0.929**	40.350/39.515	0.165/0.140	0.147/0.162
SAIHN	30.420/30.150	0.922/0.923	36.978/36.177	0.153/0.132	0.151/0.168
Real-ESRGAN	27.408/27.500	0.878/0.886	28.177/26.578	0.126/0.107	0.099/0.126
Ours	30.075/29.855	0.914/0.916	**24.176**/**22.841**	**0.078**/**0.064**	**0.075**/**0.103**

Table 1. IQ metric values computed over the validation split of the DIV2K [42] dataset and the Waterloo Exploration Database (WED) [47], computed for each of the state-of-the-art descreening algorithms. The IQ metric values on the left of each cell are for the DIV2K validation set, whereas those on the right of each cell are for the WED dataset. IQ metric values with bold text indicates the best value among the algorithms for each dataset. (↑) indicates that the IQ metric gives higher scores to preferred images, whereas (↓) indicates that the IQ metric gives lower scores to preferred images.

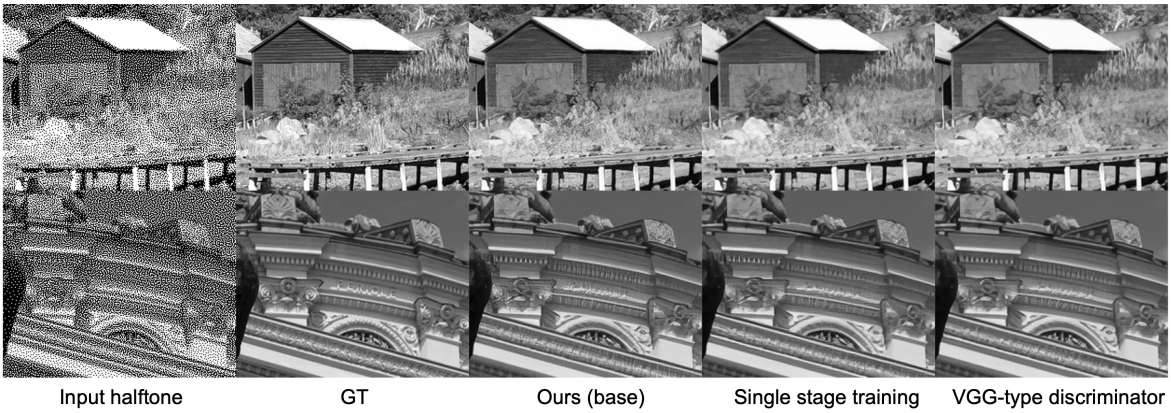

Figure 6. *Qualitative comparison for ablative study. GT stands for ground truth and the halftone generated with DBS is the input used for all descreening algorithms. For single stage training, we train the entire GAN from scratch for 20 epochs. For the version of our GAN with a VGG-type discriminator, other than the discriminator network structure, all the hyperparameters, training procedures, etc. are the same as our base model, which uses a U-net discriminator. The base model is the work described in this paper.*

Algorithm\IQ metric	PSNR (↑)	SSIM (↑)	BRISQUE (↓)	LPIPS (↓)	DISTS (↓)
Ours (base)	30.075	0.914	**24.176**	**0.078**	**0.075**
Single stage training	30.245	**0.917**	24.767	0.087	0.082
VGG-type discriminator	**30.280**	0.915	39.217	0.096	0.082

Table 2. IQ metric values computed over the validation split of the DIV2K [42] dataset, computed for different variants of our descreening algorithm. IQ metric values with bold text indicates the best value among the variants. (↑) indicates that the IQ metric gives higher scores to preferred images, whereas (↓) indicates that the IQ metric gives lower scores to preferred images.

algorithm significantly outperforms the state-of-the-art descreening algorithms.

Ablative study

We also investigate the effectiveness of two aspects of our descreening algorithm. First, we look into whether we could train the entire GAN from scratch instead of performing two-step training. We expect that the first step of our training procedure that trains only the generator gives a good starting point for the GAN and results in descreened images with better quality. Next, we compare the GANs using a U-net [30] discriminator and a VGG-type discriminator. Since a U-net discriminator enjoys a larger model capacity and can provide gradients to the generator to learn to generate images with realistic local texture [41], we expect that using a U-net discriminator for the GAN will result in better quality descreened images.

Figure 6 compares the different variations of our descreening algorithm discussed previously. While the differences are subtle, we can observe that there in fact is a benefit to both using two-step training and a U-net discriminator. This observation is supplemented by the image quality metrics in Table 2, where our base algorithm using both two-stage training and a U-net discriminator outperforms the variants. The difference is especially notable with the LPIPS [48] and DISTS [53] metrics for which the base algorithm yields values that are on average 14.75% and 8.54% lower than those yielded by the two variants.

Conclusions

In this paper, we proposed a GAN-based descreening algorithm that significantly outperforms the state-of-the-art descreening algorithms. We first observed that the current state-of-the-art descreening algorithms are trained with PSNR-oriented reconstruction loss terms and result in generating images that lack texture. Instead, we proposed using a GAN to generate descreened images with realistic texture. Furthermore, we observed that the current literature on descreening generally uses the Floyd-Steinberg error diffusion algorithm to generate the training dataset, but this is not ideal since halftone images generated with Floyd-Steinberg error diffusion tend to include visible artifacts, especially where the continuous-tone image is smooth. To address this issue, we proposed using the direct binary search (DBS) algorithm instead to generate the training dataset. Qualitative and quantitative comparison of the descreened images generated with our algorithm with the descreened images from baseline algorithms showed that our algorithm indeed significantly outperforms the state-of-the-art descreening algorithms.

References

[1] H. Prasetyo and B. Harjito, "Halftoning-based block truncation coding feature on scrambled images," in *IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, pp. 169–170, 2017.

[2] F. Wang and J. P. Allebach, "Printed image watermarking using direct binary search halftoning," in *IEEE International Conference on Image Processing (ICIP)*, pp. 2727–2731, 2016.

[3] J. Liu and J. Allebach, "Screen-based watermarking of aperiodic, clustered-dot halftones," in *IEEE International Conference on Image Processing (ICIP)*, pp. 1384–1388, 2021.

[4] T. Remez, O. Litany, and A. Bronstein, "A picture is worth a billion bits: Real-time image reconstruction from dense binary threshold pixels," in *IEEE International Conference on Computational Photography (ICCP)*, pp. 1–9, 2016.

[5] Y. Xiao, C. Pan, X. Zhu, H. Jiang, and Y. Zheng, "Deep neural inverse halftoning," in *IEEE International Conference on Virtual Reality and Visualization (ICVRV)*, 2017.

[6] M. Xia and T.-T. Wong, "Deep inverse halftoning via progressively residual learning," in *Asian Conference on Computer Vision (ACCV)*, 2018.

[7] Y. Xiao, C. Pan, Y. Zheng, X. Zhu, Z. Qin, and J. Yuan, "Gradient-guided DCNN for inverse halftoning and image expanding," in *Asian Conference on Computer Vision (ACCV)*, 2018.

[8] J. Yuan, C. Pan, Y. Zheng, X. Zhu, Z. Qin, and Y. Xiao, "Gradient-guided residual learning for inverse halftoning and image expanding," *IEEE Access*, vol. 8, 2019.

[9] H. P. A. Wicaksono, H. Prasetyo, and J.-M. Guo, "Deep learning based inverse halftoning via stationary wavelet domain," in *27th IEEE International Conference on Telecommunications (ICT)*, 2020.

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *European Conference on Computer Vision Workshops (ECCVW)*, September 2018.

[12] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial greyscale," in *Proc. S.I.D.*, vol. 17, pp. 75–77, 1976.

[13] R. Eschbach and K. T. Knox, "Error-diffusion algorithm with edge enhancement," *Journal of the Optical Society of America A*, vol. 8, no. 12, pp. 1844–1850, 1991.

[14] H. R. Kang, *Digital Color Halftoning*. Bellingham, WA, USA: SPIE Press, 1999.

[15] D. J. Lieberman and J. P. Allebach, "A dual interpretation for direct binary search and its implications for tone reproduction and texture quality," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1950–1963, 2000.

[16] T. Frank, O. Haik, A. Jumabayeva, J. P. Allebach, and Y. Yitzhaky, "New design for compact color screen sets for high-end digital color press," *IEEE Transactions on Image Processing*, vol. 29, pp. 3023–3038, 2020.

[17] R. Näsänen, "Visibility of halftone dot textures," *IEEE Trans. on Syst., Man, and Cybernetics*, vol. SMC-14, no. 6, pp. 920–924, 1984.

[18] J.-R. Liao, "Theoretical bounds of direct binary search halftoning," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3478–3487, 2015.

[19] J.-M. Guo and S. Sankarasrinivasan, "H-GAN: Deep learning model for halftoning and its reconstruction," in *IEEE International Conference on Consumer Electronics (ICCE)*, 2020.

[20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[21] M. Xia, W. Hu, X. Liu, and T.-T. Wong, "Deep halftoning with reversible binary pattern," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[22] B. Choi and J. P. Allebach, "Mimicking DBS halftoning via a deep learning approach," in *Proc. 27th Color Imag., Displaying, Process., Hardcopy, Appl. IS&T*, 2022.

[23] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[24] A. L. Mass, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *International Conference on Machine Learning (ICML)*, 2013.

[25] R. Stevenson, "Inverse halftoning via map estimation," *IEEE Transactions on Image Processing*, vol. 6, no. 4, pp. 574–583, 1997.

[26] M. Mese and P. Vaidyanathan, "Look-up table (LUT) method for inverse halftoning," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1566–1578, 2001.

[27] K.-L. Chung and S.-T. Wu, "Inverse halftoning algorithm using edge-based lookup table approach," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1583–1589, 2005.

[28] Z. Xiong, M. Orchard, and K. Ramchandran, "Inverse halftoning using wavelets," *IEEE Transactions on Image Processing*, vol. 8, no. 10, pp. 1479–1483, 1999.

[29] X. Hou and G. Qiu, "Image companding and inverse halftoning using deep convolutional neural networks," *arXiv preprint arXiv:1707.00116*, 2017.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, 2015.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[32] Q. Gao, X. Shu, and X. Wu, "Deep restoration of vintage photographs from scanned halftone prints," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4119–4128, 2019.

[33] T.-H. Kim and S. I. Park, "Deep context-aware descreening and rescreening of halftone images," *ACM Transactions on Graphics*, vol. 37, no. 4, 2018.

[34] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[35] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[36] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *European Conference on Computer Vision (ECCV)*, September 2018.

[37] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *IEEE/CVF Conference on Computer Vision*

and Pattern Recognition (CVPR), 2017.

[38] A. Lugmayr, M. Danelljan, L. Van Gool, and R. Timofte, "SRFlow: Learning the super-resolution space with normalizing flow," in *European Conference on Computer Vision (ECCV)*, pp. 715–732, 2020.

[39] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, 2018.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[41] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-esrgan: Training real-world blind super-resolution with pure synthetic data," in *International Conference on Computer Vision Workshops (ICCVW)*, 2021.

[42] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: dataset and study," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[43] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision (ECCV)*, 2016.

[44] J. H. Lim and J. C. Ye, "Geometric GAN," *arXiv preprint arXiv:1705.02894*, 2017.

[45] A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," in *International Conference on Learning Representations (ICLR)*, 2019.

[46] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[47] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2017.

[48] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[49] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[50] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[51] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*, 1996.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, p. 84–90, may 2017.

[53] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2567–2581, 2022.

Author Biography

Baekdu Choi received his B.Sc. in electrical and computer engineering from Seoul National University, Seoul, South Korea in 2017 and is currently working on a Ph.D. in electrical and computer engineering at Purdue University, West Lafayette, IN, USA. His research mainly focuses on digital image processing, digital halftoning and color management for inkjet printers.

Jan P. Allebach is Hewlett-Packard Distinguished Professor of Electrical and Computer Engineering at Purdue University. Allebach is a Fellow of the IEEE, the National Academy of Inventors, the Society for Imaging Science and Technology (IS&T), and SPIE. He was named Electronic Imaging Scientist of the Year by IS&T and SPIE, and was named Honorary Member of IS&T, the highest award that IS&T bestows. He has received the IEEE Daniel E. Noble Award, the IS&T/OSA Edwin Land Medal, and the IS&T Johann Gutenberg Prize. He is a member of the National Academy of Engineering.