Comparative Analysis of Inverse Halftoning Techniques

Hana Darling-Wolf, Jeffrey Qiu

Abstract—Halftoning, or dithering, is an image compression technique that has long been used in the print industry to reproduce tone with a limited colour palette (e.g. black and white). Inverse halftoning, on the other hand, seeks to retrieve continuous-tone images from halftone images. However, these algorithms can be energy intensive. Therefore, this paper seeks to determine which inverse halftoning algorithms are most "worth" their computational cost. To this end, we evaluate six inverse halftoning techniques based on both their reconstructed image quality (measured via PSNR and SSIM), and their computational efficiency (estimated by runtime). We find that low-pass filtering, Mese and Vaidyanathan's lookup table method [1], and Xia et. al's deep inverse technique [2] achieve the best ratio of image quality to cost. However, there is a trade-off between reconstructed image quality and runtime. These findings can help guide algorithm decisions for inverse halftoning methods which best suit project requirements.

Index Terms—Computational Photography, Inverse Halftoning, Dithering, Denoising

1 Introduction

Halftoning, or dithering, is an image compression technique that reproduces tone with a limited colour palette (e.g. black and white) via the distribution of halftone dots [2], [3]. Although halftoning typically implies the use of only two tones (e.g. black and white), a greater number of tones can be used to create a compressed colour image, or adjust the level of compression and detail. Dithering has long been used in the print industry in order to cut down on ink and cost [3]. Today, environmentally conscious web designers are also turning to dithering to radically reduce the energy consumption associated with accessing online content [4].

Inverse halftoning looks to retrieve continuous-tone images from these halftoned images. Reverting halftone images to continuous-tone images is useful not only for the recovery and preservation of historical printed media, but for various image editing and processing that is not possible on halftone images without serious image degradation [2], [5]. Indeed, even simple manipulation — such as rotation by an angle other than 90 degrees — or compression becomes difficult on haltone images [5]. However, these inverse processes can be energy intensive. It is therefore desirable to find inverse halftoning techniques which generate high quality reconstructed images that are also computationally efficient. In this paper, we explore various techniques for restoring dithered images to a continuous-tone image. In particular, we evaluate the effectiveness of these techniques given different levels of compression (the number of tones used in our dithered images) as well as the computational cost.

2 Background and Related Work

Dithered images have a very low signal-to-noise ratio (SNR) relative to their respective grayscale image due to their low bit-depth [5]. There are several types of dithering methods, which must be dealt with in different ways. Due to the popularity of Floyd-Steinberg error diffusion, we chose to focus on error diffused halftones [2]. In error diffused halftones, most noise lies in the high spatial frequencies. Therefore, the simplest method for removing dither is applying a low-pass filter. However, this results in a loss of edge information, and is prone to blur and artifacts [2]. Inverse halftoning therefore typically aims to remove this high frequency noise, known as blue noise, while retaining the original image features [5]. Some algorithms also seek to specifically enhance edge preservation [6].

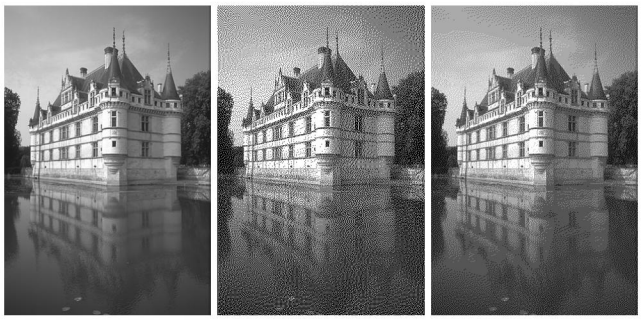
More complex techniques for inverse halftoning have also been proposed [1], [2], [6], [7], [8]. One of the most promising techniques is Mese and Vaidyanathan's use of a pre-computed lookup table (LUT) which improved both reconstruction accuracy and efficiency in comparison to previous implementations using iterative filtering or projection onto a convex set (POCS) [1]. More recently, neural networks and deep learning have been applied to the problem of inverse halftoning, with Xia and Wong achieving a state-of-the-art performance using progressive residual learning [2].

Xia et al. have also implemented a technique for reversible halftoning — via encoding information about the colour and fine details of the original image in the distribution of the halftone dots in the dithered image — which permits for the restorability of the original colour image from a dithered image [6]. However, because the images we wish to recover are typically not produced using reversible halftoning, our focus in this project remains on methods for retrieving images from traditional, non-reversible dithering.

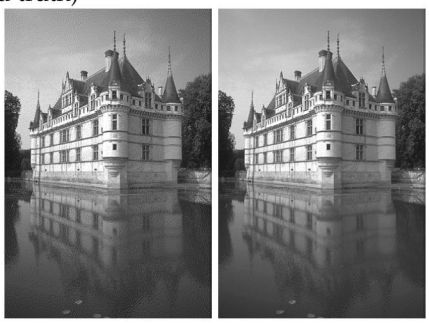
3 Theory

3.1 Dithering

We first consider the case of dithering for 2-tone (halftone) images. We define a 2-tone image as an image whose pixel values are constrained to two possible values: black or white. Thus, each pixel value in a 2-tone image can be encoded using a single bit: $\{0, 1\}$. Given a continuous-tone



(a) Grayscale Image (ground truth) (b) 2-tone Dithered Image (c) 4-tone Dithered Image



(d) 8-tone Dithered Image (e) 16-tone Dithered Image

Fig. 1: Results of Floyd-Steinberg dithering at different compression levels (n-tones). If viewing digitally, zoom in to see details.

image, dithering returns a 2-tone image that approximates the continuous pixels values of the original image by the distribution of black-and-white pixels. In practice, pixels of continuous-tone images are stored as discrete values, such as in GIF or PNG formats. We use 8-bit grayscale images (256-tone) to approximate continuous-tone images. Dithering — and inverse dithering — can be extended to colour images, for example, by applying the operation across each RGB channel separately.

There are many existing algorithms to perform the dithering operation. In this project, we employ the widely-used Floyd-Steinberg error diffusion [9] as our dithering operation. In contrast to simple thresholding, error diffusion introduces intentional noise to the halftone image to reduce banding artifacts. Figure 1b a shows a 2-tone dithered image obtained from a continuous-tone (8-bit grayscale) image.

We also implement the extension of dithering for general n-tone images. Figures 1c and 1d provide an example of 4-tone and 8-tone dithered images, respectively. In 4-tone images, each pixel can take on four discrete values and can be encoded by 2 bits: $\{00, 01, 10, 11\}$. In general, n-tone images compress each pixel into $\log_2 n$ bits, with higher n resulting in closer similarity to the continuous-tone case.

3.2 Inverse Halftoning

We now consider the *inverse* problem: given only a halftone image, how can we recover the original continuous-tone image? Intuitively, the density of pixel intensities in a given region reflects the continuous-tone pixel values in that region. In this project, we examine two approaches specifically designed for the problem of inverse halftoning: lookup tables and deep learning.

3.2.1 *Lookup Table*

The keys of a lookup table can describe halftone patterns found in image patches. The corresponding values in the LUT would then be the corresponding continuous-tone pixel values. Using such a table, we can iterate through each pixel of a halftone image, look at its neighbouring pixel values, and then estimate the continuous-tone pixel value by referencing the same neighbourhood pattern in our table.

We use the LUT method proposed in [1]. Prior to using the LUT for inferencing, we must populate the values of the table through a training step. Given a training set of halftone images and their corresponding continuous-tone (grayscale) ground truth images, the training step is as follows:

1) For each halftone image and its ground truth image, iterate through each pixel x_i in the halftone image and look at the local 4x4 image patch surrounding x_i. We call this image patch a pattern $P_i \in \{0, 1\}^{16}$.
2) For the key P_i in the LUT, store the average corresponding ground truth pixel value $LUT[P_i] = \mathbb{E}[x_i']$.

To use the lookup table for image reconstruction, we iterate through each pixel x_i in the halftone image and read off the LUT values:

1) Look at the local 4x4 image patch surrounding x_i. This image patch is a pattern $P_i \in \{0, 1\}^{16}$.
2) Find the value in the LUT corresponding to the key P_i. Call this value $LUT[P_i]$.
3) If P_i is in the LUT, set the reconstructed pixel $\hat{x}_i = LUT[P_i]$
4) Otherwise, if P_i is not in the LUT, we have a *nonexistent pattern*. In the case of a nonexistent pattern, we set $\hat{x}_i$ to be a Gaussian-weighted average of the pixels surrounding x_i.

Provided that the LUT is sufficiently trained, the occurrence of nonexistent patterns should be infrequent, as the 'typical' patterns are encoded in the LUT during training.

3.2.2 *Deep Learning*

Deep Inverse Halftoning proposed in [2] is a state-of-the-art method that uses a convolutional neural network (CNN) trained on halftone images in order to perform continuous-tone reconstruction. The model features two modules that uniquely tackle the inverse halftoning problem: (i) a Content Aggregation network to remove halftone patterns and recover continuous-tone values, and (ii) a Detail Enhancement network that restores lost features.

3.3 Image Denoising

The dithering operation, such as with Floyd-Steinberg error diffusion, can be viewed as the intentional addition of noise to an image. Consequently, we can frame the inverse halftoning problem as that of image denoising. We apply four general image denoising approaches of interest: low-pass filtering, bilateral filtering, alternating direction method of multipliers (ADMM) with total variation (TV), and ADMM with a denoising convolutional neural network (DnCNN).

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Fig. 2: A 5x5 Gaussian blur kernel for low pass filtering.

3.3.1 Low-pass Filtering

Applying a low-pass filter to an image attenuates high-frequency image details, such as sharp edges, effectively blurring the image. In the spatial domain, one way to achieve low-pass filtering is to convolve the image with a Gaussian blur kernel, e.g., Fig. 2. These convolutions can be performed efficiently as point-wise multiplication in the Fourier domain. When applied to halftone images, the low pass filter effectively calculates a Gaussian-weighted spatial average of neighbouring pixel values to estimate a continuous-tone value.

3.3.2 Bilateral Filter

The bilateral filter combines the spatial blurring of the low-pass filter with an additional weighting function to preserve image edges. The filter weight assigned to pixel x to reconstruct pixel x_i is given by

$$w(x, x_i) \propto f(\|I(x) - I(x_i)\|)G(\|x - x_i\|), \tag{1}$$

where G is a smoothing kernel, such as the Gaussian function depicted in Fig. 2. The function f is also a smoothing kernel, except its value is calculated using the difference between $I(x)$, the intensity of pixel x, and $I(x_i)$, the intensity of pixel x_i.

3.3.3 ADMM-TV

ADMM is a powerful iterative method for solving convex optimization problems [10]. We can formulate our inverse problem as

$$\min_x \quad \frac{1}{2}\|Ax - b\|_2^2 + \lambda\Psi(x), \tag{2}$$

where A is the image formation model, b is the observed image, and Ψ is a regularization operator.

For inverse halftoning, we wish to recover a continuous-tone image x from a dithered image b. We set A to be identity and Ψ be a total variation prior, which penalizes reconstructions x that have large absolute gradients.

There are two important parameters under our control when using ADMM-TV. The first is λ, which appears in Equation (2) to control the strength of the TV prior. The second is a penalty parameter ρ, which the ADMM approach introduces to aid convergence.

3.3.4 ADMM-DnCNN

The optimization problem in Equation (2) can also be solved using ADMM with a general denoising prior, such as non-local means. Here, we consider a denoising convolutional neural network to be used in iterations of ADMM. We use the pre-trained DnCNN model from [11], [12].

TABLE 1: Low-pass Hyperparameter Search Results

	Tone		
	2	4	16
σ	1	1	0.5

TABLE 2: Bilateral Filter Hyperparameter Search Results

	Tone		
	2	4	16
$\sigma_{spatial}$	2	1	1
$\sigma_{intensity}$	4	1	0.1

TABLE 3: ADMM-TV Hyperparameter Search Results

	Tone		
	2	4	16
λ	0.3	0.075	0.01
ρ	0.01	0.01	0.01

4 Analysis and Evaluation

4.1 Dataset

The images used in our evaluation of inverse halftoning techniques come from the BSDS300 dataset [13], which consists of a training set of 200 images and a testing set of 100 images. For each image, we take its grayscale version as the ground truth image. We then apply Floyd-Steinberg error diffusion (Sec. 3.1) on the grayscale images to dither them, creating 'noisy' images. For the purposes of evaluation, we generated image sets corresponding to three different compression levels: 2-tone, 4-tone, and 16-tone.

4.2 Implementation of Inverse Halftoning

Each technique is evaluated on a test set of 100 2-tone images, and their reconstructions are compared to the ground truth. Furthermore, low-pass filtering, bilateral filtering, ADMM-TV, and ADMM-DnCNN are also tested on test sets of 100 4-tone images and 100 16-tone images. Results are presented in Sec. 5.

Implementations are all done according to Sec. 3. For our implementation of the low-pass filter, we used FFT to perform the filtering operation in the Fourier domain. In the spatial domain, the filter size is set to $\lceil 9\sigma \rceil$, where σ is the standard deviation of our Gaussian kernel. To determine optimal values for the parameter σ, we performed a search (Appendix Fig. 7) over possible values. For the bilateral filter, we use a Gaussian function for both the spatial and intensity kernels. As such, there are two parameters $\sigma_{spatial}$ and $\sigma_{intensity}$ that we consider (Appendix Fig. 8).

In ADMM-TV, we optimized for the parameters λ and ρ. Heatmaps of the hyperparameter search are presented in Appendix Fig. 9. In our application of ADMM-DnCNN, we noticed improved reconstructed image quality if we modify the σ for the blur kernel in the x-update step of ADMM. Our hyperparameter search (Appendix Fig. 10) is over σ_{blur} and ρ. Hyperparameter search results for each technique are presented in Tables 1-4.

We implement the LUT algorithm introduced by Mese et Al. [1], training it on our set of 200 training images. This

TABLE 4: ADMM-DnCNN Filter Hyperparameter Search Results

	Tone		
	2	4	16
σ_{blur}	1.2	0.5	0.3
ρ	0.1	0.5	0.2

implementation is restricted to 2-tone images. Beyond 2-tone images, the size of the typical set of patterns rapidly expands. Thus, a sufficiently trained LUT becomes increasingly difficult to achieve, and its size becomes intractable. As the occurrence of nonexistent patterns increases, the output approaches that of a low-pass filter.

For the neural network model, we used the pre-trained Deep Inverse model available at https://github.com/MenghanXia/InverseHalftoning. Since this was pre-trained on halftone images only, its application is also limited to that of halftone images. In our testing, we observed that the reconstructed image quality quickly decreases as we increase the number of tones. For future investigations, we may consider retraining this model for application beyond 2-tone images, and exploring the effects of transfer learning for inverse dithering networks.

4.3 Image Quality and Runtime Evaluation

To compare image quality across our reconstructed images, we use two metrics: the peak signal-to-noise-ratio (PSNR) and the structural similarity index measure (SSIM) [14]. In image processing, PSNR is a standard measure for the quality of image reconstructions. For our project, we are particularly interested in *perceived* image quality, since the dithering operation aims to improve upon the perceived image quality. For example, consider a grayscale image and a corresponding 2-tone image. The highest possible PSNR would be achieved if the 2-tone image was generated by applying a simple thresholding function to the grayscale image — this minimizes the pixel-wise intensity difference. Dithering, therefore, *lowers* the PSNR of the 2-tone image. Nonetheless, the dithered image is often perceived to be more similar to the original and of higher image quality.

In addition to image quality, we compare the computational cost of each technique. We measure the per-image runtime of each technique in order to compare their computational costs. Since the runtimes are heavily machine-dependent, all reported runtimes in our results were measured on the same computer, a 2.40GHz Intel Core i7-5500U CPU. These runtimes suffice to be used for relative comparisons.

5 Results

For each method — and where possible, each compression level — we generated a new dataset of images, and calculated the average PSNR, SSIM, and runtime per image (see Table 5).

5.1 Quantitative Results

In order to compare results, we generated two plots (Figures 3 and 4). Fig. 3 compares the quantitative image reconstruction quality of each method for each level of compression

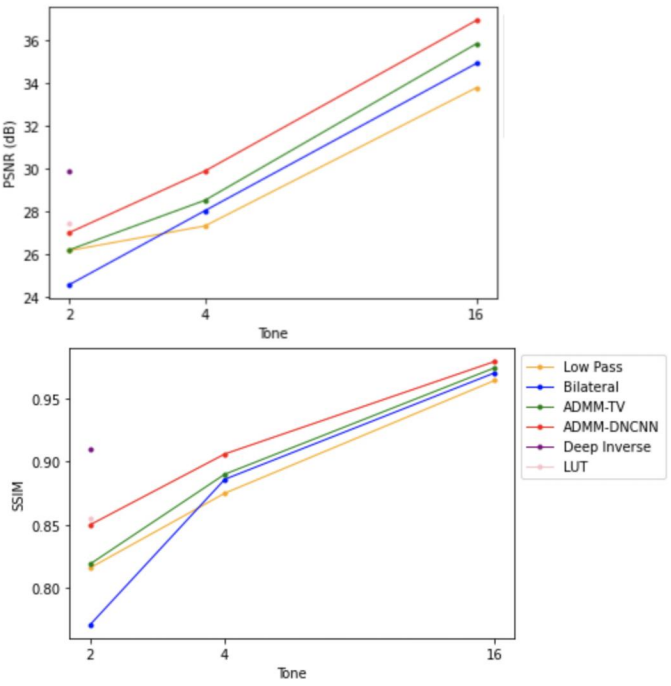


Fig. 3: Reconstructed Image Quality for each level of compression (2-tone, 4-tone, 16-tone)

(2-tone, 4-tone, and 16-tone). Fig. 4 plots the performance of each method in terms of both reconstructed image quality and runtime.

As Fig. 3 illustrates, Deep Inverse method performed the best in terms of reconstructed image quality for 2-tone dithered images (see also Table 5). The lookup table also performed well, and required half the runtime (Fig. 4, Table 5). ADMM-DnCNN achieved an average PSNR and SSIM on par with the lookup table, but was very computationally inefficient compared to the other methods tested (it was therefore omitted from Fig. 4).

For 2-tone images, there is a clear tradeoff between image reconstruction quality and computational cost. As we see in Fig. 4, three methods in particular have a strong image quality to runtime ratio: the deep learning technique, the lookup table, and the low-pass filter. These appear to share a linear relationship and demonstrate how directly both PSNR and SSIM are related to computational cost. It is not surprising that these methods perform well on 2-tone images, seeing as these are methods that have been trained for this purpose, or cited as strong solutions for inverse halftoning [2], [6], [8]. The other more general denoising methods we tested do not perform to the same standard for 2-tone images.

However, as we increase the number of tones (decrease the level of compression), the reconstructed image quality of the low-pass, bilateral, ADMM-TV, and ADMM-DnCNN methods improves significantly (Fig. 3). In fact, for 16-tone images, the SSIM for these techniques all converge to a value above 0.95. Indeed, for lower compression levels, the low-pass filter gives us strong results at a fraction of the runtime of other methods (see Fig. 4). Moreover, the low-pass filter (and the other general denoising techniques) do not require retraining for dealing with a larger number of tones (unlike

TABLE 5: Quantitative Results

	2-tone			4-tone			16-tone		
	Average PSNR (dB)	Average SSIM	Average Runtime (s)	Average PSNR (dB)	Average SSIM	Average Runtime (s)	Average PSNR (dB)	Average SSIM	Average Runtime (s)
LP	26.188	0.816	0.059	27.335	0.875	0.066	33.777	0.964	0.059
Bilateral	24.596	0.771	5.016	28.048	0.886	5.934	34.911	0.97	5.523
ADMM -TV	26.22	0.819	17.652	28.531	0.89	17.452	35.826	0.974	16.378
ADMM -DnCNN	27.023	0.85	> 60	29.895	0.096	> 60	36.916	0.979	> 60
Deep Inverse	29.90	0.910	7.65	-	-	-	-	-	-
LUT	27.477	0.855	3.28	-	-	-	-	-	-

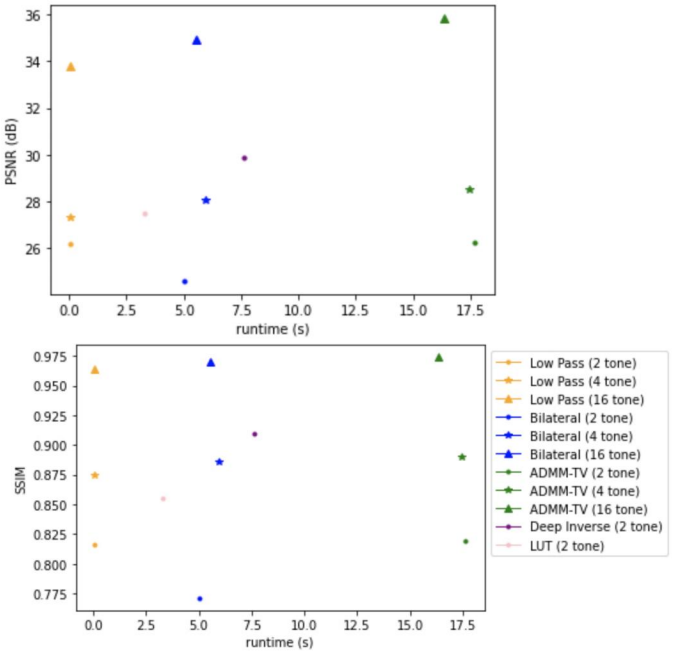


Fig. 4: Reconstructed Image Quality (PSNR and SSIM) vs. runtime (s)

the deep learning and lookup table approaches), and can be applied directly at the same computational cost.

5.2 Qualitative Results

In order to evaluate our qualitative results, we examine an example output for each method tested (see Fig. 5, Fig. 6). We chose an image which features both details and flat areas that are prone to revealing artefacts. As we can see in Figures 5 and 6, our qualitative results generally agree with our quantitative data. The low-pass filter, bilateral filter, ADMM-DnCNN, and ADMM-TV all produce compelling results for dithered images with sixteen tones. Interestingly, there is not much difference perceptually between the ADMM-TV applied on four tones and sixteen tones.

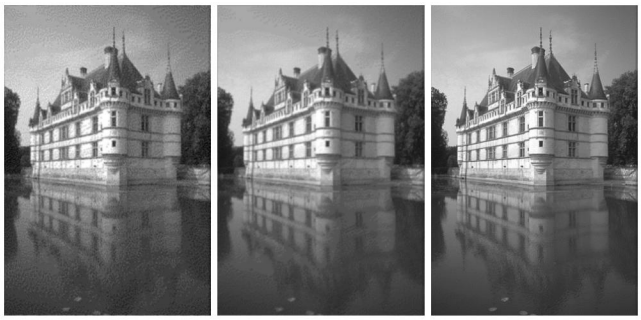
For the methods applied on 2-tone dithered images, deep inverse produces the best qualitative results. Although the lookup table actually achieves a higher PSNR value (27.395 dB) for this image, the deep inverse algorithm performs better in terms of SSIM, which lines up with our perceptual evaluation of the images. ADMM-DnCNN also performs well in terms of both perceived image quality and SSIM. All the other methods retain a significant amount of artefacts (including the lookup table). Although the lookup table seems to remove a greater number of dithering artefacts than the low-pass filter, it is not clear whether it significantly out-performs the low-pass filter in terms of qualitative results. There is also little perceptible difference between the low-pass filter and ADMM-TV, and this is reflected in the PSNR and SSIM.

6 Discussion

These results are useful to help guide people in choosing both inverse halftoning algorithms, and the level of compression at which to dither images. In particular, Fig. 4 can be used as a resource to determine which inverse halftoning method is most appropriate given the demands. For example, if your goal is to retrieve a high quality reconstructed image, the deep learning approach might be best suited. However, if you goal is to convert the image to a continuous tone image for editing purposes before returning it to a dithered image, applying a simple low-pass filter might be sufficient. On the other hand, if you are seeking to dither an image (e.g. to use on your a web page) and want a low-cost way of retrieving the image (or expect to inverse halftone frequently), it makes more sense to use produce a higher tone image (e.g. sixteen tones) which can be quickly retrieved using a low-pass filter.

Our findings can also inform future research, suggesting that new inverse halftoning algorithms should focus on achieving both higher quality images and faster runtimes than the deep inverse technique, the lookup table, and the low-pass filter (that is, they should fall above the line formed by these three methods in Fig. 4).

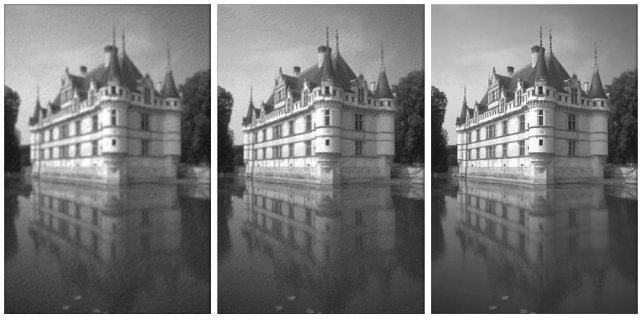
Future work could also build on this project by using alternative measures of the computational cost (such as number of floating point operations), and considering the cost of training each algorithm. We were also unable to retrain the deep inverse algorithm given the time constrains for this project. Future work could compare the performance of this algorithm to other denoising techniques at lower compression levels. Finally, additional algorithms and levels of compression could be evaluated.



(a) Low Pass (2-tone) PSNR: 25.926 dB SSIM: 0.833

(b) Low Pass (4-tone) PSNR: 26.595 dB SSIM: 0.9

(c) Low Pass (16-tone) PSNR: 33.418 dB SSIM: 0.96



(d) Bilateral (2-tone) PSNR: 24.117 dB SSIM: 0.798

(e) Bilateral (4-tone) PSNR: 26.539 dB SSIM: 0.897

(f) Bilateral (16-tone) PSNR: 35.744 dB SSIM: 0.975

(g) ADMM-TV (2-tone) PSNR: 25.997 dB SSIM: 0.842

(h) ADMM-TV (4-tone) PSNR: 28.81 dB SSIM: 0.913

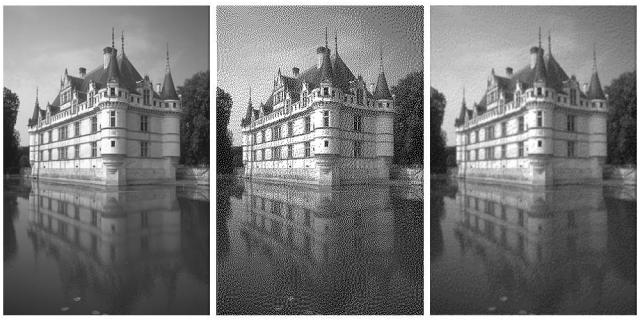
(i) ADMM-TV (16-tone) PSNR: 36.601 dB SSIM: 0.974

(j) ADMM-DnCNN (2-tone) PSNR: 26.838 dB SSIM: 0.904

(k) ADMM-DnCNN (4-tone) PSNR: 30.583 dB SSIM: 0.937

(l) ADMM-DnCNN (16-tone) PSNR: 37.341 dB SSIM: 0.977

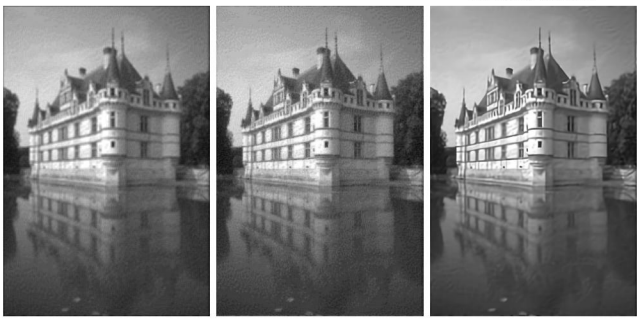
Fig. 5: Reconstructed Image Low Pass Filter, Bilateral Filter, ADMM-TV, and ADMM-DnCNN on 3 Compression Levels



(a) Grayscale Image (ground truth)

(b) 2-tone Dithered Image

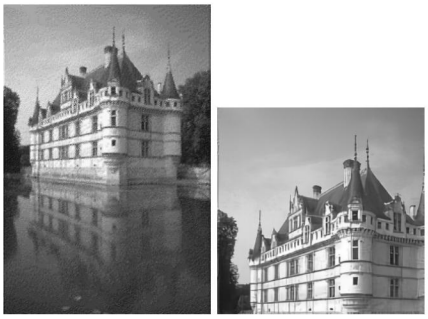
(c) Low Pass (2-tone) PSNR: 25.926 dB SSIM: 0.833



(d) Bilateral (2-tone) PSNR: 24.117 dB SSIM: 0.798

(e) ADMM-TV (2-tone) PSNR: 25.997 dB SSIM: 0.842

(f) ADMM-DnCNN (2-tone) PSNR: 26.838 dB SSIM: 0.904



(g) LUT (2-tone) PSNR: 27.395 dB SSIM: 0.896

(h) Deep Inverse (2-tone) PSNR: 26.929 dB SSIM: 0.942

Fig. 6: Reconstructed Image from 2-tone Dithered for All Methods

7 Conclusion

We have evaluated six inverse halftoning methods in terms of both recovered image quality and computational cost. For images with only two tones (black and white), there is a clear tradeoff between quality and efficiency. The algorithms which achieved the best quality (measured in terms of PSNR or SSIM) to cost (taken as the runtime) ratio were the low-pass filter, the lookup table, and Xia et al.'s deep inverse technique [2]. However, performance for the other denoising techniques — that is, bilateral filter, ADMM-TV, and ADMM-DnCNN — improved significantly for lower compression levels (4-tone and 16-tone). The low-pass filter also performed very well on 16-tone images, and was by far

the most computationally efficient. This suggests that the low-pass filter is the optimal solution for retrieving 16-tone images. These observations (see Figures 3, 4), can help guide individuals deciding which inverse halftoning method to apply given their needs. They can also help determine the ideal level of compression for dithered images, given the frequency with which they will be retrieved, or the purpose of their retrieval. Finally, these results may inform the direction of future research by providing a threshold for improvement that depends on both image quality and computational cost.

REFERENCES

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

[2] M. Xia and T.-T. Wong, "Deep inverse halftoning via progressively residual learning," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 523–539.

[3] R. A. Ulichney, "Dithering with blue noise," *Proceedings of the IEEE*, vol. 76, no. 1, pp. 56–79, 1988.

[4] K. De Decker and V. Grosjean, "Low-tech magazine," *LOW←TECH MAGAZINE https://solar.lowtechmagazine.com/about. html*, 2007.

[5] T. D. Kite, N. Damera-Venkata, B. L. Evans, and A. C. Bovik, "A high quality, fast inverse halftoning algorithm for error diffused halftones," in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*, vol. 2. IEEE, 1998, pp. 59–63.

[6] M. Xia, W. Hu, X. Liu, and T.-T. Wong, "Deep halftoning with reversible binary pattern," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 000–14 009.

[7] W. Chau, S. Wong, and S. Wan, "A critical analysis of dithering algorithms for image processing," in *IEEE TENCON'90: 1990 IEEE Region 10 Conference on Computer and Communication Systems. Conference Proceedings*. IEEE, 1990, pp. 309–313.

[8] M. Mese and P. Vaidyanathan, "Recent advances in digital halftoning and inverse halftoning methods," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, pp. 790–805, 2002.

[9] R. W. Floyd, "l. steinberg,"an adaptive algorithm for spatial grayscale,"," in *Proc. sID*, vol. 17, no. 2, 1976, pp. 75–78.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. [Online]. Available: http://dx.doi.org/10.1561/2200000016

[11] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[12] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

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

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

APPENDIX

Heatmaps for Hyperparameter Search

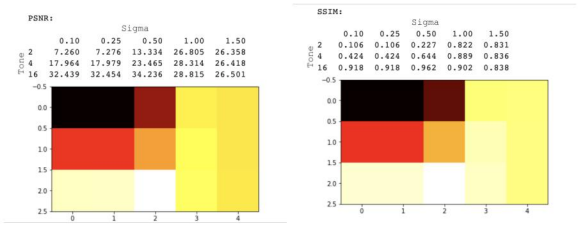


Fig. 7: Hyperparameter Search for Low Pass Filter

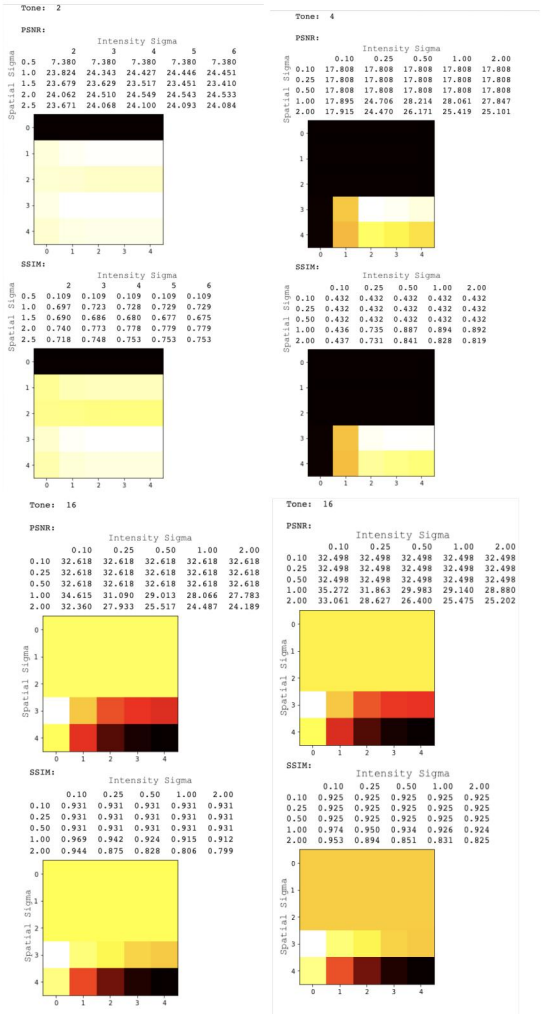


Fig. 8: Hyperparameter Search for bilateral on 2 Tone, 4 Tone, and 16 Tone images

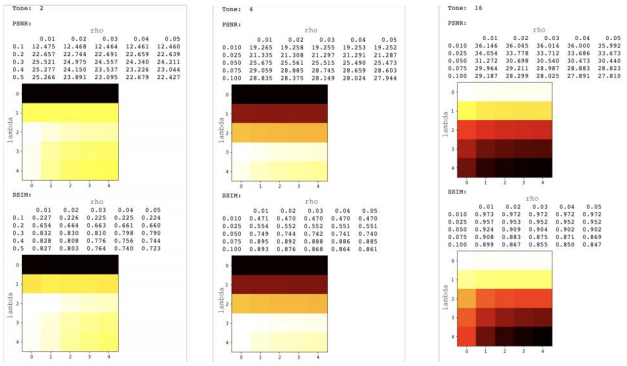


Fig. 9: Hyperparameter Search for ADMM-TV

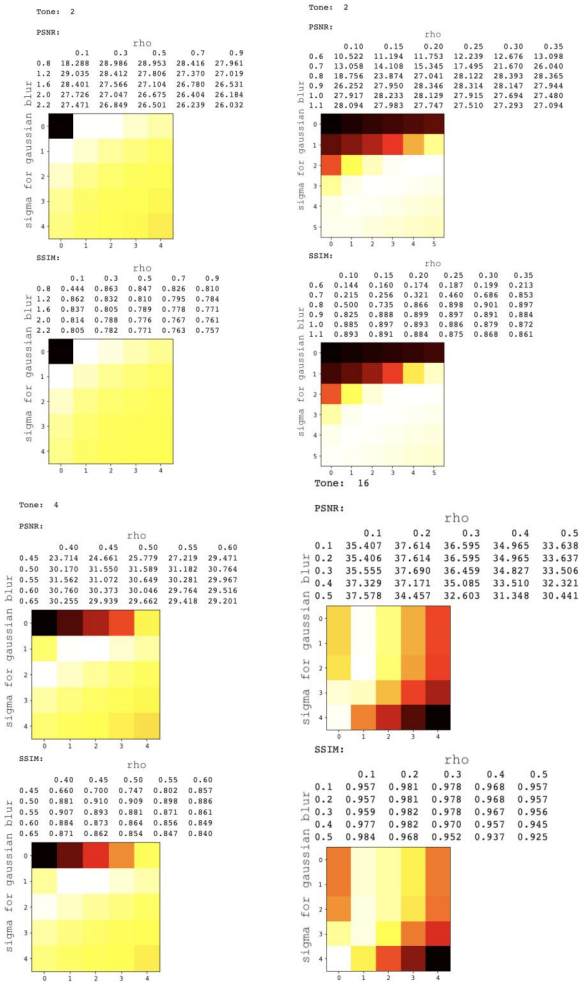


Fig. 10: Hyperparameter Search for ADMM-DnCNN on 2 Tone, 4 Tone, and 16 Tone images