STAG: Smart Tools and Applications in Graphics (2023)
K. Lupinetti, F. Banterle, N. Capece, and U. Erra (Editors)

JPEG Line-drawing Restoration with Masks

Y. Zhu[1] and Y. Yamaguchi[1]

[1]The University of Tokyo, Japan

Abstract

Learning-based JPEG restoration methods usually lack consideration on the visual content of images. Even though these methods achieve satisfying results on photos, the direct application of them on line drawings, which consist of lines and white background, is not suitable. The large area of background in digital line drawings does not contain intensity information and should be constantly white (the maximum brightness). Existing JPEG restoration networks consistently fail to output constant white pixels for the background area. What's worse, training on the background can negatively impact the learning efficiency for areas where texture exists. To tackle these problems, we propose a line-drawing restoration framework that can be applied to existing state-of-the-art restoration networks. Our framework takes existing restoration networks as backbones and processes an input rasterized JPEG line drawing in two steps. First, a proposed mask-predicting network predicts a binary mask which indicates the location of lines and background in the potential undeteriorated line drawing. Then, the mask is concatenated with the input JPEG line drawing and fed into the backbone restoration network, where the conventional L_1 loss is replaced by a masked Mean Square Error (MSE) loss. Besides learning-based mask generation, we also evaluate other direct mask generation methods. Experiments show that our framework with learnt binary masks achieves both better visual quality and better performance on quantitative metrics than the state-of-the-art methods in the task of JPEG line-drawing restoration.

CCS Concepts

• ***Computing methodologies*** → ***Reconstruction;*** *Image processing;* • ***Applied computing*** → *Media arts;*

1. Introduction

A large quantity of line drawings on the internet are uploaded after JPEG compression which is a lossy compression algorithm that introduces noise and artifacts. In this paper, we address the task of restoring JPEG-compressed digital line drawings. Specifically, the purpose is to estimate the latent undegraded line drawing $\mathbf{x}$ produced by painting software, given the JPEG-degraded observation line drawing $\mathbf{x_0} = J(\mathbf{x})$ where J is JPEG compression operation. In the case that $\mathbf{x}$ is photo, the problem can be properly solved by state-of-the-art restoration networks like FBCNN [JZT21].

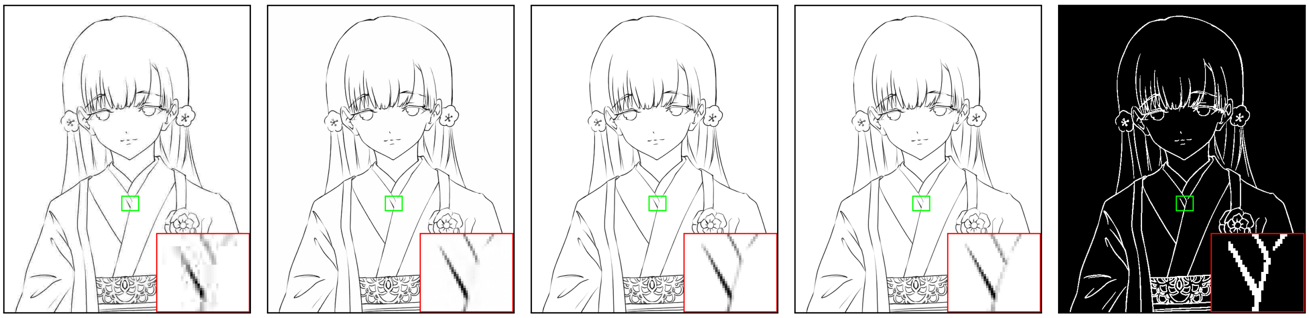
However, unlike photos, line drawings are images that are composed of distinct lines placed against a background. Currently, digital line drawings are created using a painting software like Photoshop[Ado], where artists draw strokes on a white canvas. The strokes usually have a paintbrush texture and the white canvas is plain without any texture or gradations. The strokes and white canvas form the lines and the background respectively in these line drawings. Given that the pixel brightness of a monochromatic line drawing is in the range of $[0,255]$, generally the background brightness is constantly 255 and the line brightness ranges from 0 to 254.

We find that direct use of existing image restoration networks for line drawings mainly have two limitations: (1) Restoration networks usually fail to correctly predict the background pixel brightness, even if the ground-truth brightness value is constantly 255. (2) Background contains no intensity information but occupies the majority of the area, which can affect the training efficiency on lines. (After a statistical analysis of a line-drawing dataset we collect, we find that the number of background pixels accounts for, on average, about 95% of the total line drawing pixels.)

Based on the prior that background brightness of digital line drawings is 255, we propose a thresholding method to tackle the first limitation. The method functions as a post-processing scheme to the output of an arbitrary restoration model. It is simple and fast to implement, effectively removing a large portion of errors in the background and thus improving the performance on quantitative metrics.

To simultaneously tackle the two limitations, we further propose a framework that can be applied to various existing restoration networks. The central idea of this proposed framework is utilizing a binary mask indicating the location of the lines and the background in a line drawing to guide the training. Our framework incorporates the mask into the existing networks by adding an additional input channel, without any modification of the networks' structure. We apply masked Mean Square Error (MSE) loss to guide the networks to optimize solely on the lines, and optimization on the background

DOI: 10.2312/stag.20231299



JPEG (PSNR = 27.32) *FBCNN (PSNR = 32.60)* *FBCNN with our framework & our learnt mask (PSNR = 33.43)* *Ground truth* *Learnt mask*

Figure 1: *Visual comparisons of FBCNN baseline and our framework with learnt mask on a JPEG line drawing from our line-drawing test set with QF = 10. Note the difference of the light lines. Please zoom in for better visualization.*

is removed from the training process. Finally, the output line drawing of the network is masked, with the background being set to 255 according to the prior.

The binary masks in our framework need to be generated from JPEG line drawings. While some conventional thresholding methods like traditional adaptive thresholding and Otsu's method [Ots79] exist, we propose a learning-based method for generating high-accuracy masks. The idea of our method is to view mask generation as a pixel-wise segmentation task, where each pixel in a JPEG line drawing is assigned to two predefined classes, namely line and background. In our method, we propose a mask-predicting network which takes as input a JPEG line drawing and outputs its corresponding binary mask.

We evaluate our thresholding method and our framework combined with different mask generation methods on a line drawing dataset collected from Pixiv [Pix]. We take three state-of-the-art restoration networks, namely FBCNN [JZT21], DRUnet [ZLZ*21], and Swinir [LCS*21], as baselines and make comprehensive comparisons. Quantitatively, both our thresholding method and framework with learnt masks achieve better performance than baselines in terms of PSNR and SSIM. We also show that our framework can improve the visual quality of restored line drawings comparing to the baselines. An example of restoration effect on FBCNN backbone is shown in Fig. 1.

2. Related Work

Learning-based JPEG Restoration

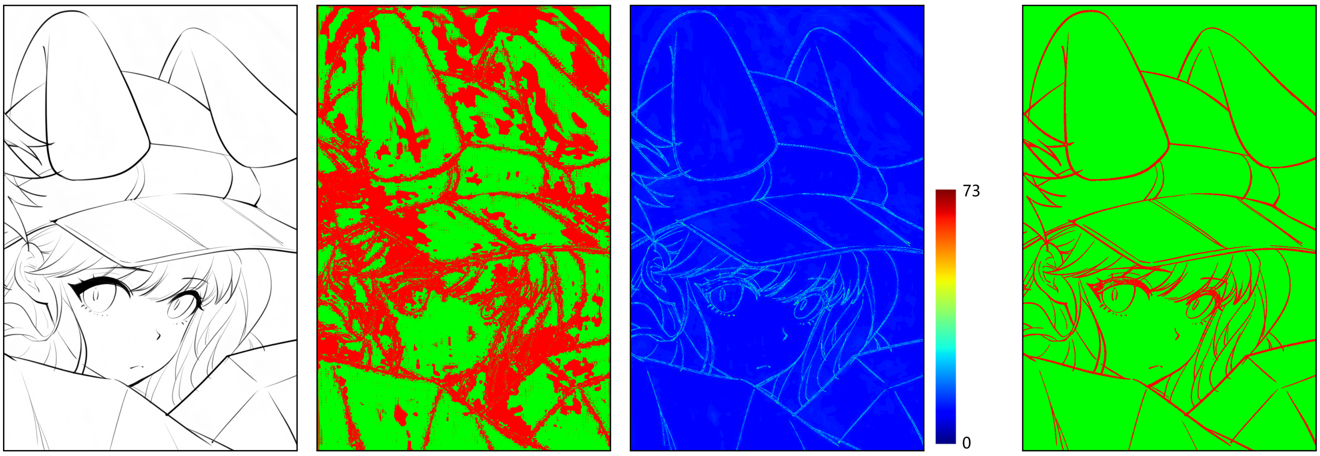
JPEG is a lossy compression algorithm that is widely used in image compression. The level of noise in the compressed image is represented by quality factor (QF), a number ranging from 0 to 100. A lower QF indicates greater information loss and the presence of more artifacts in the compressed image.

Learning-based methods for image restoration in the past few years have achieved state-of-the-art performance. JPEG restoration can be seen as a subtask of image restoration. To train JPEG restoration networks, undegraded images are used as labels and their JPEG-compressed versions are used as inputs. These networks can be mainly divided into two types, namely blind and non-blind, based on whether the QF of the input image is known.

The first learning-based JPEG restoration method is proposed by Dong *et al*. [DDLT15] after the application of deep convolutional network in image super-resolution task [DLHT14]. It is a non-blind method in which the CNN is trained separately for each QF. Guo and Chao [GC16] train a two-branch CNN to aggregate spatial and DCT domain information for JPEG restoration. Zhang *et al*. [ZYHL18] propose a dual domain multi-scale CNN which incorporate DCT domain and pixel domain. Liu *et al*. [LZZ*18] introduce wavelet transform into U-net [RFB15], which reduces the size of feature map to achieve high computational efficiency for JPEG artifacts removal. Zhang *et al*. [ZZZ18] propose a denoising CNN which can handle JPEG compression noise with a tunable noise level map as input. Inspired by that, Zhanget al. [ZLZ*21] incorporate the noise level map into U-net and further improve the performance in JPEG denoising. Fu *et al*. [FZW*19] design a deep convolutional sparse coding (DCSC) network based on learned iterative shrinkage-threshold algorithm [GL10] to reduce JPEG compression artifacts. Ehrlich *et al*. [EDLS20] take use of JPEG's quantization matrix as guidance to JPEG restoration network. Jiang *et al*. [JZT21] propose a flexible blind convolutional neural network (FBCNN) that can predict QF of the input JPEG image and control the output effect inspired by spatial feature transform [PLWZ19] [WYDL18]. Swinir [LCS*21] is proposed after the success of Swin Transformer [LLC*21]. Kawar *et al*. [KSEE22] propose a method to solve JPEG image restoration based on Denoising Diffusion Restoration Models (DDRM) [KEES22].

Mask Generation

In our proposed framework, the role of masks is to separate the line pixels and the background pixels in a line drawing. To generate these binary masks, potential solutions include thresholding and segmentation. Automatic thresholding methods like Otsu's method [Ots79], Kapur's method [KSW85] and Huang's method [HW95] aim to classify foreground and background pixels based on gray-level histograms. Learning-based semantic segmentation



(a) *The output line drawing from FBCNN (QF = 10, PSNR = 37.58)* **(b)** *Binary error map of (a)* **(c)** *Absolute error map of (a)* **(d)** *Binary error map of (a) after thresholding (PSNR = 37.82)*

Figure 2: *Visual analysis of predicting errors. The input line drawing is from our testset and the background brightness is constantly 255. (a) The output line drawing from FBCNN with input QF = 10. (b) We compute the binary error map of (a). Pixels with zero error value are marked as green, and otherwise red. (c) We compute the absolute error map of (a). (d) Our thresholding method is applied to (a) and the PSNR is improved*

using Fully Convolutional Networks [LSD15] and U-net [RFB15] [ZSTL19] also provide hints for our mask-predicting network since we view mask generation from JPEG line drawings as a segmentation task with two classes.

Masked MSE in Image Inpainting

Our proposed framework is similar to deep learning-based image inpainting methods such as [PKD*16], [YLY*18] and [XXL*21]. These methods take an input completion mask along with the input image and compute the masked Mean Squared Error (MSE) as the reconstruction loss, in addition to the adversarial loss. This scheme enables the Generators to efficiently learn the structure of the regions that need to be completed. To the best of our knowledge, such a scheme has never been used in supervised restoration networks since denoising is needed uniformly across all parts of the image. Most of the existing restoration networks we discussed prefer L_1 loss.

3. Proposed Approach

To address JPEG line drawing restoration task, we introduce two methods in this section. We first introduce a thresholding method which is directly applied to the output line drawings of exisiting restoration networks. We then present our line-drawing restoration framework and discuss several mask generation methods for our framework, including our proposed mask-predicting network.

3.1. Thresholding

As we show in Fig. 2, existing restoration networks generally fail to accurately predict the background brightness which should be constantly 255. Even though the absolute error values in the background are close to 0 (Fig. 2d), the large amount of errors (Fig. 2c) in the background still affect the performance on quantitative metrics. To tackle this problem, we propose a simple yet effective thresholding method to post-process the output line drawings from existing restoration networks. Given a grayscale line drawing $\mathbf{L}$ output by the restoration networks, the thresholding method T_θ with threshold value θ is formulated as

$$\underset{(i,j)\in\Omega}{T_\theta(\mathbf{L}_{i,j})} = \begin{cases} 255 & \mathbf{L}_{i,j} > \theta \\ \mathbf{L}_{i,j} & \mathbf{L}_{i,j} \leq \theta, \end{cases} \tag{1}$$

where $\mathbf{L}_{i,j}$ is the pixel luminance of $\mathbf{L}$ at index (i,j), and Ω is the set of all the pixel indices in $\mathbf{L}$.

Because of the low absolute error values in the background of the output line drawings, it is appropriate to select a threshold θ close to 255. In our experiment, the threshold θ is empirically set to 251. Since some line pixels in a line drawing can also have brightness close to 255, the thresholding method will inevitably introduce additional errors. However, due to the fact that the total amount of the line pixels in a line drawing is significantly smaller than that of the background pixels, and that most of the line pixels are much darker than background, this method still consistently improves the overall performance in terms of quantitative metrics.

3.2. Line-drawing Restoration Framework with Masks

The proposed thresholding method only tackles background errors, and inevitably introduces extra errors to the lines in line drawings. We thus propose a framework that not only removes background errors but also improves the performance on the lines. An overview of our framework, as compared to the baseline framework (commonly used in most learning-based image restoration methods), is illustrated in Fig. 3. Given a restoration network, the baseline framework takes as input a grayscale JPEG line drawing $\mathbf{I_j}$ and output its restored version. By contrast, suppose we have a binary mask $\mathbf{M}$

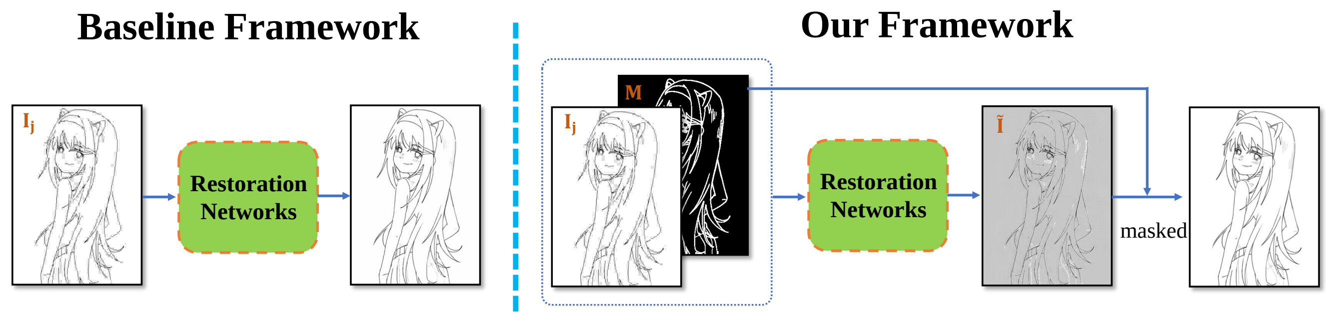


Figure 3: *Our proposed framework compared with the baseline framework. The baseline framework takes as input a JPEG line drawing* $\mathbf{I_j}$ *and directly takes the output of restoration networks as the restoration result. Our framework adds another input channel for a binary mask* $\mathbf{M}$*. In the output line drawing* $\widetilde{\mathbf{I}}$*, only the lines are restored. Then* $\widetilde{\mathbf{I}}$ *is masked by* $\mathbf{M}$*, so that background is set to white color.*

where the value of line pixels is 1, our framework takes as input $\mathbf{I_j}$ concatenated with $\mathbf{M}$ and outputs a line drawing where only lines are restored. Then the output $\widetilde{\mathbf{I}}$ is masked by $\mathbf{M}$ and the background is set to white, that is, the final restored line drawing $\mathbf{L}$ is:

$$\mathbf{L} = \max(\widetilde{\mathbf{I}}, \mathbf{M_{255}} - \mathbf{M_{255}} \odot \mathbf{M}), \tag{2}$$

where $\mathbf{M_{255}}$ is a matrix same size as $\mathbf{L}$ with all elements equal to 255, and $\odot$ is pixelwise product. In the training stage of our framework, we use the ground-truth mask $\mathbf{M}$ synthesized from a ground-truth line drawing $\mathbf{I}$ by thresholding:

$$\mathbf{M}_{i,j} = \begin{cases} 0 & \mathbf{I}_{i,j} = 255 \\ 1 & \mathbf{I}_{i,j} \leq 254, \end{cases} \tag{3}$$

where $\mathbf{I}_{i,j}$ is the pixel luminance of $\mathbf{I}$. We then use a masked MSE loss function with respect to $\mathbf{M}$ as:

$$L(\widetilde{\mathbf{I}}, \mathbf{I}) = \frac{\sum(\mathbf{I} - \widetilde{\mathbf{I}} \odot \mathbf{M})^2}{\sum \mathbf{M}}. \tag{4}$$

where $\sum$ sums up all the elements in the matrix.

Our framework can be viewed as a learning-based image inpainting approach without adversarial training, with the completion area being the lines in a line drawing. However, different from non-blind image inpainting where completion mask is specified by the user, generating masks from JPEG input line drawings can be a nontrivial task. In the next section, we introduce methods for mask generation in our framework.

3.3. Mask Generation

Mask-predicting Network

We propose a learning-based method that treats mask-generating task as semantic segmentation with two classes: line and background. Our mask-predicting network is illustrated in Fig. 4. We use an end-to-end training, where both the training data $\mathbf{I_j}$ and label $\mathbf{M}$ are generated from ground-truth line drawing $\mathbf{I}$ with JPEG compression and equation (3) respectively. Our mask predicting network outputs 2 channels representing the logits for each class. During the training stage, we use the Cross-entropy loss as the optimization function. During the inference stage, the pixel values (0 or 1) of the masks are generated based on the 'argmax' of the two channels.

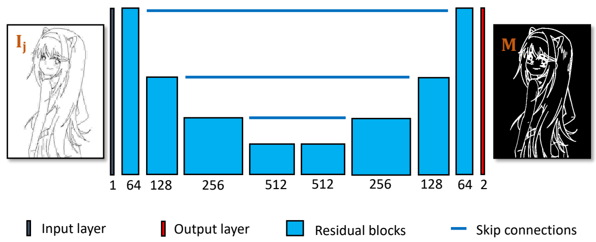


Figure 4: *Structure of our mask-predicting network.*

Direct Mask generation Methods

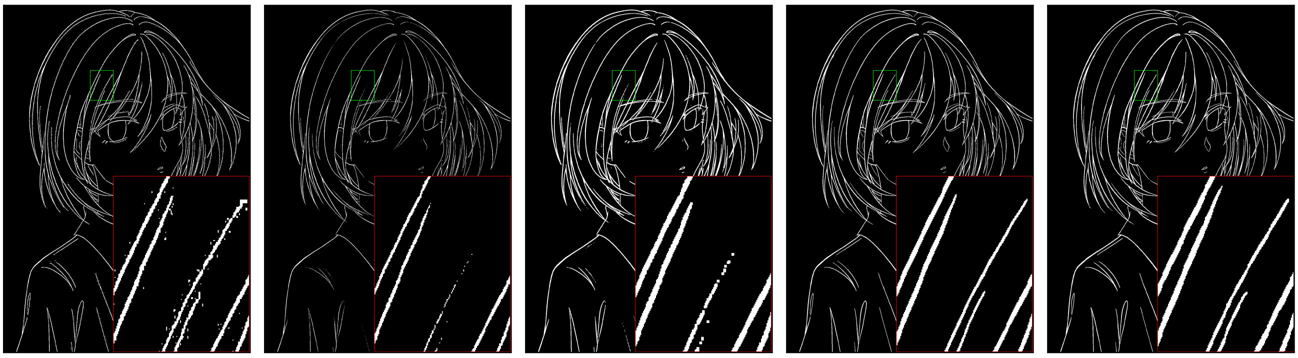
We also consider some direct methods that generate binary mask from a JPEG line drawing. In our experiment, we choose one representative method from histogram-based methods and one from adaptive thresholding methods, namely Otsu's method (referred as "Otsu") and Adaptive Gaussian Thresholding (AGT). Since Otsu's method tends to generate thinner lines compared to the ground-truth masks, we also apply a dilation operation once to the output mask of Otsu's method (referred to as "Otsu_D"). The window size and constant in AGT are set to 35 and 2. In Fig. 5 we show the visual comparisons of these methods.

4. Experiments

In our experiment, we choose three state-of-the-art restoration networks as backbones: FBCNN, DRUNet and SwinIR. FBCNN is a blind network specially designed for restoring JPEG images with unknown QFs. DRUNet encodes the QF as a noise-level map to guide the restoration. SwinIR is a transformer-based restoration network trained for a single QF. First, we introduce our training setup and our line-drawing dataset. We then show the evaluation results of our thresholding method and our line-drawing restoration framework.

4.1. Experiment Settings

The experiments are performed on a Ryzen Threadripper 3960X 24-Core Processor CPU and NVIDIA RTX A6000 GPU. All the



AGT Otsu Otsu_D Ours Ground truth

Figure 5: *Visual comparisons of direct mask generation methods and our mask-predicting network. The mask generated by our method is marked as "Ours" here. The input line drawing is from our test set and compressed with QF = 10.*

training times we mentioned in this section is mesured on a single GPU.

We use a line-drawing training set containing 297 large-sized line drawings and a test set containing 13 line drawings. The line drawings in our datasets are digital line drawings with a background brightness of 255. Most of the line drawings in our dataset are collected from Pixiv and randomly divided into a training set and a test set. During training, the line drawings are randomly cropped into patches.

4.2. Thresholding for Baselines

We demonstrate the improvement achieved by our thresholding method in terms of quantitative metrics. Fig 6 illustrates the training graph that compares baselines with our thresholding method. Training is conducted on our training set, and the plotted PSNR and SSIM curves are evaluated on our test set with QF = 10. It is apparent that our thresholding methods consistently improve the performance compared with the baselines.

4.3. Evaluations of Our Framework

We evaluate our framework combined with different mask generation methods. We first demonstrate that direct methods (AGT, Otsu and Otsu_D) for mask generation fail to achieve satisfying results using our framework. Then we give the quantitative and visual evaluations of our framework and our mask-predicting network.

Direct Mask Generation

From the visualization in Fig. 5, one can see that our method generates a more accurate mask than the direct methods do. Evaluations based on mIoU(mean Intersection over Union) and boundary F-score metrics further confirm the superior performance of our method. We demonstrate the results in Table 1, where these methods are evaluated on our line-drawing test set.

Furthermore, we use FBCNN to compare the training performance between the baseline and our framework with masks generated by direct methods. The moving averages of PSNR and SSIM

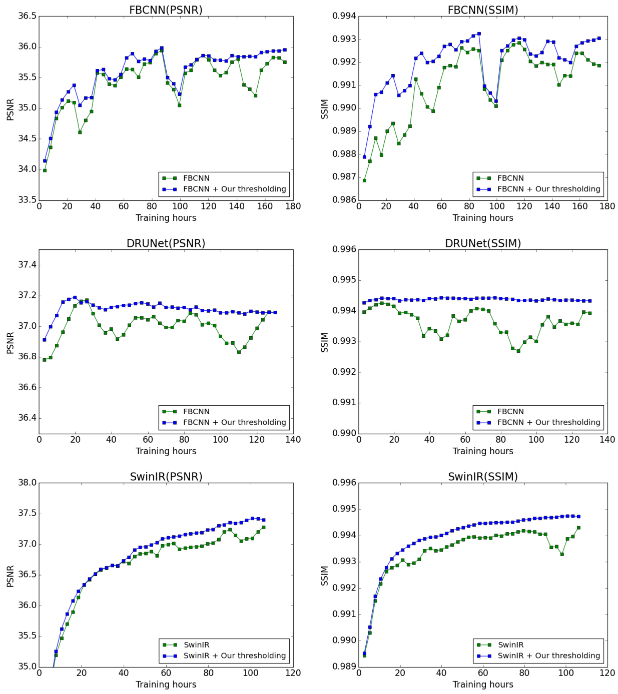


Figure 6: *PSNR and SSIM evaluations of baselines and our thresholding method for backbone FBCNN, DRUNet and SwinIR on our line-drawing test set compressed with QF = 10.*

evaluations on our test set during training are shown in Fig. 7. It is obvious that direct methods are significantly worse than baseline and thresholding. Thus, we consider it is inappropriate to use traditional direct methods for mask generation. Therefore, we only evaluate our framework with learning-based method in the following experiments.

Table 1: *Average mIoU and F-score evaluation results for masks generated from our test set.*

Method	mIoU | F-score
AGT	0.715 | 0.831
Otsu	0.644 | 0.780
Otsu_D	0.673 | 0.797
Ours	**0.887** | **0.939**

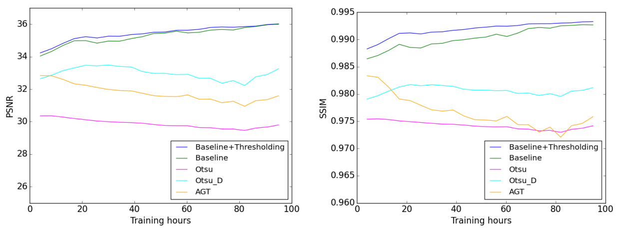


Figure 7: *PSNR and SSIM evaluations during training using backbone FBCNN. The test set are compressed with QF = 10. The performance of our framework combined with direct mask generation methods (Otsu, Otsu_D and AGT) is significantly worse than baseline and our thresholding method. All the curves are processed with moving average.*

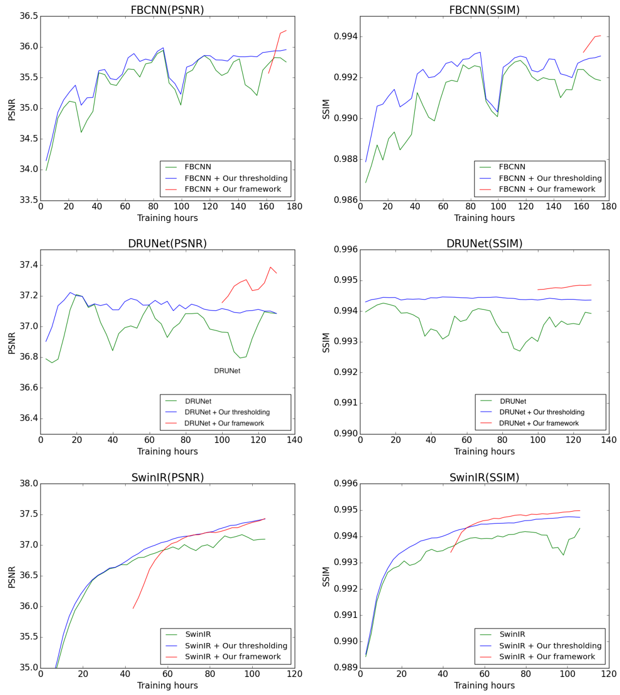


Figure 8: *PSNR and SSIM evaluations of baselines and our methods for backbone FBCNN, DRUNet and SwinIR on our line-drawing test set compressed with QF = 10. The curves of our framework are plotted after training of our mask-predicting networks. All the curves are processed with moving average.*

Mask Generation with Our Mask-Predicting Network

Our mask-predicting network is trained with the same scheme of the backbone networks. For FBCNN, which is a blind network trained on JPEG images whose QF ranges from 5 to 95, our mask-predicting network is also trained with the same QF range for 158 hours. For DRUNet, which is a non-blind network trained with the same range as FBCNN but encodes the QF as a noise-level map into the network, we also train our mask-predicting network with the same noise-level map and QF range for 97 hours. For SwinIR, which is a non-blind network trained with only one QF, our mask-predicting network is trained with QF = 10 for 38 hours. For simplicity, we refer our line-drawing restoration framework with masks generated by our mask-predicting network as "our framework".

For FBCNN, DRUNet and SwinIR backbones, we set the total training times to 174, 130 and 106 hours respectively. The learning rates are the same as that in their original papers. We plot the moving averages of PSNR and SSIM evaluations on our test set for the three baselines compared with our thresholding method and our framework in Fig. 8. Note that our framework needs time to train the mask-predicting networks. Therefore, for better visualization, we shift the training lines for our framework by 158, 97 and 38 hours. It is evident that both our thresholding method and our framework outperform the baselines, with our framework generally achieving the best performance within the given training time. We further report the numerical comparisons in Table 2.

We evaluate our framework for the three backbones in the training times aforementioned qualitatively. The results are shown in Fig. 9, Fig. 10 and Fig. 11. Since our thresholding method makes little difference in visual results, we only show comparisons of baseline and our framework with the three backbones. As one can see, our framework shows a better visual restoration effect for light lines in line drawings. The mask predicted for the three baselines also fairly show the shape of light lines. We infer that the excellent prediction of mask may guide the backbones to perform better for restoring light lines.

4.4. Limitations

For each single lined drawing, our framework consistently achieves better quantitative results in terms of SSIM, which measures the structual similarity with the ground truth. We attribute this to the fact that our mask-predicting network fairly predicts the shapes of lines as we demonstrated in Fig. 9, Fig. 10 and Fig. 11. However, our framework may get slightly worse PSNR performance when the background of original line drawings is unconsciously contaminated by the artists, which results in the background not being constantly 255.

Another limitation exists when our framework is applied in real-world situation. When the deteriorated line drawings contain other type of unknown noise besides JPEG artifacts, our blind mask-predicting network (which we used for FBCNN backbone) may sometimes fail to generate a clean line mask. Since our framework heavily rely on the quality of the line mask, this may result in worse visual quality compared with the baseline. In this case, non-blind mask-predicting network (which we used for DRUNet backbone) with a low noise-level parameter specified by the user is required in order to get a clean line mask.

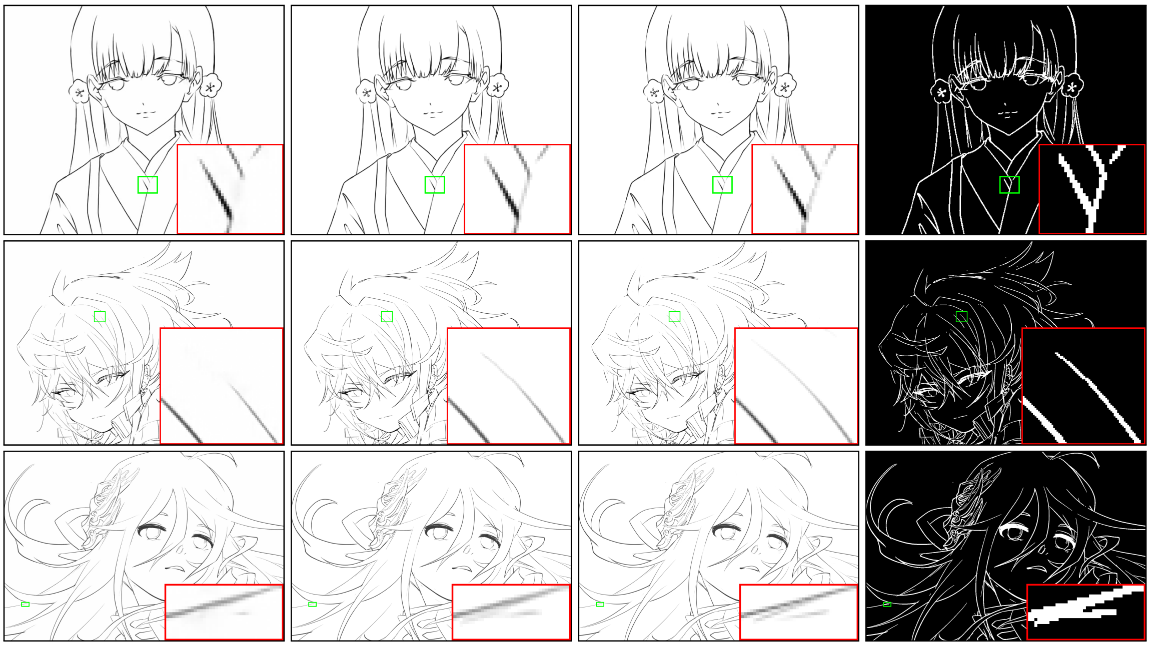


Figure 9: *Visual comparisons of* ***FBCNN*** *backbone with and without our framework. The input JPEG line drawings are from our line-drawing test set compressed by QF = 10. Please zoom in for better visualization.*

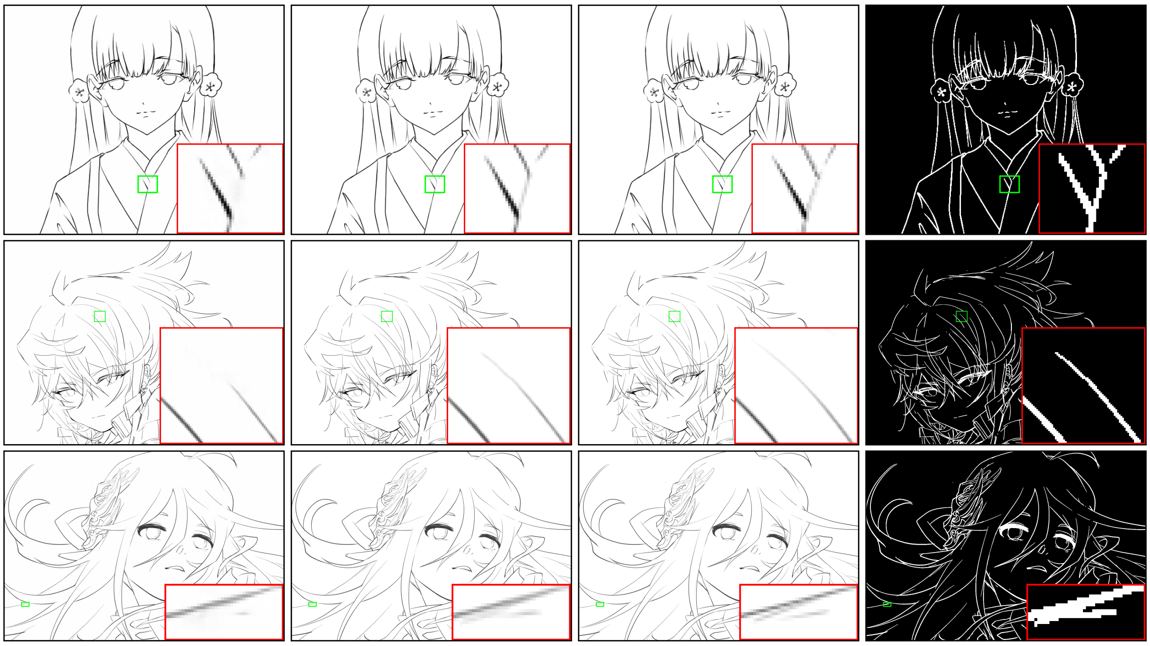
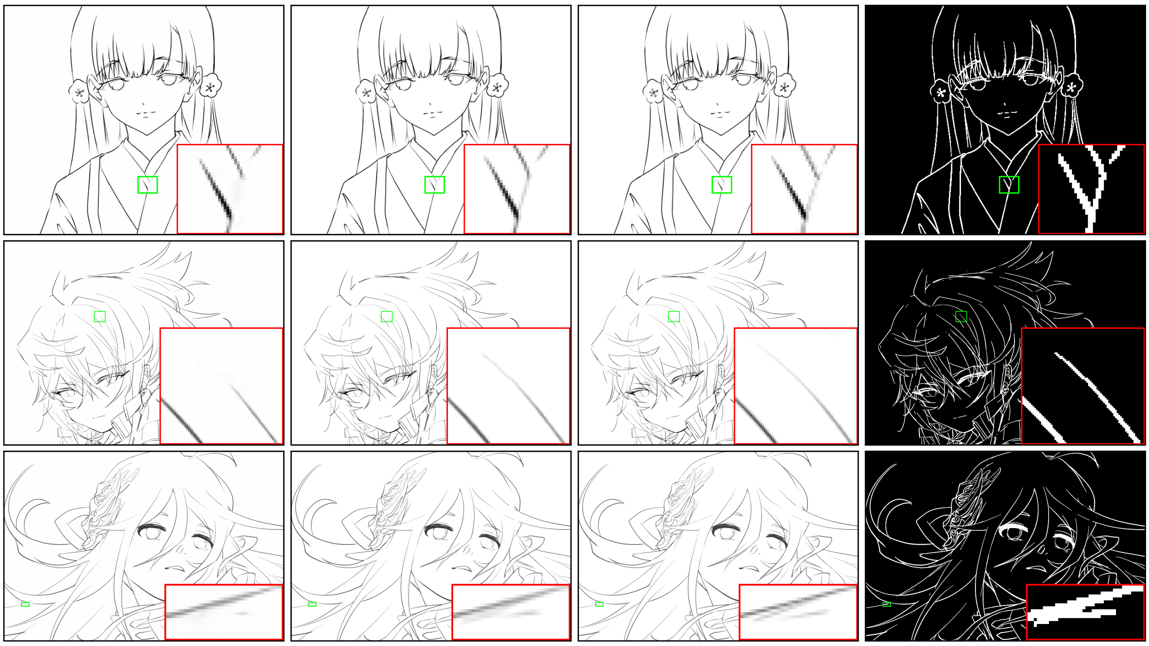


Figure 10: *Visual comparisons of* ***DRUNet*** *backbone with and without our framework. The input JPEG line drawings are from our line-drawing test set compressed by QF = 10. Please zoom in for better visualization.*



SwinIR *SwinIR + our framework* *Ground truth* *Learnt mask (SwinIR)*

Figure 11: *Visual comparisons of **SwinIR** backbone with and without our framework. The input JPEG line drawings are from our line-drawing test set compressed by QF = 10. Please zoom in for better visualization.*

Table 2: *PSNR | SSIM results of baseline and our methods for FBCNN, DRUNet and SwinIR. The training and evaluation are performed on our line-drawing training set and test set.*

Backbone	QF	Baseline	Baseline + our thresholding	Baseline + our framework
FBCNN	10	35.42 | 0.9915	35.66 | 0.9927	**36.47** | **0.9941**
	20	37.56 | 0.9948	38.45 | 0.9961	**39.20** | **0.9969**
	30	39.45 | 0.9959	40.08 | 0.9973	**40.80** | **0.9978**
	40	40.42 | 0.9965	41.22 | 0.9979	**41.85** | **0.9982**
DRUNet	10	37.09 | 0.9941	37.10 | 0.9943	**37.39** | **0.9949**
	20	39.64 | 0.9963	40.11 | 0.9971	**40.49** | **0.9974**
	30	41.87 | 0.9978	41.93 | 0.9981	**42.23** | **0.9982**
	40	43.09 | 0.9982	43.17 | 0.9986	**43.43** | **0.9987**
SwinIR	10	37.09 | 0.9944	37.43 | 0.9947	**37.46** | **0.9950**

5. Conclusion

In this paper, we propose a thresholding method and a line-drawing restoration framework with learning-based mask generation method for JPEG line-drawing restoration. Our thresholding method effectively corrects most of the prediction errors in the background of line drawings. Our framework utilizes a binary mask of line drawings to avoid training on the plain background without intensity information in line drawings and to guide the restoration.

We conduct experiments comparing different mask generation methods for our framework, and results show that our proposed mask-predicting network is more suitable compared to Otsu's method and Adaptive Gaussian Thresholding. Extensive experiments prove superior performance of our proposed framework with our mask generation method for line-drawing restoration. When training time is limited, our thresholding method can be easily applied to existing baselines to get better quantitative results.

Due to the wide spread of JPEG line drawings on the internet, we only consider JPEG restoration in this work. However, we consider that the potential of our framework is not limited to JPEG restoration. In future work, exploiting the effectiveness of our framework for other noise removal or super resolution of line drawings is worth trying.

References

[Ado] ADOBE INC. *Adobe Photoshop*. URL: https://www.adobe.com/products/photoshop.html 1.

[DDLT15] Dong, Chao, Deng, Yubin, Loy, Chen Change, and Tang, Xiaoou. "Compression artifacts reduction by a deep convolutional network". *Proceedings of the IEEE international conference on computer vision*. 2015, 576–584 2.

[DLHT14] Dong, Chao, Loy, Chen Change, He, Kaiming, and Tang, Xiaoou. "Learning a deep convolutional network for image super-resolution". *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*. Springer. 2014, 184–199 2.

[EDLS20] Ehrlich, Max, Davis, Larry, Lim, Ser-Nam, and Shrivastava, Abhinav. "Quantization guided jpeg artifact correction". *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*. Springer. 2020, 293–309 2.

[FZW*19] Fu, Xueyang, Zha, Zheng-Jun, Wu, Feng, et al. "Jpeg artifacts reduction via deep convolutional sparse coding". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, 2501–2510 2.

[GC16] Guo, Jun and Chao, Hongyang. "Building dual-domain representations for compression artifacts reduction". *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, 628–644 2.

[GL10] Gregor, Karol and LeCun, Yann. "Learning fast approximations of sparse coding". *Proceedings of the 27th international conference on international conference on machine learning*. 2010, 399–406 2.

[HW95] Huang, Liang-Kai and Wang, Mao-Jiun J. "Image thresholding by minimizing the measures of fuzziness". *Pattern recognition* 28.1 (1995), 41–51 2.

[JZT21] Jiang, Jiaxi, Zhang, Kai, and Timofte, Radu. "Towards flexible blind JPEG artifacts removal". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 4997–5006 1, 2.

[KEES22] Kawar, Bahjat, Elad, Michael, Ermon, Stefano, and Song, Jiaming. "Denoising diffusion restoration models". *arXiv preprint arXiv:2201.11793* (2022) 2.

[KSEE22] Kawar, Bahjat, Song, Jiaming, Ermon, Stefano, and Elad, Michael. "Jpeg artifact correction using denoising diffusion restoration models". *arXiv preprint arXiv:2209.11888* (2022) 2.

[KSW85] Kapur, Jagat Narain, Sahoo, Prasanna K, and Wong, Andrew KC. "A new method for gray-level picture thresholding using the entropy of the histogram". *Computer vision, graphics, and image processing* 29.3 (1985), 273–285 2.

[LCS*21] Liang, Jingyun, Cao, Jiezhang, Sun, Guolei, et al. "Swinir: Image restoration using swin transformer". *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, 1833–1844 2.

[LLC*21] Liu, Ze, Lin, Yutong, Cao, Yue, et al. "Swin transformer: Hierarchical vision transformer using shifted windows". *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, 10012–10022 2.

[LSD15] Long, Jonathan, Shelhamer, Evan, and Darrell, Trevor. "Fully convolutional networks for semantic segmentation". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, 3431–3440 3.

[LZZ*18] Liu, Pengju, Zhang, Hongzhi, Zhang, Kai, et al. "Multi-level wavelet-CNN for image restoration". *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, 773–782 2.

[Ots79] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms". *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), 62–66 2.

[Pix] Pixiv Inc. *Pixiv*. https://www.pixiv.net/ 2.

[PKD*16] Pathak, Deepak, Krahenbuhl, Philipp, Donahue, Jeff, et al. "Context encoders: Feature learning by inpainting". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 2536–2544 3.

[PLWZ19] Park, Taesung, Liu, Ming-Yu, Wang, Ting-Chun, and Zhu, Jun-Yan. "Semantic image synthesis with spatially-adaptive normalization". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, 2337–2346 2.

[RFB15] Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. "U-net: Convolutional networks for biomedical image segmentation". *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, 234–241 2, 3.

[WYDL18] Wang, Xintao, Yu, Ke, Dong, Chao, and Loy, Chen Change. "Recovering realistic texture in image super-resolution by deep spatial feature transform". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 606–615 2.

[XXL*21] Xie, Minshan, Xia, Menghan, Liu, Xueting, et al. "Seamless manga inpainting with semantics awareness". *ACM Transactions on Graphics (TOG)* 40.4 (2021), 1–11 3.

[YLY*18] Yu, Jiahui, Lin, Zhe, Yang, Jimei, et al. "Generative Image Inpainting with Contextual Attention". *CoRR* abs/1801.07892 (2018). arXiv: 1801.07892. URL: http://arxiv.org/abs/1801.07892 3.

[ZLZ*21] Zhang, Kai, Li, Yawei, Zuo, Wangmeng, et al. "Plug-and-play image restoration with deep denoiser prior". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (2021), 6360–6376 2.

[ZSTL19] Zhou, Zongwei, Siddiquee, Md Mahfuzur Rahman, Tajbakhsh, Nima, and Liang, Jianming. "Unet++: Redesigning skip connections to exploit multiscale features in image segmentation". *IEEE transactions on medical imaging* 39.6 (2019), 1856–1867 3.

[ZYHL18] Zhang, Xiaoshuai, Yang, Wenhan, Hu, Yueyu, and Liu, Jiaying. "DMCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal". *2018 25th IEEE international conference on image processing (icip)*. IEEE. 2018, 390–394 2.

[ZZZ18] Zhang, Kai, Zuo, Wangmeng, and Zhang, Lei. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising". *IEEE Transactions on Image Processing* 27.9 (2018), 4608–4622 2.