Document De-Blurring using Image De-Blurring Techniques

Apoorva Kumar
akumar255@wisc.edu

Vaibhav Nitnaware
nitnaware@wisc.edu

Varun Kaundinya
kaundinya@wisc.edu

1. Introduction

Image Deblurring is a fundamental but challenging computer vision problem of recovering a sharp image from a blurred image. This problem has been of highlight lately with Google and Adobe working towards implementing Deep Learning based algorithms to deblur images. While multiple state-of-the-art techniques [1,4,8] have come up in the deep learning space to deblur pictures we see that most of these techniques focus on deblurring photographs and images.

On the other hand, text deblurring has been handled as a unique problem in itself, and it has generally not been considered a subset of classical image deblurring. Our primary focus will be to evaluate the state-of-the-art image deblurring techniques to tackle the problems and see how it fares for the text deblurring task. Our first task is to generate a document deblurring dataset, as there weren't any previously available. Next, we plan to evaluate our chosen state-of-the-art models to deblur these document images and present the results on the same. Finally, we plan to evaluate the model on real-world blurred document images and look at its response.

2. Related Work

There has been some work dealing with or handling document deblurring. Using a simple convolutional network as an encoder-decoder to deblur text images was published in [2]. While results are promising on text with unidirectional motion blur or simple Gaussian blur, it failed on images with multiple blurs. They also had issues with the complex text of low occurrence, like mathematical symbols. Image Super-Resolution was also used by [9] to super-resolve text images, which falls somewhat in the space of document and text image recovery but doesn't directly address the problem of deblurring.

While there are multiple papers in the space, some mark drastic evolution in the deblurring technique, and we chose to pick some for our evaluation. Our selected papers [5, 6, 10] are from three different years to estimate when the deblurring space could have been assimilated and draw a conclusion based on the same.

Deep-Deblur [6] was published in 2017 and presented a multi-stage image deblurring technique. The model uses a feature pyramid network with upsampling to generate deblurred images. The output from each layer of the FPN is passed through a different convolution head to recover from coarser to finer details. The paper is one of the first to use multi-stage recovery in the deblurring domain, unlike others that tried one-shot recovery.

MPRNet [10], published in 2020, was another remarkable development in deblurring space. Rather than resizing the image like DeepDeblur, they split it into equal parts at each level and recovered the images from them. UNet [7] was used as an encoder-decoder-based recovery network with a Supervised Attention Module to predict the actual issues in the image and generate a feature map which, when added to the blurred image, returns the deblurred image. They also used a combination of MSE and EdgeLoss [3] to recover the images on multiple fronts.

Deep-RFT [5] was published in 2022 and proposed a new deblurring method wherein they focus on pixel-level and kernel-level feature estimation. They developed a new block called Res-FFT block with the idea that frequency selection on the image helps understand the blur information. Thus they finally suggest that the recovery can be significantly enhanced by just replacing any Resblock with a Res-FFT block in an image recovery network. They also improved upon the idea of using multiple losses from MPRNet and used MSE and Frequency Reconstruction loss.

3. Blur Pipeline

Due to the absence of any dataset in the Document deblurring domain, we had to create our dataset using artificial blurring techniques from OpenCV. We assessed different methods for generating real-life blurs and settled on three blur methods.

- Gaussian: Gaussian kernel to blur the image.
- Motion: A Directional kernel to blur the image.
- Mean: A kernel that averages all values convolved with the kernel.

3.1. Motion Blur

Regular motion blur kernels move the image with equal intensity along the direction. But such motion blurs are very easy to recover from and don't exist that ideally.

To create a more realistic Motion Blur, we multiplied the standard blur kernel with a random array of the same shape and generated a new kernel. The shifts here look like they are in stages which is how actual cameras move rather than a smooth unidirectional motion. The comparison of the kernels can be seen in Fig 1

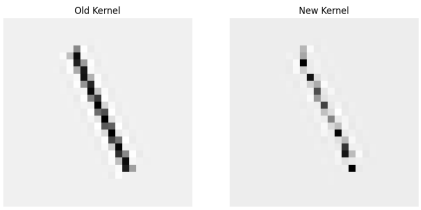


Figure 1. Comparison of Kernels

3.2. Blur generation pipeline

We developed our own technique and compiled it as a Pytorch DataLoader. The technique is as follows:

1. Pick the image for the given index
2. Apply a set of random transforms from flipping and rotation
3. Randomly crop a 256×256 or 512×512 image from the image
4. Pick a two random blurs from the above mentioned blurs with replacement
5. Apply these blurs on the cropped image and return the sharp and original image

3.3. Dataset

The publicly available document dataset by IBM called PubLayNet [11] containing 335,703 training images and 11,245 validation images are used to generating the dataset.

Some samples for the Dataset can be seen in Fig 2

4. Evaluation Method

For evaluation we used Peak Signal to Noise Ratio (PSNR) as a training and pretraining evaluation metric.

$$\text{PSNR} = 10 * \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

$$\text{where, MSE} = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}||f(i,j) - g(i,j)||^2$$

Generally, a value above 30 is considered a good restoration level.

We also run the models with PSNR>30 during training time on real-world images to see how it fares on them. Actual world blur kernels are much more complex than the ones generated mathematically though they might not be visible to the naked eye. Thus we wanted to understand if the model didn't just learn the mathematical kernels we produced or tried to unblur the image.

5. Results and Inference

First we picked the models with their best trained model weights provided by the authors' themselves and using them evaluated our dataset. The hyperparamters for the models which varied for each pretrained model are as follows:

Mertrics	Epochs	Learning Rate
Deep-DeBlur	1000	1e-4
MPRNet	3000	2e-4 to 1e-6
Deep-RFT	3000	2e-4 to 1e-6

And here are the results for the same. $PSNR_{avg}$ is the average of PSNR across the validation dataset while $PSNR_{image}$ are their results on non-text images.

Figure 2. Results of our blur

Mertrics	$PSNR_{avg}$	$PSNR_{image}$
Deep-DeBlur	15.5299	30.12
MPRNet	13.67	32.66
Deep-RFT	14.77	33.12

We can see the models trained on normal images don't work very well on their document counterpart.

Next we then moved on to transfer learn the model using the pre-trained weights and here aree the training parameters and design:

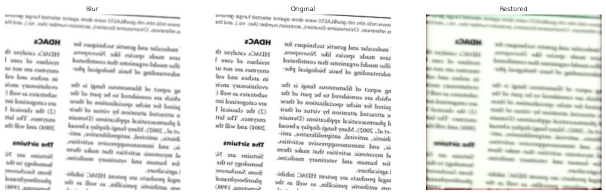
Mertrics	Epochs	LRate	Loss
Deep-DeBlur	110	1e-4	L1
MPRNet	90	2e-4 to 1e-6	Ch+Edge
Deep-RFT	40	2e-4 to 1e-6	Ch+FR

Here, *L1* is L1 Loss , *Ch* is Charbonnier Loss , *Edge* is Edge Loss, *FR* is Frequency Reconstruction Loss. The results produced are as shown below:

Mertrics	$PSNR_{max}$	$PSNR_{avg}$
Deep-DeBlur	21.26	17.89
MPRNet	18.01	15.54
Deep-RFT	35.89	28.89

We observe that Deep-RFT shows very promising results according to the PSNR metric while the other models still suffer, with Deep-Deblur and MPRNet improving by only 2dB on average PSNR and MPRNet.

As we observe for both Deep-Deblur and MPRNet, results are entirely washed out and look like the model cannot even understand the tasks in the case of text images. We can attribute it to the fact that these models use losses that don't seem suitable for the defined task. For MPRNet, the Supervised Attention Module captures text as the actual blemish or error in the image and tries to remove that. It considers the white space in the image as the actual regions of recovery

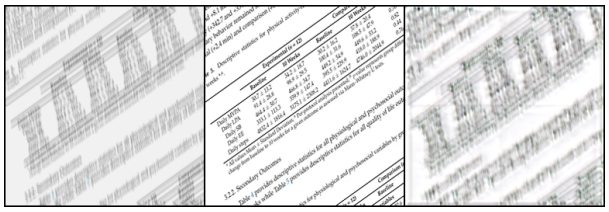


(a) Pretrained

(b) After transfer learning

Figure 3. Deep-Deblur results

(a) Pretrained



(b) After transfer learning

Figure 4. MPRNet results

and thus tries to apply the Edge Loss to recover those white spaces rather than the black characters. On the other hand, for Deep-Deblur for pretrained results, we see that the feature pyramid outputs needed to be com-

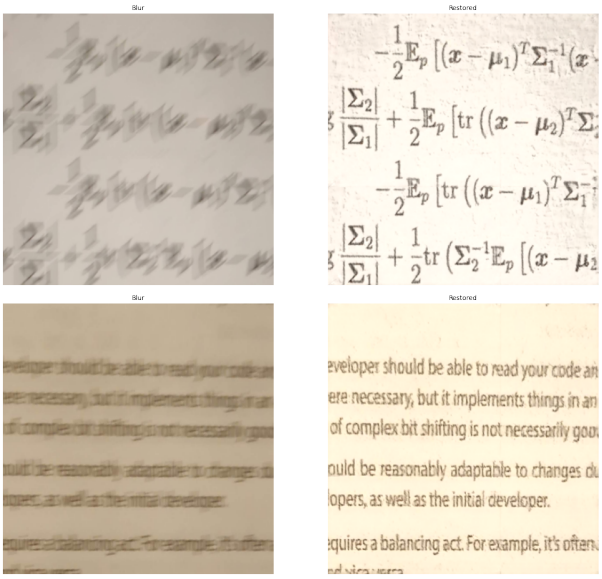


Figure 5. DeepRFT camera results

bined better and gave a weird shift. On training, it completely washed out the image, which technically helps reduce L1 loss the fastest again due to much white space. It also uses a feature Pyramid which is trained with image sizes of ((256, 256) with feature pyramid scaling it down by 2 per level) this is an issue when testing on high-resolution blurry images as they need to be scaled to this level.

We also noted that both these models have a lot of training time and computation power, with an epoch taking more than 12 to 13 hours, even on a good GPU with optimized code. Even after expending lot of resources and computation power we saw no improvement in the model unlike Deep-RFT about which you will see in a while.

Moving to Deep-RFT, we saw an outstanding result which can be seen in the PSNR Table. We directly went on to evaluate this model on real-world images. The results for the same are shown in Figure 5

We can see how well the images are recovered and look. The model can even work well on nonstandard text, unlike [2] and gives fantastic results.

6. Conclusion and Future Work

While we felt earlier that the homogenization of the deblurring problem space wouldn't be possible, Deep-RFT, with a single tweak to the Residual Block, could achieve the same. Deep-RFT could learn spacial and kernel-level features without adding much computation time using simple models. The model achieved a PSNR of ∼35 just by transfer learning for 40 epochs, and we believe it can be further improved if the deblurred image dataset was organically part of the initial training.

Deep-RFT, being a novel model (2022), can solve the document deblurring problem that has been a challenge for past models. Thus, progress has been made in this field, and the deblurring problem can be homogenized as a single research entity.

As for our future work, we can replace the Residual block in Deep-Deblur and MPRNet model with the Deep-RFT results and check for outcomes using the same. Replacing the Loss function with a correct combination of loss functions like Edge Loss and Frequency reconstruction loss can also be performed. The absence of a dataset was another big hurdle for our work, and preparing a good dataset for the same is another task. Moving forward, a dataset consisting of both text-heavy and non-text images can be created, and the deblurring problem can be tested using newer and improved models.

Our final work can be viewed at: https://github.com/VarunThfc/771FinalProject

Members	Contributions	
Apoorva Kumar *Varun Kaundinya* *Vaibhav Nitnaware*	**MPRNet** **Deep-Deblur** **Deep-RFT**	Inference and Reports

References

[1] Xiaojie Chu, Liangyu Chen, Chengpeng Chen, and Xin Lu. Improving image restoration by revisiting global information aggregation, 2021. 1

[2] Michal Hradiš, Jan Kotera, Pavel Zemčík, and Filip Šroubek. Convolutional neural networks for direct text deblurring. In Mark W. Jones Xianghua Xie and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 6.1–6.13. BMVA Press, September 2015. 1, 4

[3] Shuo Liu, Wenrui Ding, Chunhui Liu, Yu Liu, Yufeng Wang, and Hongguang Li. Ern: Edge loss reinforced semantic segmentation network for remote sensing images. *Remote Sensing*, 10(9), 2018. 1

[4] Xintian Mao, Yiming Liu, Fengze Liu, Qingli Li, Wei Shen, and Yan Wang. Intriguing findings of frequency selection for image deblurring, 2021. 1

[5] Xintian Mao, Yiming Liu, Wei Shen, Qingli Li, and Yan Wang. Deep residual fourier transformation for single image deblurring, 2021. 1

[6] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring, 2016. 1

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 1

[8] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool. Event-based fusion for motion deblurring with cross-modal attention, 2021. 1

[9] Xiangyu Xu, Deqing Sun, Jinshan Pan, Yujin Zhang, Hanspeter Pfister, and Ming-Hsuan Yang. Learning to super-resolve blurry face and text images. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 251–260, 2017. 1

[10] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration, 2021. 1

[11] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. *arXiv preprint arXiv:1908.07836*, 2019. 2