6.819 Final Project
A Deep Generative Adversarial Network for Blind Image Deblurring

Josh Hellerstein
joshh@mit.edu

Christie Hong
cshong@mit.edu

Abstract

Non-homogeneous image deblurring has been an unsolved computer vision challenge for decades. Blurring in images occur most often because of camera motion, and object/subject motion. These blurs often involve a non-homogeneous spatially varying blur, which is difficult to estimate directly from the blurred image alone. In this paper, we propose a modified Wasserstein Generative Adversarial Neural Network with Gradient Penalty (WGAN-GP) for "blind" image deblurring – no external knowledge about the image. We also propose a large synthetically generated collection of object motion blur images that surpasses existing blurred datasets in quantity and quality, as it emulates "long-exposure" shots to capture a more natural representation of motion kernels. We propose several modifications to the WGAN, and show that training on our dataset improves image deblurring. The experiments conducted in this study prove the effectiveness of our proposed method.

1. Introduction

Motion blur in real images stem from a variety of causes such as camera shaking [6, 11] and object motion [12], leading to complex variations of blur patterns.

Earlier studies focused on removing blurs caused by camera motion via blur kernel estimations. These models were only trained on translational and rotational blurs, and were not generalizable for all blurs. Recent works attempt to handle general non-uniform blurs caused by camera shake, object motions, and depth variation by parametrizing blurs with simple assumptions on the blur sources. Kupyn *et al*. optimizes this part of the problem with an end-to-end learning approach which we implicated and modified [7].

In past papers [5], there are no pairs of raw real blurry image and ground truth sharp image available for supervised learning, and Sun *et al*. created blurry images by convolving synthetic blur kernels from high definition images. To remedy this problem we replicate Nah *et al*.'s methodology for creating a better data set, with stronger correlation to that of true motion blur, by utilizing video footage and aggregating consecutive frames. With this method we simulate the same effects of taking a photo on a camera with a slow shutter speed and thus replicating object motion blur. We implement our WGAN deblurring model by training on this proposed data set, and propose several modifications to the model by [7].

2. Related Work

Most of the previous work on this particular topic address the issue of camera motion blur by modelling that aspect of this problem as a blind deconvolution problem, and estimating blur kernels. These results have been extraordinary, however their assumptions, and consequentially, their methodologies, limits general usage. For example, Gong *et al*. utilizes a fully-convolutional deep neural network (FCN) to recover the unblurred image from an estimated motion flow [3]. They directly estimate the motion flow by estimating the blur kernel and recover a blur-free latent image via non-blind deconvolution. More recently, Generative Adversarial Networks have also been applied to this problem as well. While GANs have been utilized for different computer vision problems such as super resolution, image translation, and much else, Kupyn *et al*. presents an end-to-end learning approach for motion deblurring based on a conditional GAN [7]. They treat deblurring as a special case of "image-to-image" translation, in which they generate a data set for motion deblurring and a loss function to minimize errors in generating the correct deblurred image. Similarly, Ramakrishnan *et al*. also used an end-to-end GAN architecture with loss functions, which performs blind restoration for shaken images [9]. On the other hand, Nah *et al*. utilized an end-to-end deep multi-scale convolutional neural network (CNN) [10]. This particular model utilizes a multi-scale loss that enhances convergences to a clear image greatly compared to those models estimating blur kernels.

A few significant takeaway from these works is that estimating blur kernels is not sufficient enough for the general problem of blur, as blur kernel estimation does not cover the case of object motion blur. Additionally, the idea of an end-to-end model has proven to be significant in improving deblurring models. Therefore, we worked further on Kupyn *et al*. and Nah *et al*.'s methodology of generating more accurate image data sets via video files, and an end-to-end GAN model to deblur dynamic images.

3. Proposed Methodology

The goal of our research is to recover a sharp image given a blurred image as the input. We utilize a conditional WGAN and modify its inputs and hyperparameters and propose a new conditioning variable to output a better resultant sharp image [7]. Further, we extend the model by Kupyn *et al*., which was inspired by Isola *et al*.'s pix2pix network [4] by training it on our custom data set.

A blurred image I^B can be represented as the convolution of a sharp image I^S with a blurring kernel K, and added noise N.

$$I^B = K * I^S + N$$

Instead of estimating the burring kernel K, we will have the WGAN directly generate the sharp image, conditioned on the blurred input image. We propose a modification to the network, by conditioning on the Laplacian kernel K_L convolved with the blurred image:

$$K_L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\hat{I}^S = G_\theta(I^B, K_L * I^B)$$

where $\hat{I}^S$ is the generated image and $G_\theta(z)$ is the generator network.

We can formulate the GAN training process in terms of a discriminator and generator, where the generator learns how to generate real-looking data, and the discriminator learns how to differentiate between real data and fake data. The architecture optimizes the parameters according to the following loss function:

$$\mathcal{L} = \underbrace{\underbrace{\mathcal{L}_{GAN}}_{\text{adv loss}} + \underbrace{\lambda \times \mathcal{L}_X}_{\text{content loss}}}_{\text{total loss}}$$

Which is a composition of traditional WGAN loss and perceptual loss (MSE difference between the generated and target images summed over the outputs of the feature layers ϕ_i of a VGG-19 network) proposed and specified by Kupyn *et al*. in his paper [7].

Additionally, we collected more information on techniques of generating blurs, and built a 300,000 image data set. Our data set is a larger and far better representation of true motion blur in-the-wild compared to those generated in past studies.

4. Motion Blur Generation

We wanted to generate a dataset containing realistic blur patters on diverse images for training. [3, 5, 2, 1] all look specifically at camera motion blurs and simulates motion flow by considering the translation and rotational blurs along x, y, and z axis. The x and y axial translations portray horizontal and vertical shaking of the camera, whereas translation/rotion along the z-axis causes a radial motion blur pattern [3]. These four research papers used various non-uniform motion kernels to replicate camera motion, but simply examining x, y and z translational and z rotational blurs and predicting those motion flows did not seem to have a high enough variance of outputs.

This evaluation led us to another finding: producing blurred images from high definition video files, creating long exposure shots by compounding and averaging sequential frames of a video into one photo [10]. Averaging frames from a video allows us to generate high quality "natural" object motion blur.

Nah *et al*. explains this idea of compounding and average frames together as an "integrated signal" that is "transformed into pixel value by nonlinear Camera Response Function (CRF)," which essentially maps the scene's irradiance (brightness) to intensity. Thus our blur accumulation process is modeled as the following:

$$blur = g\left(\frac{1}{M}\sum_{i=0}^{M-1} S[i]\right) \tag{1}$$

where $M, S[i]$ are the number of consecutive sampled frames and the i-th sharp frame signal capture during the exposure time. g is the CRF that maps a sharp latent signal into an observed image $S[i]$ such that $S[i] = g(S[i])$ [10]. In practice, we only have the collected frames, and the original signal and ground truth CRF is unknown, hence we approximate CRF as a gamma curve with $\gamma = 2.2$ such that $g(x) = x^{1/\gamma}$. Therefore, by correcting the gamma function we can find the latent frame signal $S[i]$ from the observed image, $S'[i]$ by applying the inverse gamma to each frame.

$$S[i] = g^{-1}(S'[i]) \tag{2}$$

To generate our data set we downloaded over 3000 high definition Youtube videos using YouTubeDB, an open source database. Hence, we selected videos from categories that had a lot of motion, such as: racing, sports, driving etc. We generated 100 samples per video with the number of frames to average over equal to 8. We divided our images into two folders called "targets" and "blurred" corresponding the sharp ground truth images to their generated blurred counterparts. We were able to successfully build and generate 300,000 true motion blur images via this method.

Figure 1: Example of generated motion blur. Left image is the generated motion blur image, and right image is our identified target image

5. Our Model

For our model (architecture shown in Figure A in Appendix), we decided to build on the conditional WGAN by Kupyn *et al*. [7]. There is evidence to indicate that blurred images have less high-frequency components in their Fourier spectra [13]. Trying to leverage this information, we decided to investigate extracting features from the frequency domain. Though extracting features from a Fourier transform of the image would yield a detailed analysis, we seek to extract a more local metric in the image, as PatchGAN is used in Kupyn's architecture and looks only at local image regions (the final image is reconstructed from local patches of images).

We extract the "frequency" features instead, by convolving the input image with a Laplacian kernel (described in section 3) – this is known to approximate the second derivative of the image. The variance of this kernel is shown to be a good indicator of "blur" in an image [8], and the resulting tensor itself highlights regions of an image containing rapid intensity changes (edges), which should occur with less variance in blurred images.

Figure 2: Convolving the Laplace kernel with blurred and sharp images. Variance of sharp image: 767.9, Variance of blurred image: 176.4

We thus computed $L_K * I^B$, and fed the result into a fourth channel of the input image. This effectively conditions the GAN on this result, as well as the normal blurred image, as in [7].

We also decided to add noise to the labels during the discriminator portion of training. We gave the discriminator a 10% chance of receiving the wrong label – we intentionally make our model intentionally confused in differentiating between fake and real images occasionally. We do this because the discriminator in the GAN is known the converge faster than the generator. Implementing this stabilizes training.

We trained the model on 2 x NVIDIA Tesla K80 12GB GPUs on 6000 additional images from our 300,000 image data set. According to Kupyn *et al*., the original model took 6 days to train using 3000 images and 300 epochs. Thus, we couldn't utilize the entirety of the dataset due to time constraints on training.

Please look at Appendix figure A for the model architecture diagram, as well as image results.

6. Results

We ran our model on our custom dataset for qualitative inspections, and the GoPro test set of 1100 images to evaluate the results in comparison with other models. Figure 3 is an example of running our model on one of the images generated from our dataset. We can see that the resulting image appears less blurry. We computed SSIM and PSNR metrics for the GoPro test data and our results. These metrics are commonly used to compare perceptual similarity of images.

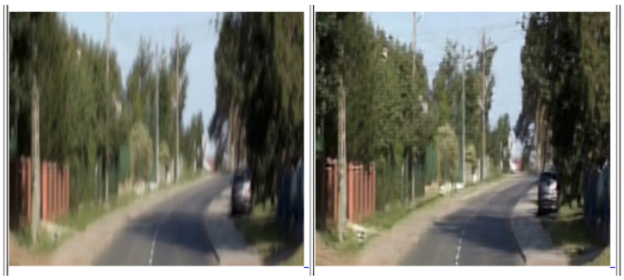

Figure 3: Left image is our generated blur image inserted into our model, right image is the resultant deblurred image

The results are shown in the table below:

GoPro Test Set Evaluation		
Paper	PSNR	SSIM
Sun *et al*. [5]	24.64	0.842
Nah *et al*. [10]	29.23	0.916
Xu *et al*. [2]	25.19	0.896
Gong *et al*. [3]	27.199	0.908
Nah *et al*. [10]	28.45	0.917
Ramakrishna *et al*. [9]	28.94	0.922
Kupyn *et al*. [7]	33.16	0.954
Ours	**33.41**	**0.958**

We can see that our model achieves a modest improvement compared to the model on which it was based (Kupyn *et al*.). However, it is currently state-of-the-art. For more results please take a look at our Appendix, Figures B, C, and D.

7. Analysis

The modifications made to the model and the increased quantity of natural data proved to be key to improving the model's accuracy. Specifically, the inclusion of conditioning on the Laplacian kernel convolved with the image can be thought of as pre-selecting image features that we know to be important to determining image blur.

As mentioned in the paper that proposed this architecture, residual blocks, dropout, leakyReLU, and normalization were all key to achieving high accuracy. Further, our proposed random noise in the discriminator likely helped the model explore the entire parameter space by slowing down the training of the discriminator.

Our dataset funtioned to increase the possible natural images accumulated, similar to the GoPro dataset. The fact that the images were generated in the same fashion, means that it likely expanded the number of possible kernel combinations that occur naturally in camera video.

8. Limitations

We learned about our implementation of the GANs, optimizing hyperparameters, and working with the details of the PyTorch code.

Our generation of blurred images had several limitations. The biggest was the quality of scenes – specifically when a sample pair would be created from a time in the video where the scene changes. This produces an invalid sharp-blur pair. We attempted to address this case by taking the difference between images and setting a threshold to identify a change in scene. The problem with this fix is that that threshold is not constant for all videos across the board.

9. Conclusion

In conclusion, our model modestly improved state-of-the-art blind image deblurring. Further, our blurred image data set that we generated surpasses most past papers generated blur images in quantity and quality. Compared to Nah *et al*., our data set is two orders of magnitude larger than his GoPro data set. Additionally, our data set is not generated by estimations of blur kernels, but rather an accumulation of many video frames to generate true object motion blur. Hence, the quality of blur is nontrivial for our model to correct, and allows our model to be far more generalizable than past papers.

Moreover, the pre-selection of features (in our case, the Laplacian) of the image allows for guided parameter selection, which can be especially useful when fine tuning a pretrained model. Future work could involve run the model on the entire 300,000 image dataset, if time constraint is not an issue. To increase accuracy of the model, we could ensemble different past image deblurring models to produce a better and more varied output.

10. Individual Contribution (Josh)

For this project, I set up the deep learning environment on Google Cloud with 2x Tesla K80 GPUs, CUDA, Pytorch, Tensorflow, and Keras, and maintained git / executed dataset generation code in the cloud. I acquired the WGAN model implementation in Pytorch, and modified it to include "confusing the discriminator," and the conditioning on the Laplacian of the image. I evaluated the model against other models on the GoPro dataset, and executed the model on our dataset. I selected the videos from YoutubeDB, and wrote a script to scrape them at 1280x720p, 30fps. I checked the image blurring code, and modified it to run fast.

References

[1] T. Z. Ayan Chakrabarti, Willian T. Freeman. Analyzing spatially-varying blur, 2010.

[2] T. C. M. L. Bing Xu, Naiyan Wang. Empirical evaluation of rectified activations in convolution network, 2015.

[3] L. L. Y. Z. I. R. C. S. A. v. d. H. Q. S. Dong Gong, Jie Yang. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur, 2016.

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[5] Z. X. J. P. Jian Sun, Wenfei Cao. Learning a convolutional neural network for non-uniform motion blur removal, 2015.

[6] A. Z. J. P. Oliver Whyte, Josef Sivic. Non-uniform deblurring for shaken images.

[7] M. M. D. M. J. M. Orest Kupyn, Volodymy Budzan. Deblurgan: Blind motion deblurring using conditional adversarial networks, 2017.

[8] J. L. Pech-Pacheco, G. Cristóbal, J. Chamorro-Martinez, and J. Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 3, pages 314–317. IEEE, 2000.

[9] A. G. S. R. Sainandan Ramakrishnan, Shubham Pachori. Deep generative filter for motion deblurring, 2017.

[10] K. M. L. Seungjun Nah, Tae Hyun Kim. Deep multi-scale convolutional neural network for dynamic scene deblurring, 2016.

[11] J. J. Shicheng Zheng, Li Xu. Forward motion deblurring. *CVPR*, pages 1465–1472, 2013.

[12] K. M. L. Tae Hyun Kim, Byeongjoo Ahn. Dynamic scene deblurring. 2013.

[13] R. Yan and L. Shao. Blind image blur estimation via deep learning. *IEEE Transactions on Image Processing*, 25(4):1910–1921, 2016.