Indian Institute of Technology Kharagpur
Department of Electrical Engineering

Blur Kernel Estimation using Deep Learning

Arun Patro | 13EE35026
Under the Guidance of Dr. Rajiv Ranjan Sahay

A report
Submitted in partial fulfilment of requirements for the degree of

Bachelor of Technology

in

Electrical Engineering

Submitted on 3rd May 2017

Signature: ....

Declaration

I hereby declare that:

1) The work contained in this thesis is original and has been done by me under the guidance of my supervisor, Dr. Rajiv Ranjan Sahay
2) The work has not been submitted to any other institute for any degree or diploma.
3) I have followed the guidelines provided by the institute in preparing the thesis.
4) I have conformed to the norms and conditions given in the Ethical Code of Conduct of the Institute.

(Signature of the Student)

Arun Patro

13EE35026

Acknowledgment

I would like to express my gratitude towards my guide, Dr. Rajiv Ranjan Sahay, Electrical Engineering for offering me the opportunity to work on this project and continuously guiding me throughout the work. He has been a constant source of inspiration and the work in this project is far from complete without his guidance. I am also thankful to him for his constant support and for his invaluable time he had provided me with. I would also like to thank him for providing every resource that was required during the work.

Arun Patro

13EE35026

Abstract

Image Processing is a very vital technology in the modern age, with far fetched applications in all sectors like Healthcare, Military, Cinema, Photography, Engineering, etc. Often images get distorted and degraded. It is of prime interest to analyse these images and attempt to reconstruct and restore then. Blur, a type of distortion, can occur in images due to optical defocusing, camera shake or relative motion between camera and object. Due to varying object distances for a constant focal length of a camera lens, different parts of the image can be blurred with different intensities. This is a problem of non-uniform blur in images, also referred to as spatially variant blur. Similarly, when a camera shakes, different parts of the image have moved in different directions with different amounts. If we know the blur kernel at every pixel, we can deblur the image with a better accuracy. This project tries to estimate the blur kernel of a blurred image (motion and defocus) at patch-level of 32x32 in original images. This project is modelled as a classification problem[1] with finite classes that are labeled as different kernels. We use Convolutional Neural Networks and the deep learning methodology to predict the blur kernel using a softmax criterion.

Contents

1 Introduction

Digital Images that are captured by a camera often come out blurry and noisy. This can happen due to many factors like camera shake during exposure, movement of objects in the scene in long exposure timings, inaccurate focal length adjustment of the lens, etc. Removing the effect of these disturbances from images has long been an important research topic in computer vision and image processing. The blurred image can be modelled by:

$$G(x,y) = h(x,y) * F(x,y) + n(x,y)$$

where h represents the Blur Kernel (spatially invariant) that causes the image to get distorted, and n represents white gaussian noise.

Deblurring blurred images to obtain sharper images and restoring its features has wide applications in medical imaging, video conferencing, space exploration, x-ray imaging, cinema, and every other engineering discipline.

The goal of image deblurring is to recover the sharp image F given its blurry observation G. One method for deblurring is to use blind deconvolution, but this is a severely under-constrained problem since there are more unknowns than available measurements. For a blurry image G, there are infinitely many pairs of h and F that satisfy. Blind Deconvolution algorithms use iterative methods like maximum a posteriori estimation and expectation-maximization algorithms. These are used to estimate a good blur kernel (Point Spread Function).

However, in a practical scenario where the motion of the object, or the camera jitter is not linear, the motion blur experienced at different parts of the image will be different and will have a unique blur kernel at every pixel. For defocused images, when the camera tries to capture a 3-D scene onto a 2-D screen, objects that are out of focus will appear blurred. All of these are an example of Non-Uniform Blur that have spatially varying blur kernel. Therefore it is important to deblur parts of the image individually and reconstruct the entire image, as the blur kernel is varying from pixel to pixel.

There have been many non-uniform deblurring techniques [3][4][5] over the years. Blur kernels can also be estimated by looking at image statistics[2], the blur spectrum[6], or with a learning approach using hand-crafted features[9]. Other techniques like[10][11] jointly estimate the sharp image and blur kernels using a sparsity prior.

In this project we use try to estimate the blur kernel for the spatially blurred images for optical gaussian blur using Deep Learning similar to the solution proposed by Sun et. al.[1]. The task of blur kernel estimation is modelled as a mutli-class classification problem.

2 Aim and Approach

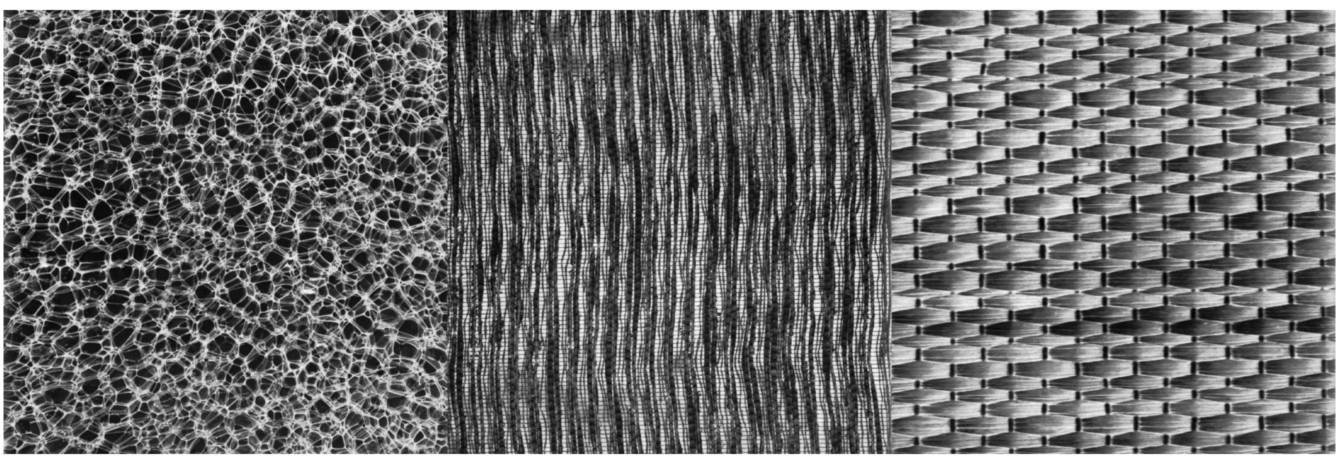
The aim of this thesis is to develop a model for estimating the blur kernels using Convolutional Neural Networks (CNNs), and an optimal classifier to estimate the blur kernel at each patch/pixel. We start with estimating the kernel value at each 32x32 patch. After obtaining the predicted sigma map for the the blurred images, we may apply a smooth function to correct the transitions. However this smoothing doesn't work that well for natural images.

3 Implementation

3.1 Dataset

Since we want to train our model to learn the different blur intensities, we need a dataset of sharp images that are not blurred from before. This is achieved by selecting the Brodatz Textured dataset that is sharp and contains 112 grayscale images. As some of the images have blank spaces of size greater than 32x32, we remove them out. We choose only 80 images of the Brodatz dataset. Each image is blurred using a sigma value from [0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3.0] globally, using *imgaussfilt(I,sigma)* function in MATLAB.

Non overlapping patches of 32x32 are sampled from these images, that are used to train a Convolutional Neural Network with 10 classes [1, 2, … 10] that correspond to sigma [0.3, 0.6, … , 3.0] respectively.



Some textures from the Brodatz dataset

3.2 Framework

The neural networks were implemented on Torch, a scientific computing framework with wide support for machine learning algorithms. Computation was accelerated using GPUs from Nvidia.

3.3 Network Architecture

We used a pre trained CNN i.e. LeNet[7] trained on MNIST data to finetune our model. As the number of layers proved to be insufficient to learn the dataset, we add more layers to the net. We ended up with 8

Convolutional layers and 4 Fully Connected Layers. This network has 3.27 Million parametes. We choose a shallow net over a large network like AlexNet[8] because it would unnecessarily overfit.

4 Work Progress

4.1 Results

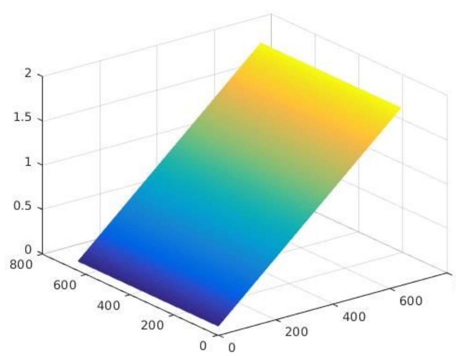
The model of 3 Million parameters was able to learn upto 98% training accuracy but performed bad on the test dataset at 50% accuracy. It performed poor on the validation dataset consistently stagnated at 45%. By increasing the generalizing and introducing dropouts[12] and batch norm[13] layers for reducing overfit, the training time increased tremendously. We decided to adopt larger convolution kernels and more layers. The larger network had 5 Million parameters and 1.5 longer. This too converged fast and up to 100% accuracy. It however failed on the sigma map generation. We conclude that larger nets are able to largely overfit the data and shallower nets can achieve the purpose more efficiently.

4.2 Evaluation on Textured images

On evaluation on ramp and sin blurred images, we see better results with finer quantizations. Hence, we move on with 30 classes.

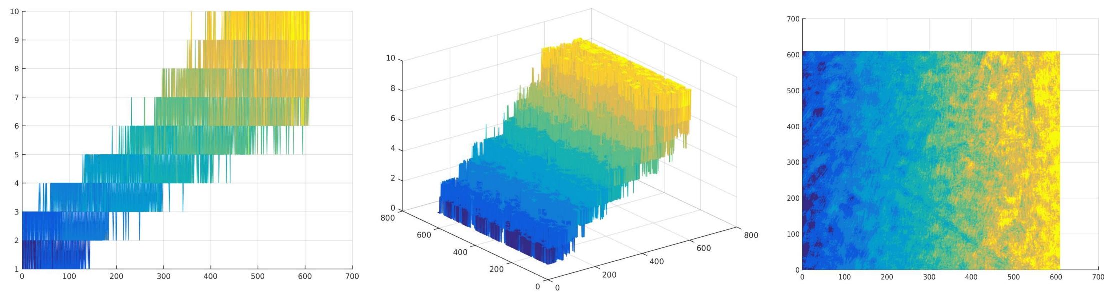
4.2.1 Ramp Blurred

Blurred Image



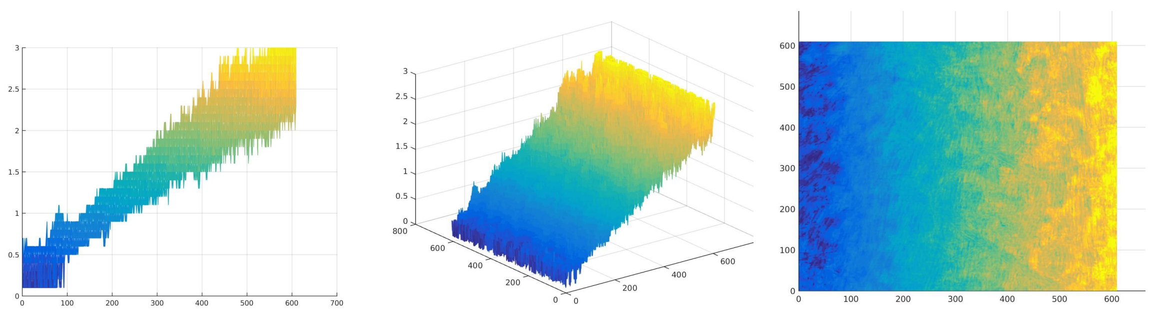
Ground Truth

Model with 10 classes:



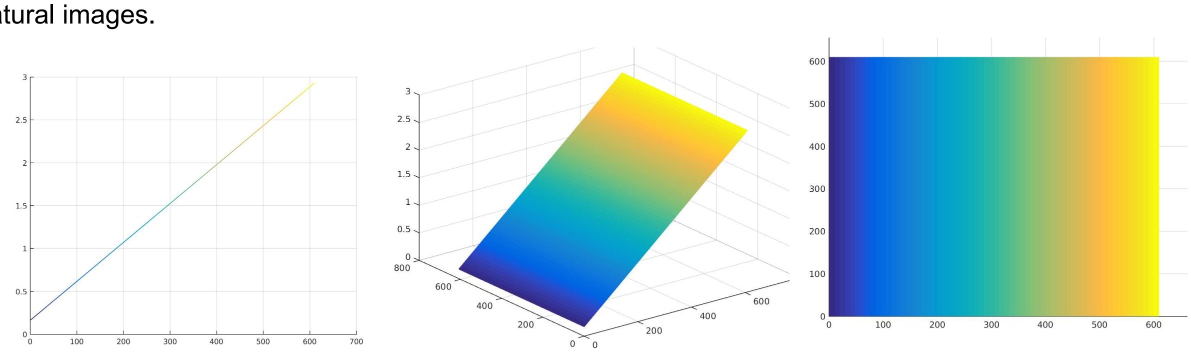
MSE: 0.0552 | **RMSE**: 0.2350

Model with 30 classes:



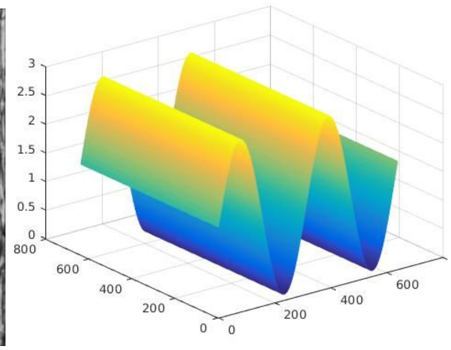
MSE: 0.0270 | **RMSE**: 0.1645

The coarse predictions can be smoothed using the *smooth* function in MATLAB that averages along the gradient of the image. This can give good results only for ramp blurred images and performs poorly on natural images.



MSE: 1.3506e-08 | **RMSE**: 1.1621e-04

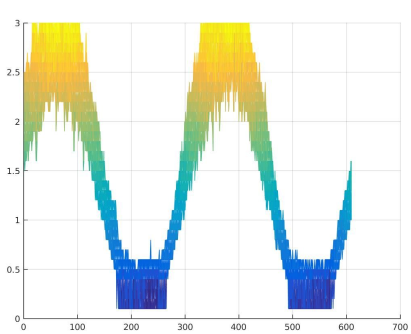
4.2.2 Sin Blurred

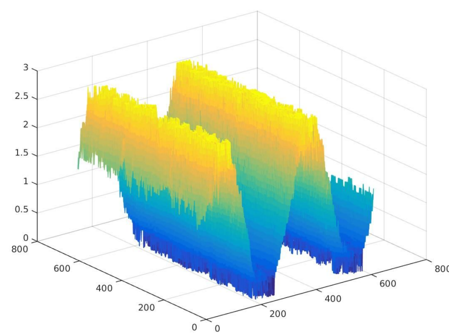


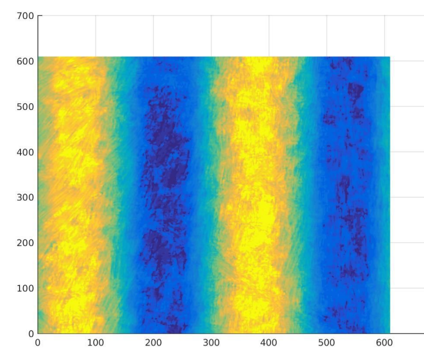
Blurred Image

Ground Truth

Before Smoothing:

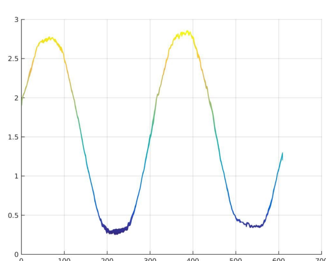


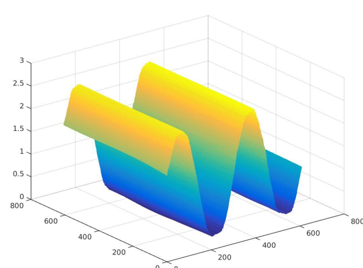


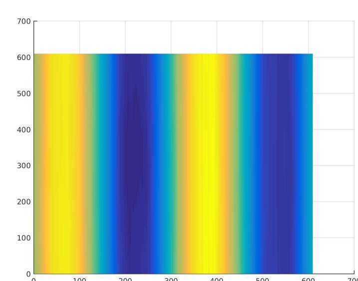


MSE: 0.0397 **| RMSE**: 0.1991

After Smoothing:





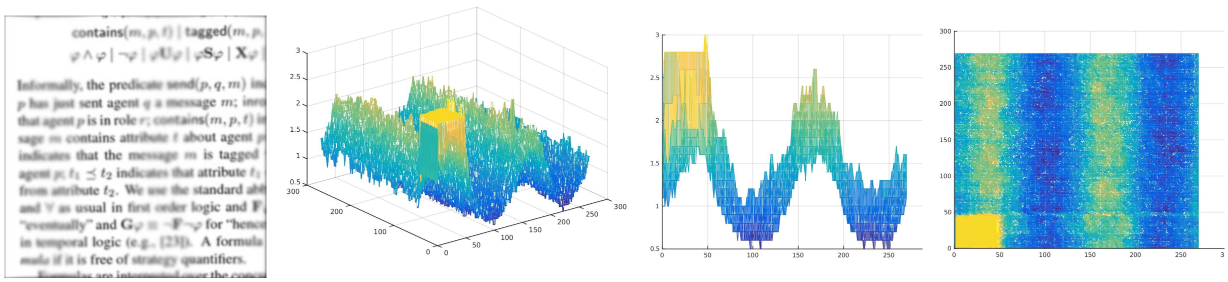


MSE: 0.0177 | **RMSE**: 0.1331

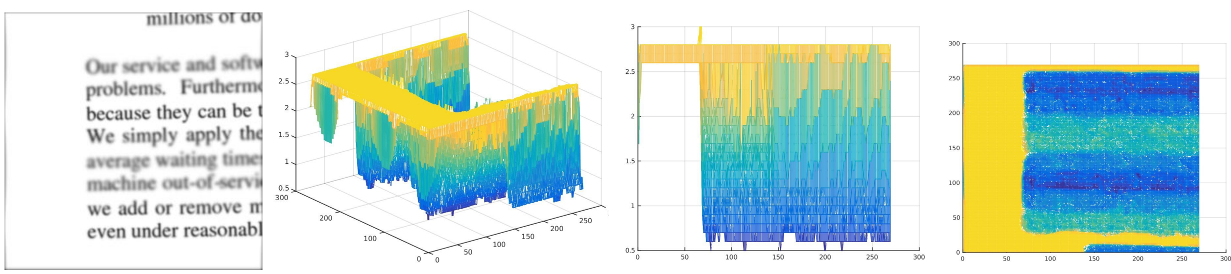
4.3 Evaluation on Text Images

The model performs well on text images for different sin blurs in both x and y axes. We observe that for white backgrounds, the model predicts it as having large sigma.

Example 1:



Example 2:



4.4 Evaluation on Natural Images with defocused background

We see that the model is able to coarsely predict the sigma map. Since the evaluation is done patchwise, we lose the sharpness of the images. Hence, object boundaries are lost.

4.5 Evaluation on Natural Images with defocused foreground

The model is able to quantify the blur of the defocused foreground. However we will not be able to distinguish between the defocused foreground (fences) and far background (sky).

5 Conclusion

We can see that using a CNN, we are able to coarsely predict the sigma map of a space variantly blurred image. This can act as an initial estimate in deblurring the blurred image. As this is modelled as a classification problem, the resolution of sigma is 0.1. This is not too low, as a mismatch of sigma value of 0.1 can introduce ringing effects while deblurring. This coarse prediction can be smoothened using a moving average filter to produce more continuous results. One may use Markov Random Fields for correcting.

Ideally Blur Kernel Estimation is a regression problem. In the future, we will be trying to estimate the Blur Kernel Value using Fully Convolutional Networks.

6 References

[1] Sun, J., Cao, W., Xu, Z. and Ponce, J., 2015, June. Learning a convolutional neural network for non-uniform motion blur removal. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 769-777). IEEE.

[2] Levin, A., 2006. Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems* (pp. 841-848).

[3] Ji, H. and Wang, K., 2012, June. A two-stage approach to blind spatially-varying motion deblurring. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (pp. 73-80). IEEE.

[4] Kim, T.H. and Lee, K.M., 2014, June. Segmentation-free dynamic scene deblurring. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2766-2773). IEEE.

[5] Paramanand, C. and Rajagopalan, A.N., 2013. Non-uniform motion deblurring for bilayer scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1115-1122).

[6] Chakrabarti, A., Zickler, T. and Freeman, W.T., 2010, June. Analyzing spatially-varying blur. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (pp. 2512-2519). IEEE.

[7] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), pp.2278-2324.

[8] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[9] Couzinie-Devy, F., Sun, J., Alahari, K. and Ponce, J., 2013. Learning to estimate and remove non-uniform image blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1075-1082).

[10] Kim, T.H. and Lee, K.M., 2014, June. Segmentation-free dynamic scene deblurring. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2766-2773). IEEE.

[11] Xu, L., Zheng, S. and Jia, J., 2013. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1107-1114).

[12] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), pp.1929-1958.

[13] Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.