Non-uniform Blur Kernel Estimation via Adaptive Basis Decomposition

Anonymous Authors[1]

Abstract

This research focuses on comparing, testing and improving deblurring algorithms. We have implemented the following approaches: 1) Non-uniform Blur Kernel Estimation via Adaptive Basis Decomposition, 2) DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks. In our project, we have conducted several experiments to deblur images from GoPro and Kohler data sets. This article presents all the steps we have taken and all the results obtained, as well as our conclusions about the effectiveness of each approach and some suggestions for further improvements.

1. Introduction

Motion blur estimation and restoration are fundamental problems in digital image processing and computer vision. Motion blur is typically produced by camera or subject movement, inaccurate focusing, low resolution of an aperture, out-of focus and so on. Blurring causes significant degradation in image quality. There may be consequences such as aesthetically blurry images and performance degradation of subsequent computer vision tasks: tracking, detection, classification, etc. In order to get a clearer picture we can either reshoot the same photo or use our knowledge of Machine Learning and reproduce a deblurred image.

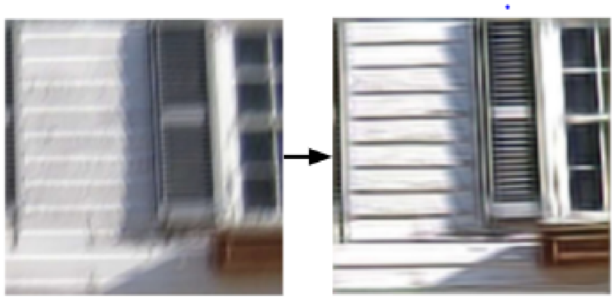


Figure 1. An example of blureed and deblurred images.

In this work we focus on the realistic non-uniform motion blur estimation. Major goal of our study: provide dense, accurate estimates of non-uniform motion fields via local kernel estimation. Estimate kernels at the pixel level. Secondary goal: once the motion field has been estimated, perform non-blind image deblurring.

The literature on motion kernel estimation is extremely vast. We limit the current analysis to spatially varying kernel estimation methods from a single image. Early methods attempting to estimate non-uniform motion blur kernels are limited to the case of camera egomotion.

The common formulation of non-uniform blur model is the following:

$$I_B = k(M) * I_S + N$$

where I_B is a blurred image, $k(M)$ are unknown blur kernels determined by motion field M. I_S is the sharp latent image, * denotes the convolution, N is an additive noise.

Among all known digital image processing methods, in this study we will examine the following:

- Non-uniform Blur Kernel Estimation via Adaptive Basis Decomposition;
- DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks.

In this study, we solved the following tasks:

- Use pre-trained Non-uniform model and compute PSNR and SSIM metrics for both Kohler and Gopro datasets. Compare results with the reference.
- Use pre-trained DeblurGAN model and follow the same steps. In case if results are worse, we can try to retrain a model.
- Write a code to train a Non-uniform model. Plot training and validation metrics depending on number of epoch. Compare results.
- *In case of failure in task 3, try to reproduce tables 1 and 2 from the DeblurGAN paper and train the networks from scratch.

1.1. Non-uniform Blur Kernel Estimation via Adaptive Basis Decomposition

The detailed exploration of this method is described in (G. Carbajal, 2021). Shortly, given a blurry image, a neural network predicts a set of kernel basis and corresponding pixel-wise mixing coefficients allowing to deblur the corresponding sharp image by first convolving it with each basis kernel and performing a weighted sum using the mixing coefficients.

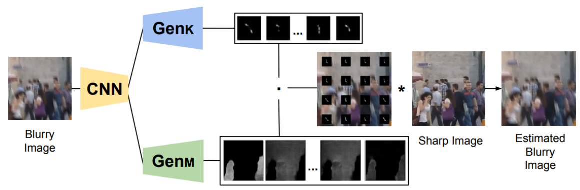


Figure 2. Overview of an Adaptive method.

Fig.2 demonstrates a principle of operation of an adaptive method, we can observe that the model calculates mixing coefficients and different kernels for different objects in a given image. We want to emphasize that this model works well for digital images in which some objects are moving, some are stable.

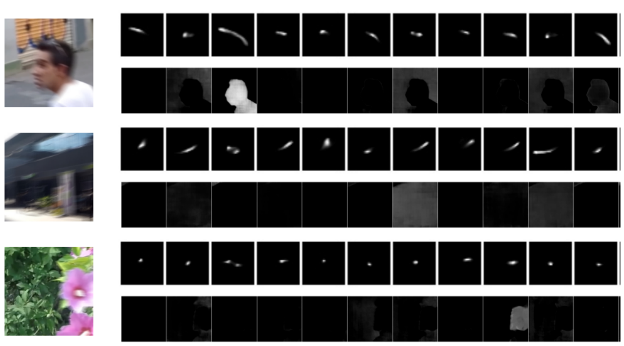


Figure 3. Examples of generated kernels and corresponding mixing coefficients.

Fig.3 shows examples of generated kernel basis and corresponding mixing coefficients predicted from the blurry images shown on the left. The adaptation to the input is more notorious for the elements that have significant weights.

The proposed network is composed by an encoder and 2 decoders. The encoder takes as an input a blurry image. Decoders output the motion kernel basis and corresponding mixing coefficients. Fig.4 demonstrates an architecture of the network:

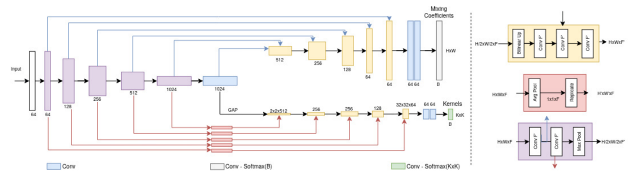


Figure 4. Architecture Details.

The proposed network is composed by an encoder and two decoders. The encoder takes as an input a blurry image. The decoders output the motion kernel basis and corresponding per-pixel mixing coefficients

1.2. DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks

DeblurGAN – an approach based on conditional generative adversarial networks and a multi-component loss function. For this method, Wasserstein GAN with the gradient penalty and perceptual loss are used. This solution allows to restore finer texture details than if using traditional MSE or MAE as an optimization target. A detailed ixplanation of this method is presented in a paper (Orest Kupyn, 2018).

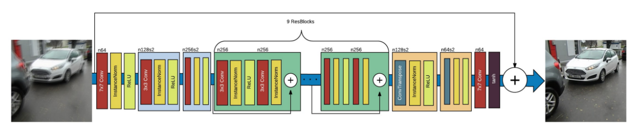


Figure 5. DeblurGAN generator architecture.

Fig.5 demonstrates a DeblurGAN generator architecture. DeblurGAN contains two strided convolution blocks with stride 12, nine residual blocks and two transposed convolution blocks. Each ResBlock consists of a convolution layer, instance normalization layer, and ReLU activation.

2. Conducted Experiments

2.1. Research Tasks

In this task we were asked to write a code for a training process of a Non-uniform Blur Kernel Estimation Neural Network, since authors of a corresponding work have not provided an initial training code for an open access. The task can be divided into two parts:

- Creating a non-uniform blurred dataset for training;
- Reproduction of the training model itself.

The training process of a neural network takes place as follows: a synthetically blurred image, corresponds sharp image and blurring kernels are fed to the input. The model is processing the blurred image and estimates the set of baseline blurring kernels and mixing coefficients, linear combination of which reproduces the set of per-pixel blurring kernels of the image. After that, reblur and kernel loss functions are calculated. For the repaired model, these two functions were repaired using PyTorch.

The reproduction of the non-uniform training model and the generation of a training dataset was too long, so we decided to train the DeblurGAN Neural Network in parallel. It was easier to do so, because the initial code was available in the developers' github repository. To perform a training model, the training dataset was generated. The program provided by developers takes at an input a blurred image and a corresponding sharp image, after that merges them together. The resulting image will be used for the training. 2000 image pairs of the GoPro training dataset were taken as the basis. Blurred images with the modified gamma correction factor were taken into consideration, because they have greater blurring degree. Due to the structure of the GoPro dataset, which consists of many subfolders with images of different nature, the dataset making program provided by the developers was modified to work with the specified structure. As the result, we have obtained the following dataset:

Figure 6. The synthesized dataset.

The following figures demonstrate an example of blurred, restored and sharp image during one epoch. We took to samples do demonstrate the difference in the model accuracy during its learning:

Figure 7. An example of obtained results for the 1st epoch.

Figure 8. An example of obtained results for the 58th epoch.

The resulting dataset was loaded to the training model, which was aslo modified for a multi-fold structure. The training lasts for 300 epis, during which various model quality metrics were captured, including PSNR (Peak Signal to Noise Ratio). We trained our model during 58 epochs. WE have concluded that the model quality increases during the learning process.

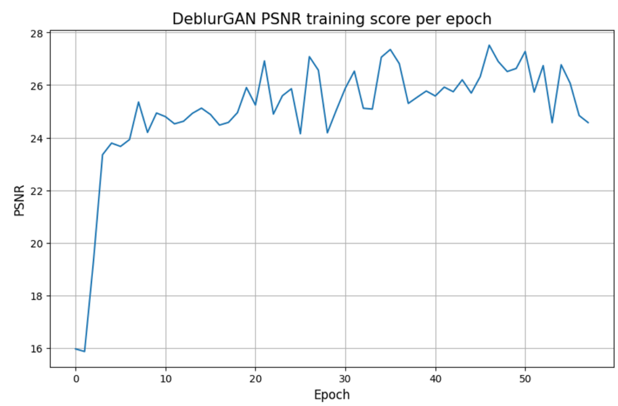


Figure 9. The model training process.

2.2. Replication Tasks

Firstly, we decided to measure the deblur performance of an adaptive model. For this goal, we used a kohler dataset.

Kohler is a dataset that was created using static objects (printed photo posters) and a monopod located on a moving platform and having 6 axes of movement (3 rotating and 3 translating). It would not be entirely correct to compare the original images with those obtained due to some algorithm, since different resolutions and lighting would contribute to the variability of the metrics. In this way, reliable images along the trajectory were generated by the mobility of the platform, reproducing the camera trajectory step by step and taking a picture. Using this strategy, a dataset was formed of 12 camera shakes, 4 images and 200 ground truth images along the trajectory. The resulting kernels (1,3,5,7,9,11) are non-uniform, (2,4,6,8,10,12) are approximately uniform.

To compare similarity between 2 images A and B (represented as vectors), we first estimate the optimal scaling a and translation T such that L2 norm between A and B becomes minimal, after that we calculate PSNR as:

$$PSNR(A, B) = 10log_{10} * (m^2/||A - T(aB)||^2)$$

where m is a maximal possible intensity value, i.e. m =255 as we work with 8bit encoding. Given a sequence of ground truth images U*, along the trajectory, we define the PSNR similarity between an estimated image U and the ground truth as the maximum PSNR between U and any of the images along the trajectory.

$$SIM = maxPSNR(U, U*)$$

GoPro is a dataset created by cutting video into frames. It is important to note that most of the photos were rendered with motion blur due to camera movement, but there are also frames with objects moving in different directions, it is difficult to assess the ratio of uniform and non-uniform. Also, there were no intermediate ground truth images, so standard formulas of MSE and PSNR were used.

SSIM was calculated using the following formula:

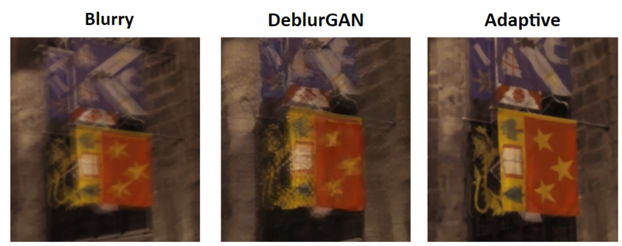
$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

The results of our calculation are presented in the following table:

Method	KOHLER		GOPRO	
	PSNR	SSIM	PSNR	SSIM
DeblurGAN	25.97	0.75	32.12	0.98
Adaptive	26.86	0.80	31.21	0.91
Reference value	28.39	0.82	NA	

According to the results obtained, it can be seen that on KOHLER, the Adaptive non-uniform dataset showed better results than DeblurGan in both metrics. But the result obtained by us still did not coincide with that indicated in the article. To validate the implemented algorithm for calculating PSNR and SSIM, we took ready-made Images From Hirch Deblurred Dataset and Blurry image From Kohler and calculated metrics for them. The result coincided with the expected, so we tried to run the Adaptive model with a different gamma parameter (in the article it was indicated that gamma = 1.0 on the Kohler dataset, the standard value for the model is 2.20), this also did not allow us to achieve the result indicated by the authors. Received values picture by picture

PSNR Expected	27.58	33.16	NA	35.19
PSNR Calculated	27.58	33.16	26.98	34.47



On the Gopro dataset, the adaptive model showed a worse result than deblurgan, this can be explained by the fact that there is a smaller percentage of images with non-uniform blur than in the above-mentioned Kohler.

lpuhjg

Discussion of the Results

Acknowledgements

References

G. Carbajal, P. Vitoria, M. D. P. M. J. L. Non-uniform blur kernel estimation via adaptive basis decomposition. Technical report, 2021.

Orest Kupyn, Volodymyr Budzan, M. M. D. M. J. M. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.