DeepBlur

Andy Gilbert, Shai Messingher, Paul Westhoff

emails {adgil, shaim, westhoff}@stanford.edu

Introduction

- Image deblurring is a highly relevant problem: **~1.2 trillion photos taken in 2017**, 85% on smartphones.
- Neural Networks give the ability to infer complex priors on images.

Dataset

- Used subset of ImageNet dataset (1153 images)
- Created four **blur kernels** per image using Gaussian sampling from a mechanical motion process [1][3].
- Added Gaussian and Poisson noise to each image

Method

1. Computed 15 deconvolved versions of the input blurry image using Wiener filtering (Fig 1). Stacked them.
2. Added original blurry image to the stack. Used this 16-level stack into the CNN (see Fig 2)
3. Used **SSIM loss** and **PSNR** for measure of accuracy
4. Applied **bilateral filter** to output of NN

Features

- 15 filtered images and one blurred image each with 3 color channels

Training

Hyperparameter	Min	Max	Optimal
Learning Rate	1e-5	1e-2	5 e-4
Filter Size	3	5	5
Dropout Rate	.75	.9	.8
Num ResBlocks	12	19	12
Batch size	2	32	32
Batchnorm	Equivalent results, removed to reduce necessary parameters		
Input normalization	Better without input normalization		
Bilateral Filter	window size = 7 $\sigma_{color} = 40$ $\sigma_{space} = 10$		

Loss

$$L_{SSIM} = \frac{1}{n*c*w*h} * \sum_{i=0}^{n} \sum_{p \in P} -SSIM(p)$$

Where:

$$SSIM(p) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \times \frac{2\sigma_{xy} + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \quad (3)$$

μ_x = average of first image window
μ_y = average of second image window
σ_x = standard deviation of first image window
σ_y = standard deviation of second image window
$c_1 = (k_1 R)^2$
$c_2 = (k_2 R)^2$
R = dynamic range of pixels within the window

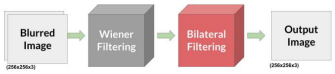

Fig 3. Control Architecture

Results

- Our network provides higher PSNR when compared to traditional Wiener Filtering (26.1 dB vs 16.6 dB)

Discussion

- Proved hypothesis that non-linear combination of stack of simply-deconvolved versions of an image can form a sharper reconstruction.

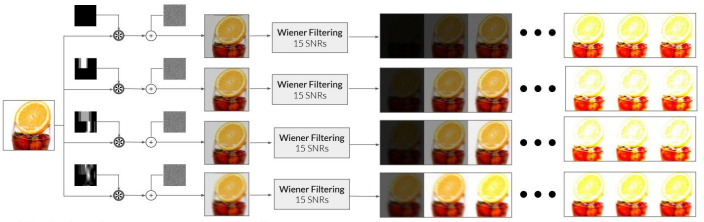

Fig 1. Our feature creation pipeline

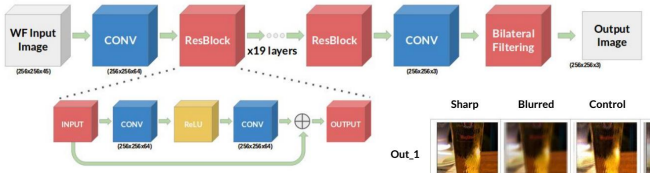

Fig 2. Our NN Architecture

Future Work

- We attempted to incorporate a GAN into the project with the hopes that the loss function could contain both SSIM and an adversarial loss component (proposed by [2])
- We would like to investigate alternative architectures and transfer learning as our GAN failed to learn the difference between sharp and blurry images
- Additionally our project would benefit from a thorough hyperparameter search as ours was limited by the project timeline.

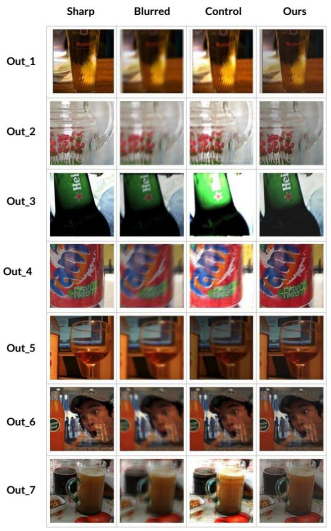

Fig 4. Our results

1. Schuler, Christian J., et al. "Learning to deblur." IEEE transactions on pattern analysis and machine intelligence 38.7 (2016): 1439-1451.
2. Nah, Seungjun, Tae Hyun Kim, and Kyoung Mu Lee. "Deep multi-scale convolutional neural network for dynamic scene deblurring." arXiv preprint arXiv:1612.02177 3 (2016).
3. http://home.deib.polimi.it/boracchi/Projects/PSFGeneration.html