Andy Gilbert
adgil

Anirudh Patel
anirudh2

Shai Messingher
shaim

Training a DNN to Deblur Images

Motivation

Imaging research groups at universities and consumer electronic companies have been inclined to develop systems that automatically implement the image processing pipeline (IPP) using DNNs. For this project, we will focus on the deblurring block of the IPP, and build a DNN-based algorithm that is able to deblur images without knowing the blur kernel. That is given a blurry image y, our goal is to estimate the latent image x given by:

$$\mathbf{y} = \mathbf{k} * \mathbf{x} + \mathbf{n}.$$

Where k is the blur kernel and n is the noise estimate.

Related Work

Pre-neural network-based research in this field has been compiled succinctly by Wipf et al., who compared the Variational Bayes and MAP estimation methods from earlier papers and showed that, contrary to popular opinion at the time, in an ideal setting, Variational Bayes and MAP estimation have an identical underlying cost function [2]. We refer the interested reader to this paper.

Here, we focus on how neural networks (NNs) have been previously used for deconvolution. Image reconstruction has been approached via NNs for a while now. These approaches vary in scope. For example, Khare et al. use NNs to identify the type of blur [3], and Tansley et al. propose an architecture to learn an inverse filter to deblur text [4]. However, all of these were trained on and apply to only one image class. We are basing our project on the work by Schuler et al. [1]. The work done in this paper attempts to learn how to deblur for a general image, not just a different image within the same class of images.

Project Overview

We will use a subset of images from imageNet with similar content as the data set for this project. Using content specific training will allow us to realize the effectiveness of this algorithm while minimizing the expense associated with training. We have yet to finalize the exact subset but will use one with the most images. In order to augment the size of this dataset we will randomly crop the images. After creating the augmented dataset of sharp images, we will apply randomly generated blur kernels to create the blurred image inputs. The blurred kernel will be estimated by randomly generating x and y motion coordinates from a Gaussian process [1]. We need to determine the maximum blur kernel we will apply as this algorithm has proven ineffective on blurs from exceedingly large kernels [1].

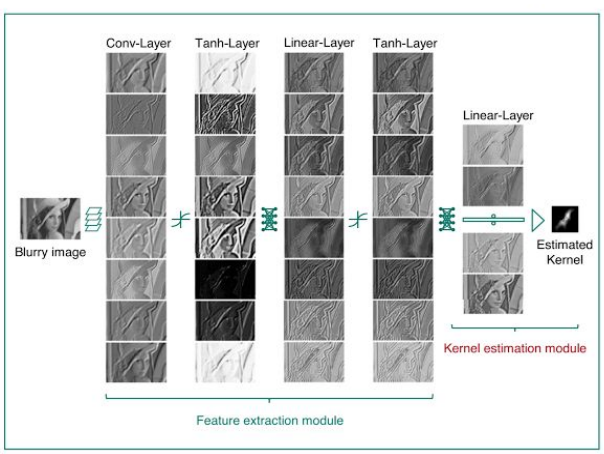

Fig 1: Feature extraction pipeline from [1] used to estimate the blur kernel of a blurred image. This extraction is iterated along with image processing operations..

The deblurring block consists of 3 steps. The first step uses a DNN to extract relevant features of the image. It extracts the features and then approximates multiple copies of the blurred and latent images as separate linear combinations of the output of a tanh activation layer. The second step is the kernel estimation module. Given the estimates obtained from the previous step the kernel is obtained by minimizing the cost function show below where $\overline{x_i}$ and $\overline{y_i}$ are the estimates of the latent and blurred images respectively.

$$\sum_i \|\tilde{\mathbf{k}} * \tilde{\mathbf{x}}_i - \tilde{\mathbf{y}}_i\|^2 + \beta_k \|\tilde{\mathbf{k}}\|^2$$

The final step is the image reconstruction piece. Once we know the blur kernel this can simply be estimated from:

$$\|\tilde{\mathbf{k}} * \tilde{\mathbf{x}} - \mathbf{y}\|^2 + \beta_x \|\tilde{\mathbf{x}}\|^2$$

The output of this stage is then compared to the known latent image and we can backpropagate through each stage of the pipeline to learn the appropriate feature extraction weights.

We will be basing our algorithm on Schuler et al's research [1]. They provide a review of relevant techniques in the field, as well as an explanation of how to "unroll" the IPP. Schuler et al implemented a system processing 10 million images using a system developed in C++/CUDA. We will implement a replica of their system using PyTorch and content specific images rather than images from the entire ImageNet Dataset.

We will evaluate the success of our algorithms by comparing to deblurring results obtained by traditional signal processing methods.

Milestone, Timeline, & Goals

Due date	Milestone
02/21	Get input data processing pipeline
03/01	Implement CNN
03/07	Implement the full stage
03/12	Training, hyperparameter Tuning & Presentation Preparation.

We will evaluate the results of our deblurring system using several metrics. Qualitatively, we will analyze selected results to determine if we can visualize the image better. We will plot several example results from our dataset as well as plots of how training and testing accuracy changed across hyperparameters such as blur kernel size, number of stages, learning rate, and learning decay. Quantitatively, we will measure the blur using the Peak Signal to Noise Ratio (PSNR) to evaluate the output of our network compared to the input blurred image and ground truth original image.

One challenge of this project will be the amount of time it will take to train it. From literature, we learned it takes ~ 2 days per layer to train. We're using 3 layers, implying ~6 days per iteration [1]. We plan to simplify the deblurring block and number of images to shorten training time. Another challenge we will have to overcome is implementing a DNN, a Kernel Estimation Module, and an Image Estimation Module, all included in the deblurring block, within the timeline available.

At Least 3 Scientific References

[1] Schuler, Christian J., et al. "Learning to deblur." IEEE transactions on pattern analysis and machine intelligence 38.7 (2016): 1439-1451.

[2] Wipf and H. Zhang. “Revisiting Bayesian Blind Deconvolution”. In: ArXiv e-prints (2013).

[3] C. Khare and K. K. Nagwanshi. “Image Restoration in Neural Network Domain using Back Propagation Network Approach”. In: Int. J. Computer Information Systems 2.5 (2011), pp. 25–31.

[4] E. Tansley, M. J. Oldfield, and D. J. MacKay. “Neural network image deconvolution”. In: Maximum Entropy and Bayesian Methods. Fundamental Theories of Physics. Springer, 1996, pp. 319–325.

Note

The DNN feature extraction piece of this project overlaps with the project that Shai Messingher and Andy Gilbert will be completing for CS230 (Deep Learning).