Deep Convolutional Neural Network for Image Deconvolution

Mudit Goyal
Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
Email: goyalmudit24@gmail.com
Roll no. 200070045

Navneet
Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
Email: krnavneet30@gmail.com
Roll no. 200070048

Srushti Bangde
Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
Email: srushti.bangde@iitb.ac.in
Roll no. 200070081

***Abstract*—In this report, we present our implementation of a deep convolutional neural network for solving the problem of image deconvolution. The traditional linear convolution model for image blur is often insufficient due to factors such as camera noise, saturation, and image compression. The paper's approach is to leverage the power of deep learning to capture the characteristics of the degradation process while introducing a novel, separable structure to enhance the robustness of the deconvolution process against artifacts. The network comprises two submodules, both of which are trained in a supervised manner and yield fair performance compared to previous generative-model-based methods. Through our work, we establish a connection between traditional optimization-based approaches and neural network architectures, paving the way for more robust image restoration techniques.**

***Index Terms*—Compression, deconvolution, neural network, robustness, saturation**

I. Introduction

The capacity to recover blurred pictures has long been a topic of interest in computer vision, as it has multiple applications in fields such as surveillance, medical imaging, and satellite imagery. Traditional picture deblurring methods eliminate blur from deteriorated images using mathematical models. These models, however, frequently fail to account for real-world characteristics such as camera noise, picture compression, and saturation, which can result in artifacts and restrict the usefulness of these approaches. Previous research has shown that improperly handling these problems could raise a broad set of artifacts related to image content, which are very difficult to remove. To solve this issue, various past works are dedicated to modeling and addressing each particular type of artifact in non-blind deconvolution for suppressing ringing artifacts, removing noise, and dealing with saturated regions. Deep learning-based techniques have recently demonstrated promising results in picture deblurring applications. The study "Deep Convolutional Neural Network for Image Deconvolution" introduces a unique deep convolutional neural network design to handle picture deblurring difficulties caused by camera noise, saturation, and compression artifacts in this context. The authors create a data-driven approach that uses picture samples that can be easily acquired online to teach the CNN the deconvolution operation without the need for any pre-processing to deblur the image.

However, using existing deep neural networks directly to deconvolution issues yields insufficient results. As a result, the authors link classic optimization-based approaches to a neural network architecture, where a unique, separable structure is presented as a trustworthy support for robust deconvolution against artifacts. The suggested network consists of two submodules, both of which were trained in a supervised manner with correct initialization and produced good results on non-blind picture deconvolution when compared to earlier generative-model-based techniques.

The research offers a promising deep learning technique to address the issues of picture deblurring, which has the potential to outperform existing image deblurring approaches. We want to deploy the suggested deep convolutional neural network architecture and assess its performance on benchmark datasets in this project to further demonstrate its usefulness in real-world applications.

II. Proposal

A. Problem Statement

The objective is to achieve an original clear image from a degraded image and to minimize unwanted artifacts by using a pseudo-inverse kernel implemented via Convolution Neural Network.

B. Mathematical model

1) Blur Model: The blur model is given by:

$$\hat{y} = \psi_b[\varphi(\alpha x * k + n)] \tag{1}$$

In this equation, αx represents the latent sharp image. Also, we have $\alpha \geq 1$ to signal the fact that αx may have values that surpass the dynamic range of camera sensors and so be clipped. k is the known convolution kernel, or typically referred to as a point spread function (PSF), n models additive camera noise. $\varphi(.)$ is a clipping function to model saturation, defined as $\varphi(z) = min(z, zmax)$, where zmax is a range threshold. $\psi_b[.]$ is a nonlinear (e.g., JPEG) compression operator. The generation of blurry image from an input x is quite straightforward by image synthesis according to the convolution model taking all kinds of possible image degradation into generation.

The goal is to train a network architecture $f()$ that minimizes

$$\frac{1}{2|N|} \sum_{i \in N} ||f(\hat{y}_i) - \hat{x}_i||^2 \tag{2}$$

where $|N|$ is the number of image pairs in the sample set.

2) Image deconvoluton: Image Deconvolution CNN (DCNN) We describe our image deconvolution convolutional neural network (DCNN) based on the separable kernels. This network is expressed as

$$h_3 = W_3 * h_2 \;\; h_l \quad = \sigma(W_l * h_{l-1} + b_{l-1}), \quad l \in 1,2 \;\; h_0 = \hat{y} \tag{3}$$

where W_l is the weight mapping the $(l-1)$th layer to the lth one and b_{l-1} is the vector value bias. $\sigma(\cdot)$ is the nonlinear function, which can be sigmoid or hyperbolic tangent.

III. Methodology

A. Defocus Blur

We have implemented a function `defocus_blur` that applies a defocus blur effect on an input image. The function takes two parameters as inputs:

- image: The input image that needs to be defocused.
- Amount: The amount (in variance) of the defocus blur effect applied on the image.
  The function first creates a kernel matrix of size `kernel_size x kernel_size` and uses Gaussian blur depending on the variance input to create defocus blur.

In our approach, we are loading image files from a specified directory using the glob library. It then reads each image as an RGB image using OpenCV (cv2) and scales the pixel values to be between 0 and 1. Next, it applies a defocus blur to each image using the "`defocus_blur`" function. The original and defocus-blurred images are stored in separate numpy arrays (`original_images` and `defocus_blurred_images`) for use in getting back the original image.

B. Model

We made multiple patches and split them into 8:2 for training and validating the model. There are two subnets in the model: **Deconvolution sub-network**, and **Outlier Rejection Sub-Network**. These two subnets are trained independently.

1) Deconvolution sub-network:
   This sub-network implements the separable kernels using 2 hidden layers. First, we make patches of the image of size (`64 x 64`) and then perform zero padding to produce a patch of (`184 x 184`) which is given as an input to this sub-network. The first layer is produced by passing through a horizontal kernel of size (`1 x 121`). Then this layer is passed through a vertical kernel of size (`121 x 1`). Then we finally convolve it with a (`1 x 1`) kernel to produce an image of size (`64 x 64`).

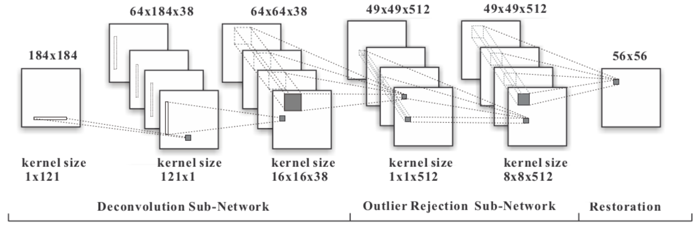

Fig. 1. Complete network architecture for deep deconvolution, Source: [1]

2) Outlier Rejection Sub-Network:
   The two network modules are concatenated in the system by combining the last layer of deconvolution CNN with the input of denoise CNN. This is done by merging the (`1 x 1 x 36`) kernel with 512 (`16 x 16`) kernels to generate 512 kernels of size (`16 x 16 x 36`). Note that there is no nonlinearity when combining the two modules. Training the second sub-network requires patches of size (`64 x 64`) which is increased to (`192 x 192`) using zero padding and is then fed into the network.

C. Training Model

We first trained both models using the patches that we created from the images and then transferred the weights of the models to a different model with the same architecture but with only one difference is that the input shape is now according to a full-scale image of dimension (`800 x 600`. This transfer of weights can be done because the CNN network we used depends only on filters and kernel sizes and is not affected by the input shape of the image.

IV. Results and Observations

After training the above model, a new model is defined which will take zero padded blurred image of size (`800 x 600`) and will output a deblurred image. This model will not be trained and the weights will be directly extracted from the above pre-trained model on patches. This will be our testing of the model. The following are the results:

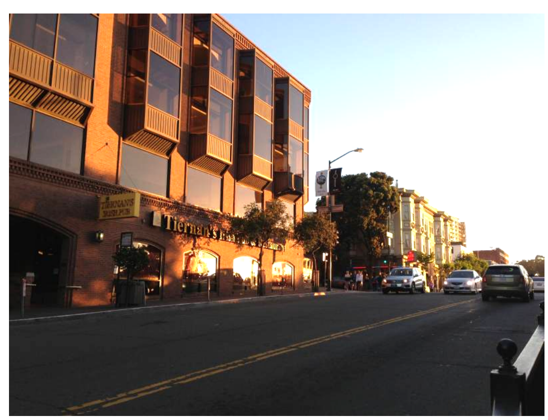

Fig. 2. Sample image from their dataset

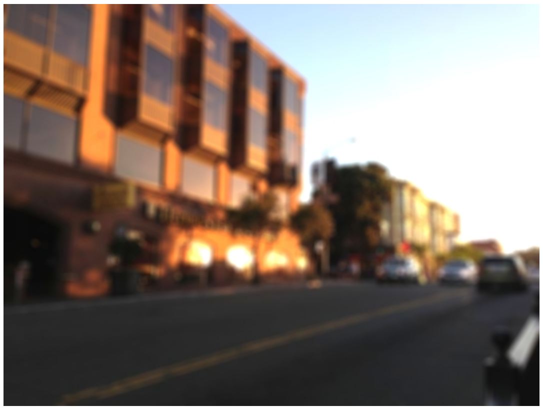

Fig. 3. Input image

Fig. 4. DCNN output without denoise for Image deconvolution

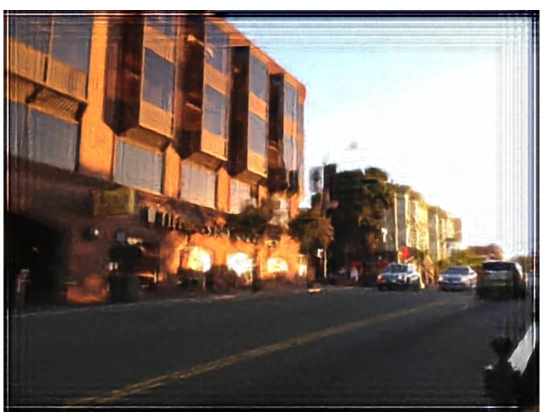

Fig. 5. DCNN for Image deconvolution

V. Future Work

The paper we reviewed uses the point-spread-function to perform custom padding of images, but this part is not well explained in the paper, so for our model, we used zero padding on all the images and patches, and that caused the borders to get a little distorted but the overall image is good enough. This aspect of proper pre-processing of images can be worked upon in the future. We also faced issues with training because of computation constraints as we were using google colab, this model can be improved in the future by increasing the number of epochs and increasing the training dataset

VI. Conclusion

We implemented an existing paper and studied several novel approaches. The approach involved using relatively large kernel support for convolutional neural networks to deal with deconvolution. We proposed a supervised pre-training on the sub-network that corresponds to a reinterpretation of Wiener deconvolution and applied traditional deconvolution to network initialization, which significantly improved performance. Through our work, we demonstrated that a new convolutional neural network architecture can deal with deconvolution. However, without a good understanding of the functionality of each sub-net and performing supervised pre-training, it is difficult to make the network work very well. We showed that the use of sparsely regularized deconvolution can be used to extract useful middle-level representation in our deconvolution network, unifying the process in a deeper convolutional neural network

VII. Key Links

- Project demo video
- Project code

VIII. Acknowledgment

The report has been written for the course EE 769 - Introduction to Machine Learning under the valuable guidance of Prof. Amit Sethi.

References

[1] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/1c1d4df596d01da60385f0bb17a4a9e0-Paper.pdf