Brenda Wang, Yifan Wang
21 February 2023
COEN 344 Computer Vision I
Final Project Proposal

Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising

1. Abstract

The goal of this project is to implement the denoising convolutional network from the TIP2017 paper "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising" using TensorFlow. We'll focus on the first part of the experiment, which is to create the DnCNN (denoising convolutional neural networks) and train it for Gaussian denoising with images of known noise level. After implementing this, we can try different experiments, where we will compare the PSNR (Peak Signal-to-Noise Ratio) of the DnCNN's results on testing data with the algorithm the authors describe in the paper.

We plan to first explore the robustness of this algorithm by testing it with images corrupted by noises beyond Gaussian noise to see how well the proposed method would perform on images with different types of noise. Since the authors mention that "using a larger training dataset can only bring little improvement," we want to investigate the DnCNN's performance when trained with smaller sized datasets to find the highest ratio of performance to number of images trained with. Lastly, since the layers of the DnCNN described in the paper have high computational cost, we will try to adjust the layers of the CNN to observe the trade-off between time complexity and accuracy. We will discuss the problem in detail in the Problem Statement section. We expect to complete this project within three weeks.

2. Problem Statement

Image denoising is a fundamental problem in computer vision as the quality of the denoising can have impacts on the performance of later tasks. Many denoising algorithms have been proposed over the years, but deep learning-based methods have recently shown great promise in achieving state-of-art performance. The paper, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for image Denosing," proposes a novel approach using deep convolutional neural networks that achieved top results on several benchmark datasets. Gaussian noise is a simplified form of noise commonly used in denoising research. The authors only used Gaussian noise to train and evaluate their model, but, in the real-world, images can be affected by a variety of types of noise, such as salt and pepper noise, quantization noise, or signal-dependent noise. It is unclear how well the proposed method would perform on these types of noise.

The authors trained their model with a total of 400 images and noted that training with a larger dataset would only make slight improvements to the result. We want to investigate how the model will perform when it is trained with a smaller number of images to see if there is a peak in performance at a certain sized dataset. If the model trained with smaller datasets still has slightly different results, we can use this algorithm with less datasets to save time.

The proposed method uses a deep residual network with many layers (17), which results in a high computational cost. This can be a limitation for applications that require real-time processing or low-power devices. Thus, we will try to decrease the number of layers or features in the CNN to study the trade-off between time complexity and accuracy.

We aim to investigate these questions in our project.

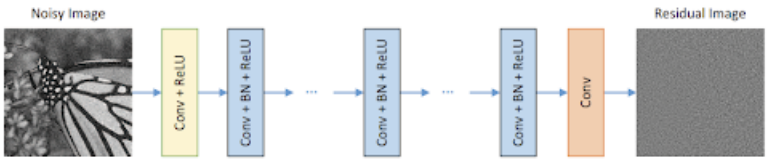

Figure 1: Architecture of the DnCNN. (Reference [1])

3. Methodology

Architecture

The paper, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for image Denosing," proposed a novel approach to image denoising using deep convolution networks. The author introduces a deep residual network called DnCNN, which consists of several convolutional layers with batch normalization and ReLU activation. The architecture is shown in Figure 1.

There are three types of layer for this architecture:

- Conv+ReLU: for the first layer, 64 filters of size $3 * 3 * C$ are used to generate 64 feature maps. C = 1 for gray image.
- Conv+BN+ReLU: for layer 2-(D-1) (D: Depth), 64 filters of size $3 * 3 * 64$ are used, and batch normalization is added between convolution and ReLU.
- Conv: for the last layer, C filters of size $3 * 3 * 64$ are used to reconstruct the output.

Unlike a traditional denoising algorithm which tries to find the latent clean image, the DnCNN algorithm adopts the residual learning formulation to train a residual mapping. This algorithm also adopts batch normalization to speed up and stabilize the training process.

Dataset

We will implement the experiment described in the paper using TensorFlow.

The dataset we will use is BDS500 (Berkeley Segmentation Data Set and Benchmarks 500) published by the UC Berkeley Computer Vision Group. It contains 500 RGB images, 400 for training and 100 for testing. As preprocessing, we will resize the images to 180*180 and add (Gaussian) noise to them. For later exploration, we'll add other types of noise on the images.

Software Requirements

Tensorflow >= 1.4

Numpy

Opencv

Experiments

We will conduct several experiments to evaluate the performance and robustness of the algorithm. First we will train and test the algorithm to implement the model described in the paper. We'll report the measured PSNR of the testing dataset by using TensorFlow to ascertain that the paper's model was implemented correctly.

Next, we will change the noise on the testing data to other kinds and will use the images to test the model in order to evaluate the performance of the original model on different types of noise and investigate how the measured PSNR changes as the noise types change.

Then, we will train from scratch and test the model with a reduced size of the training dataset (for example 200 images) while keeping the size of the testing dataset (100 images) the same. We will

compare the performance of the model with the original algorithm, and investigate how the resulting PSNR measurements change as a function of the training set size.

Finally, we will reduce the layer of the existing DnCNN architecture and train it with the original dataset. We will compare the performance of the model on the same testing dataset with the original algorithm, and investigate how the PSNR measurements change as a function of the number of layers in the DnCNN architecture.

4. Solution

Our solution is to implement the "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for image Denosing" algorithm using TensorFlow, with a focus on the known noise level part. We will use the same architecture and parameters as described in the paper, train and test on the datasets we mentioned above. We will use standard performance metrics such as peak signal-to-noise ratio (PSNR) to evaluate the performance of the algorithm and compare the PSNR value with other algorithm mentioned in the paper. To test the robustness of the algorithm, we will do some exploration as we mentioned above.

5. Timeline

This project is expected to finish within three weeks:

Week 1: Paper research and environment setup; Data preparation and pre-processing; Algorithm understanding and design.

Week 2: Algorithm implementation; Experiment design.

Week 3: Experiments implementation and results analysis; Presentation preparation.

6. Deliverables

The final deliverables will be:

- A working Python implementation and Tensorflow model of the "Beyond a Gaussian Denoiser" algorithm, focusing on the known noise level part.
- Results of the different experiments: Add variety types of noise, change the size of dataset, adjust the layer of CNN, with result analysis.
- A report summarizing the project, including the problem statement, solution, methodology, results, and conclusions.
- A final presentation summarizing the project and key findings.

7. References

[1] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, Lei Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising", vol. 26, no. 7, pp. 3142-3155, Jul. 2017. http://www4.comp.polyu.edu.hk/~cslzhang/paper/DnCNN.pdf

[2] Berkeley Segmentation Data Set and Benchmarks 500 https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html#bsds500

[3] Review: DnCNN - Residual Learning of Deep CNN for Image Denoising (Denoising & Super Resolution & JPEG Deblocking https://sh-tsang.medium.com/review-dncnn-residual-learning-of-deep-cnn-image-denoising-super-resolution-jpeg-deblocking-cbf464b03130

[4] "Deep Convolutional Neural Networks" https://www.run.ai/guides/deep-learning-for-computer-vision/deep-convolutional-neural-networks

[5] "Understand Deep Residual Networks — a simple, modular learning framework that has redefined state-of-the-art" https://medium.com/@waya.ai/deep-residual-learning-9610bb62c355

[6] "Residual Networks" https://sleebapaul.github.io/resnets-tutorial/