THE ANDREW & ERNA VITERBI FACULTY OF ELECTRICAL ENGINEERING

SIPL Signal and Image Processing Lab

TECHNION Israel Institute of Technology

Residual Learning of Deep CNN for Image Denoising

Elimelech Har-Sheffer and Menachem Amar, Supervised by Dr. Meir Bar-Zohar

Introduction

- In recent years, deep convolution neuronal networks (CNN) have been widely used to achieve Image Denoising, one being the DnCNN network which focuses on AWGN.
- In this project we are exploring several ways to improve the proposed network by changing the loss function, training the network on a known image dataset containing images with real noise. We will examine the improvements compared to several leading works in the field.

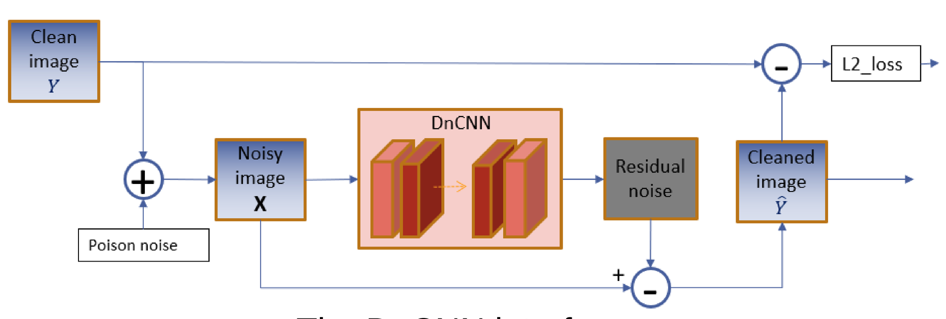
Clean and denoised image

Goals

- Examine and compare our denoising network with other methods known as state-of-the-art (SOTA) in the field.
- Working with known datasets, training and testing our network with them.
- Examination of different loss functions and their effect on our denoising network output according to a few different criteria.

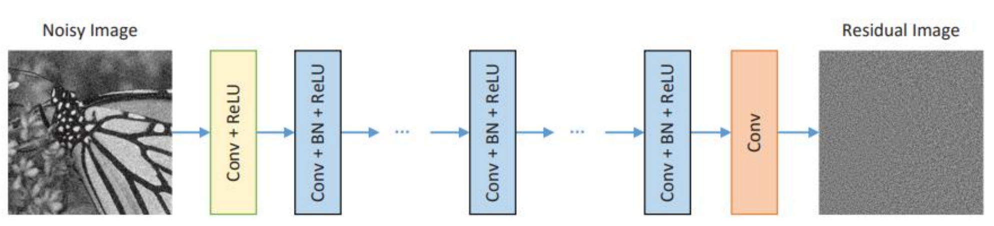
The DnCNN

- Denoising Convolutional Neural Network – a Deep Neural Network for cleaning noise from images:
  - Introduced by Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang in their article "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising" in 2017.
  - Given the original clean image Y:
    - Add Synthetic noise to it in order to get the noisy image X.
    - Pass the noisy image X in the DnCNN
    - Output: Image of the noise Z.
    - Get the cleaned Image: $\hat{Y} = X - Z$
    - Loss = L2_loss($\hat{Y}$ - Y)



The DnCNN interface

- The Architecture:
  - First layer: 64 filters of size 3×3×c to generate 64 features maps (c=1 for gray images, c=3 for RGB images).
  - Layers 2-16: Conv+BN+ReLU, 64 filters of size 3×3×64 between convolution and ReLU
  - Last layer: c filters of size 3×3×64 to reconstruct the output.



Network Architecture

- We compared our results with the RIDNet presented in the article "Real Image Denoising with Feature Attention" by Anwar, S., & Barnes, N. (2019).

Datasets

- Smartphone Image Denoising Dataset – SIDD
  - Real noisy images from smartphone cameras in different sizes (like 5000×3000).
  - Ground-truth images obtained by the mean of 500 noisy images of the same scene.
  - 5 representative smartphone cameras.
  - **Our training was done on this dataset.**
- Nam Dataset
  - Real noisy images.
  - 17 images from 11 static scenes.
  - Noise-free images obtained by the mean of 500 noisy images of the same scene.
  - We cropped the images in 512×512 patches and randomly selected 17 from those for testing.
- Real Noisy Images – RNI15
  - 15 Real noisy images.
  - The clean images are not given for this dataset.
  - The performance of the network is diagnosed by qualitative comparison.

loss functions

1) L1 loss

$$\text{L1\_loss} = \alpha \cdot \text{mean}\left(|\text{cleanIm} - \text{outputIm}|\right) + (1-\alpha)\cdot \text{mean}\left(|\text{Luma}(\text{cleanIm}) - \text{Luma}(\text{outputIm})|\right)$$

- $\alpha = 0.5$

2) L2 loss

$$\text{L2\_loss} = \alpha \cdot \text{mean}\left((\text{cleanIm} - \text{outputIm})^2\right) + (1-\alpha)\cdot \text{mean}\left((\text{Luma}(\text{cleanIm}) - \text{Luma}(\text{outputIm}))^2\right)$$

- $\alpha = 0.5$

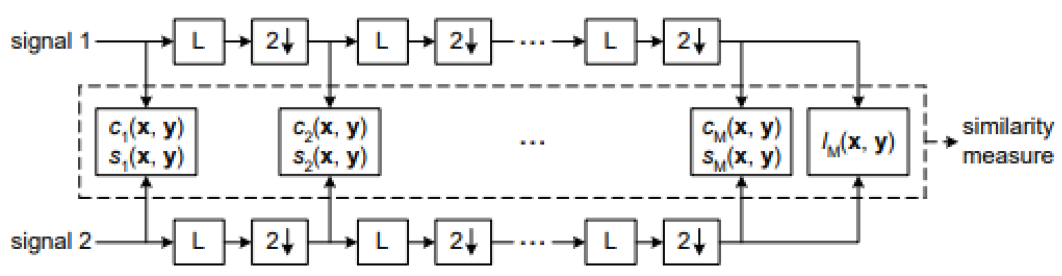
3) SSIM loss

$$\text{SSIM\_loss} = 1 - \text{mean}\left(\text{ssim}(\text{cleanIm}, \text{outputIm})\right)$$

- evaluates images accounting for the fact that the HVS is sensitive to changes in local structure.

4) Multi-SSIM loss

- The signals go through a process when at each stage the signal is passed through a low pass filter and downsamples the filtered image by a factor of 2.



Multi-SSIM structure

$$\text{Multi\_SSIM\_loss} = 1 - \text{mean}\left(\text{MultiSSIM}(\text{cleanIm}, \text{outputIm})\right)$$

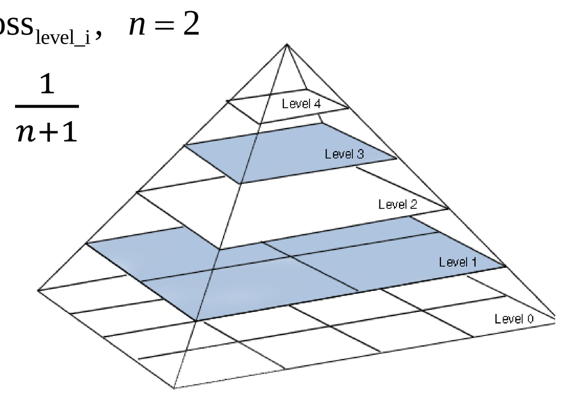
5) L2-Pyramid loss

$$\text{L2\_Pyramid\_loss} = \sum_{i=0}^{n} \lambda_i \cdot \text{L2\_loss}_{\text{level\_i}}, \quad n = 2$$

6) L1-Pyramid loss

$$\text{L1\_Pyramid\_loss} = \sum_{i=0}^{n} \lambda_i \cdot \text{L1\_loss}_{\text{level\_i}}, \quad n = 2$$

- For both we took $\lambda_i = \frac{1}{n+1}$



Pyramid levels structure

7) L1-Chroma + Multi-SSIM loss

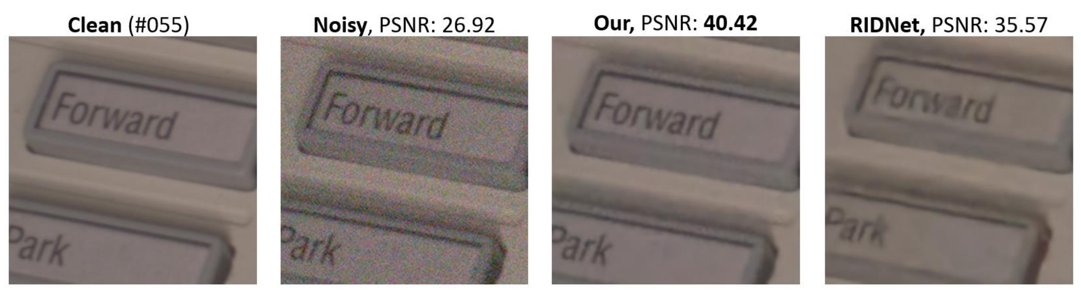
$$0.5 \cdot \text{mean}\left(|\text{cleanIm} - \text{outputIm}|\right) + 0.5 \cdot \left(1 - \text{mean}\left(\text{MultiSSIM}(\text{Luma}(\text{cleanIm}), \text{Luma}(\text{outputIm}))\right)\right)$$

8) L2-Chroma + Multi-SSIM loss:

$$0.5 \cdot \text{mean}\left((\text{cleanIm} - \text{outputIm})^2\right) + 0.5 \cdot \left(1 - \text{mean}\left(\text{MultiSSIM}(\text{Luma}(\text{cleanIm}), \text{Luma}(\text{outputIm}))\right)\right)$$

Results

- Comparison between our results and RIDNet results:



Performance comparison between our best results (L1-Pyramid) and RIDNet over SIDD example.

- Comparison between our different methods for loss functions:

L1-Pyramid vs L2-Pyramid loss functions over SIDD image example

- Train on SIDD and test on 20 images from SIDD benchmark and 17 from Nam benchmark:

Loss Function	PSNR	SSIM
L2	40.04	0.9381
L1	**40.51**	**0.9403**
SSIM	39.31	0.9381
Multi – SSIM	40.38	0.9317
L2 - Pyramid	39.99	0.9375
L1- Pyramid	**40.50**	**0.9407**
L1-Chroma + Multi-SSIM	**40.50**	**0.9403**
L2-Chroma + Multi-SSIM	40.14	0.9395

Performance comparisons of different Loss function methods over SIDD

Datasets	BM3D	DnCNN	FFDNet	CBDNet	RIDNet	Our-best
Nam	37.30	35.55	38.70	39.01	**39.09**	38.44
SIDD	30.88	26.21	29.20	30.78	38.71	**40.51**

Performance comparisons of different Networks over SIDD and Nam Dataset

Conclusions

- We tested our network on a variety of known datasets in order to compare our network performance with the performance of other SOTA known networks.
- Unlike the RIDNet, we only trained on images with natural (non-synthetic) noise, in addition we trained on 60×60 size patches and batch size of 128 and in RIDNet they worked with 80×80 and batch size of 32.
- According to our results based on tests done on SIDD and Nam datasets the L1-Pyramid loss function has slightly better performance compared to other loss-functions.
- Our best network performance over SIDD dataset is better than the other SOTA known algorithms i.e. RIDNet, FFDNet and CBDNet at 1.8 dB on average. But keep in mind that our network training does not contain images with synthetic noise.
- Our best network performance over Nam dataset are less good compared to RIDNet in about 0.65 dB on average.
- Our Network is based on the DnCNN and our performance are better than the results presented in the RIDNet article at about 3.5 dB for Nam dataset and 14.3 dB for SIDD.

October 2020