Supervised Image De-noising in Presence of Spatially Varying Noise

Rupali Gupta
Dept. Computing Science
University of Alberta
Edmonton, Canada
rupali1@ualberta.ca

Ava Sehat Naiki
Dept. Computing Science
University of Alberta
Edmonton, Canada
asehatni@ualberta.ca

Priyanka Kaplish
Dept. Computing Science
University of Alberta
Edmonton, Canada
kaplish@ualberta.ca

Abstract—In recent years, significant amount of work has been done in the area of discriminative model learning for image restoration and image denoising because of its promising results. In this paper, we are using a Convolutional Neural Network for denoising the image by construction of a feed-forward network. This method uses blind and non-blind approach for noise removal. To speed up the training process and performance, residual learning and batch normalization are used. The residual learning is used to remove the latent image in the hidden layers. The proposed model can be used for de-noising the image with white Gaussian noise of an unknown noise level. Experimental results show the effectiveness of the proposed method for denoising the image with spatially varying noise level.

Keywords—Image Denoising, Convolutional Neural Networks, Spatially Varying Noise, Batch Normalization, Residual Learning.

I. INTRODUCTION

Image denoising has been a crucial low-level problem in computer vision. Noise is an unwanted by-product of an image which mars the quality of it and hides the desired information. De-noising the image deals with reconstructing the original image from the noisy version. It removes the noise while conserving the quality.

The search for an efficient image de-noising method is still a persistent challenge. The two major drawbacks of most of the denoising methods are, first, they make the denoising process time-consuming and second, they involve several manually chosen parameters. To overcome these challenges, Artificial Intelligence can be used to develop intelligent models in context of truncated inference. Convolutional Neural Network can be used for image denoising as with its deep architecture it can increase the capacity and flexibility for harnessing image characteristics. With the help of Rectifier Linear Unit, batch normalization and residual learning considerable advances can be made to enhance the performance. Also, CNN can be used for parallel computation which is useful for better run time performance.

The proposed method is designed to predict the residual image from the noisy counterpart with spatially varying noise. Spatially varying noise in this case means the white Gaussian noise across the image with varying noise level across the pixels of the image. The noise is not uniform throughout the image. We are dealing with images having different levels of noise in different parts of image. Residual image is the difference between the noisy and latent clean image. Further, batch normalization is done to stabilize and improve the training performance. The integration of residual learning and batch normalization is used for speeding up the performance. The results are compared based on PSNR ratio and MSE. The proposed method primarily focuses on spatially varying white Gaussian noise.

II. LITERATURE REVIEW

Vast number of models have been introduced to optimize the process of denoising images in areas of complexity, run time and noise type. In this project we have compared our proposed method to following other models;

- BM3D is a 3-D block-matching algorithm in which Image fragments are grouped together based on similarity. BM3D is a less computationally demanding method which addresses specific uniform noise.
- Trainable Nonlinear Reaction Diffusion (TNRD) is a flexible learning framework based on the concept of nonlinear reaction diffusion models. The mentioned models are based on the analysis model which is limited in capturing the full characteristics of image structures; this limits TNRD and BM3D in blind denoising of images.

TABLE I
Available Methods Comparison

	Existing Models			
	BM3D	*TNRD*	*DnCNN- S*	*DnCNN- B*
Noise Type	Uniform	Uniform	Uniform	Uniform
Noise Range	Specific	Specific	Specific	Blind
Data Type	Grayscale	Grayscale	Grayscale / RGB	Grayscale / RGB

Per Table I, it is evident that none of the models that we have examined addresses the spatially varying noise. In oppose to TNRD and BM3D, DnCNN model relies on retrieving the residual image form the noisy version; this enables DnCNN model to execute on images with unknown noise level.

In addition to BM3D and TNRD models we have examined other models to select and develop a comprehensive solution for denoising degraded images in presence of spatially varying noise.

- Plain Multi Layer Perceptrons (MLP) is directly applied to image patches to estimate mapping from a noisy image to a noise-free image. MLP is easily adapted to less extensively studied types of noise, such as mixed Poisson-Gaussian noise, JPEG artifacts and salt-and-pepper.
- Cascade of Shrinkage Fields (CSF) have been introduced by Tappen. Through this model, restored image x is predicted from a corrupted observation y after training on a set of sample images S.
- The Nuclear Norm Minimisation (NNM) regularises each singular value equally to pursue the convexity of the objective function. However, this restricts its capability and flexibility in dealing with many practical problems where the singular values should be treated differently. WNNM adaptively assigns weights on different singular values.
- Expected Patch Log-Likelihood (EPLL) algorithms is presented to fully employ image prior, where the Gaussian mixtures model is employed to estimate the noisy image patches. GMM is a probabilistic model for representing the presence of subpopulations within an overall population. The basic idea of the EPLL is to maximise the likelihood of patches.

III. IMPLEMENTATION DETAILS

A. Model Design

Noisy image is provided as an input to the model. The output of the model is registered as the difference between residual mapping and noisy image. The loss function of the model can be defined as:

$$\ell(\mathbf{\Theta}) = \frac{1}{2N}\sum_{i=1}^{N} \|\mathcal{R}(\mathbf{y}_i;\mathbf{\Theta}) - (\mathbf{y}_i - \mathbf{x}_i)\|_F^2$$

Fig. 1: Loss Function

The DnCnn model architecture with depth D consists of three types of layers, (*i*)Conv+ReLU, (*ii*)Conv+BN+ReLU and (*iii*)Conv.

(*i*)Conv+ReLU: For first layer, 64 filters of size 3x3x*c* are used to generate 64 feature maps and rectifier linear units which are then utilized for nonlinearity. c is the number of image channels, i.e. c=1 for gray scale and c=3 for colored.

(*ii*)Conv+BN+ReLU: For layers 2 to (D-1), 64 filters of size 3 x 3 x 64 are used, and batch normalization is added between convolution and ReLU.

(*iii*)Conv: For the last layer, c filters of size 3 x 3 x 64 are used to reconstruct the image.

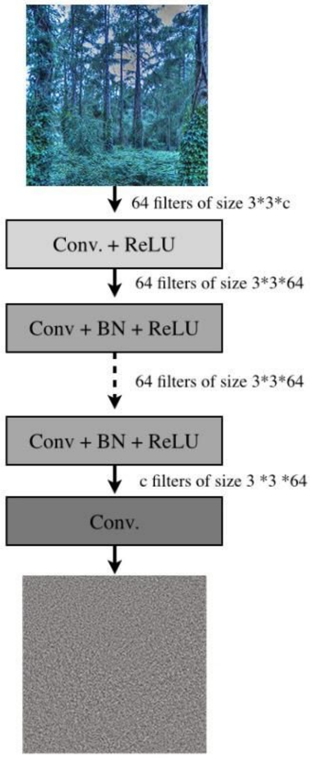


Fig. 2: System Architecture

With ReLU, DnCnn can separate image structure from noisy observation through hidden layers which is similar to iterative noise removal in methods like EPLL and WNNM.

To get the output image size same as the input image, zeroes are padded before convolution so that the feature map has the same size the input image.

DnCnn can be described as the generalization of one-stage TNRD. TNRD trains a discriminative solution for the following problem:

$$\min_{\mathbf{x}} \Psi(\mathbf{y}-\mathbf{x}) + \lambda \sum_{k=1}^{K}\sum_{p=1}^{N} \rho_k((\mathbf{f}_k * \mathbf{x})_p),$$

Fig. 3: Denoising Problem

from set of noisy-clean training image pairs. Here *N* is the image size, λ is the regularization parameter, $f_k * x$ stands for the convolution of the image x with the *k*-th filter kernel f_k, and $\rho k(\cdot)$ represents the *k*-th penalty function which is adjustable in the TNRD model. For Gaussian denoising, we set $\Psi(\mathbf{z}) = \frac{1}{2}\|\mathbf{z}\|^2$.

B. Dataset Creation

We have created dataset using MATLAB functions. We have added noises to the dataset from http://places2.csail.mit.edu/.

Dataset is based on two approaches mentioned before -blind and non-blind which contains both grayscale and colored images.

- Grayscale Images having uniform noises(non-blind).

i)Original Image ii)Noisy Image

Fig. 4: Dataset A

- Colored Images having uniform noises(non-blind).

i)Original Image ii)Noisy Image

Fig. 5: Dataset B

- Grayscale images having non-uniform noises(blind).

i)Original Image ii)Noisy Image

Fig. 6: Dataset C

- Colored images having non-uniform noises(blind).

i)Original Image ii)Noisy Image

Fig. 7: Dataset D

C. Implementation

Implementation of DnCnn model contains training and testing. The denoising of noisy image is done using Supervised learning. The model is trained on noisy image and clean image and predicts the output based on the training. We have divided the model into two models i.e. Model A and Model B based on the blind and non-blind approach. Non-Blind approach is used by Model A and Blind approach is used by Model B.

In Blind approach, model is trained on varying noise level from 0 to 55 across the image. When the model is provided with clean images and noisy images, it learns through supervised learning to predict the output.

a) Training: During training phase, we train the model for 900 images of size 180x180. The patch size is set as 50×50, and crop 128×3,000 patches to train the model. Please find below the details of each model as follows:

Model A: Model A consists the implementation of Supervised de-noising model for the images having uniformly distributed noise which is non-blind approach. The model is trained for noise level 15, 20 and 25 for 50 epochs. The execution has two phases that includes:

- Supervised De-noising for channel 1 i.e. grayscale images having uniform noise
- Supervised De-noising for channel=3 i.e. colored images having uniform noise.

Model B: Model B is the enhancement of Model A which contains implementation of Supervised de-noising model for the images having spatially varying noise which is non-blind approach. The model is trained for noise level in range of 0 to 55 for 30 epochs. The execution has two phases that includes:

- Supervised De-noising for channel=1 i.e. grayscale images having spatially varying noise.
- Supervised De-noising for channel=3 i.e. colored images having uniform noise.

b) Testing: After training the model, the next step in our implementation is to test the model for the images of spatially varying noise. The testing is done for 100 images.

The testing dataset is processed under the trained model and the results are stored for further evaluation.

c) Evaluation: This is the final step of our implementation. During evaluation phase, we have evaluated our models by comparing our obtained results with the existing clear images in the dataset. During testing phase, PSNR ratios and MSE of the images are calculated. This information helped our system in the process of calculating the final accuracy as well.

TABLE II
Model A and Model B Details

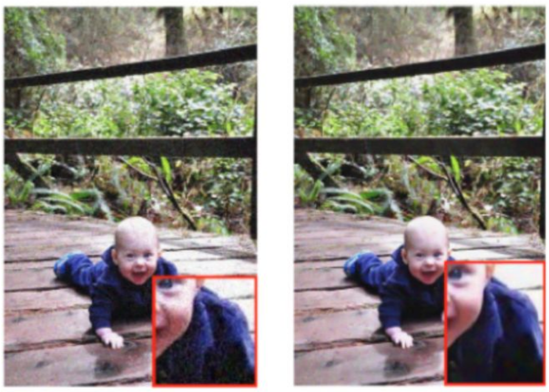
	Proposed Models		
	Dataset	*Architecture*	*Training (epoches)*
Model A	Uniform Noise / Grayscale	DnCNN	50
	Uniform Noise / RGB	DnCNN	50
Model B	Spatially Varying/ Grayscale	DnCNN	30
	Spatially Varying/ RGB	DnCNN	30

D. Error Analysis

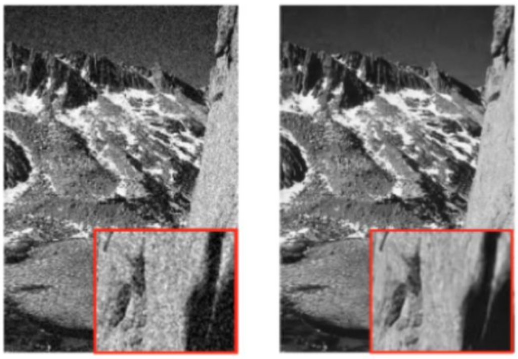
The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error[5]. We have calculated the results on the bases of PSNR and MSE values.

E. Implementation Results

*a) Model A :*In Model A, we have created and implemented model for images having uniform noise level (constant sigma noise). As you can see in Fig 6, we have obtained denoised images for colored and grayscale images. We have observed that we have successfully preserved most of the minute detail of the images.

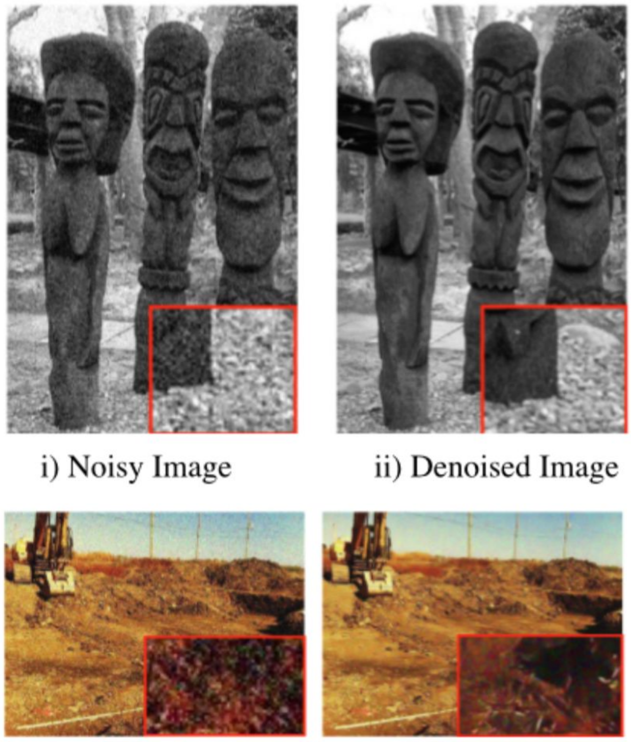


i) Noisy Image ii) Denoised Image



i) Noisy Image ii) Denoised Image

Fig. 8: Model A Results



i) Noisy Image ii) Denoised Image

i) Noisy Image ii) Denoised Image

Fig. 9: Model B Results

b)Model B : In Model B, we have created and implemented model for images having non-uniform noise level (varying sigma noise). As you can see in Fig 7, we have used our testing dataset and have managed to reduce the noise from the image for the images having noise of multiple sigma values.

III. PROPOSED METHOD COMPARISON

To measure the efficacy of the proposed models and compare the model's performance to BM3D, TNRD and DnCNN, we have leveraged the Peak Signal-to-Noise Ratio(PSNR) as an Image Quality Metric. PSNR is a quantitative method to compare image enhancement models. In this method the mean squared error(MSE) represents the average for the squares of the "errors" between the degraded images and the enhanced images.The limitation of MSE calculation is that mean-squared error depends strongly on the image intensity scaling and does not capture the quality difference between two images with different scaling efficiently. PSNR measures the ratio between the maximum possible value (power) of a signal and the power of the noise.

$$MSE = \frac{1}{mn} \sum_{0}^{m-1} \sum_{0}^{n-1} \|f(i,j) - g(i,j)\|^2$$

$$PSNR = 20 \log_{10} \left(\frac{MAX_f}{\sqrt{MSE}} \right)$$

Fig. 10: Peak Signal-to-Noise Ratio(PSNR)

TABLE III

Sample Images Accuracy Calculation

	Model A- grayscale					
	test001	test002	test003	test004	test005	test006
PSNR	21.56	21.85	20.82	21.94	21.18	21.02
MSE	453.94	424.61	538.26	415.91	495.45	514.04

TABLE IV

Sample Images Accuracy Calculation

	Model A - RGB					
	test001	test002	test003	test004	test005	test006
PSNR	26.85	26.68	26.79	26.70	27.40	26.59
MSE	134.27	139.63	136.14	138.99	118.30	142.56

TABLE V

Sample Images Accuracy Calculation

	Model B - grayscale					
	test016	test017	test018	test019	test020	test021
PSNR	21.08	23.70	24.20	24.01	22.80	21.79
MSE	506.99	277.33	247.17	258.23	341.19	430.53

TABLE VI

Sample Images Accuracy Calculation

	Model B - RGB					
	test011	test013	test015	test016	test019	test027
PSNR	29.34	26.64	26.72	28.64	26.68	28.47
MSE	75.68	140.93	138.36	88.92	139.63	90.36

Table III Model A and B both exhibit promising results for retrieving a clean image; in this table the higher value of PSNR proves of having better performance on the model. Given the complexity of spatial varying noise in oppose to uniform noise, lower PSNR value of developed models can be explained. We have majorly evaluated our results based on the Error Analysis.

TABLE VII

Proposed Approach in Comparison to Existing Methods

	Performance Comparison					
	BM3D	*TNRD*	*DnCNN - S*	*DnCNN - B*	*Model A*	*Model B*
Noise Type	Uniform σ = 25	Uniform σ = 25	Uniform σ = 25	Uniform σ = 25	Uniform σ = 25	Spatial Varying Noise
PSNR -Grayscale -RGB	28.93	31.63	29.43	29.35	21.20 26.90	22.91 30.81
MSE -Grayscale -RGB	83.17	44.66	75.13	74.50	493.17 131.82	322.65 53.95

CONCLUSION

To conclude, this project addresses images that contain gaussian noise as well as varying noises. For the images with the Gaussian noise, we can filter them by using Spatial filters. For the images with the varying noise, we can train the model with the help of existing algorithms on different level of noises.This system has been developed while considering major factors affecting de-noising of noisy images. We have used white Gaussian mixture model to add noise in the images. Using this we have created models for all the phases of implementation. After the data is created, we have proceeded towards developing the models. The models which are developed in the training phase are tested and evaluated based on the obtained results. The accuracy of this system is being calculated based on two different strategies in which it has successfully achieve 30.81 PSNR ratio in Model B Phase II.The accuracy was calculated on the bases of the automated system developed in the training and testing model. We can do further modification by training the model for other types of noises and applying different image restoration techniques for testing. This project is very useful to develop industrial application like removing noises from old photos, studying satellite images and examining old artifacts/manuscripts.

ACKNOWLEDGEMENT

This project was supported by **Dr. Lihang Ying.** We would like to show our gratitude to him for sharing their pearls of wisdom with us during the course of this project. We also thank our course project mentor **Mr. Subhayan Mukherjee** from University of Alberta who has provided insights and expertise that greatly assisted our project. We are grateful to all of those with whom have had the pleasure to work during this project.

REFERENCES

[1] hang, K., Zuo, W., Chen, Y., Meng, D. and Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE Transactions on Image Processing, 26(7), pp.3142-3155.

[2] Bertalmío, M. (2018). Denoising of Photographic Images and Video. 1st ed. Barcelona: Springer International Publishing.

[3] Zhang, K., Zuo, W. and Zhang, L. (2018). FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. IEEE Transactions on Image Processing, 27(9), pp.4608-4622.

[4] Zhang, L., Zhang, K. and Zuo, W. (2018). Learning a Single Convolutional Super-Resolution Network for Multiple Degradations. In: CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Salt Lake City.

[5] Compute peak signal-to-noise ratio (PSNR) between images - Simulink
https://www.mathworks.com/help/vision/ref/psnr.html