OVERVIEW OF DEEP LEARNING METHODS FOR DENOISING

Dhruv Jayesh Dholakia, Mohammed Saquib Suglatwala, Najeebuddin Mohammed

Department of Electrical and Computer Engineering,
University of Waterloo

ABSTRACT

Various deep learning methods have been proposed over the past few years for denoising of images. In this paper, we propose a comparative study of 3 convolutional neural network-based methods specifically designed for removing noise from natural images. We first analyse the theoretical aspects of the papers that propose the networks, followed by a comparison of performance using standard metrics on a synthetically noised FashionMNIST dataset.

***Index Terms*—** Image Denoising, FFDNet, DnCNN, Denoising Autoencoder

1. INTRODUCTION

Image denoising is a classical problem that has plagued visual communication ever since the advent of efficient computing for compression and transmission in a large scale. The noise present in natural images is of an unknown distribution [1], which makes it a challenging issue. Various image denoising techniques based on different theoretical concepts have been proposed and formulated for over 40 years [2].

Traditionally, filters were used for denoising. These included various types like linear, adaptive, non-linear and non-adaptive. Non-adaptive filters proved to the best out of them all based on peak signal to noise ratio (PSNR), but still had several drawbacks like over-smoothing, blurring and loss of fine detail. As the average computing power of the academia world grew at an exponential pace, various statistical and machine learning based methods were formulated to better solve the problem without the disadvantages of filtering. These new methods however had a new set of downsides, including lack of generalization, manual tuning of parameters and considerable computing time for optimization.

Recently, more flexible and complex algorithms based on deep learning techniques have been proposed to overcome the above drawbacks and be a comprehensive solution for most noising issues. CNNs have been the go-to for nearly all image-based tasks, so it is no surprise that various modifications and augmentations have been proposed to create custom CNN architectures designed specifically for denoising. Modified generative models based on autoencoders and GANs have also been formulated in recent years. Real-time denoising of video streams have also been a trending topic of research in this area.

In the following sections, we analyse the following 3 methods; Denoising CNN (DnCNN) [3], Fast and Flexible Denoising CNN (FFDNet) [4], Denoising Autoencoder (DAE) [5]. The first two are CNN-based and the last one is convolutional autoencoder based. Section-2 will go over the theoretical aspects and methodology followed by the experimentation details of our comparative study in section-3. Section-4 and 5 will present our results and conclusions respectively.

2. METHODOLOGY

2.1. FFDNet

Favorable characteristics for an ideal image denoiser could be its efficiency and effectiveness of handling spatially variant noise of images with a single model. Even though the model-based approaches for denoising are quite efficient with varying noise levels, their optimizing algorithms still struggle with the time efficiency and inability to use them directly on the problems containing spatially variant noise.

Some other well-known approaches are based on discriminative learning which includes MLP and CNN-based methods [6] towards denoising. However, these denoising methods are limited in flexibility and the learned model is usually tailored to a specific noise level. Among all of the known effective discriminative denoising methods, DnCNN [3] has shown promising performance. The model parameters of such method are trained for noisy images corrupted by additive white Gaussian noise with a fixed noise the level which makes them less effective when applied directly on the real images with other noise levels.

The authors of the paper [4] presented a fast and flexible denoising convolutional neural network to overcome the above-stated pitfalls of existing CNN-based approaches. From the perspective of regression, DnCNN [3] learns mapping function $x = F(y; \Theta\sigma)$ between input noisy image y and the desired output x. In which, the model parameters $\Theta\sigma$ are trained for noisy images corrupted by AWGN with a fixed noise level σ. In the paper [4], the authors have proposed a different mapping function x = F(y, M; Θ), where M is a noise level map. While the parameters $\Theta\sigma$ was varying with the change of noise level in DnCNN [3], in the FFDNet [4], the noise level map is modeled as an input and the model parameters Θ are invariant to the noise level. Thus, with a single network, proposed architecture provides flexibility in handling the various noise levels.

The authors have stated that the input noise level map plays an important part in controlling the trade-off between noise reduction and detail preservation. They have emphasised the problem of heavy visual quality degradation caused by setting a larger noise level to smooth out the details. Also, to mitigate this they have adopted a method of orthogonal initialization on convolutional filters. They have pointed out that the FFDNet [4] works on downsampled sub-images, which largely accelerates the training and testing speed, and enlarges the receptive field as well.

The authors have shown quantitative comparisons of FFDNet [4] with state-of-the-art denoising methods, including model-based methods such as BM3D [7] and WNNM [8] and discriminative learning-based methods such as TNRD [9] and DnCNN [3] which used noisy images corrupted by AWGN. The outcomes clearly showcase the higher potential of FFDNet [4] where it outranks the other stated methods in both denoising performance and computational efficiency. Furthermore, images having signal-dependent, non-Gaussian and spatially variant noise was also evaluated by the authors where FFDNet [4] still achieved perceptually convincing results with setting proper noise level maps.

2.2. DnCNN

Usually, high denoising images have two drawbacks [3]. One of them is time inefficient because of the complex optimization problem. On the other hand, this effort is manual and have to tune the correct parameters to gain the efficiency. Here, to mitigate these problems, the authors have used Convolutional Neural Network (CNN) for denoising data because of the three main reasons. Firstly, this process is effective because of the capacity and flexibility as it has a very deep architecture. Secondly, it has a learning methods which includes Rectified Linear Unit (ReLU), Batch Normalization and Residual Learning to speed up the training process and boost up the denoising performance. Furthermore, the third reason is the parallel computation to enhance the run time performance.

The authors have designed to predict the output $\hat{v}$ residual image - difference between noisy image and latent clear image rather than the denoised image $\hat{x}$ in order to extract the latent clean image from the functions in the hidden layers. The combination of batch normalization with the residual learning improves the speed of training.

Moreover, the authors extend this DnCNN to handle denoising tasks apart from Gaussian tasks such as SISR (Single image super-resolution) and JPEG image deblocking by analyzing the connection between DnCNN and TNRD [9]. These denoising tasks can be modeled by the same image degradation model.

Now adding Gaussian Noise and training the CNN model can outperform other methods such as BM3D [7], WNNM [8] and TNRD [9] according to the authors. Not only this, the model can also give optimistic results when multiple upscaling factors and different quality factors are added to a single CNN model for SISR and JPEG deblocking respectively.

2.3. DAE

We recall that a traditional autoencoder [10] maps a given input to a hidden latent space of a modifiable dimensionality through a deterministic mapping in the encoding stage and the uses the latent representation to generate a reconstructed output in the decoding stage. The parameters of the network are learnt to minimize the reconstruction error between the input and output. Usually, this error function is in the form of mean squared error (MSE) [11] and the learning process is unsupervised as target labels for the input are unnecessary when the goal is to accurately reconstruct the input by learning its features in a hidden latent space.

The authors of [5] propose a modification to the input of the autoencoder in order to hypothesize on the robustness of the feature representation learnt in the latent space. It was motivated on the informal reasoning that a good performing autoencoder should be able to learn the implicit characteristics of the inherent unknown distribution of natural images through partial observation and will be mostly unaffected by slight occlusion or corruption, similar to how the human visual system can recognise the information in a noisy image just as well as a perfect image. To test this, they proposed "blackening" (forcing pixel values to zero) a certain number of random pixels in the input image such that the autoencoder will have to "fill in" those blanks from the information stored in the learnt latent representation. While the paper only tackles this way of adding noise, it does state as a footnote that the approach is independent to the method of corruption.

While the authors clearly mention that their main objective is not denoising but rather the explicit robustness to corrupting noise, it has proven to be a formidable technique for the denoising of natural images. They also specifically mention that the proposed method cannot be categorised as a regularization technique despite small noise addition being considered so [12]. This is because the noise added to the input of a denoising autoencoder is large enough to be corrupting and should cause a visually noticeable destruction of information.

The model is experimented and compared against SVMs with Gaussian and polynomial kernels, 1 and 3 hidden layers deep belief network (DBN-1 and DBN-3) and a 3 hidden layer deep network initialized by stacking basic autoencoders [5] as a preprocessing stage on the image classification benchmark proposed in [13]. It performs the best on all but one of the classification tasks.

The authors also present their design think in the perspective of manifold learning, top-down generative modelling and information theory, which are all deeply rooted in the field of deep learning and statistics. We choose not to review these perspectives as the original explanation on comparison with the human visual system provides a much better understanding of the technique and ingenuity behind it.

3. EXPERIMENTATION

3.1. Data

This study uses the FashionMNIST [14] dataset, published by Zalando's Research team, for performance comparison. The training dataset is divided further in a 3:1 ratio to get a validation set. Random Gaussian noise with a mean of 0 and variance of 0.01 is added to the images in all 3 sets (training, validation, test) to get our inputs. The original images are passed as the expected output.

3.2. Metrics

For the training of all 3 models, **mean squared error** (MSE) is used as the loss function. In reality, it is not an ideal measure of the difference in images as observed by the human visual system [15]. So, we also compute the **Structural Similarity Index** (SSIM) [16] between the denoised and original images in the test set. **Peak signal to noise ratio** (PSNR) is also calculated to quantify the remaining noise, For both of these performance metrics, the higher the better.

3.3. Software

All programming was done using **Python3** and **TensorFlow** v2 in Google Colab using a GPU accelerator. For array computation and image processing, we use *NumPy, Scikit-Learn and Scikit-Image* libraries. The python variant of the *Matplotlib* library was used for plotting of sample images and the final results. The code and the data can be viewed at `https://git.uwaterloo.ca/613_w22/review_project_denoising`

3.4. Architecture

3.4.1. FFDNet model

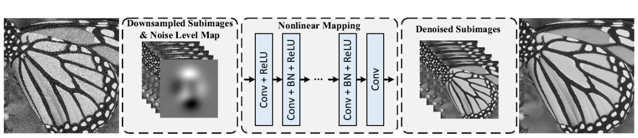

Fig. 1. FFDNet Architecture (image taken from [4])

As shown in Fig 1, the input image is reshaped into four sub-images, which are then input to the DnCNN together with a noise level map. DnCNN is using Rectified Linear Unit with kernel size 3x3 and has 13 convolutional layers. The final output is reconstructed by four denoised sub-images. The source code for the FFDNet has been implemented by [17].

3.4.2. DnCNN model

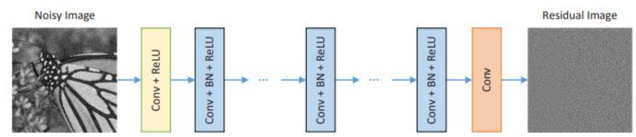

Fig. 2. DnCNN Architecture (image taken from [3])

In this architecture, above image has three different colours because different combination of learning methods have been applied. In the first layer, Convolutional and Rectified Linear Unit with 16 filters and a kernel size of 3 x 3 is there. In the intermediate stages, there are five layers of Convolutional, Rectified Linear Unit and Batch Normalization with 16 filters and the same kernel size. In the final stage, one layer of Convolutional with 1 filter and the same kernel size is there to reconstruct the output.

3.4.3. DAE model

The denoising autoencoder's architecture is the same as a deep convolutional autoencoder. We use two convolutional layers in the encoder followed by the deep layer and 3 convolutional layers in the decoder. We use the ADAM [18] optimizer for training with early stopping.

4. RESULTS

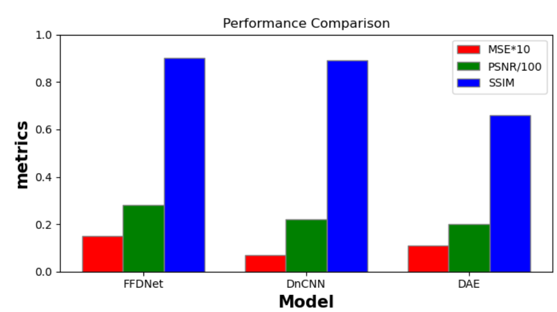

Fig. 3. Final performance comparison of all models

As seen in Fig 3, both FFDNet and DnCNN obtain very close results in terms of all metrics. DAE performs poorly but reaches so in the fastest training time. In terms of ease of implementation, DnCNN is the best with FFDNet being the most complex as observed from our code. All three networks needed extensive hyperparameter tuning to reach their results, with the FFDNet being significantly dependent on the noise variance, the DnCNN significantly dependent on the filter size and the DAE significantly dependent on the depth of the convolution.

5. CONCLUSION

In our study, we have successfully presented an analytical comparison between FFDNet, DnCNN and DAE in terms of their theoretical foundations, architecture implementations and performance on a synthetically noised dataset. FFDNet, the most recently proposed technique, performed the best. This is unsurprising considering they are a modification of the DnCNN to allow for faster training by performing denoising on downscaled sub-images, which provides a faster and better memory efficiency. The denoising autoencoder performs adequately but requires extensive thought behind the convolution design as well as hyperparameter tuning without providing any significant improvements in training time when compared to FFDNet.

6. REFERENCES

[1] Jun Xu, Lei Zhang, and David Zhang, "External prior guided internal prior learning for real-world noisy image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2996–3010, 2018.

[2] R. Bernstein, "Adaptive nonlinear filters for simultaneous removal of different kinds of noise in images," *IEEE Transactions on Circuits and Systems*, vol. 34, no. 11, pp. 1275–1291, 1987.

[3] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[4] Kai Zhang, Wangmeng Zuo, and Lei Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

[5] Pascal Vincent, Hugo Larochelle, Y. Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, 01 2008.

[6] Viren Jain and Sebastian Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. 2008, vol. 21, Curran Associates, Inc.

[7] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[8] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng, "Weighted nuclear norm minimization with application to image denoising," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2862–2869.

[9] Yunjin Chen and Thomas Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2017.

[10] Yann Lecun, *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)*, Universite P. et M. Curie (Paris 6), June 1987.

[11] Wikipedia contributors, "Mean squared error — Wikipedia, the free encyclopedia," 2022, [Online; accessed 12-March-2022].

[12] Christopher M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural Computation*, vol. 7, no. 1, pp. 108–116, Jan. 1995.

[13] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th International Conference on Machine Learning*, New York, NY, USA, 2007, ICML '07, p. 473–480, Association for Computing Machinery.

[14] Han Xiao, Kashif Rasul, and Roland Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[15] Zhou Wang and Alan C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[16] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[17] Matias Tassano, Julie Delon, and Thomas Veit, "An Analysis and Implementation of the FFDNet Image Denoising Method," *Image Processing On Line*, vol. 9,

pp. 1–25, 2019, https://doi.org/10.5201/ipol.2019.231.

[18] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun, Eds., 2015.