Image Denoising Using a U-net

Paavani Dua
Department of Electrical Engineering
Stanford University
paavanid@stanford.edu

Abstract

The purpose of this project is to use a U-net to denoise images instead of traditional denoising imaging techniques such as spatial filtering, wavelet thresholding and transform domain filtering. A good denoising model should remove noise as much as possible along with preserving edges. CNNs give better results than traditional techniques and are more computationally efficient.

1. Introduction

In the last two years, deep convolutional networks have outperformed the state of the art traditional image processing techniques in many visual recognition tasks, e.g. [1,2] While CNNs have been around for a while, they have gained popularity in recent years due to an explosion in available data for training as well as computational power.[3] Krizhevsky et al. [4] was considered a breakthrough due to supervised training of a large network with 8 layers and millions of parameters on the ImageNet dataset with 1 million training images. Deeper networks are in use at present. While CNNs can be used for classification tasks, visual tasks are also trained for. The typical use of convolutional networks is on classification tasks, where the output to an image is a single class label. However, in many visual tasks, especially in biomedical image processing, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel.

While image denoising techniques require parameters to be manually set and complex optimization increases computational cost, in recent years, CNNs have proven to be more computationally efficient while producing better results. In medical imaging, U-nets are extensively used instead of the traditional image processing pipeline.

The U-net is a convolutional neural network developed for biomedical image segmentation. The main idea is to supplement a usual contracting network by successive layers, where pooling operations are replaced by upsampling operators. Upsampling increases the resolution of the output.

2. Related Work

Various CNN architectures are frequently implemented. GANs have been used to remove unknown noise from images [6] while various CNN models on different deep learning frameworks have been trained for medical applications for CT and MRI scans [7]. Every CNN model has different number of layers and hyperparameters tuned based on computational power and time needed to run. Some have traditional denoising techniques along with CNNs fused.

Enhanced convolutional neural denoising network-ECNDNet utilises residual learning technique and is shown in Figure 1.

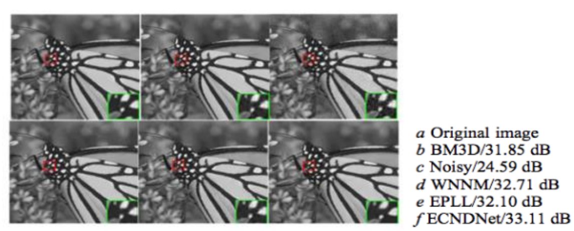

Figure 1: Results for ECNDNet with other CNN and traditional denoising techniques[1]

Traditional techniques are also fused with CNNs to denoise images well such as the BCNN.[8]

3. Methods

The CIFAR-10 dataset is used as the training and test set. The dataset has 60,000 images with training set with 50,000 images and test set having the remaining 10,000 images. The resolution of the color images are 32*32.

The PyTorch framework is used to code it in with a U-net being directly pulled from a public Github repo. The model is seen below of the U-net architecture.

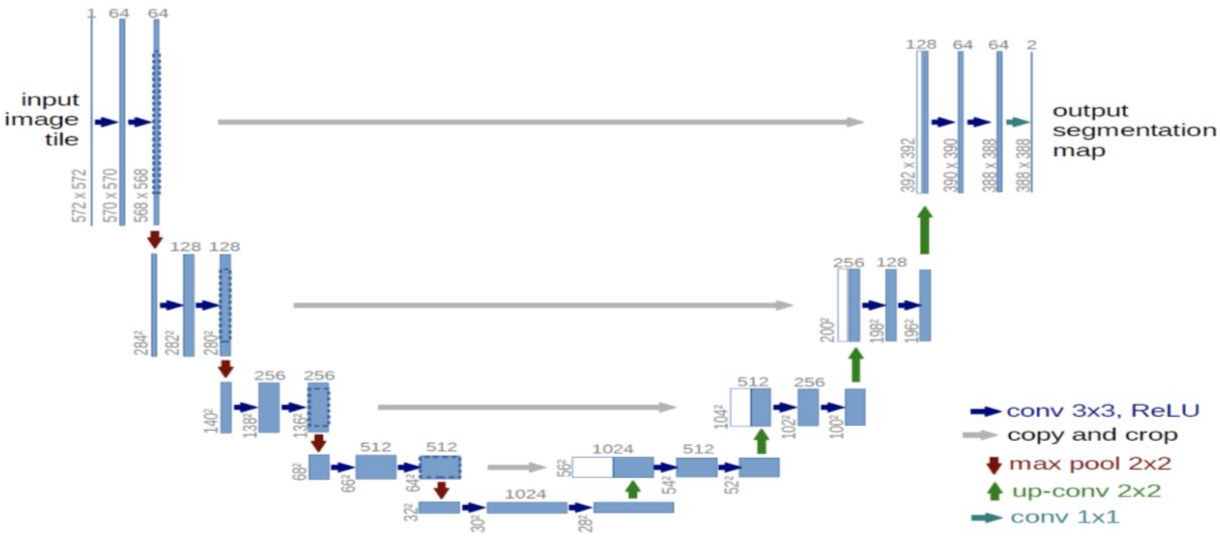

Figure 2: U-net architecture

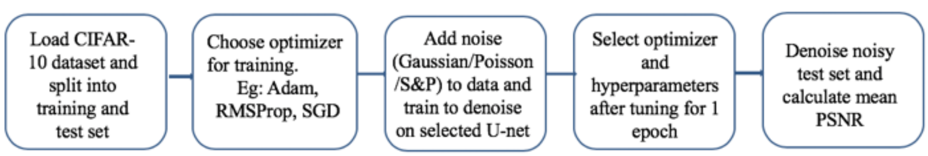

Figure 3: Flowchart of algorithm

The model can be tuned for various hyperparameters such as different optimizers and regularizers to choose the best hyperparameters that give the highest PSNR. Due to training the model on the CPU, very few models were tested but on a GPU, better tuning can be done.

Different noises can be added to the original images such as Gaussian noise, Poisson noise, Salt and Pepper Noise etc to denoise.
Based on the minimum loss from the optimizer on the training set, choose the best hyperparameters and denoise for the test set
Various U-nets and datasets can be plugged in to train and tune hyperparameters.

After training 3 models with Gaussian noise of sigma = 0.05, 3 optimizers-Adam, RMSprop and SGD were used. Adam gave the lowest loss and so was chosen as the optimal optimizer.

Using Adam as the optimizer, 2 models were trained with different noises-Gaussian and Poisson- to obtain the PSNRs after denoising. Each model has an approximate run time of an hour. The PSNR results along with the losses can be seen in the Results section.

The batch size used for the training model was 16 with a dropout rate of 0.2 but these are all hyperparameters that can be tuned along with the sigma of the noises that can be chosen based on user preferences.4.

4. Results

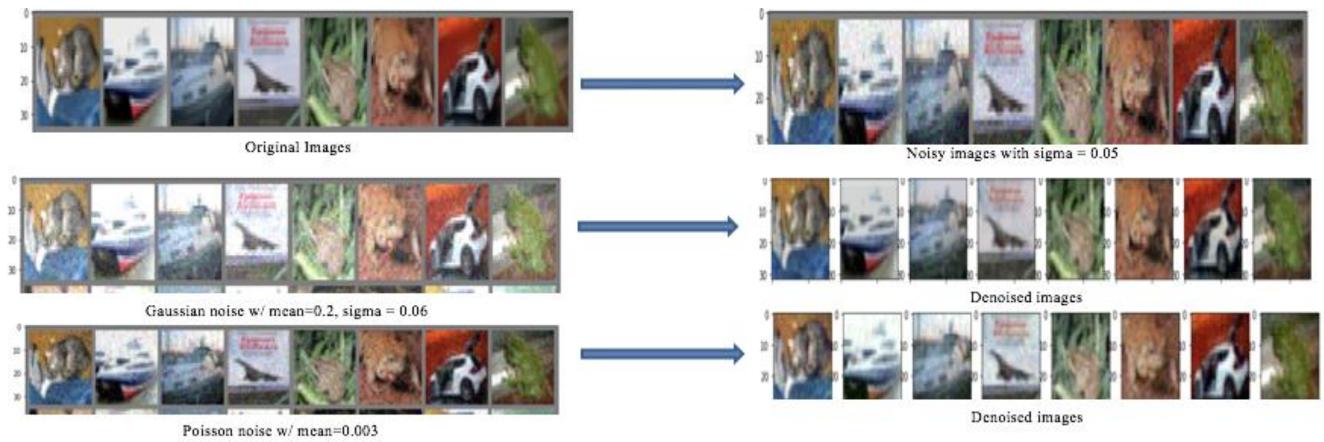

Figure 4: Noisy and denoised images

Optimizer & parameters	Mean noisy PSNR	Mean PSNR
Optimizer: Adam with learning rate=0.001 Noise: Gaussian noise with sigma = 0.05	26.18	28.38
Optimizer: RMS with learning rate=0.001 Noise: Gaussian noise with sigma = 0.05	26.18	28.18
Optimizer: SGD with learning rate=0.001, momentum = 0.9 Noise: Gaussian noise with sigma = 0.05	26.18	24.32
Optimizer: Adam with learning rate=0.001 Noise: Gaussian noise with mean=0.2, sigma = 0.06	14.14	24.38
Optimizer: Adam with learning rate=0.01 Noise: Poisson noise with mean=0.003	25.39	25.43

Figure 5: Noisy PSNR vs Denoised PSNR

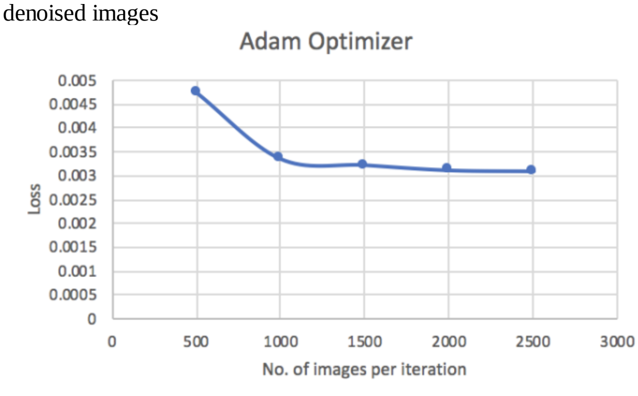

Figure 6: Adam optimizer loss over 1 epoch

Table 1: Various Optimizer losses

Iteration	Adam	RMSProp	SGD
	Loss		
1	0.00474	0.00402	0.01431
2	0.00334	0.00335	0.00778
3	0.00321	0.00300	0.00662
4	0.00310	0.00306	0.00577
5	0.00308	0.00309	0.00517

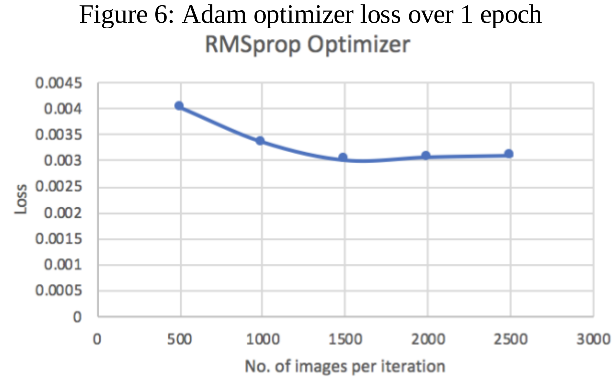

Figure 7: RMSprop optimizer loss over 1 epoch

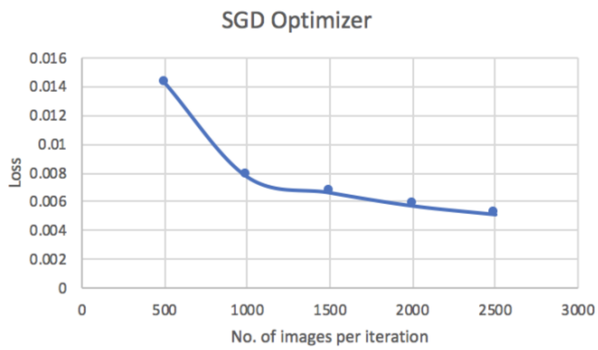

Figure 8: SGD optimizer loss over 1 epoch

5. Discussion & Future Work

While the results only show an implementation of an existing U-net with different hyperparameters, more comparison needs to be done with traditional imaging techniques such BM3D and compare the PSNR and run-time to see if the U-net does a better job. CBM3D is the most widely used algorithm with best results that can be downloaded by anyone in MATLAB but requires an older version of MATLAB, and so I wasn't able to run it due to compatibility issues and failed troubleshooting.

While the runs were done on my personal laptop on the CPU, with services such as GCP and AWS using GPUs, hyperparameter tuning can be done easier and faster for increased epochs to get the best hyperparameters that produce the highest PSNR.

Another idea that would be good for future integration and testing would be super-resolution by using different datasets, downsampling the images to pass through the U-net and upsampling them to then get the PSNR to decide if super-resolution works fine through this U-net.

6. References

[1] Ronneberger, Fischer, and Brox., "U-net: Convolutional Networks for Biomedical Image Segmentation", *arXiv.org*, 2015
[2] Krizhevsky, A., Sutskever, I., Hinton, G.E.: "Imagenet classification with deep convolutional neural networks", NIPS, 2012.
[3] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
[4] Lefkimmiatis, S.: 'Universal denoising networks: a novel CNN architecture for image denoising'. *IEEE Conf. Computer Vision Pattern Recognition,* June 2018.
[5] Fei,Luo, Tian, et al., "Enhanced CNN for Image Denoising", *The Institution for Engineering and Technology*, 2018
[6] Chao, Chen, Chen, et al., "Image Blind Denoising With Generative Adversarial Network Based Noise Modeling", *CVPR*, 2018
[7] Buzuk, Heinrich, and Stille, "Residual U-Net Convolutional Neural Network Architecture for Low-Dose CT Denoising", *Current Directions in Biomedical Engineering*, 2018
[8] Ahn, B., Cho, N.I.: 'Block-matching convolutional neural network for image denoising', *arXiv preprint arXiv*, 2017.
[9] Simonyan, K., Zisserman, A. "Very deep convolutional networks for large-scale image recognition", *arXiv*, 2014
[10] Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J., "Deep neural networks segment neuronal membranes in electron microscopy images", *NIPS*, 2012