2021

AI Denoising with MATLAB's Pretrained CNN

Henderson Van Hua, Chad Raymor, Lizthebett Valdivia
University of Houston, hvhua@cougarnet.uh.edu, cdraymor@cougarnet.uh.edu, lvaldiv2@cougarnet.uh.edu
COSC 4372, Medical Imaging

Dr. Nikolaos V. Tsekos
University of Houston, nvtsekos@central.uh.edu

Hosted on GitHub
https://github.com/chadraymor/COSC4372

Introduction

This work serves as a report for the provided Convolutional Neural Network (CNN) by MATLAB used to denoise images. The necessary functions were pulled from MATLAB's Imaging Processing Toolbox and Deep Learning Toolbox. The neural network was primarily tested on the language's phantom() function, a function that generates a basic image of a Shepp-Logan head phantom for processing. The results of this work will be found below along with thorough explanations of each part of the project.

Objectives

- **AI Denoising:** Test the pretrained CNN network to remove Gaussian noise without the existence of a training network.
  - Noise removal works only with 2-D single-channel images. Multiple color channels and 3-D images will require treating each channel or plane separately.
  - Pretrained network currently only recognizes Gaussian noise, with a limited range of standard deviation.
- **Image Analysis:** Determine how effective the pretrained network is at denoising.
  - Analyze the breaking point of the network. At what point does the amount of noise become too much to handle?

Algorithms

Input

- Modified Shepp-Logan Head Phantom image *logan1* with size of 256
  - *logan1* can be MRI images of the brain, CT images, and so forth.
    - *logan1* must be a 2-D, grayscale image for the pretrained network to operate. Otherwise must be operated for each color channel and plane.
- Noise Type and Noise Density *logan2*
  - Pretrained network limited to Gaussian Noise
    - Possible to have Poisson, Salt & Pepper, and Speckle Noise, but required retraining of the network.
  - Noise density specified as a numeric scalar, default of 0.05, and can be increased or decreased.
- Chosen denoising network *net*
  - Currently loaded with pretrained Convolutional Neural Network

Image Processing Toolbox

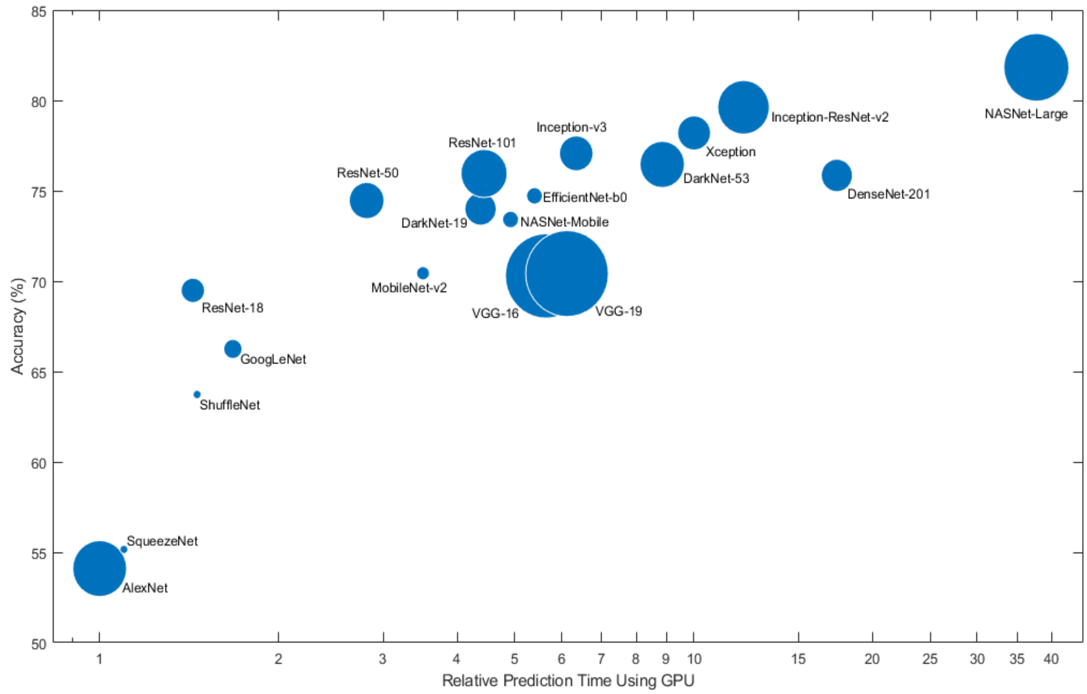
MATLAB's Image Processing Toolbox provides a comprehensive set of standardized algorithms and tools for image processing, analysis, visualization, and algorithm development. While the toolbox is primarily used for noise reduction in the project, it also includes other standard processing tools such as segmentation, image enhancement, geometric transformation — providing a base for further deep learning and image processing techniques.

Deep Learning Toolbox

The Deep Learning Toolbox created by MATLAB serves as a framework for designing and implementing deep neural networks with algorithms, and pretrained models. The framework has pretrained CNNs and long short-term memory (LSTM) networks to perform processing techniques on data. Not only that, the toolbox allows users to learn and build other network architectures such as generative adversarial networks (GANs) and Siamese networks using automatic differentiation, and custom training loops. Though, the Deep Learning Toolbox was primarily used to serve as a look towards AI Denoising and its effectiveness from a pretrained network's perspective.

Pretrained Image Denoising Network DnCNN and other Neural Networks

Our group tried several methods of training our own denoising network, but after discussions with Professor Tsekos, we decided to continue using MATLAB's multitude of pretrained networks. While there are networks such as GoogLeNet, SqueezeNet, Xception CNN, we stuck primarily with the dedicated denoising network DnCNN. However, we also found that MATLAB provides a cycle-consistent generative adversarial network (CycleGan) that is used to provide unsupervised medical image denoising. Although we focused on DnCNN, this would have been an alternative for AI Denoising.

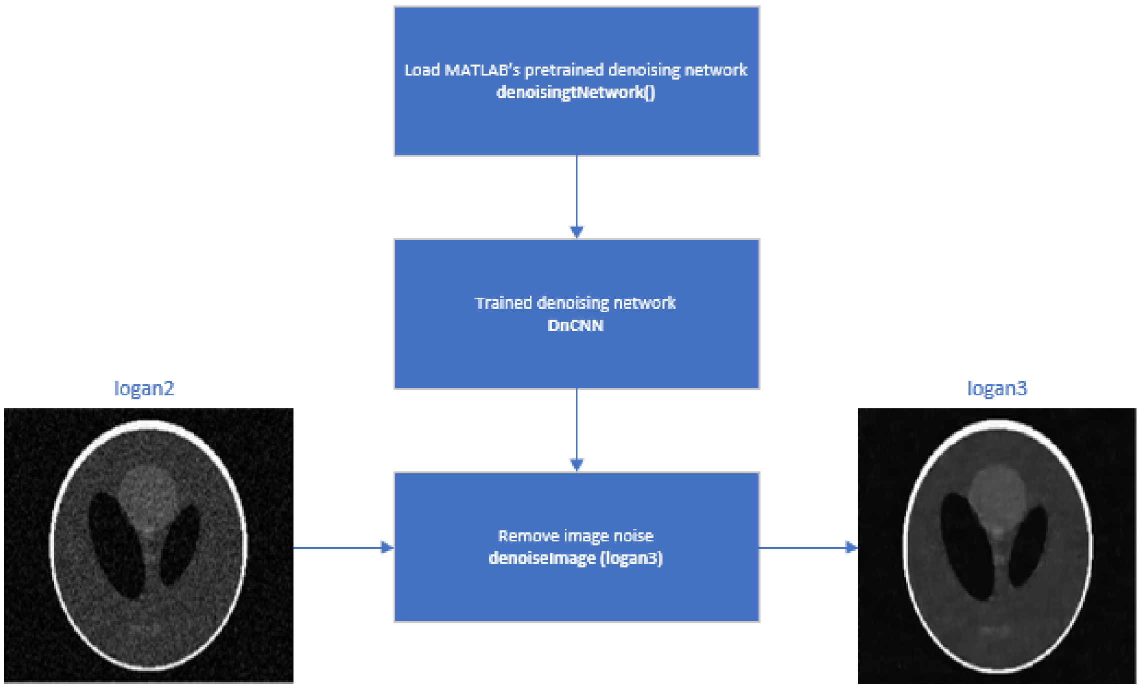


Graph comparing MATLAB's existing Neural Networks

DnCNN Characteristics

- By default the network has 20 convolution layers to its total of 59 layers.
- Limited to 2-D grayscale unless applied per channel and plane.
- DnCNN is trained using only the luminance channel because human perception is more sensitive to changes in brightness than changes in color.
- The DnCNN network can be trained to increase image resolution and remove JPEG compression artifacts.
    - Residual learning strategy, the DnCNN network learns to estimate the residual image. The residual contains information about the distortion, being blocking artifacts.

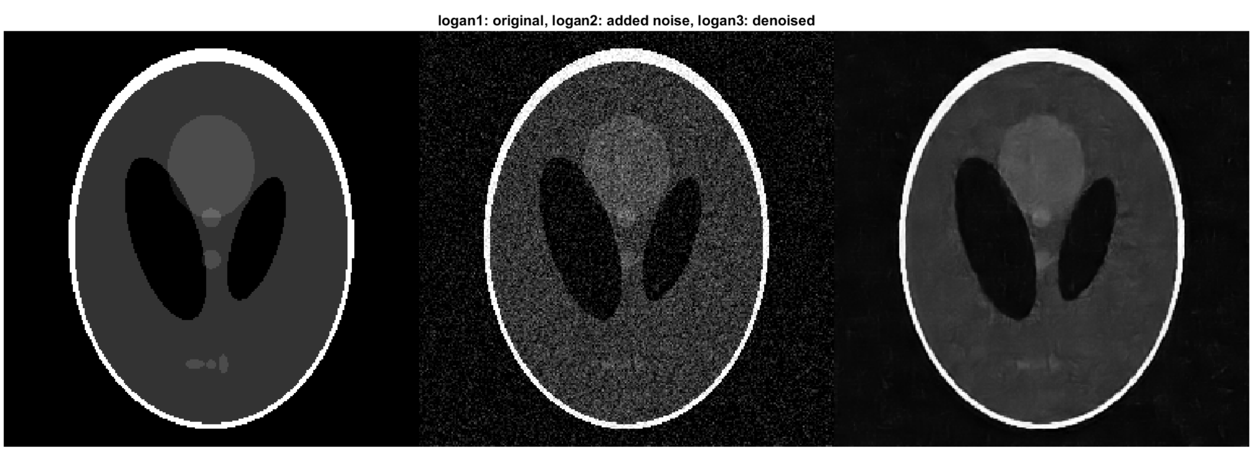
Simple Flow Diagram



Showcases flow of the pretrained network DnCNN

Image Analysis Showcase

To test MATLAB's pretrained network DnCNN, and locate its breaking point, we tested several noise density levels — their results found below:

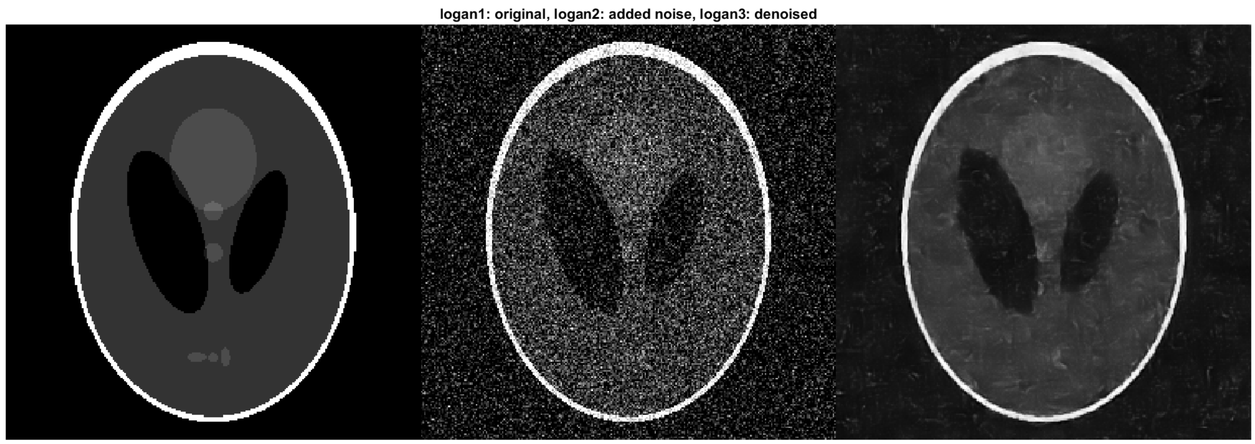


Noise Density: 0.01

Peak SNR: 22.85014

SNR: 10.43534

Denoised Image *logan3* holds most of the detail from the original image *logan1,* primarily losing detail towards the bottom ellipticals.

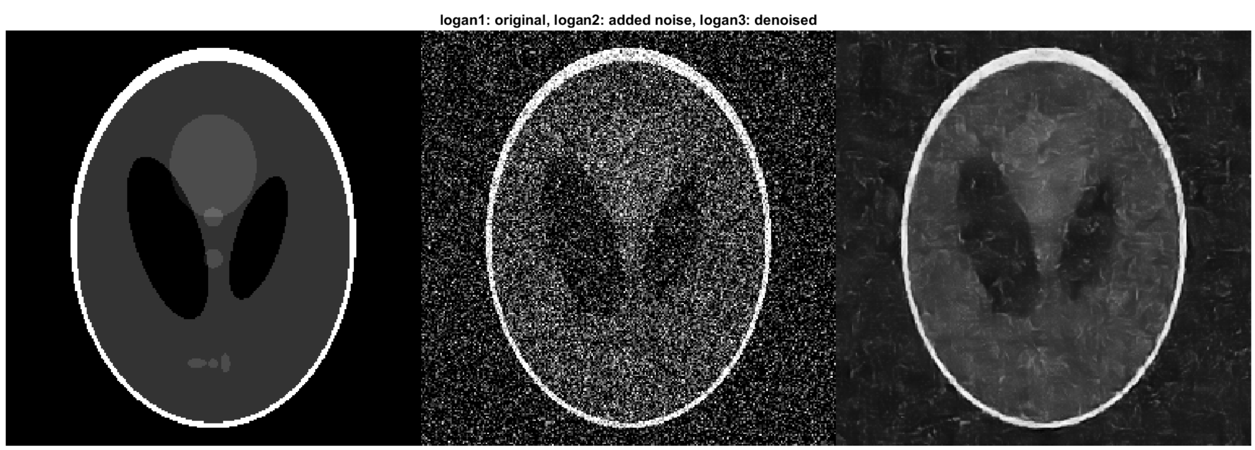


Noise Density: 0.05

Peak SNR: 16.95184

SNR: 4.75969

Denoised Image *logan3* continues to derive detail from *logan1*, the upper-light circle now blending more with the surrounding area, and the middle circles remaining visible through their outlines. At the bottom, the third standing elliptical remains partially visible, though the first two have been assimilated with the area.

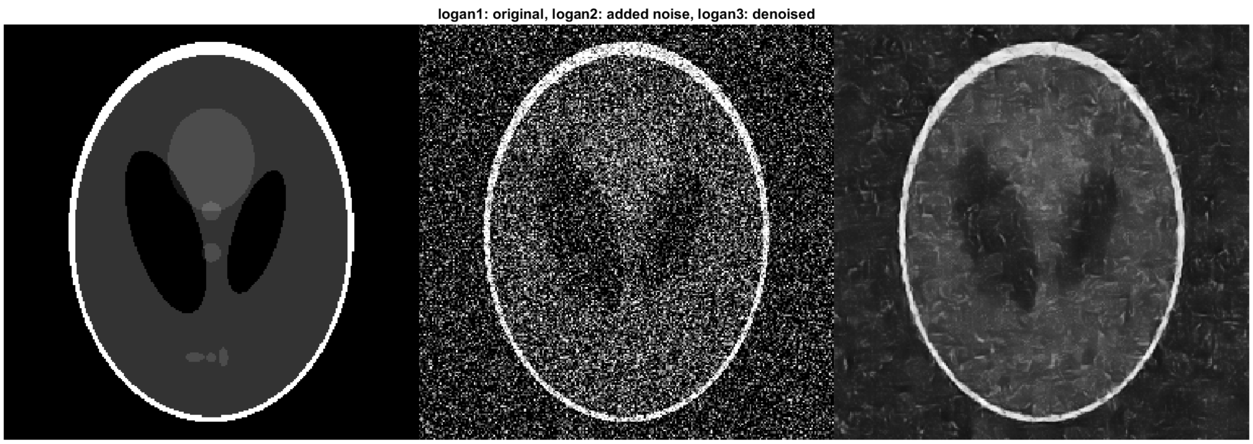


Noise Density: 0.10

Peak SNR: 14.44606

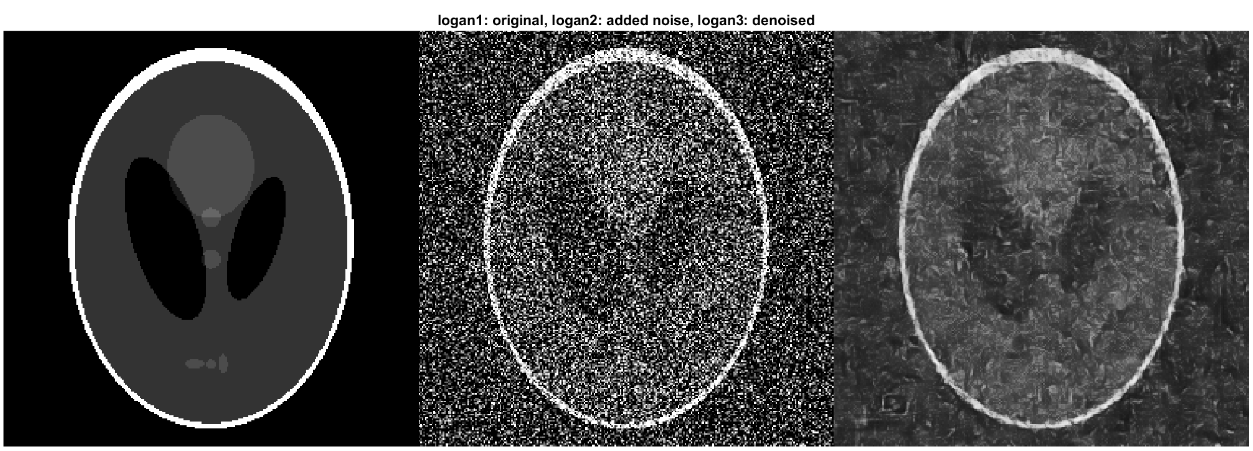
SNR: 2.73451

The upper-light circle just barely remains visible as an outline, and the top part of the middle circles has blended in with its surrounding, the bottom part still exists with its figure. The bottom ellipticals have completely lost their detail in comparison to the original. Compared to the first two results, the two black ellipticals have started to lose much of their shape.



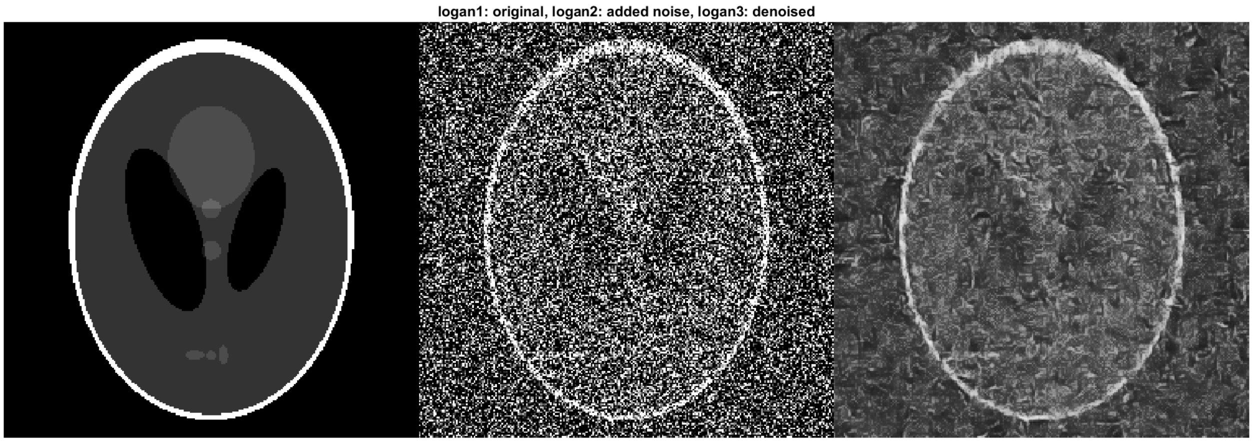
Noise Density: 0.15
Peak SNR: 13.19383
SNR: 1.89840

The two main, dark ellipticals remain visible from the original, albeit with great loss to its detail and outline. Surprisingly, the bright middle circle remains partially visible even though it blends almost entirely with the background.



Noise Density: 0.30
Peak SNR: 11.42544
SNR: 1.15389

Main ellipticals remain visible, but again continuing to lose detail. It arguably looks smaller and skinnier as the network struggles to denoise *logan2*. Bright, middle circle has been assimilated with the rest of the image now.

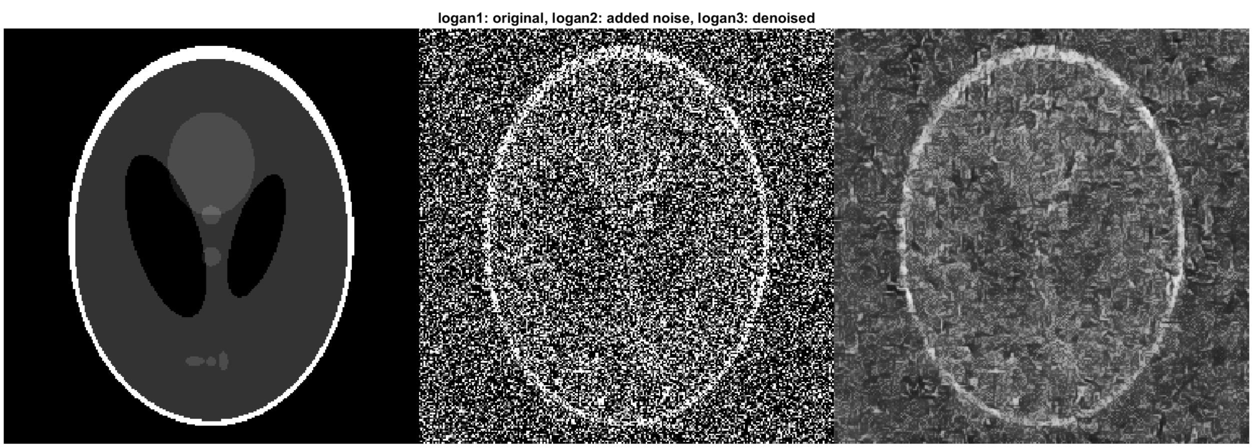


Noise Density: 0.60

Peak SNR: 10.42985

SNR: 1.35393

Main ellipticals are just barely visible, primarily appearing as slightly darker blobs in comparison to the brighter scale sections.

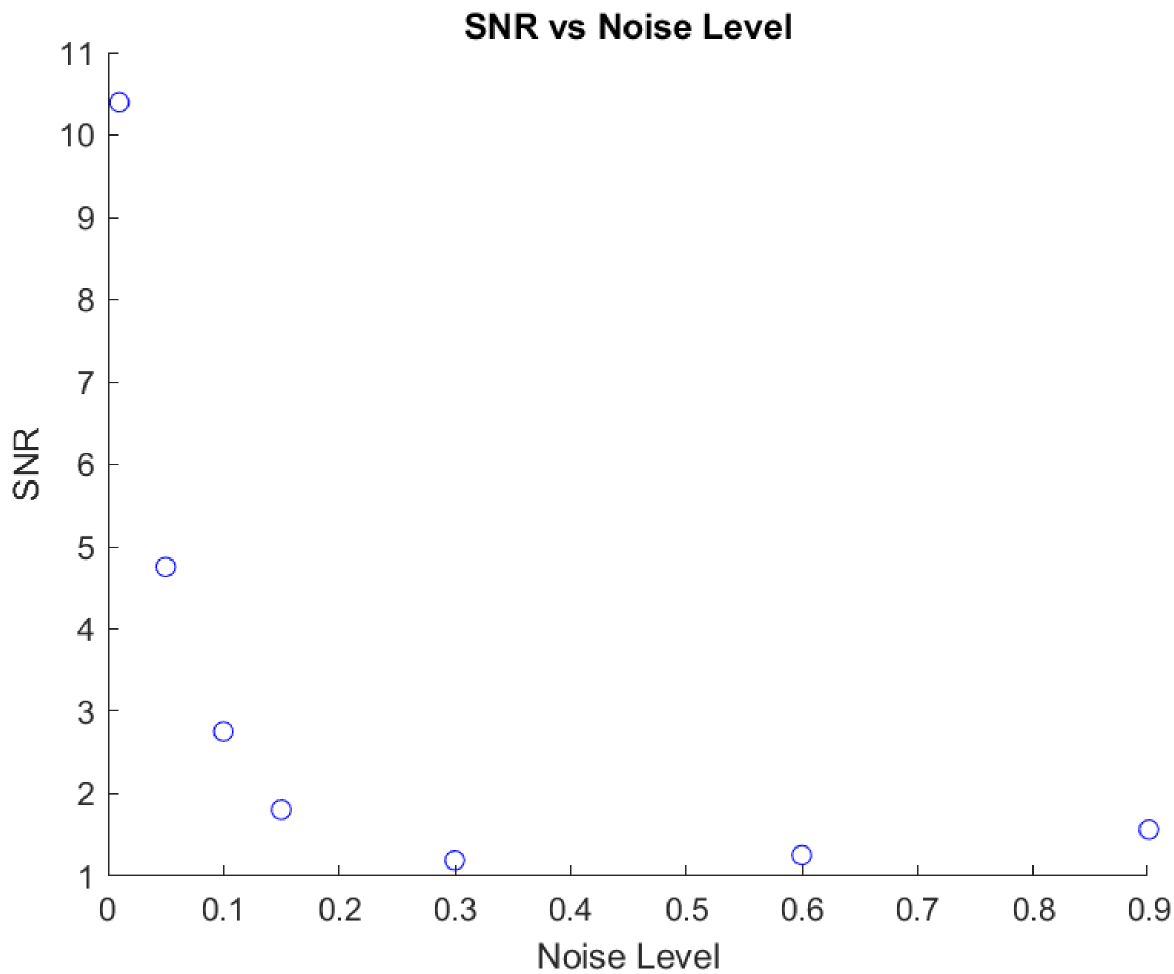


Noise Density: 0.90

Peak SNR: 10.05811

SNR: 1.50717

Denoised image has insuringly been lost in comparison to its original form, the pretrained network is incapable of denoising such an intense density of noise.



Found above is a graph showcasing the Single-to-Noise Ratio against the varying levels of Noise Densities. As expected, the SNR decreases as noise increases, though interestingly there is a slight bounce after noise density levels of 0.3.

Conclusion

This work serves as a showcase of MATLAB's pretrained denoising CNN and its effectiveness at numerous levels of noise density. While the network can not certainly perform effective levels of denoising at densities such as 0.90, most of the detail is already lost when approaching the 0.50 noise density mark. Prior to 0.50, the pretrained network arguably performs decently and can serve as a basis for other denoising networks.

Appendix A

The MATLAB code is hosted on GitHub.

https://github.com/chadraymor/COSC4372

- Denoise_test.m: The base of the project. Simply run for the results and data. Used for individual observation and testing.
- denoise_test_withGraph.m: Alternative that displays all models and noise levels along with SNR vs. Noise Level graph.