Samuel Adkins

Z23557914

Digital Image Processing

Due: April 10th, 2024

MATLAB Assignment 5 – Report

Part 1

Step 3 of Guidelines (optional)

I expanded "Part 1" to include other test images, noise type, and quality metrics.

The two additional pictures I use throughout this assignment are "TreeFogForest.jpg" and "colorbuilding.png", Im1 and Im2 respectively.

Before I can use TreeFogForest (Im1), I had to resize it and then I save the new RGB and grayscale version:

```
img_og = imread('TreeFogForest.jpg');
img_RGB = imresize(img_og, [648, NaN]); % Res
imwrite(img_RGB, 'IM1_RGB.png');

img_Gray = im2gray(img_RGB);
imwrite(img_Gray, 'IM1_Gray.png');

```

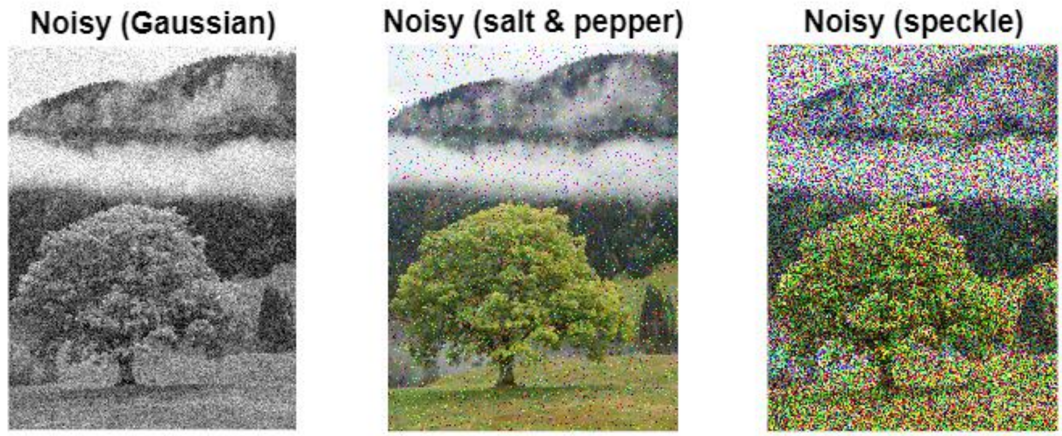
For an additional noise, I chose "speckle" with an intensity of 0.6. Then I added the different types of noise to the new image:

```
% Part 1 - different grayscale image / same (default) noise types
Im1_gaussian = imnoise(Im1_G,'gaussian');

% Part 2 - color image / same (default) noise types
Im1_sp = imnoise(Im1_RGB, 'salt & pepper');

% Part 3 - color image / different noise types/default values
% Add speckle noise
Im1_speck = imnoise(Im1_RGB, 'speckle', 0.6); % Adjust intensity as needed
```

The result is:



For an additional metric, I included Full-Reference MSE.

the human perception of quality.

```
IM1_orig_err = immse(Im1_RGB, Im1_RGB);
fprintf('\nIMMSE score for original image is %0.4f.\n',IM1_orig_err)
```

```
IMMSE score for original image is 0.0000.
```

```
IM1_gaus_err = immse(Im1_gaussian, Im1_G);
fprintf('\nIMMSE score for noisy (Gaussian) image is %0.4f.\n',IM1_gaus_err)
```

```
IMMSE score for noisy (Gaussian) image is 622.2289.
```

```
IM1_salt_err = immse(Im1_sp, Im1_RGB);
fprintf('\nIMMSE score for noisy (Salt & Pepper) image is %0.4f.\n',IM1_salt_err)
```

```
IMMSE score for noisy (Salt & Pepper) image is 1001.1901.
```

```
IM1_speck_err = immse(Im1_speck, Im1_RGB);
fprintf('\nIMMSE score for noisy (Speckle) image is %0.4f.\n',IM1_speck_err)
```

```
IMMSE score for noisy (Speckle) image is 7228.8409.
```

The highest IMMSE score is Speckle with a whopping 7216 with the intensity 0.6.

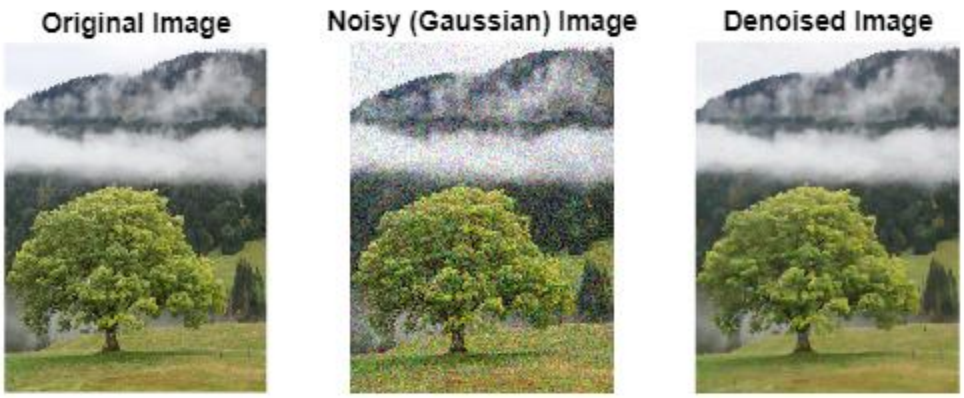
Part 2

(step 5 of the guidelines)

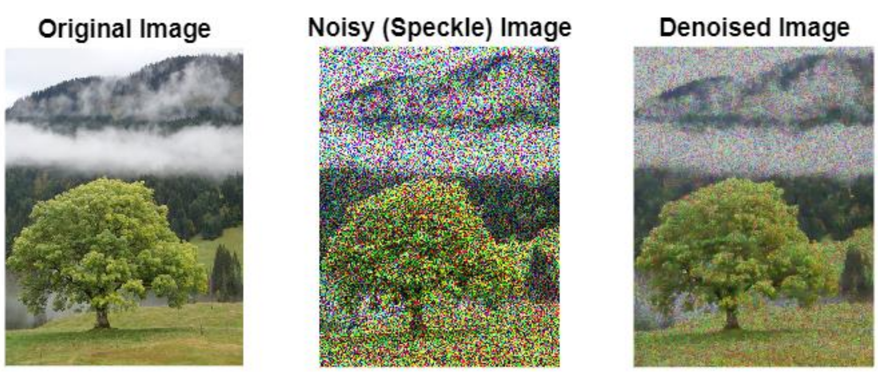
Additional Image 1

With the noisy versions of Im1, done above, I split the color channels and use the pre-trained DcCNN to denoise it, and the results are quite good.

First I did the Gaussian:



Then I used the DcCNN denoising technique to denoise the “speckle” version of Im1:



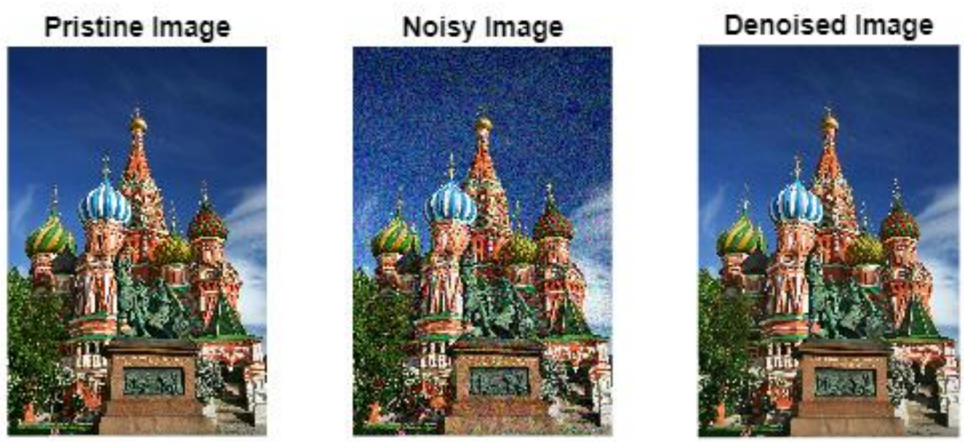
To compare the results, I made a table of the different metrics as columns, the first row as the Gaussian version, and the 2nd row as the Speckle version:

Image	PSNR	SSIM	MSE
"Im1_Gaus"	29.328	0.88399	0.0011672
"Im1_Speckle"	17.388	0.5363	0.018246

The results in the tabe above show that the denoise technique worked better on the Speckle version.

Additional Image 2

In the same fashion as Image 1, I created a gaussian noise version of "colorbuilding.jpg" and then denoised it using DcCNN:



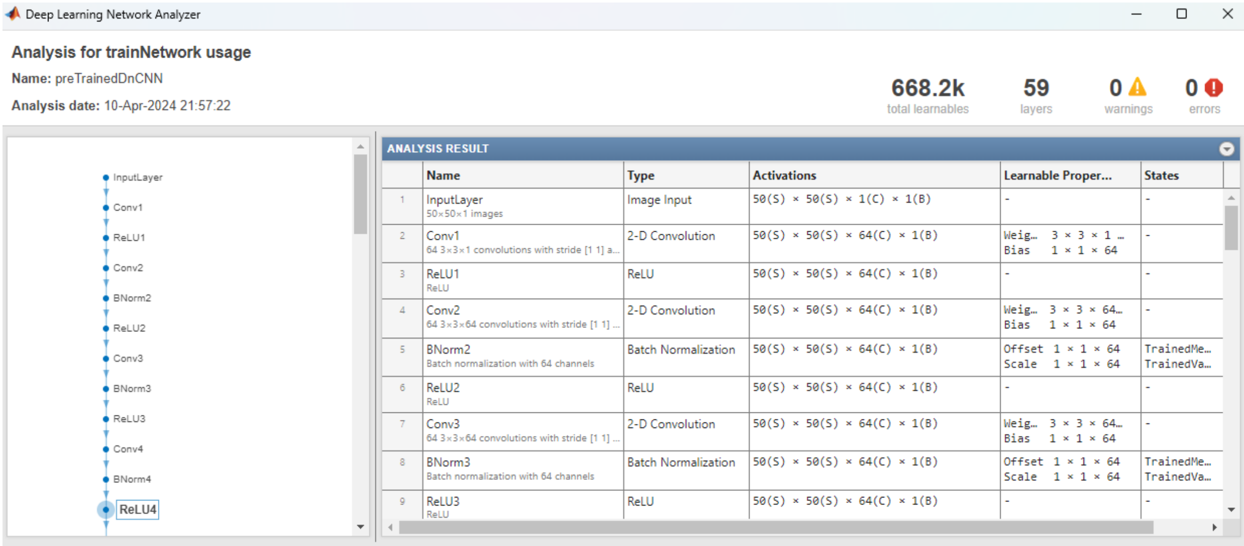
Then I compared the results of the speckle Im2 vs the denoised speckle Im1:

Image	PSNR	SSIM	MSE
"Im1"	17.388	0.5363	0.018246
"Im2"	28.828	0.9476	0.0013097

I was wondering if the amount of colors in the picture would have a significant impact on the performance, but it did not seem to.

Your turn (step 7) (optional)

To analyze the network, I just used analyzeNetwork(preTrainedDnCNN) and looked through the result:

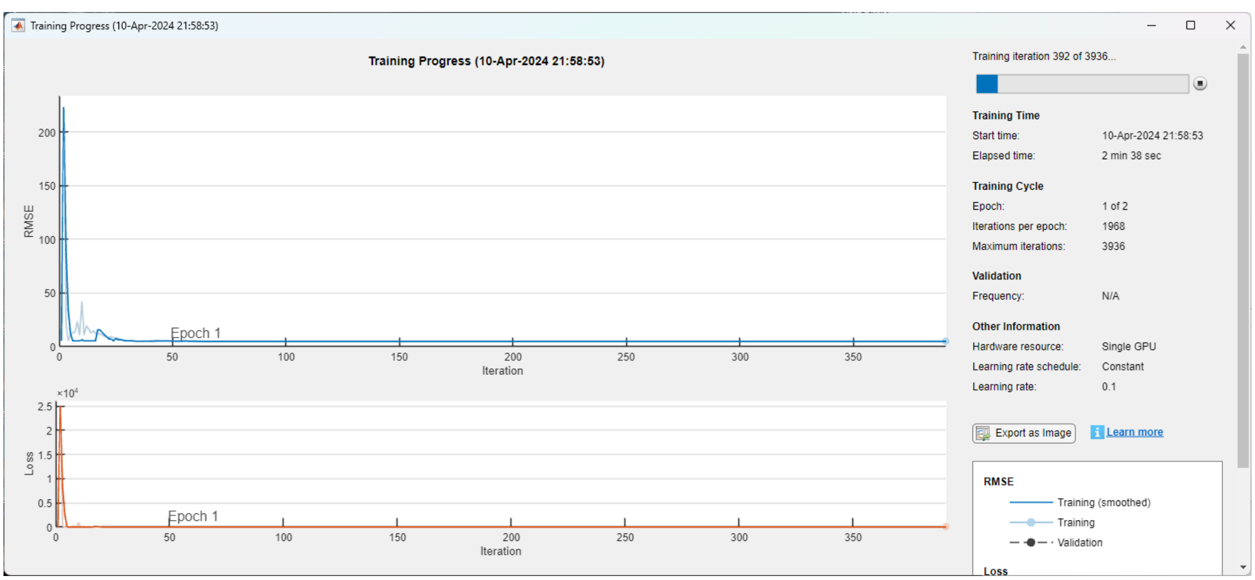


Part 3 Optional

I imported the tulip data folder from GitHub: https://github.com/ogemarques/data-files/blob/main/tulip.zip

Make sure to add the files to your MATLAB Set Paths.

The training converges very fast with the given parameters:



Guideline Step 8 – Questions

Question 1

How did the deep-learning-based noise reduction technique perform according to the selected image quality metrics? Did you notice anything unusual?

The performance of the denoising met my expectations and I found nothing unusual:

```
The MSE value is 0.0182
```

```
T = table(["Im1_Gaus"; "Im1_Speckle"], [psnrVal_Im1Gaus; psnrVal_Im1Speck], [ssimVal_Im1Gaus; ssimVal_Im1Spe
    'VariableNames', {'Image', 'PSNR', 'SSIM', 'MSE'});

% Display the table
disp(T);
```

Image	PSNR	SSIM	MSE
"Im1_Gaus"	29.328	0.88399	0.0011672
"Im1_Speckle"	17.388	0.5363	0.018246

Comparing the NIQE scores:

```
niqeIm1_original = niqe(Im1_RGB);
niqeIm1_gaus = niqe(Im1_gaussian);
niqeIm1_gaus_deno = niqe(denoisedRGB_Im1Gaus);
niqeIm1_speck = niqe(Im1_speck);
niqeIm1_speck_deno = niqe(denoisedRGB_Im1Speck);


% Create a table
TN = table(niqeIm1_original, niqeIm1_gaus, niqeIm1_speck, niqeIm1_gaus_deno, niqeIm1_speck_deno,...
    'VariableNames', {'Original', 'Gaussian', 'Speckle', 'Denoised Gaussian', 'Denoised Speckle'});

disp(TN);
```

Original	Gaussian	Speckle	Denoised Gaussian	Denoised Speckle
2.6233	10.791	26.959	2.9516	5.0612

Question 2

Did you notice any significant difference between the quality of the denoised images for Gaussian versus salt-and-pepper (a.k.a. speckle) noise? If so, what is your explanation for the differences in quality?

I did notice a significantly better result in the Gaussian images than the speckle, but the speckle also had a higher intensity.

QUESTION 3:

Based on your observations so far, which recommendations would you give to someone who is new to this topic and wants to use deep-learning-based solutions to remove/reduce noise from images for a particular application? Hint: Simpler is often better.

Models do not train well on Apple Silicon; use Windows if you can.

Question 4:

Was the choice of hyperparameters (learning rate, number of epochs, etc.) appropriate?

Question 5:

Is the network overfitting? Explain.

Probably, because there is not enough random variation and in the grand scheme, the training set is small.

Question 6:

Which general principle of autoencoders explains why they work so well, in general, for denoising?

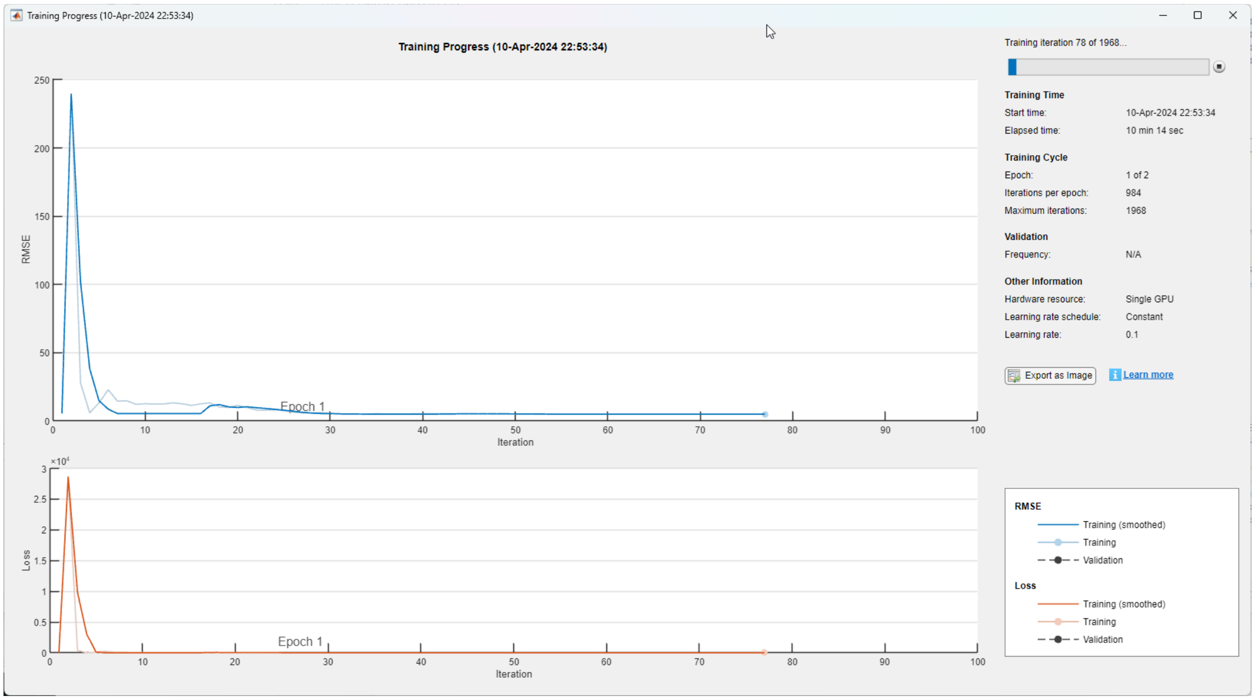
Autoencoders learn to recognize important parts of an image while ignoring others, such as the noise. This ability helps them clean up the image by focusing on the key features and reducing the unwanted noise.

Question 7:

Did you notice any significant difference between the quality of the denoised images for Gaussian versus salt-and-pepper (a.k.a. speckle) noise? If so, what is your explanation for the differences in quality?

I noticed the gaussian had better denoised results across the board and I believe it was due to my speckle image having a high intensity when making the noise.

OPTIONAL DnCNN Training



Difficulties

1. Implementing early stopping was challenging and I never got it right.
2. Training does not perform well on Apple Silicon.
3. I got a lot of graphical timeout warnings when working with ssimMap.