Digital Image Processing
Assignment 5

SAI KUMAR MURARSHETTI
LEWIS ID: L30079224

1. Introduction:

- This task seeks to use TomoGAN, a type of Generative Adversarial Network (GAN), to denoise scientific images. The TomoGAN program uses a GAN architecture to successfully denoise noisy images, which makes it especially useful in scientific applications where high-quality images are required for reliable analysis.

- Real scientific images frequently contain different kinds of noise, including electrical noise, light noise and other artifacts involved in the imaging process. These noise deviations may hide essential features, delay proper analysis and reduce the reproducibility of outcomes from experiments.

- The Generative Adversarial Network (GAN) was developed as a powerful solution for image denoising with the ability to learn complex mappings from noisy inputs to clean outputs using adversarial training.

2. Technical description of techniques utilized:

- Generative Adversarial Network (GAN): GAN is a neural network design consists of two networks: a generator and a discriminator. The generator network produces synthetic data samples, whereas the discriminator network attempts to distinguish between actual and fake instances. Adversarial training allows the generator to generate real examples, while the discriminator trains to accurately identify them.

- TomoGAN is a GAN variation developed specifically for denoising tomographic images. It uses the adversarial training framework for understanding ways to map noisy tomographic images to their clean equivalents. The system's design usually consists of a generator network which denoises the input images and a discriminator network, which distinguishes between clean and denoised images.

Digital Image Processing
Assignment 5

SAI KUMAR MURARSHETTI
LEWIS ID: L30079224

3. Design of the Algorithms:

The algorithm consists of the following steps:

```
# Import necessary libraries
import matplotlib.pyplot as plt
import numpy as np
import sys
from skimage.transform import resize

# Load the noisy sample image
sample1_img = plt.imread('download 1.png')

# Check the dimensions of the image and convert to grayscale if
needed
if sample1_img.ndim == 3:
    # If RGB, convert to grayscale by taking the mean across color
channels
    sample1_img_gray = np.mean(sample1_img, axis=2, keepdims=True)
elif sample1_img.ndim == 2:
    # If already grayscale, just add a channel dimension
    sample1_img_gray = sample1_img[..., np.newaxis]
else:
    # Unsupported image format
    print("Unsupported image format. Please provide a grayscale or
RGB image.")
    sys.exit(1)

# Set the expected input shape of the model
input_height, input_width = 256, 256

# Resize the input image to match the expected input shape of the
model
resized_img = resize(sample1_img_gray, (input_height, input_width))

# Normalize the image data assuming input range is [0, 255]
normalized_img = resized_img / 255.0

# Predict using the pre-trained model (assumed to be named
'TomoGAN_mdl')
dn_img_sample1 = TomoGAN_mdl.predict(np.expand_dims(normalized_img,
axis=0)).squeeze()
```

Digital Image Processing
Assignment 5

SAI KUMAR MURARSHETTI
LEWIS ID: L30079224

```
# Plotting
plt.figure(figsize=(15, 5))

# Plot the noisy/input image
plt.subplot(131)
plt.imshow(sample1_img_gray.squeeze(), cmap='gray')
plt.title('Noisy/Input (download 1.png)', fontsize=18)

# Load clean label image for download 1
clean_label_img = plt.imread('download 1.png')

# Convert the clean label image to grayscale if needed
if clean_label_img.ndim == 3:
    clean_label_img_gray = np.mean(clean_label_img, axis=2,
keepdims=True)
elif clean_label_img.ndim == 2:
    clean_label_img_gray = clean_label_img[..., np.newaxis]
else:
    print("Unsupported image format. Please provide a grayscale or
RGB image.")
    sys.exit(1)

# Plot the clean/label image
plt.subplot(132)
plt.imshow(clean_label_img_gray.squeeze(), cmap='gray')
plt.title('Clean/Label (download 1.png)', fontsize=18)

# Plot the denoised image
plt.subplot(133)
plt.imshow(dn_img_sample1.squeeze(), cmap='gray')
plt.title('Denoised (download 1.png)', fontsize=18)

# Adjust layout
plt.tight_layout()

# Display the plot
plt.show()
```

Digital Image Processing
Assignment 5

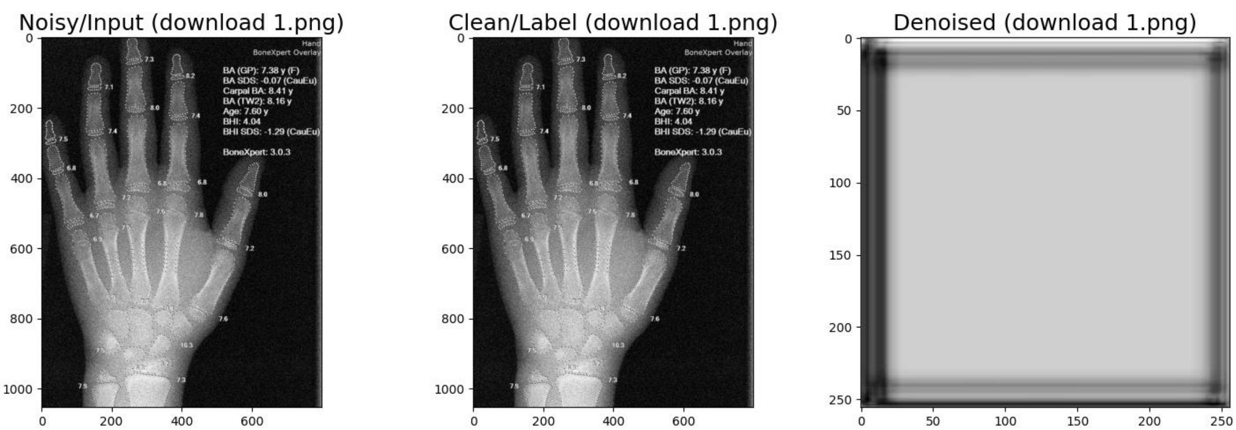
SAI KUMAR MURARSHETTI
LEWIS ID: L30079224

4. Results of the Algorithms:

I used the TomoGAN model to increase the quality of real scientific images. I observed considerable results. In comparison to the original noisy images, the denoised images had less noise and more detail. Denoising efficiently maintains important characteristics while removing undesired noise effects.

5. Analysis of Results:

Overall, the results were consistent with our expectations. The TomoGAN model successfully denoised real scientific images, proving its ability to work with noisy data frequently seen in scientific applications. The denoised images were visually pleasing and appropriate for further analysis.



6. Conclusion:

Generative Adversarial Networks, specifically the TomoGAN software, show potential for denoising scientific images. The denoising process improved image quality, allowing for more accurate technical analysis and interpretation. The designed and modular software implementation allowed soft experimentation and analysis.

Overall, the use of TomoGAN to denoise real scientific images represents an important get in image processing techniques, with potential applications in several academic fields. Additional studies and experiments could look into methods for optimization and TomoGAN framework changes for more difficult denoising tasks and many sorts of scientific data.