1 Image Processing

In this section, we will explain a Python code that performs basic image processing operations using the NumPy, Matplotlib, and scikit-image libraries.

1.1 Code Explanation

The following code performs the following operations:

- Import necessary libraries:

```
import numpy as np
import matplotlib.pyplot as plt
from skimage.transform import resize
```

- Load an input image:

```
img = plt.imread('CHNCXR_0004_0.png')
```

- Resize the image to a new shape of 64x64:

```
img_resized = resize(img, (64, 64), anti_aliasing=True)
```

- Convert the image to grayscale if necessary:

```
if len(img_resized.shape) > 2 and img_resized.shape[2] == 3:
    img_resized = np.dot(img_resized[..., :3], [0.2989, 0.5870, 0.1140])
```

- Plot a histogram of the pixel values in the resized image:

```
plt.hist(img_resized.ravel(), bins=256, range=(0.0, 1.0))
plt.xlabel('Pixel value')
plt.ylabel('Frequency')
plt.title('Histogram of resized image')
plt.show()
```

The resulting code is as follows:

```
import numpy as np
import matplotlib.pyplot as plt
from skimage.transform import resize

Load the input image

img = plt.imread('CHNCXR_0004_0.png')
```

```
Resize the image to 64x64

img_resized = resize(img, (64, 64), anti_aliasing=True)

Convert the image to grayscale if it's not already

if len(img_resized.shape) > 2 and img_resized.shape[2] == 3:
img_resized = np.dot(img_resized[..., :3], [0.2989, 0.5870, 0.1140])

Plot the histogram

plt.hist(img_resized.ravel(), bins=256, range=(0.0, 1.0))
plt.xlabel('Pixel value')
plt.ylabel('Frequency')
plt.title('Histogram of resized image')
plt.show()
```

1.2 Explanation of the Code

The code starts by importing three libraries: NumPy, Matplotlib, and scikit-image. NumPy is a Python library that provides support for multi-dimensional arrays and matrices, as well as a large collection of mathematical functions to operate on them. Matplotlib is a plotting library that provides functions for creating a variety of charts and plots. scikit-image is a Python library that provides a collection of algorithms for image processing.

Next, the code loads an input image using the `imread` function from the Matplotlib library. The image is stored in the `img` variable.

The image is then resized to a new shape of 64x64 using the `resize` function from the scikit-image library.