6CS012 - Artificial Intelligence and Machine Learning. Image Compression and Decompression using PCA.

Name: Surendra Giri
Student ID: 2357810

Task: Workshop-1

——————— Worksheet - 1. ———————-

2 Getting Started with Image Processing with Python.

Introduction to Python Imaging Library(PIL)

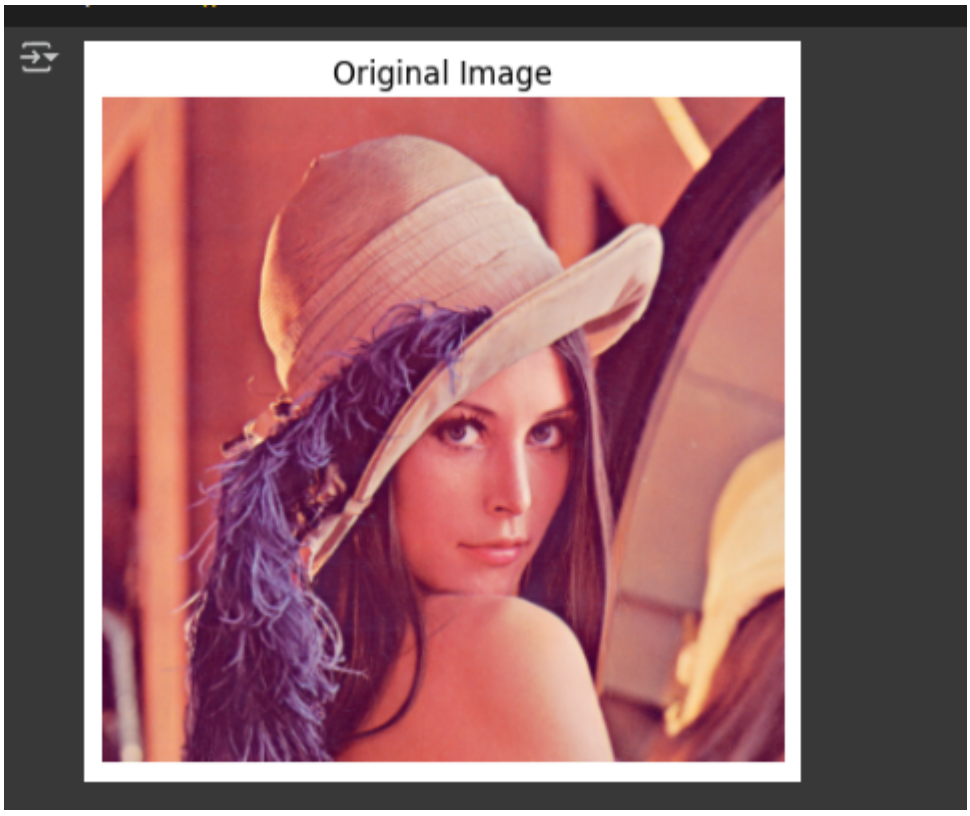
2.1 Exercise - 1:

Complete all the Task.

1. Read and display the image.

    • Read the image using the Pillow library and display it.
    • You can also use matplotlib to display the image.

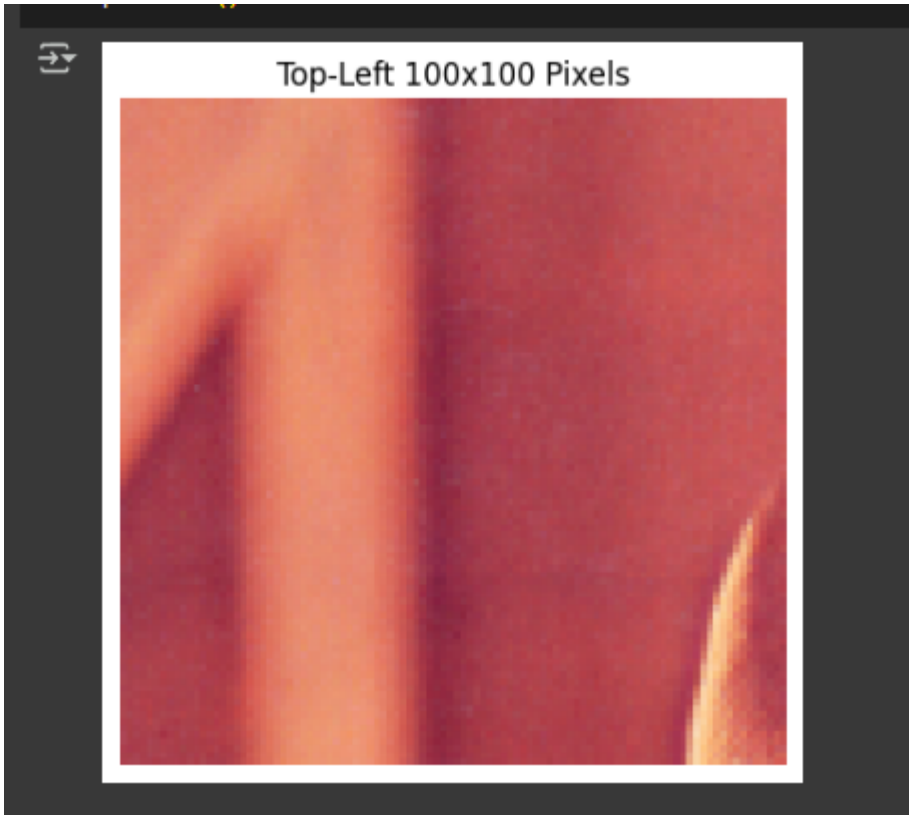
    Output Image



2. Display only the top left corner of 100x100 pixels.

    • Extract the top-left corner of the image (100x100 pixels) and display it using NumPy and Array Indexing.

    Output Image



3. Show the three color channels (R, G, B).

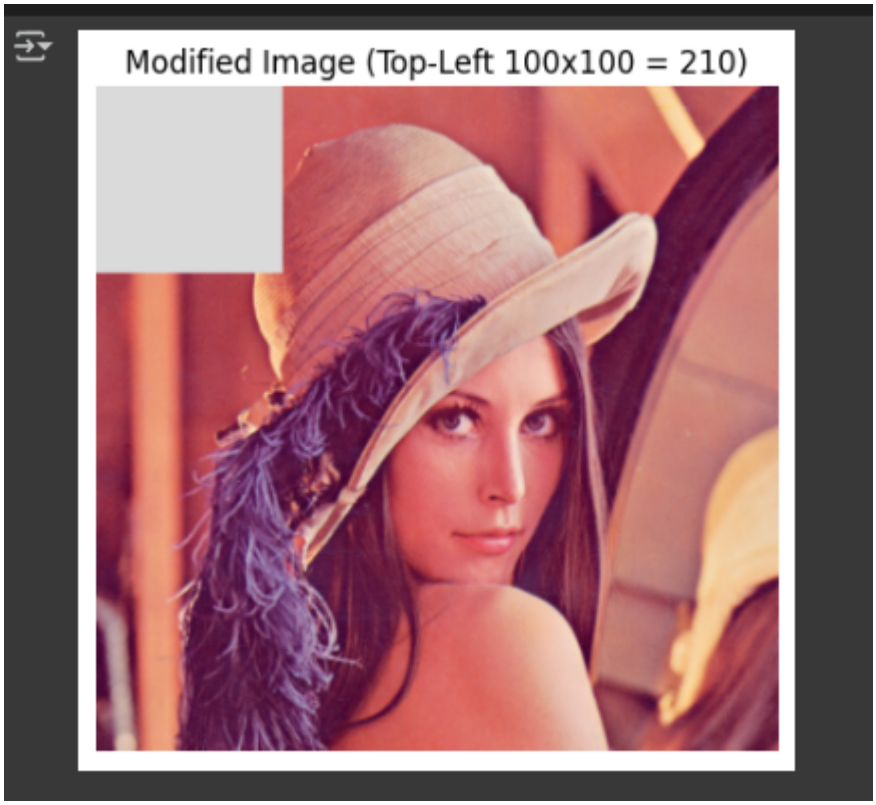
   • Separate the image into its three color channels (Red, Green, and Blue) and display them individually, labeling each channel as R, G, and B.*{Using NumPy.}*

   Output Image

4. Modify the top 100 × 100 pixels to a value of 210 and display the resulting image:

   • Modify the pixel values of the top-left 100 × 100 region to have a value of 210 (which is a light gray color), and then display the modified image.

   Output Image


Modified Image (Top-Left 100x100 = 210)

6CS012 Worksheet - 1 Siman Giri

2.2 Exercise - 2:

Complete all the Task.

1. Load and display a grayscale image.

    • Load a grayscale image using the Pillow library.

    • Display the grayscale image using matplotlib.

    Output Image

2. Extract and display the middle section of the image (150 pixels).

    • Extract a 150 pixel section from the center of the image using NumPy array slicing. • Display this cropped image using matplotlib.

    Output Image

3. Apply a simple threshold to the image (e.g., set all pixel values below 100 to 0).

    • Apply a threshold to the grayscale image: set all pixel values below 100 to 0, and all values above 100 to 255 (creating a binary image).

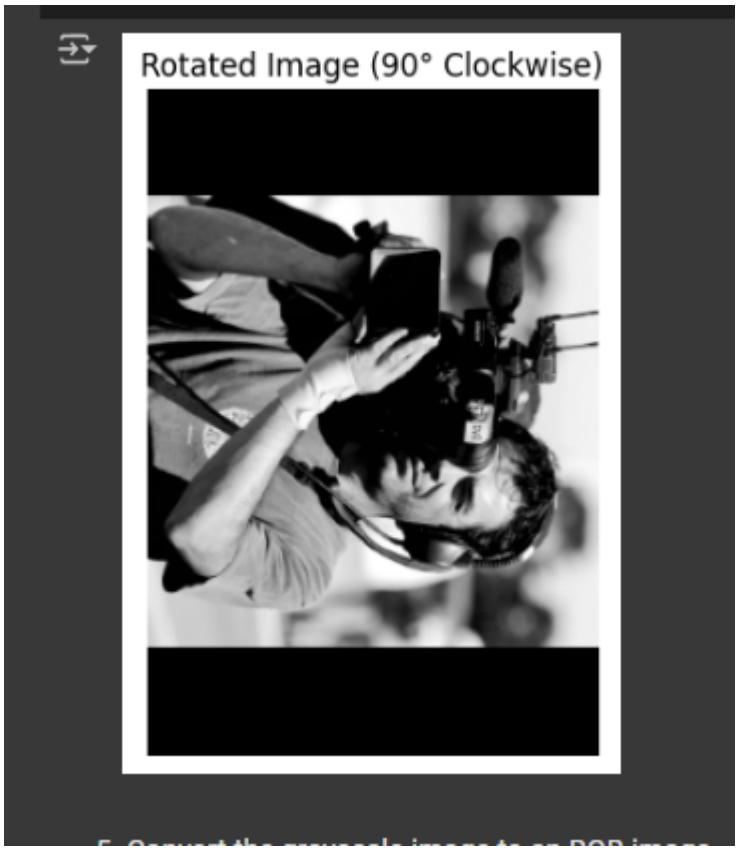
    • Display the resulting binary image.

    Output Image

4. Rotate the image 90 degrees clockwise and display the result.

- Rotate the image by 90 degrees clockwise using the Pillow rotate method or by manipulating the image array.
- Display the rotated image using matplotlib.

Output Image



5. Convert the grayscale image to an RGB image.

- Convert the grayscale image into an RGB image where the grayscale values are replicated across all three channels (R, G, and B).
- Display the converted RGB image using matplotlib.

3 Image Compression and Decompression using PCA.

In this exercise, build a PCA from scratch using explained variance method for image compression task. You are expected to compute the necessary matrices from the scratch. Dataset: Use image of your choice.

1. Load and Prepare Data:

    • Fetch an image of you choice.*{*If colour convert to grayscale*}*

    • Center the dataset - Standaridze the Data.

    • Calculate the covaraince matrix of the Standaridze data.

    Output Image

2. Eigen Decomposition and Identifying Principal Components:

    • Compute Eigen Values and Eigen Vectors.

    • Sort the eigenvalues in descending order and choose the top k eigenvectors corresponding to the highest eigenvalues.

    • Identify the Principal Components with the help of cumulative Sum plot.

    Output Image

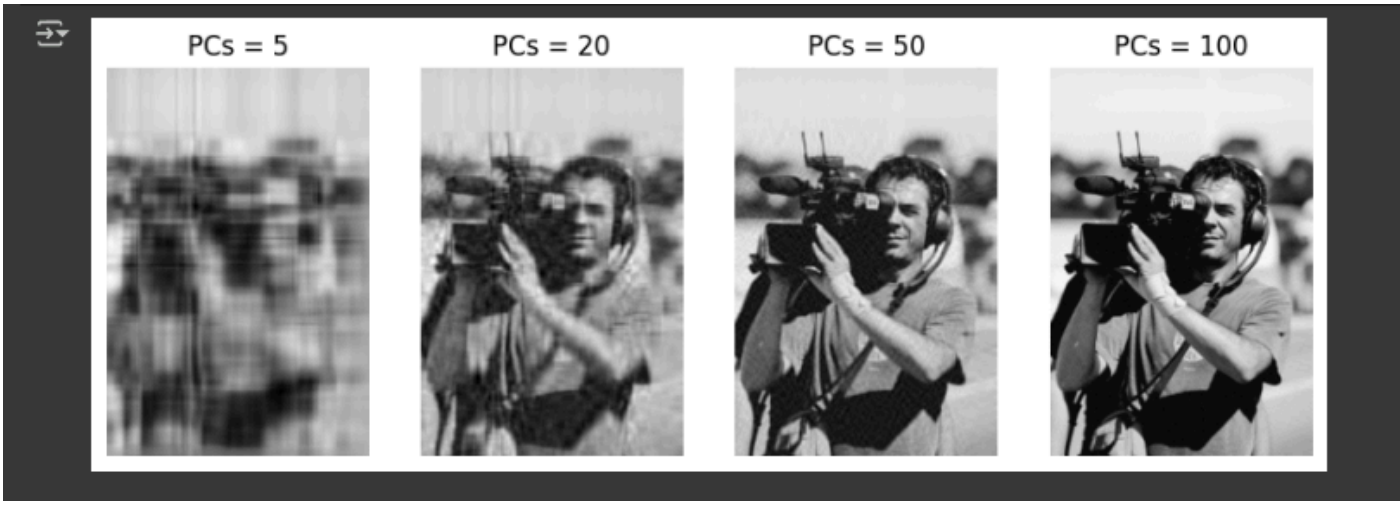

PCs = 5
PCs = 20
PCs = 50
PCs = 100

Figure 3: Sample Output for various principal Components.