EC7212: COMPUTER VISION AND IMAGE PROCESSING

TAKE HOME ASSIGNMENT 1

NAME : ABERUWAN R. M. M. P.

REG. NO. : EG/2020/3797

SEMESTER : 07

DATE : 21/06/2025

Contents

List of Figures

1 Introduction

This report presents solutions to four fundamental image processing tasks using Python and Google Colab. The tasks demonstrate intensity quantization, spatial filtering, geometric transformation, and spatial resolution reduction, as outlined in the assignment guidelines.

EC7212 – Computer Vision and Image Processing

Take Home Assignment 1

Write Python programs to perform the following image processing operations.

1. To reduce the number of intensity levels in an image from 256 to 2, in integer powers of 2. The desired number of intensity levels needs to be a variable input to your program.

2. Load an image and then perform a simple spatial 3x3 average of image pixels. Repeat the process for a 10x10 neighborhood and again for a 20x20 neighborhood.

3. Rotate an image by 45 and 90 degrees.

4. For every 3×3 block of the image (without overlapping), replace all the corresponding 9 pixels by their average. This operation simulates reducing the image spatial resolution. Repeat this for 5×5 blocks and 7×7 blocks.

Submission: Report including the code and the results (Include the GitHub link).

Figure 1.1: Take Home Assignment 1

2 Task 1: Intensity Level Reduction

2.1 Objective

- Reduce the number of intensity levels in a grayscale image from 256 to a user-specified value (power of 2).

2.2 Methodology

- The image is quantized by dividing the intensity range into equal intervals.
- Each pixel is mapped to the nearest lower interval value.

2.3 Results

- The original and quantized images are displayed for visual comparison.
- As the number of levels decreases, image detail is lost but storage requirements are reduced.

2.4 Code and Output

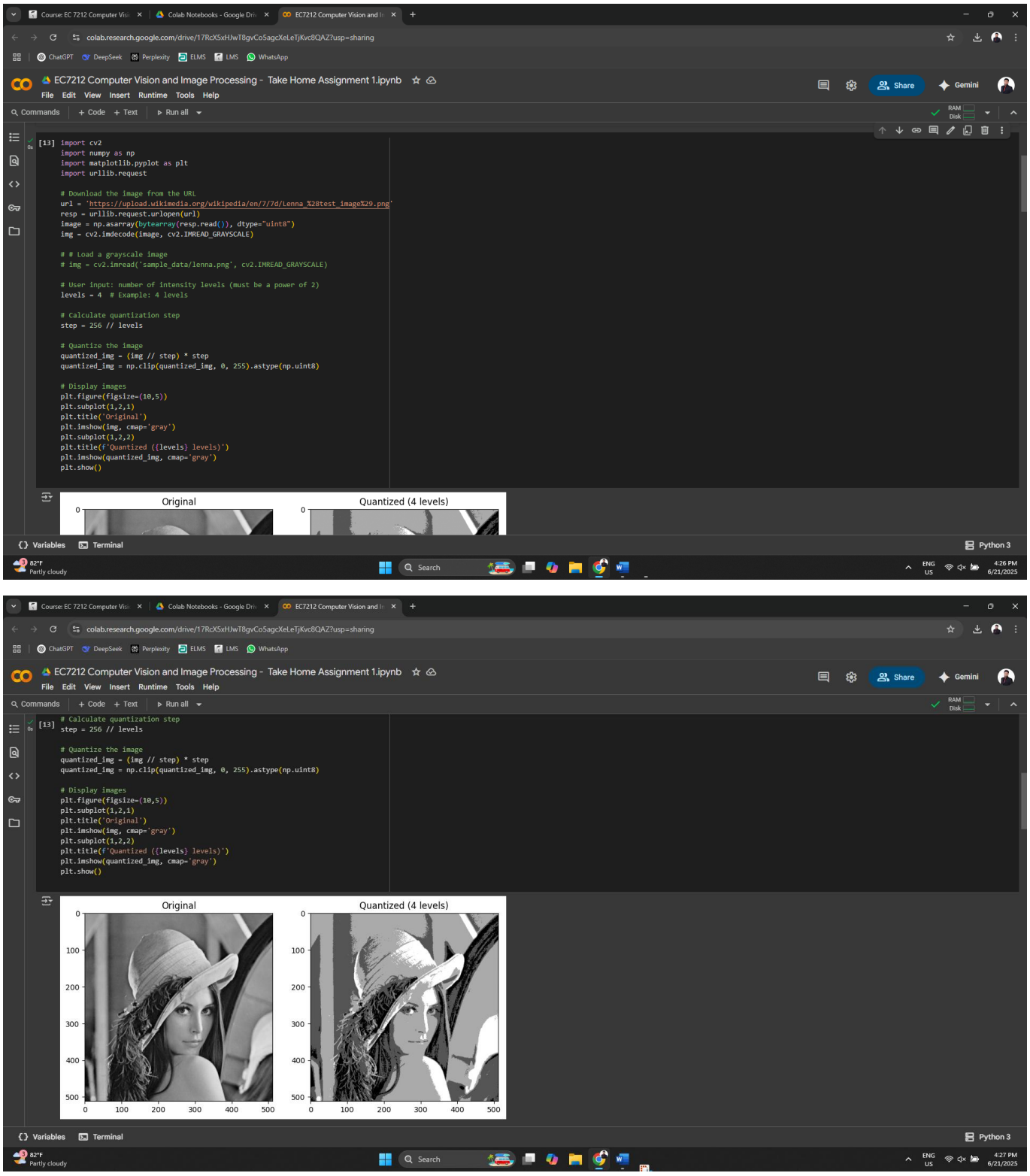

Figure 2.1: Implementation and Results for Task 1

3 Task 2: Spatial Averaging with Different Neighborhoods

3.1 Objective

- Apply mean filtering with 3x3, 10x10, and 20x20 kernels.

3.2 Methodology

- Mean filtering is performed using OpenCV's cv2.blur function.
- Larger kernels result in stronger smoothing.

3.3 Results

- Images filtered with increasing kernel sizes show progressive blurring.

3.4 Code and Output

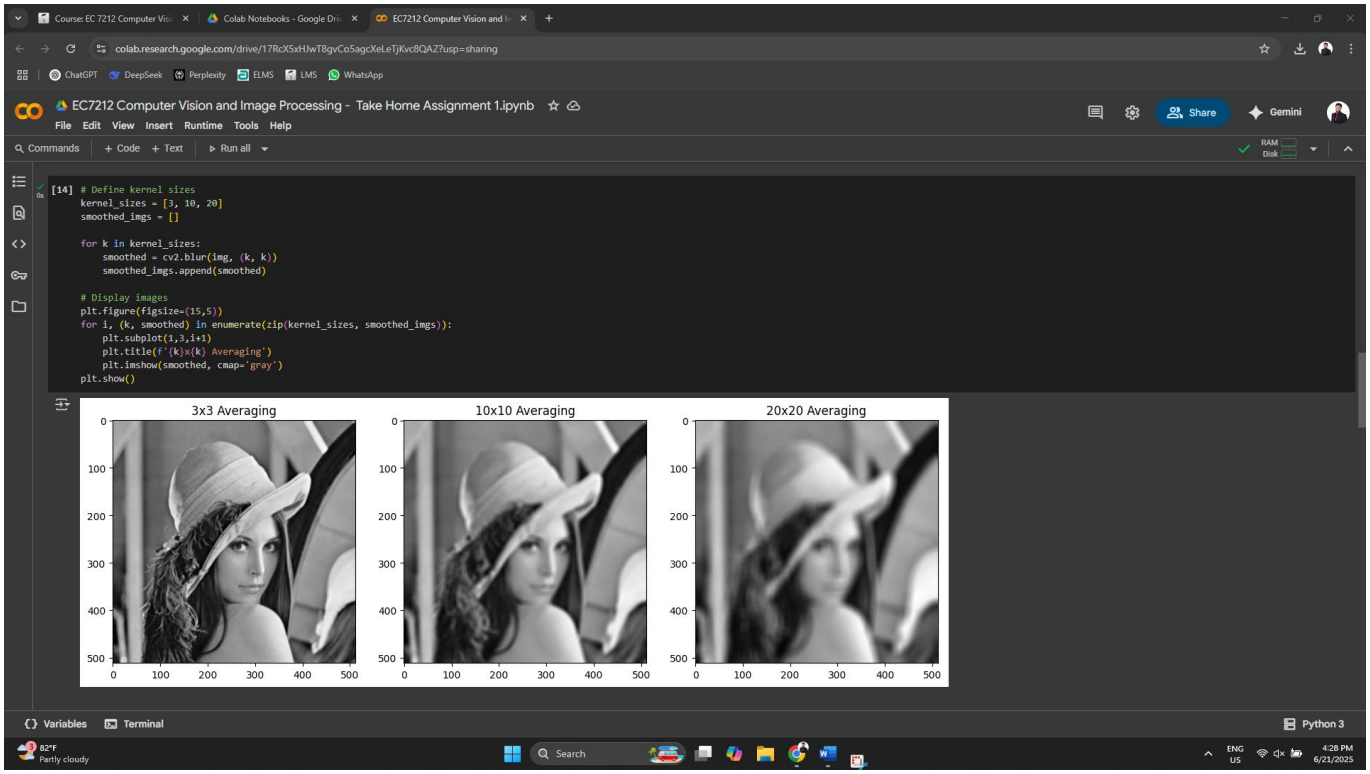

Figure 3.1: Implementation and Results for Task 2

4 Task 3: Image Rotation

4.1 Objective

- Rotate the image by 45° and 90°.

4.2 Methodology

- Rotation is accomplished using affine transformations.
- The center of the image is used as the pivot point.

4.3 Results

- Rotated images are displayed, confirming correct geometric transformation.

4.4 Code and Output

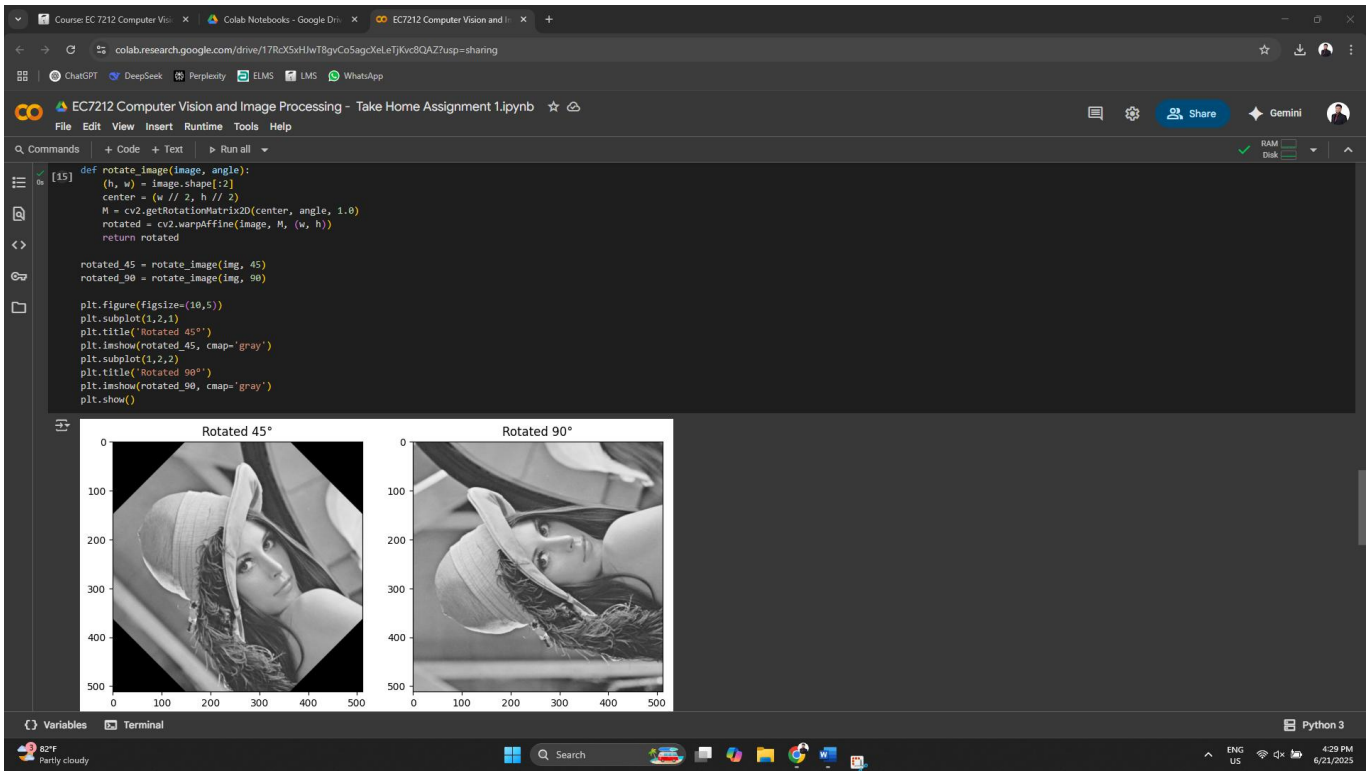

Figure 4.1: Implementation and Results for Task 3

5 Task 4: Blockwise Averaging (Spatial Resolution Reduction)

5.1 Objective

- Replace each non-overlapping block (3x3, 5x5, 7x7) with its average, simulating spatial resolution reduction.

5.2 Methodology

- The image is processed in non-overlapping blocks.
- Each block's pixels are set to the average value of the block.

5.3 Results

- Images with larger block sizes appear more pixelated, simulating lower resolution.

5.4 Code and Output

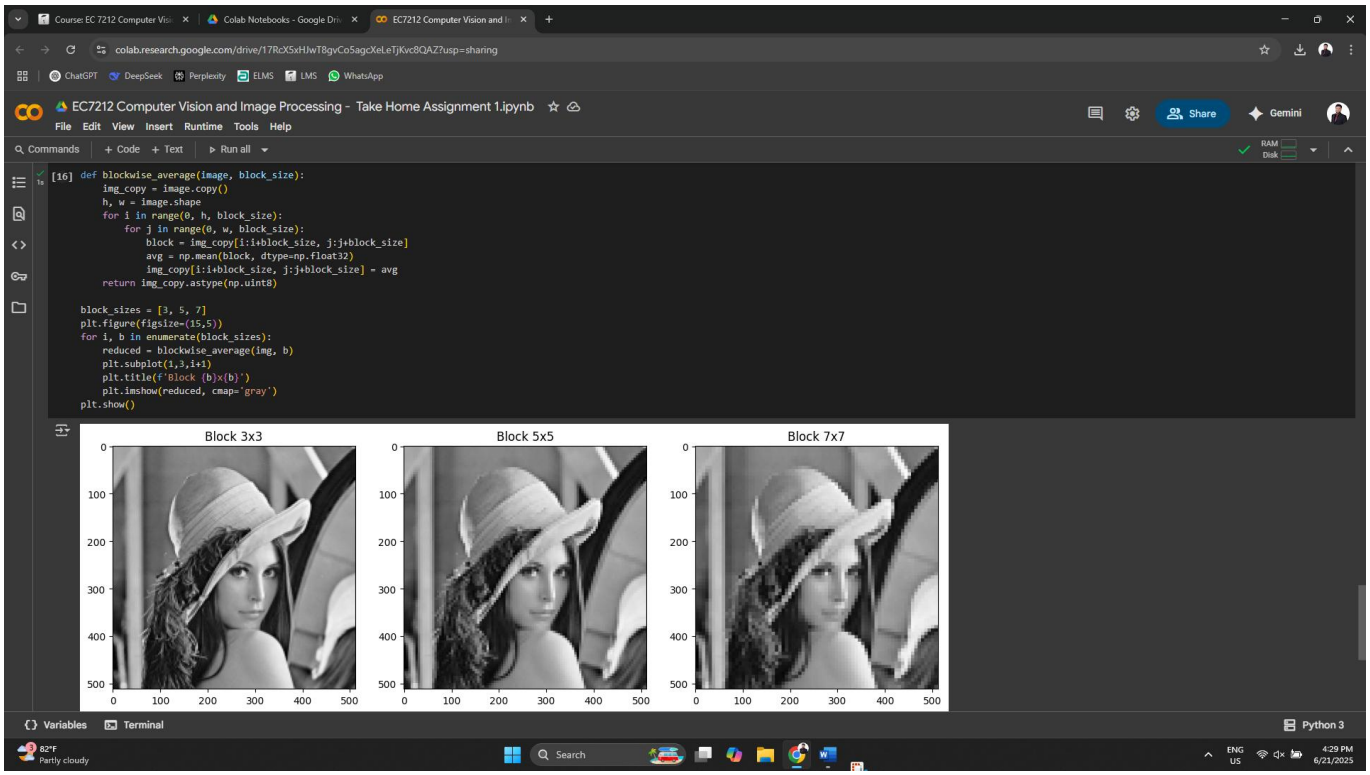

Figure 5.1: Implementation and Results for Task 4

6 Conclusion

This assignment demonstrates key image processing operations: quantization, smoothing, rotation, and resolution reduction. The provided Python implementations and visual outputs confirm the effectiveness of each method in manipulating image characteristics as required by the assignment.

7 Submission

GitHub Link:
https://github.com/MalithPramoditha/EC7212_Computer_Vision_and_Image_Processing_Take_Home_Assignment_1

Google Colab Link:
https://colab.research.google.com/drive/17RcX5xHJwT8gvCo5agcXeLeTjKvc8QAZ?usp=sharing