Report

Bitmap Image Processing

This report provides an overview of a Python script designed for processing bitmap images. The script utilizes the Python Imaging Library (PIL), also known as Pillow, to perform various operations on a bitmap image file. These operations include displaying image dimensions, calculating the total number of pixels, converting the image to grayscale, resizing it, comparing file sizes, and displaying the original, grayscale, and resized images. This script serves as a practical tool for basic image processing tasks.

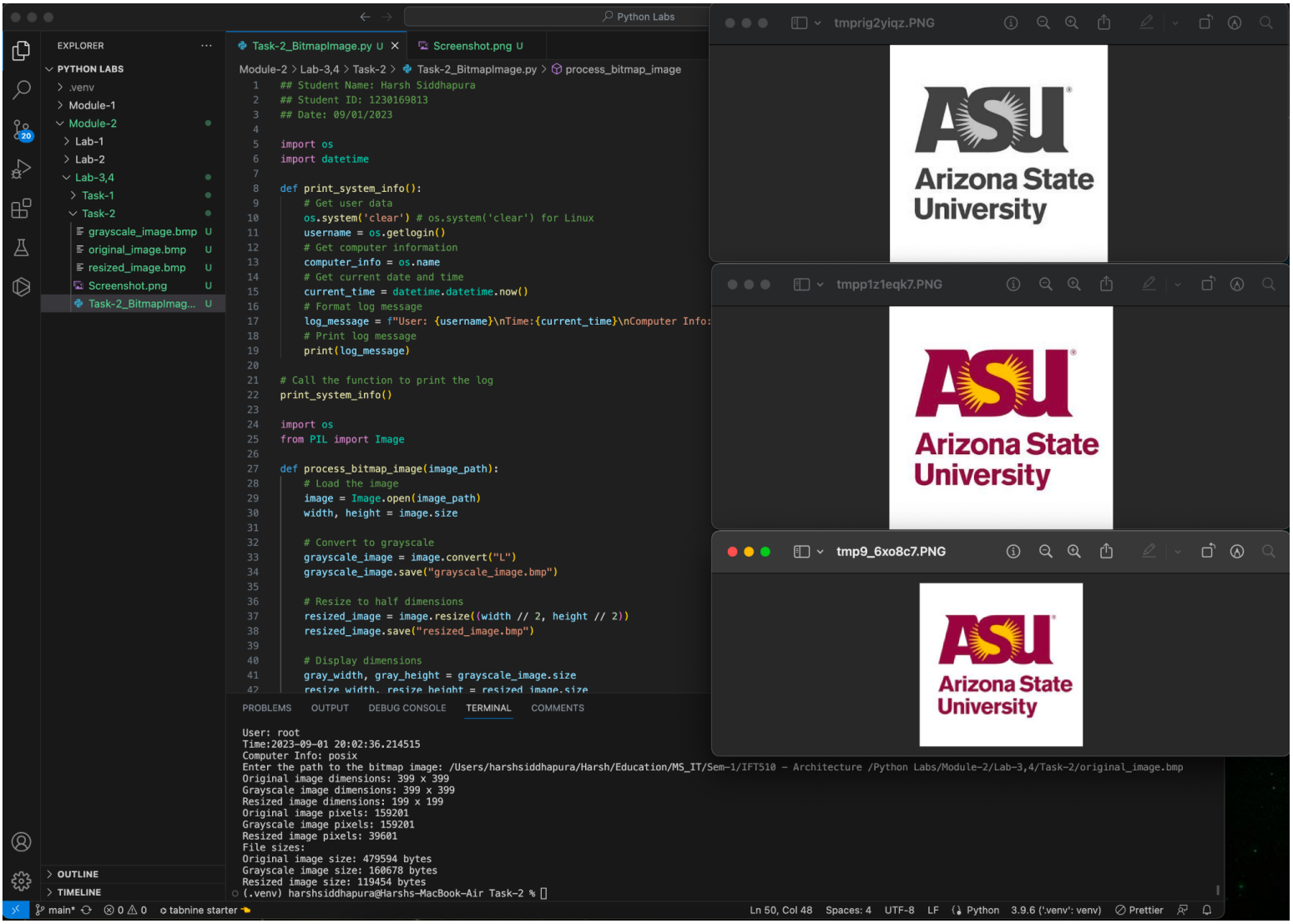
The script prompts the user to enter the path to the bitmap image they want to process. This user-provided input serves as the input for the image processing operations which includes:

- **Loading the Image:** The script uses Pillow to load the specified image from the provided path.
- **Displaying Dimensions:** It calculates and prints the dimensions (width and height) of the image, providing insights into its size.
- **Total Number of Pixels:** By multiplying the width and height, the script calculates and displays the total number of pixels in the image, a critical metric for image analysis.
- **Converting to Grayscale:** The script converts the original image to grayscale using the `.convert("L")` method and saves it as a new image called "grayscale_image.bmp."
- **Resizing the Image:** It resizes the original image to half of its dimensions using the `.resize()` method and saves it as "resized_image.bmp."
- **Comparing File Sizes:** The script retrieves and compares the file sizes of the original, grayscale, and resized images, providing a quantitative assessment of the impact of these operations on file size.
- **Displaying Images:** Finally, the script displays the original, grayscale, and resized images using the `.show()` method. This allows users to visually compare the image transformations.

This script can be useful for various applications, including:

- Image pre-processing for machine learning and computer vision projects.
- Analyzing the dimensions and size of images in a dataset.
- Visualizing the effects of common image operations, such as resizing and conversion to grayscale.

In conclusion, the Python script presented in this report provides a straightforward yet effective means of processing bitmap images. By combining user input, image analysis, file operations, and visualization, it serves as a valuable tool for introductory image processing tasks. Its ease of use and the immediate display of results make it accessible to users looking to perform basic operations on bitmap images.



Original Image

Resized Image

Grayscale Image