

Name: Abubakar Waziri

ID: 4220056

Image Processing Final Lab Exams Report

Report for Task 1: Dynamic Color Modification

1. Summary of Steps

1. Loaded the input image (image1.png) using skimage.
2. Collected user inputs:
 - a. **RGB Adjustments:** R:+50, G:-30, B:0
 - b. **Color Inversion:** Yes
 - c. **Brightness Scaling:** 1.5
3. Adjusted RGB channels while clipping values to ensure they remain in range [0, 1].
4. Applied color inversion (1 - pixel value).
5. Scaled image brightness by multiplying pixel values with 1.5 and clipped them again.
6. Saved the result as task1_output.png.
7. Printed the user inputs and steps using the report() function.

2. Challenges Faced

- Ensured pixel values remained valid after adjustments.
- Correct sequence of operations for inversion and brightness scaling.

3. Use of AI Assistance

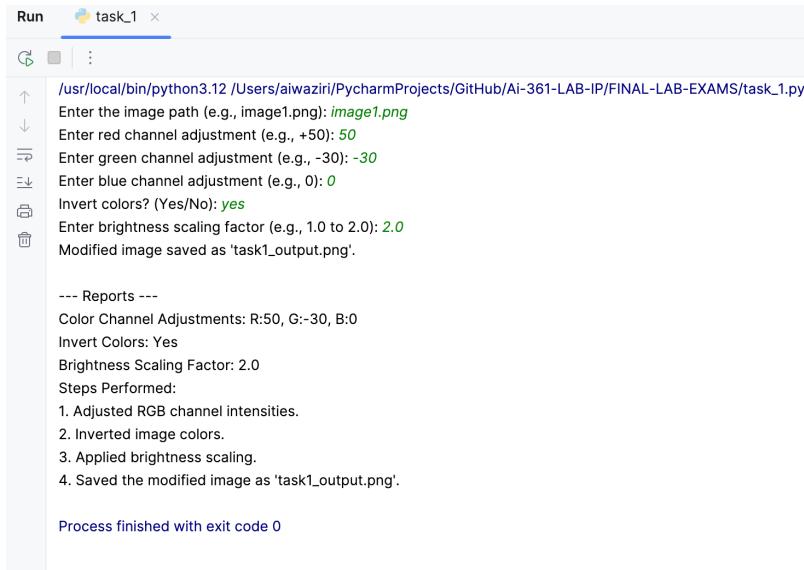
- Debugged RGB adjustments and pixel clipping.
- Verified the correct order of operations.

4. Output

- Final image saved as: task1_output.png.

In summary:

I modified the colors of the input image (image1.png) based on user-defined parameters. I adjusted the RGB channel intensities, applied color inversion if selected, and scaled the image brightness using a user-provided factor. To ensure the image remained valid, I clipped all pixel values to stay within the range [0, 1]. Finally, I saved the modified image as task1_output.png and implemented a report() function to display the user inputs and the steps performed.



The screenshot shows the PyCharm Run window titled 'task_1'. It displays the command run: /usr/local/bin/python3.12 /Users/aiwaziri/PycharmProjects/GitHub/Ai-361-LAB-IP/FINAL-LAB-EXAMS/task_1.py. Below the command, several user inputs are listed: Enter the image path (e.g., image1.png): image1.png; Enter red channel adjustment (e.g., +50): 50; Enter green channel adjustment (e.g., -30): -30; Enter blue channel adjustment (e.g., 0): 0; Invert colors? (Yes/No): yes; Enter brightness scaling factor (e.g., 1.0 to 2.0): 2.0. A message at the bottom states: Modified image saved as 'task1_output.png'. Under the heading '--- Reports ---', it lists: Color Channel Adjustments: R:50, G:-30, B:0; Invert Colors: Yes; Brightness Scaling Factor: 2.0; Steps Performed: 1. Adjusted RGB channel intensities. 2. Inverted image colors. 3. Applied brightness scaling. 4. Saved the modified image as 'task1_output.png'. At the very bottom, a message says: Process finished with exit code 0.

Report for Task 2: Custom Region Enhancement

1. Summary of the steps

In this task, I enhanced a user-defined region of an image by applying either a Gaussian blur or a sharpening filter. The process included extracting the region from the image, applying the selected enhancement, and blending the processed region back into the original image using a transparency factor. The following steps were performed:

- Loaded the image and obtained user inputs (region coordinates, enhancement type, and transparency level).
- Extracted the specified region and applied the enhancement (either blur or sharpen).
- Blended the enhanced region with the original image using the given transparency level.
- Saved the processed image as task2_output.png.
- Generated a report detailing the user inputs and the steps performed.

2. Challenges Faced

- **Region Extraction:** Ensured the correct region was selected and extracted based on the user-defined coordinates.
- **Processing Enhancement:** Handled both multi-channel (RGB) and single-channel (grayscale) images appropriately during enhancement.
- **Blending the Region:** Applied transparency blending effectively to maintain the visual integrity of the original image.

3. Use of AI Assistance

- Used AI to debug the Gaussian blur and sharpening function, ensuring correct handling of multi-channel images.
- Helped with understanding the process of transparency blending and how to apply it efficiently to the image region.

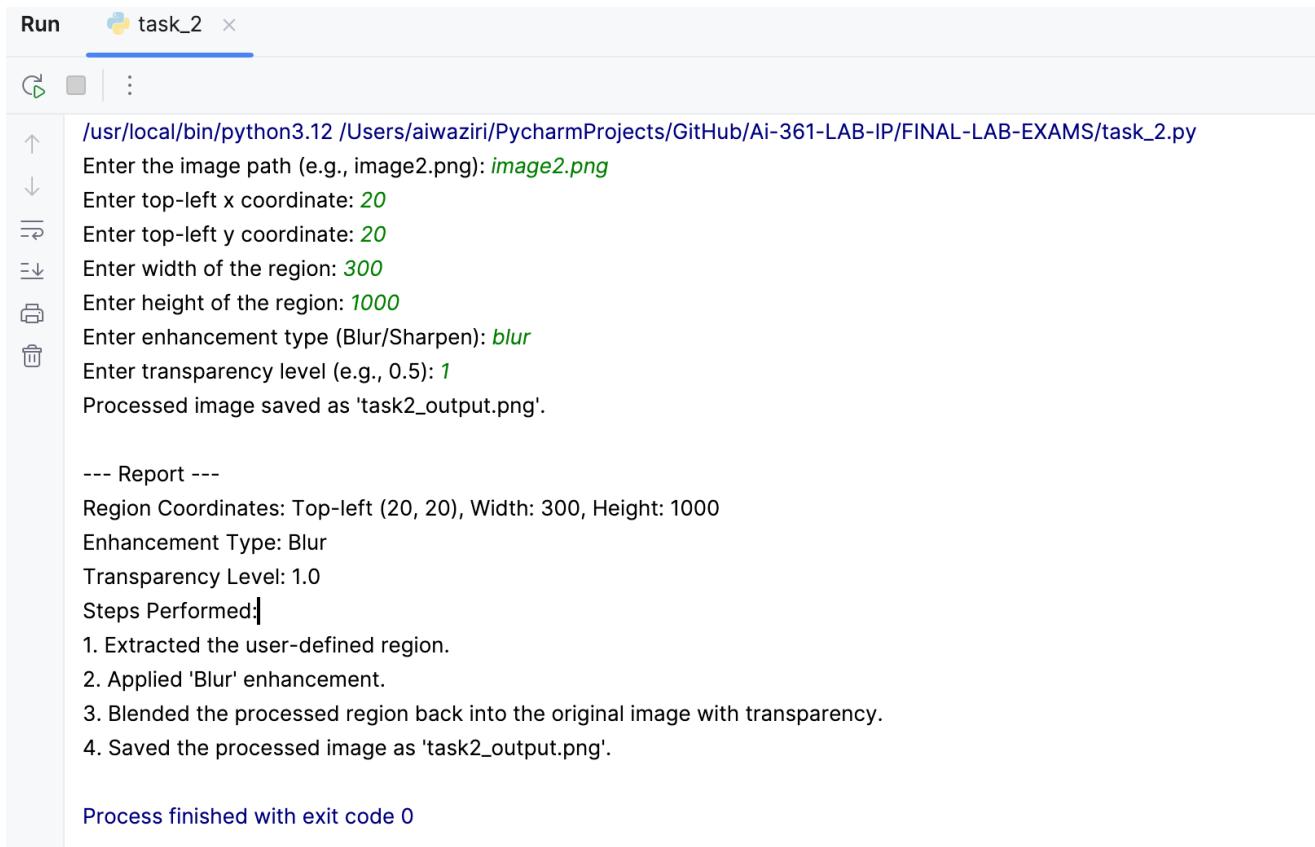
4. Output

- The final processed image was saved as task2_output.png.

In summary:

I developed a program that allows users to enhance a specific region of an image by applying either a Gaussian blur or a sharpening filter. The program first extracts the user-defined rectangular region from the image based on the given coordinates and dimensions. It then applies the selected enhancement, either by blurring the region with a Gaussian filter or sharpening it using a convolution kernel. Afterward, the enhanced region is blended back into the original image using the specified transparency level. Finally, the processed image is saved as task2_output.png, and a report detailing the steps and user inputs is generated.

Blur Example:

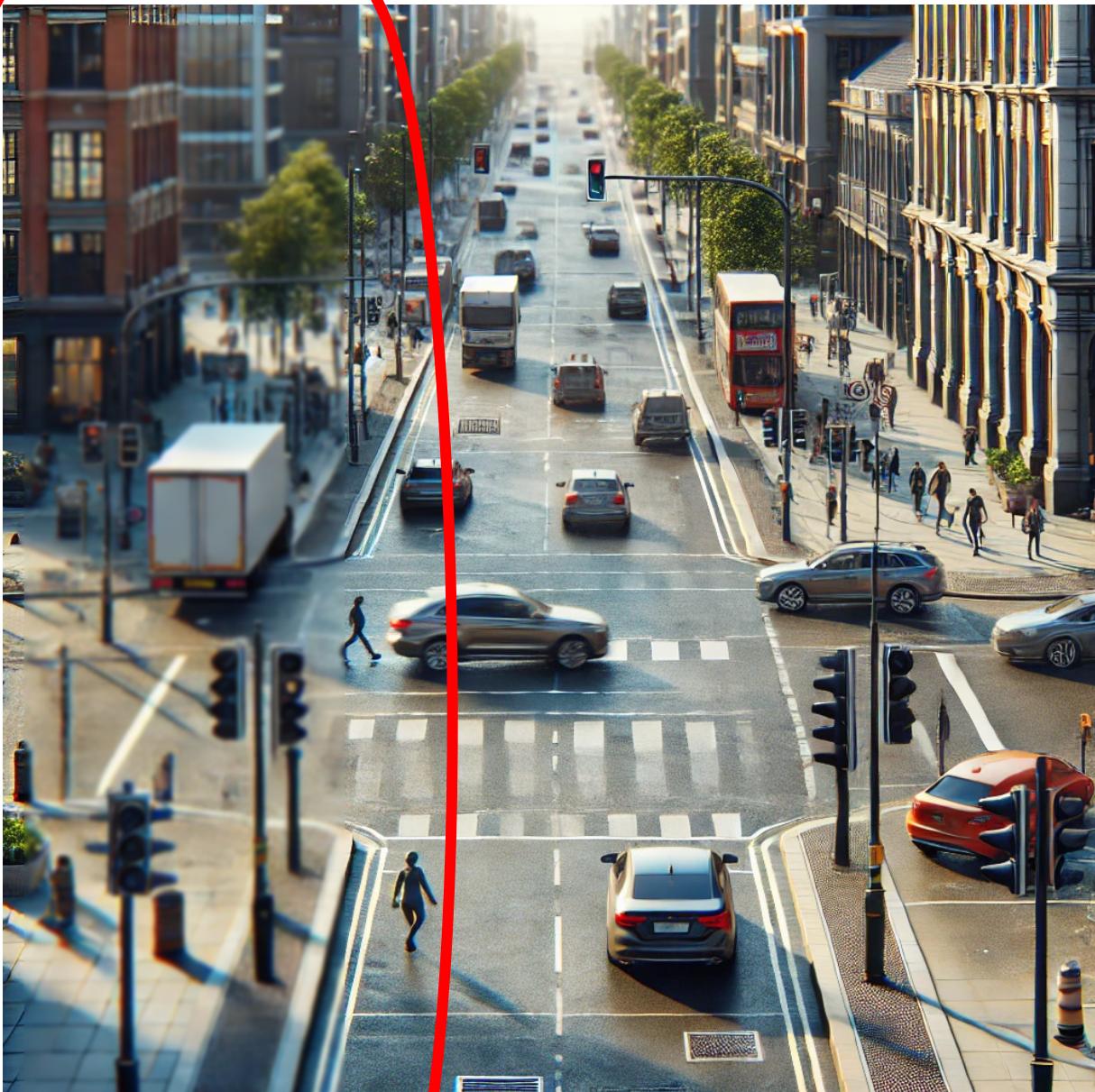


The screenshot shows the PyCharm Run console for a project named 'task_2'. The console output is as follows:

```
Run task_2 ×
Up | Down | ... | Run | Stop | Close
/usr/local/bin/python3.12 /Users/aiwaziri/PycharmProjects/GitHub/Ai-361-LAB-IP/FINAL-LAB-EXAMS/task_2.py
Enter the image path (e.g., image2.png): image2.png
Enter top-left x coordinate: 20
Enter top-left y coordinate: 20
Enter width of the region: 300
Enter height of the region: 1000
Enter enhancement type (Blur/Sharpen): blur
Enter transparency level (e.g., 0.5): 1
Processed image saved as 'task2_output.png'.

--- Report ---
Region Coordinates: Top-left (20, 20), Width: 300, Height: 1000
Enhancement Type: Blur
Transparency Level: 1.0
Steps Performed:
1. Extracted the user-defined region.
2. Applied 'Blur' enhancement.
3. Blended the processed region back into the original image with transparency.
4. Saved the processed image as 'task2_output.png'.

Process finished with exit code 0
```



Sharpening Example:

```
Run task_2 ×
↻ ⌂ : 

/usr/local/bin/python3.12 /Users/aiwaziri/PycharmProjects/GitHub/Ai-361-LAB-IP/FINAL-LAB-EXAMS/task_2.py
Enter the image path (e.g., image2.png): image2.png
Enter top-left x coordinate: 20
Enter top-left y coordinate: 20
Enter width of the region: 300
Enter height of the region: 1000
Enter enhancement type (Blur/Sharpen): sharpen
Enter transparency level (e.g., 0.5): 1
Processed image saved as 'task2_output.png'.

--- Report ---
Region Coordinates: Top-left (20, 20), Width: 300, Height: 1000
Enhancement Type: Sharpen
Transparency Level: 1.0
Steps Performed:
1. Extracted the user-defined region.
2. Applied 'Sharpen' enhancement.
3. Blended the processed region back into the original image with transparency.
4. Saved the processed image as 'task2_output.png'.

Process finished with exit code 0
```

