

1. Toy problem

Max error is: 0.0009477229279480914

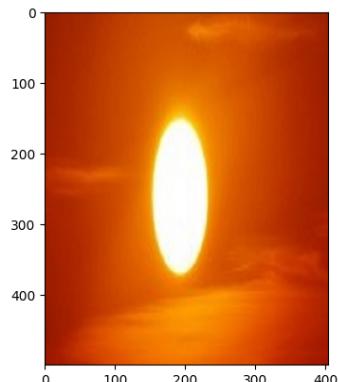
2. Poisson blending

Favorite Blending Result:

- Background Image



- Object Image



- Pasted image with source pixels directly copied onto target background region

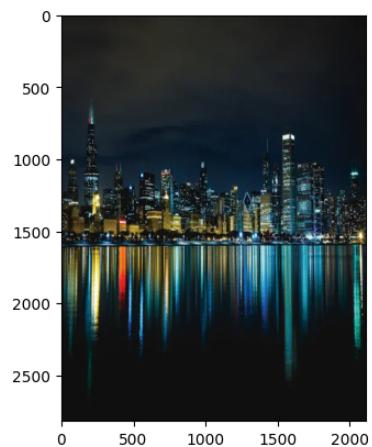


- Final blend result

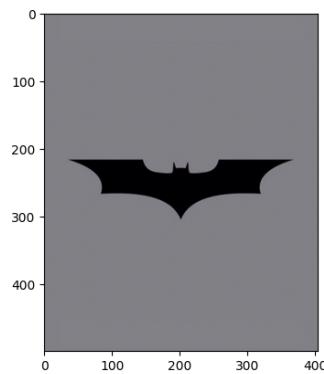


One More Good Result:

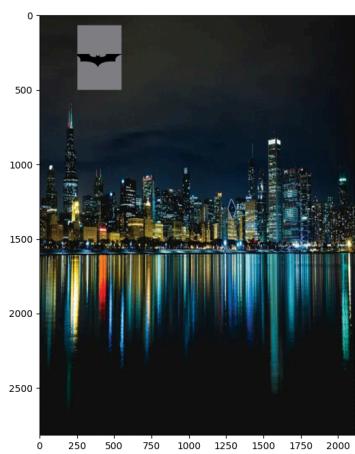
- Background Image



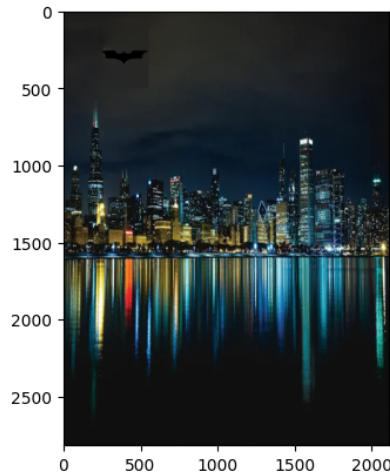
- Object Image



- Pasted image with source pixels directly copied onto target background region

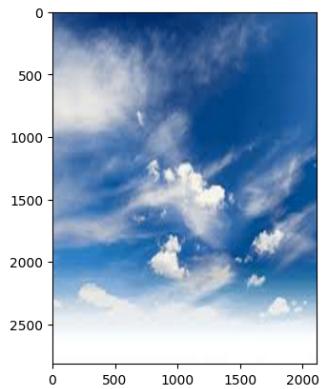


- Final blend result

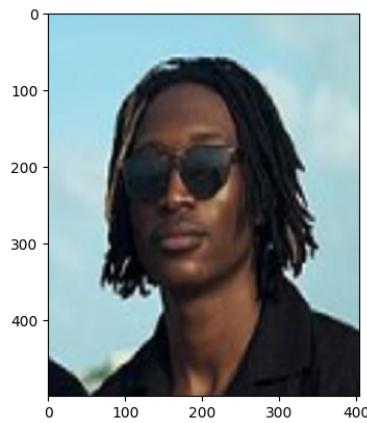


One failure case, where the result is bad:

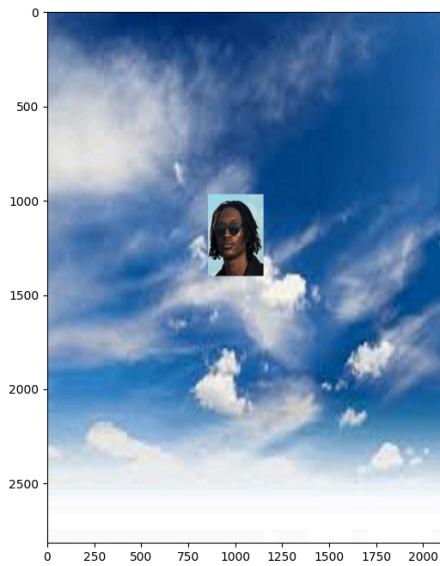
- Background Image



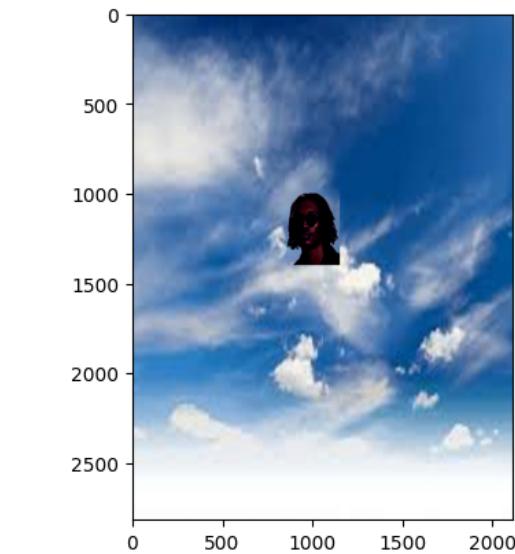
- Object Image



- Pasted image with source pixels directly copied onto target background region



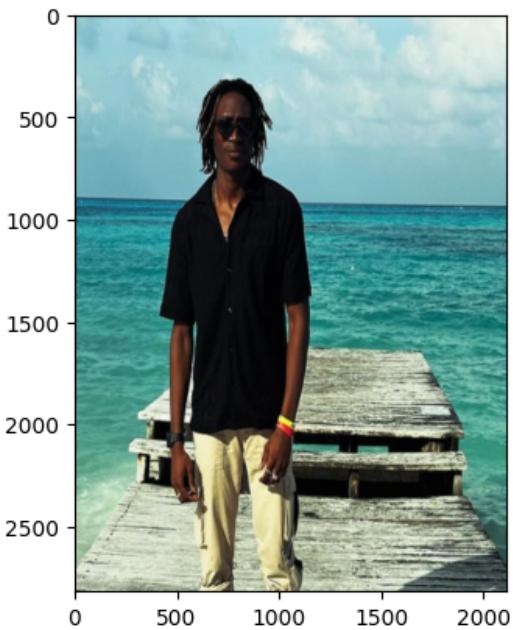
- Final blend result



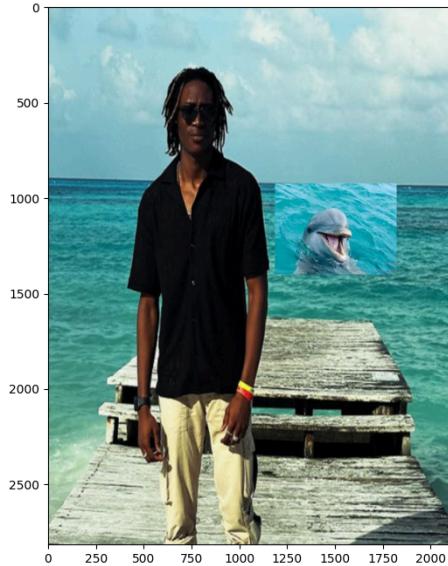
- Why it doesn't work: I think the reason it didn't work is because of a mismatch in lighting and contrast between the object and background image. Especially because the object image has many different color tones ranging between very light and dark. This makes it hard for the blending to be seamless. This can be seen in the discoloration and darkening of the face and rest of the details in the final object image.

3. Mixed gradients

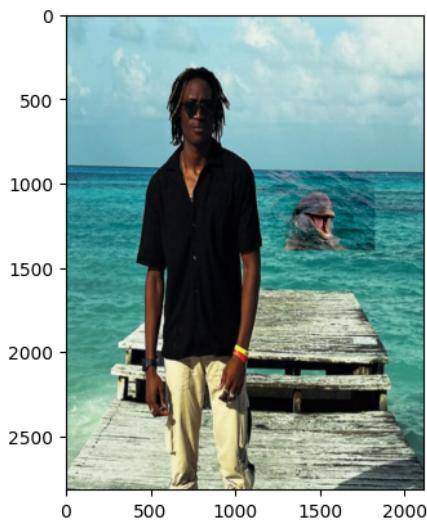
- **Background and object images**



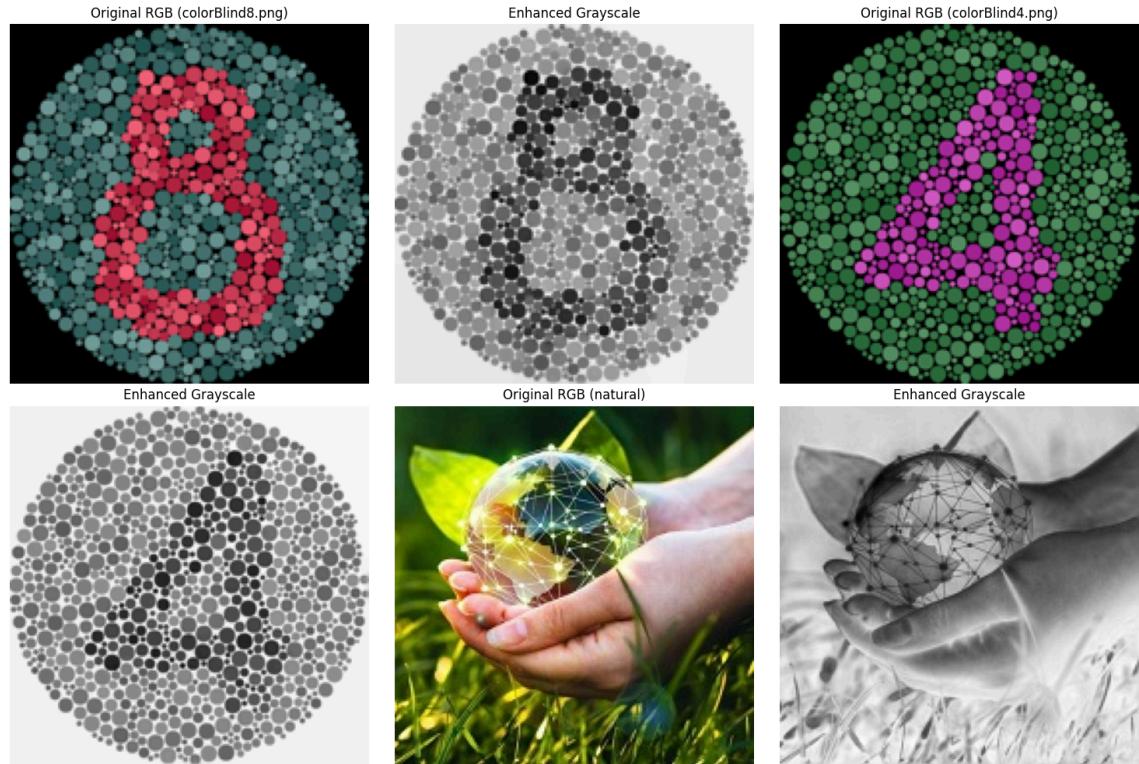
- Pasted image with source pixels directly copied onto target background region



- **Final blend result**



4. Color2Gray (B&W)



- **Method/constraints**
- My main goal with the color2gray function was to convert an RGB picture to grayscale while keeping edge features using gradient constraints. I started by calculating the gradients across all color channels for the Image and then choosing the maximum gradient to capture the most important edges. I then implemented a parse linear system to use with the gradients to create the grayscale image with enhanced contrast. This method required a lot of computation but I thought it was the best way because it helped to keep the minor details of the image in the grayscale conversion.

Acknowledgments / Attribution

Pexels Red Sun Image

"Red Sun." Pexels, <https://www.pexels.com/search/red%20sun/>.

PNGKey Transparent Image

"Chicago Transparent Background PNG." PNGKey,
<https://www.pngkey.com/maxpic/u2a9o0i1u2t4u2i1/>.

iCanvas Chicago Nighttime Skyline

"Chicago Nighttime Skyline Reflections." iCanvas,

<https://www.icanvas.com/canvas-print/chicago-nighttime-skyline-reflections-agp261#1PC6-40x26>.

Pinterest Image

Pinterest. "Image of Chicago." <https://www.pinterest.com/pin/210472982571749993/>.

National Geographic Dolphin Facts

"Dolphins." National Geographic,

<https://www.nationalgeographic.com/animals/mammals/facts/dolphins>.

Gradient Domain Processing

evin, A., Zomet, A., Peleg, S., & Weiss, Y. (2004). "Seamless Image Stitching in the Gradient Domain." Proceedings of the European Conference on Computer Vision (ECCV)

Poisson Image Editing

Pérez, P., Gangnet, M., & Blake, A. (2003). "Poisson Image Editing." ACM Transactions on Graphics (TOG), 22(3), 313-318.

Mixed Gradients and Image Fusion

Agrawal, M., Raskar, R., & Chellappa, R. (2007). "Gradient Domain Manipulation Techniques in Vision and Graphics." IEEE Signal Processing Magazine, 24(1), 68-77.

Color-to-Grayscale Conversion with Gradient Preservation

Gooch, B., & Gooch, A. (2004). "Non-Photorealistic Rendering." *AK Peters/CRC Press*.