Name (netid): Minghao Shi (mshi18)
CS 445 - Project 2: Image Quilting

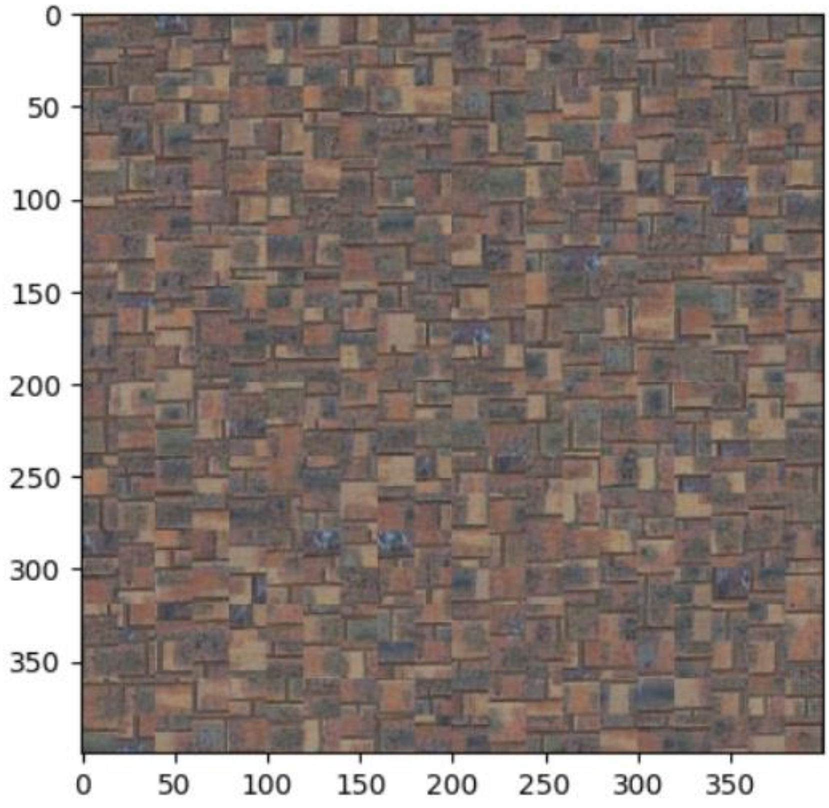
Complete the claimed points and sections below.

Total Points Claimed	**[115] / 175**
Core	
1. Randomly Sampled Texture	[10] / 10
2. Overlapping Patches	[20] / 20
3. Seam Finding	[20]/ 20
4. Additional Quilting Results	[10] / 10
5. Texture Transfer	[30] / 30
6. Quality of results / report	[10] / 10
B&W	
7. Iterative Texture Transfer	[10] / 15
8. Face-in-Toast Image	[5] / 20
9. Hole filling w/ priority function	[ ] / 40

1. Randomly Sampled Texture

Include

- Sample and output images
- Parameters: patch size, output size

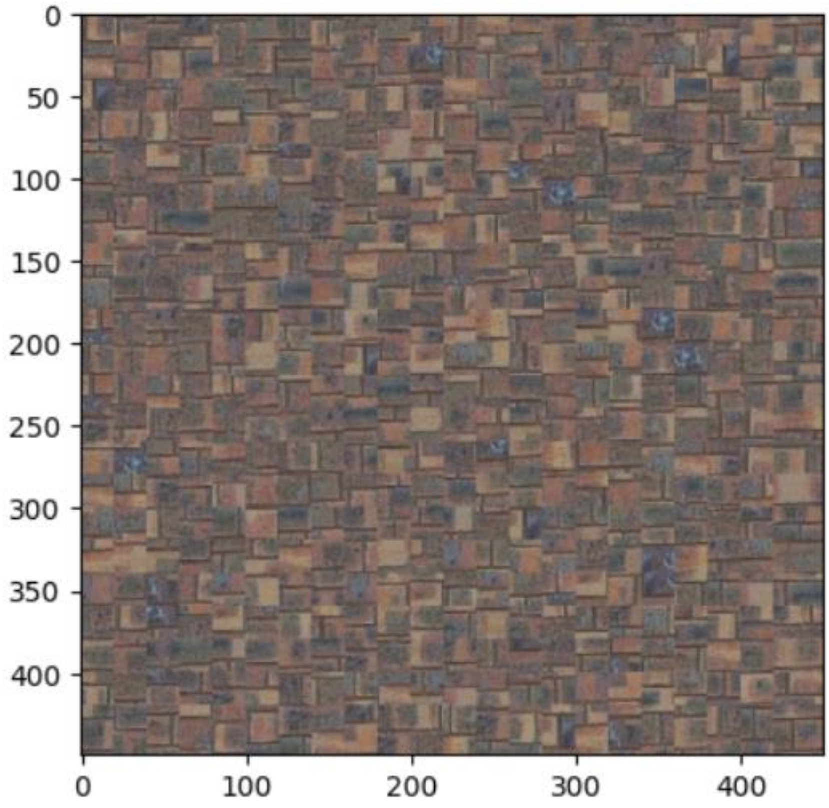


```
out_size = 400
patch_size = 20
```

2. Overlapping Patches

Include

- Output image for same sample as part 1
- Parameters: patch size, overlap size, tolerance

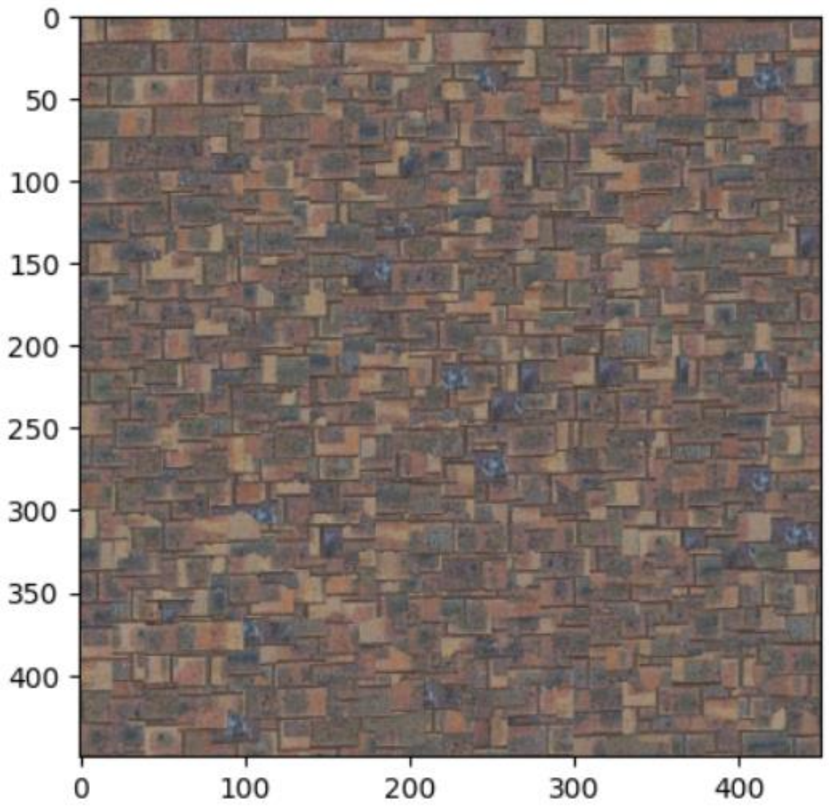


```
out_size = 450
patch_size = 30
overlap = 10
tol = 10
```

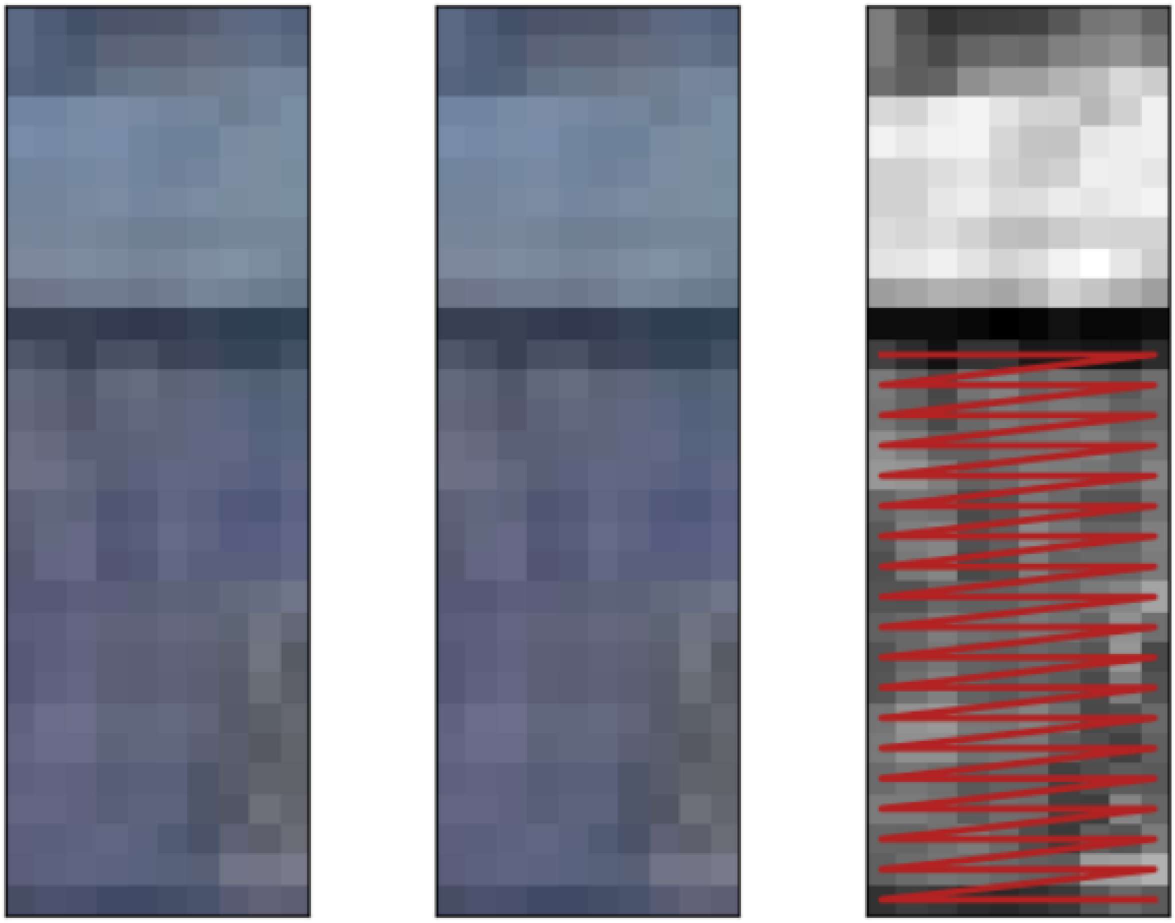
3. Seam Finding

Include

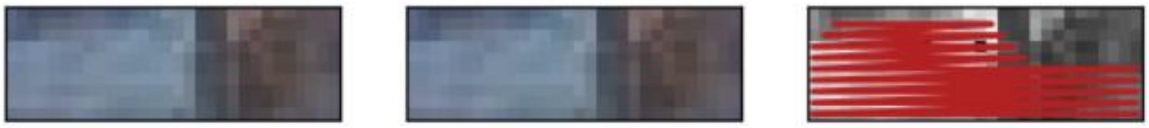
- Output image for same sample as part 1
- Illustration: for a selected patch, display (a) the two overlapping portions; (b) pixelwise SSD cost; (c) horizontal mask; (d) vertical mask; (e) combination mask. The mask is binary and tells which pixels come from which patch.
    - Note: we'll accept anything that looks like a genuine attempt to meet illustration instructions. (a) was intended to mean the two RGB patches (template and selected) that are being cut; (b) can be the SSD values of all the overlapping pixels (i.e. per-pixel SSD masked by template mask), or either one of the SSDs that you feed into cut.



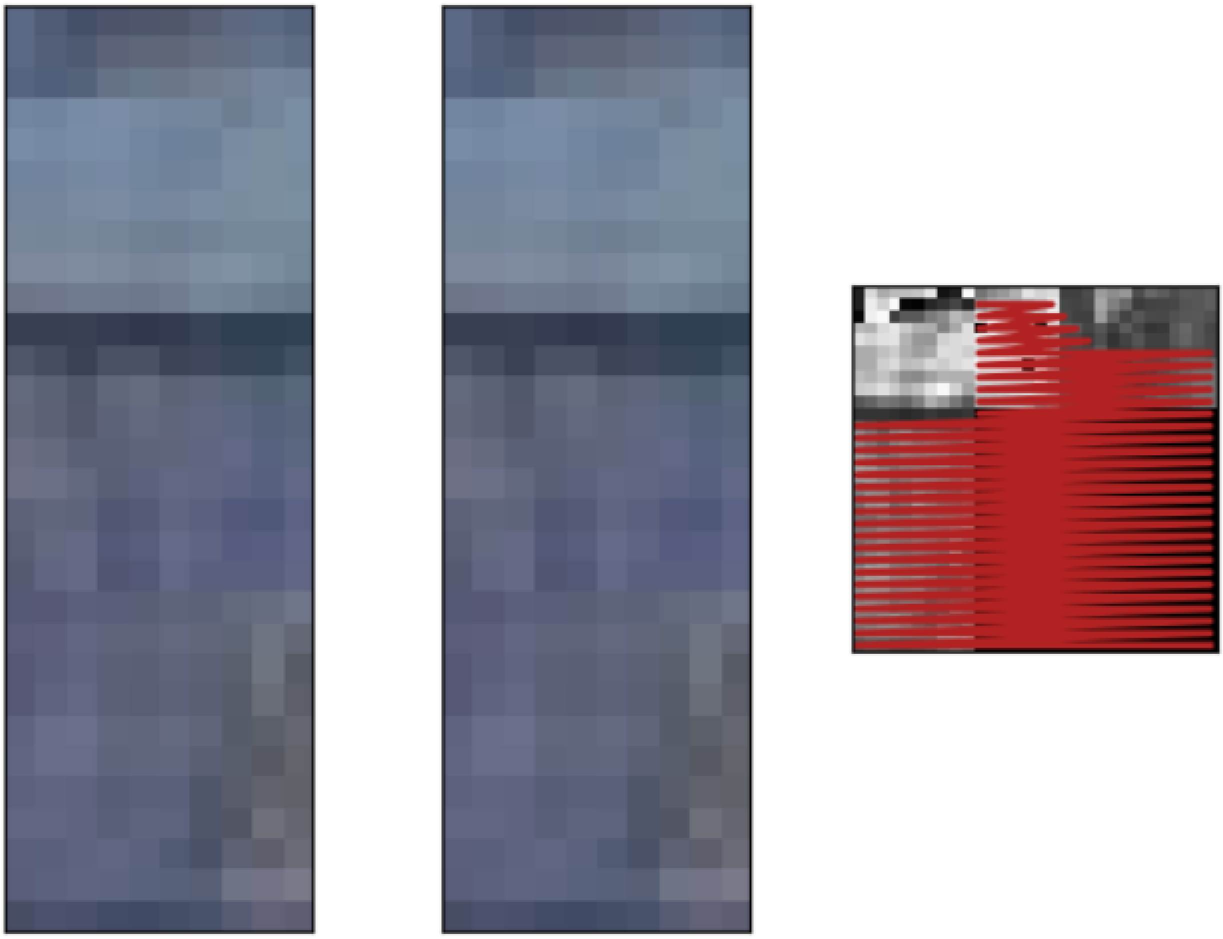
The Left Mask, overlapping, and SSD cost:



The Top Mask, overlapping, and SSD cost:



The Combine Mask, overlapping, and SSD cost:



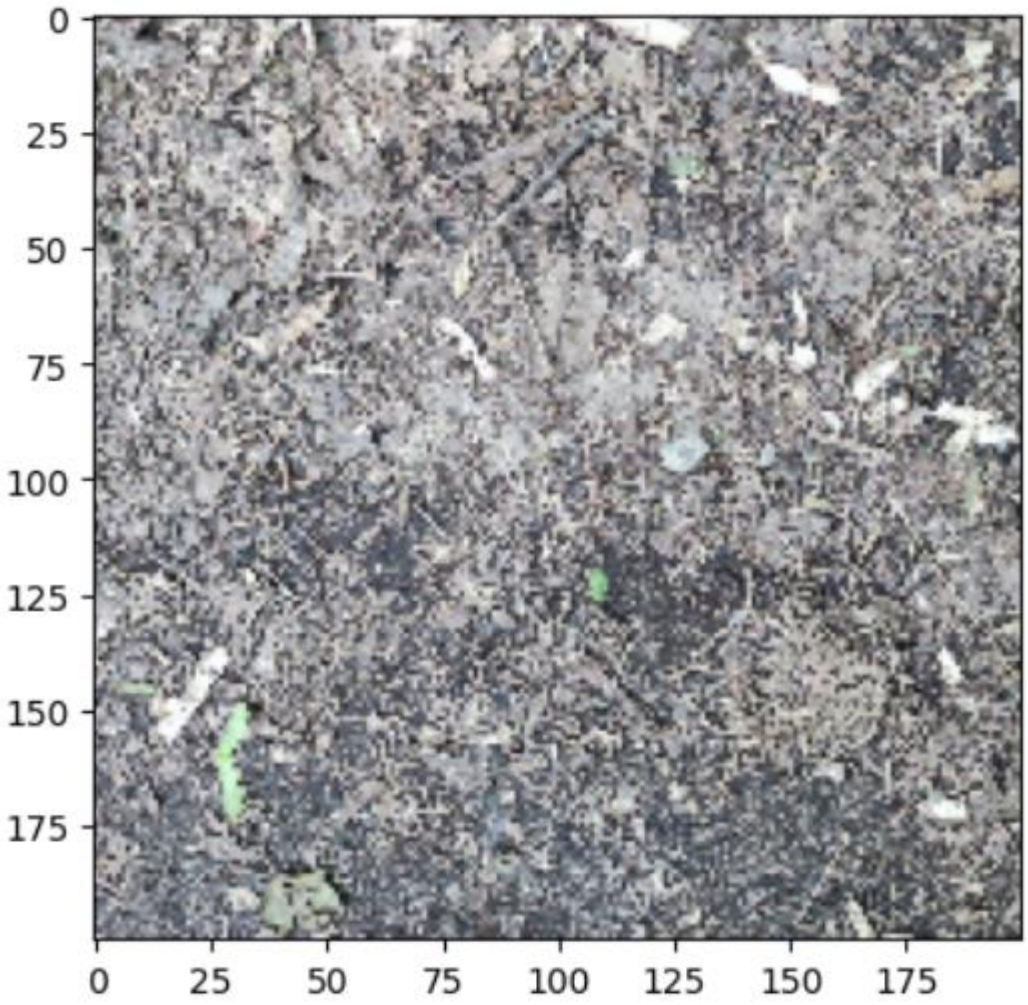
4. Additional Quilting Results

Include

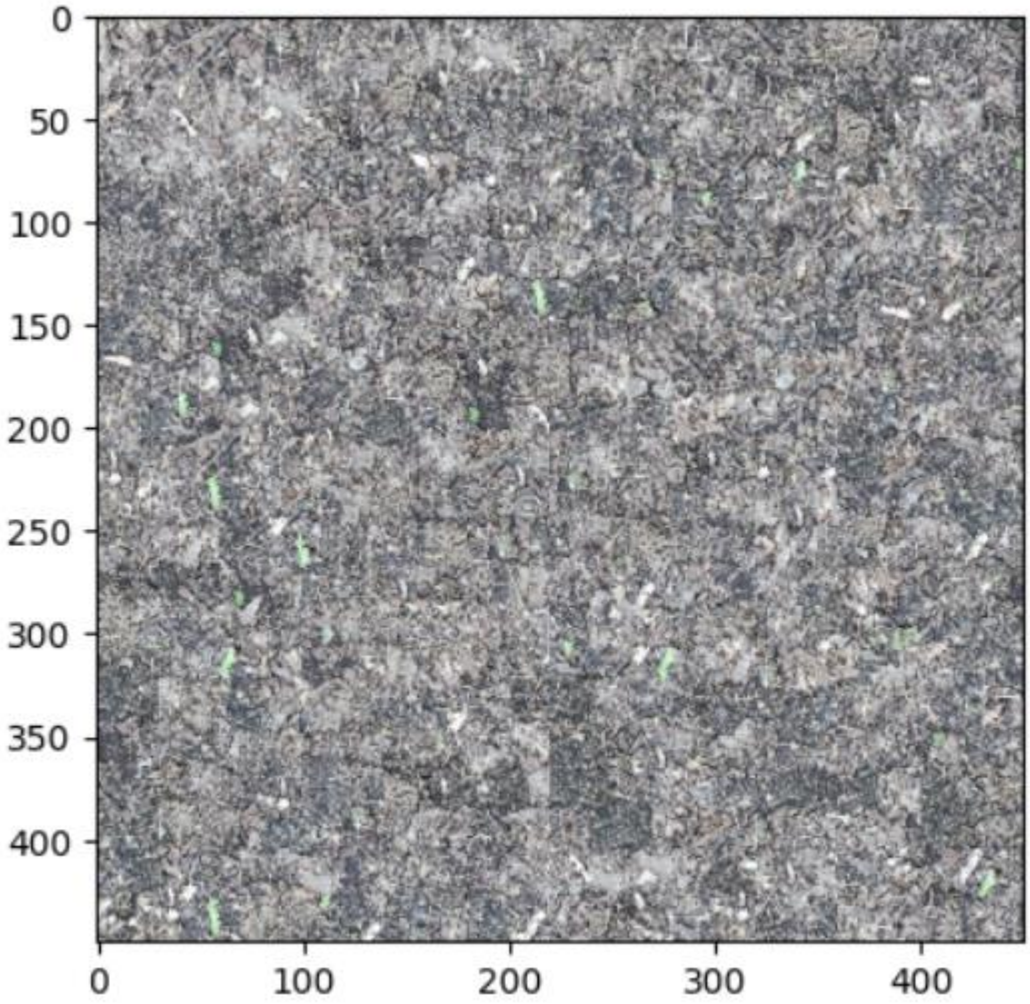
- At least two quilting results on your own images (excluding provided samples). Each result should show input texture image and output, and output should be more pixels than input.

First Image:

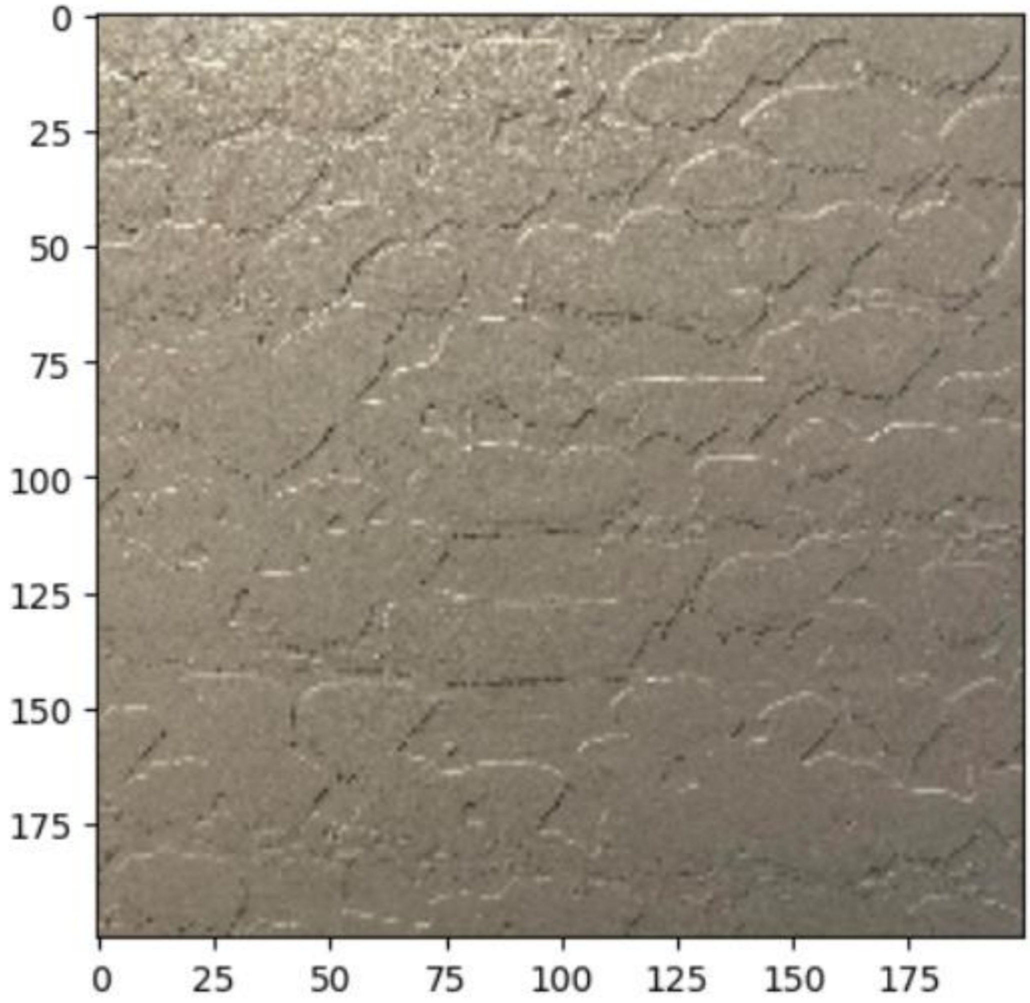
Input:



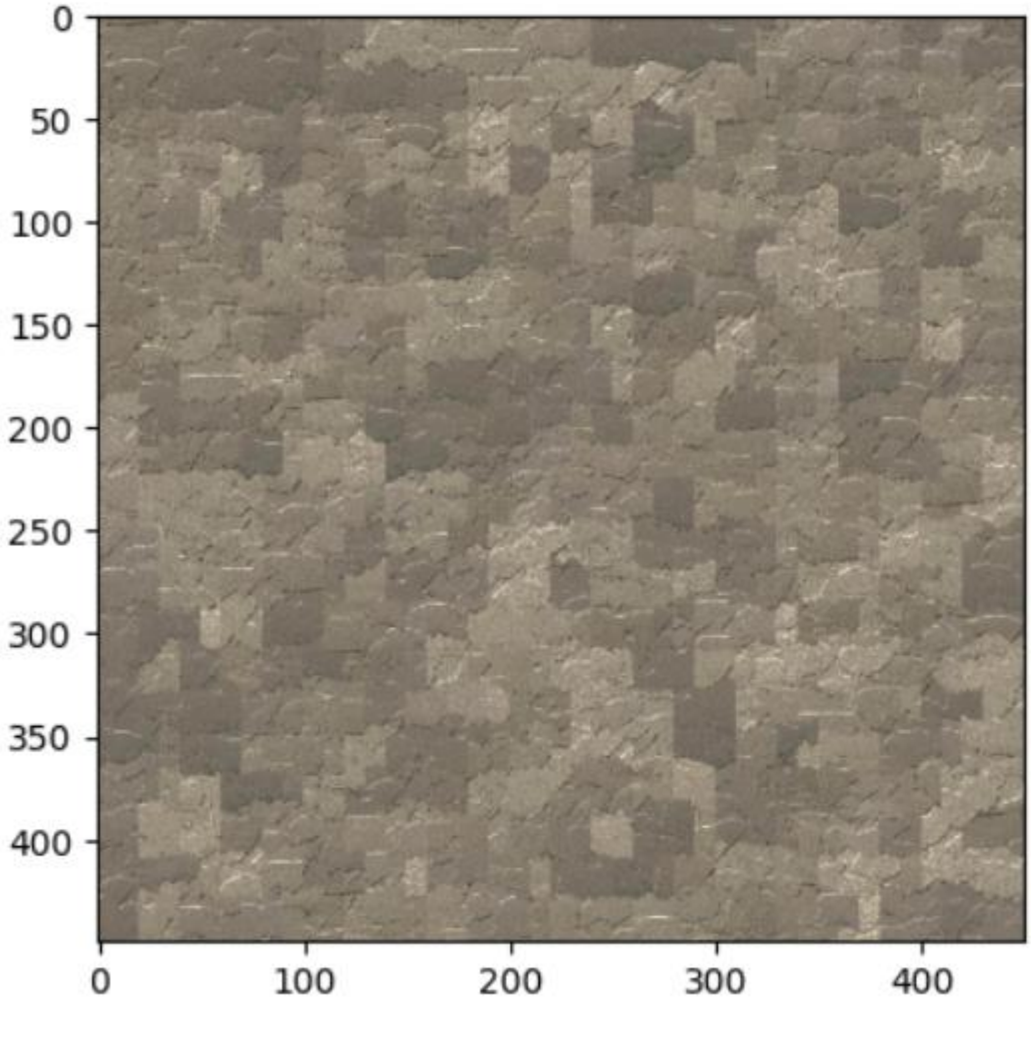
Output:



Second Image:
Input:


0
25
50
75
100
125
150
175
0
25
50
75
100
125
150
175

Output:


0
50
100
150
200
250
300
350
400
0
100
200
300
400

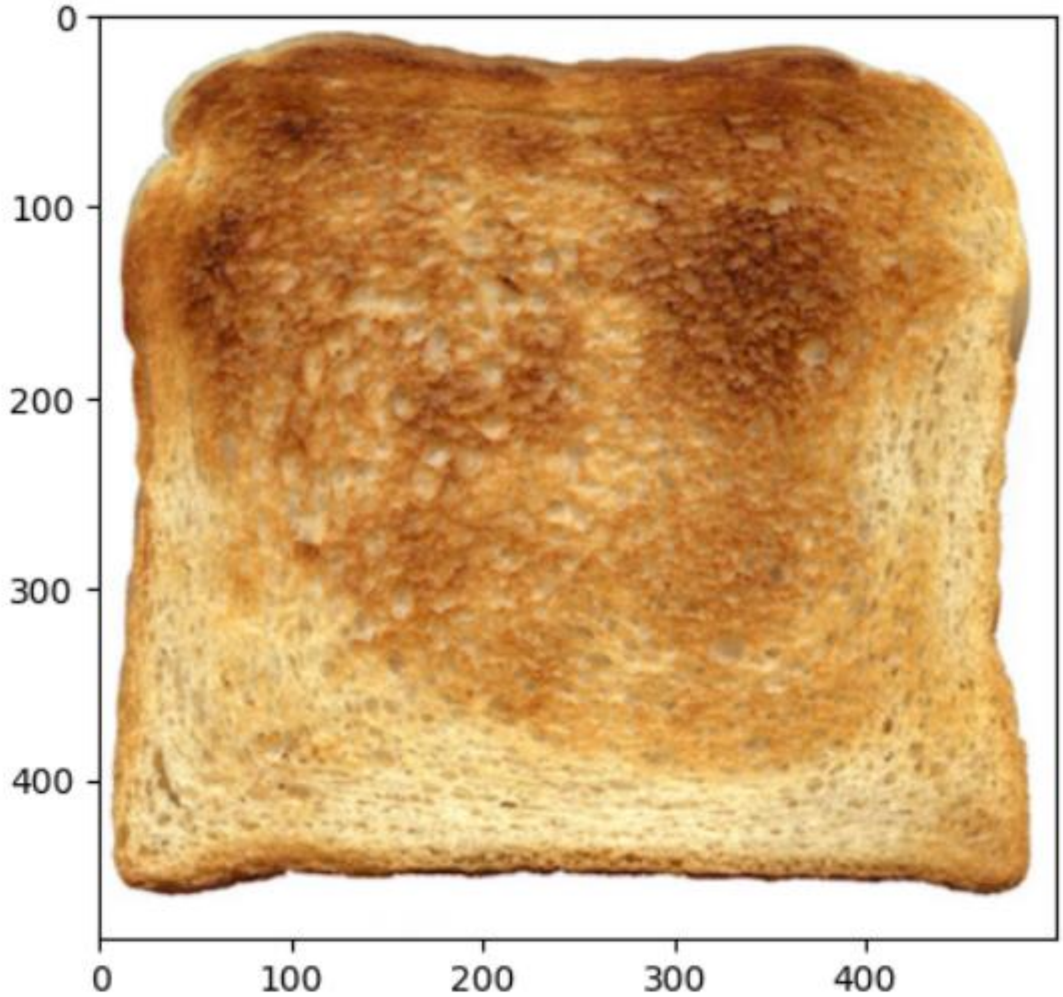
5. Texture Transfer

Include

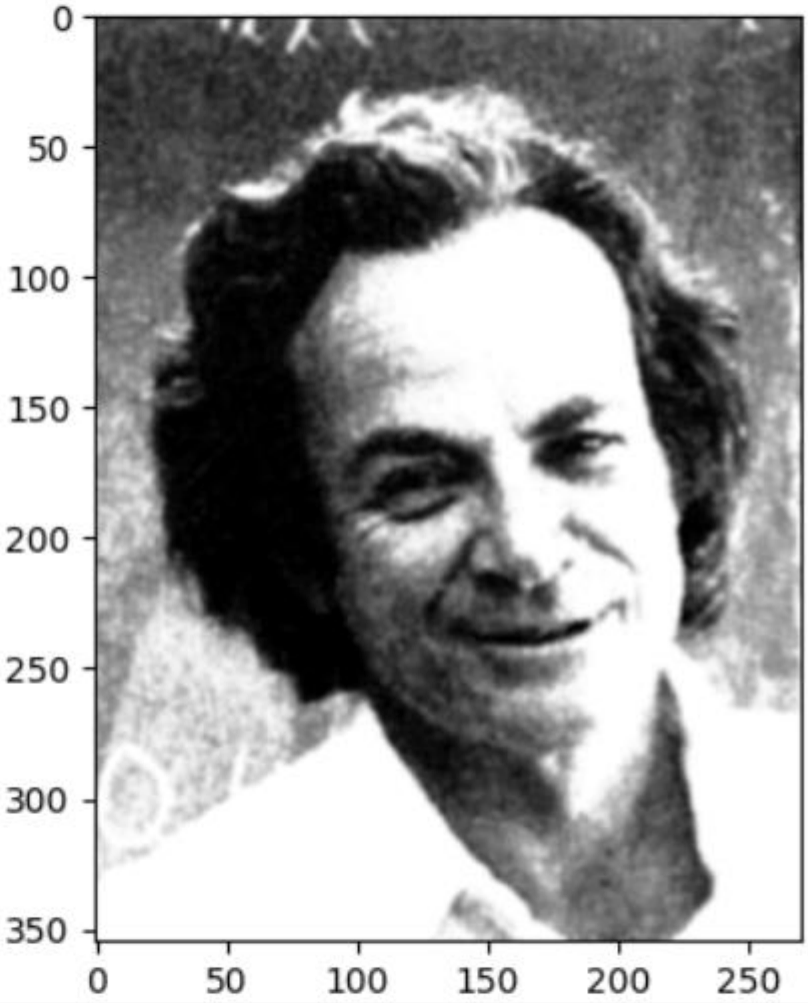
- Brief description of texture transfer method and parameters
- At least two texture transfer results (one result can use provided samples). Include the input texture and target images and the output (output should be the same size as target image)

The texture function is implemented to transfer the given texture to the target image, maintaining the characteristic of the texture, but also maintaining the appearance of the target image. The “sample” is the texture image we input. The “guidance_img” is the target image we input to the function. “patch_size” is the size of the patches we used. “overlap” is the size of the overlapping area between the new patches and the placed already on the output image. “alpha” is the weight that controls the balance between the target and sample image. Closer to 1 will give us more about the original texture and vice versa. “tol” is the maximum error tolerance when we select the samples.

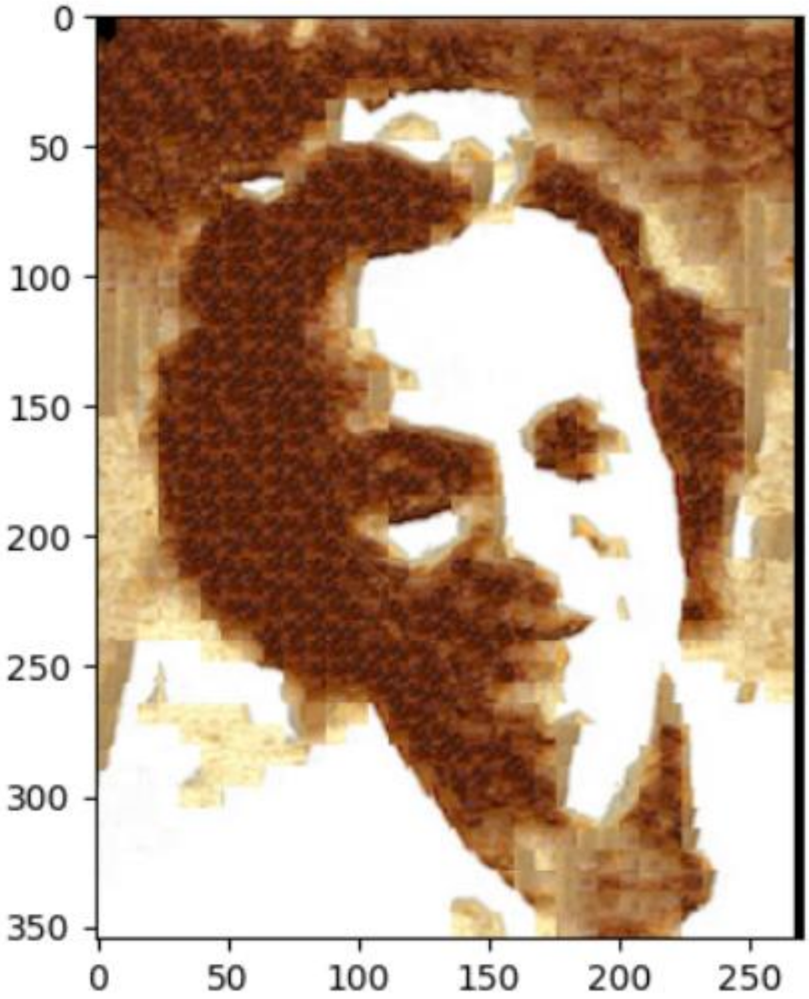
The first result:
Input texture:



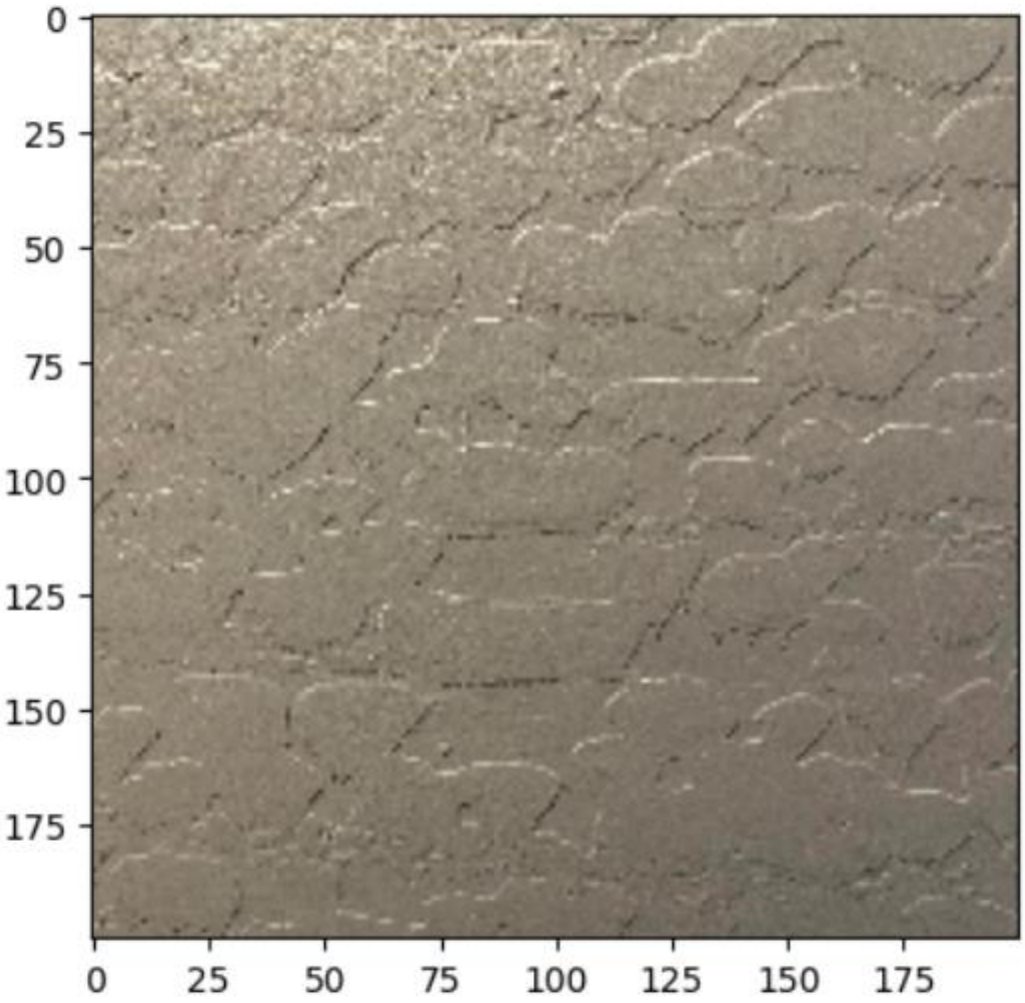
Target Image:



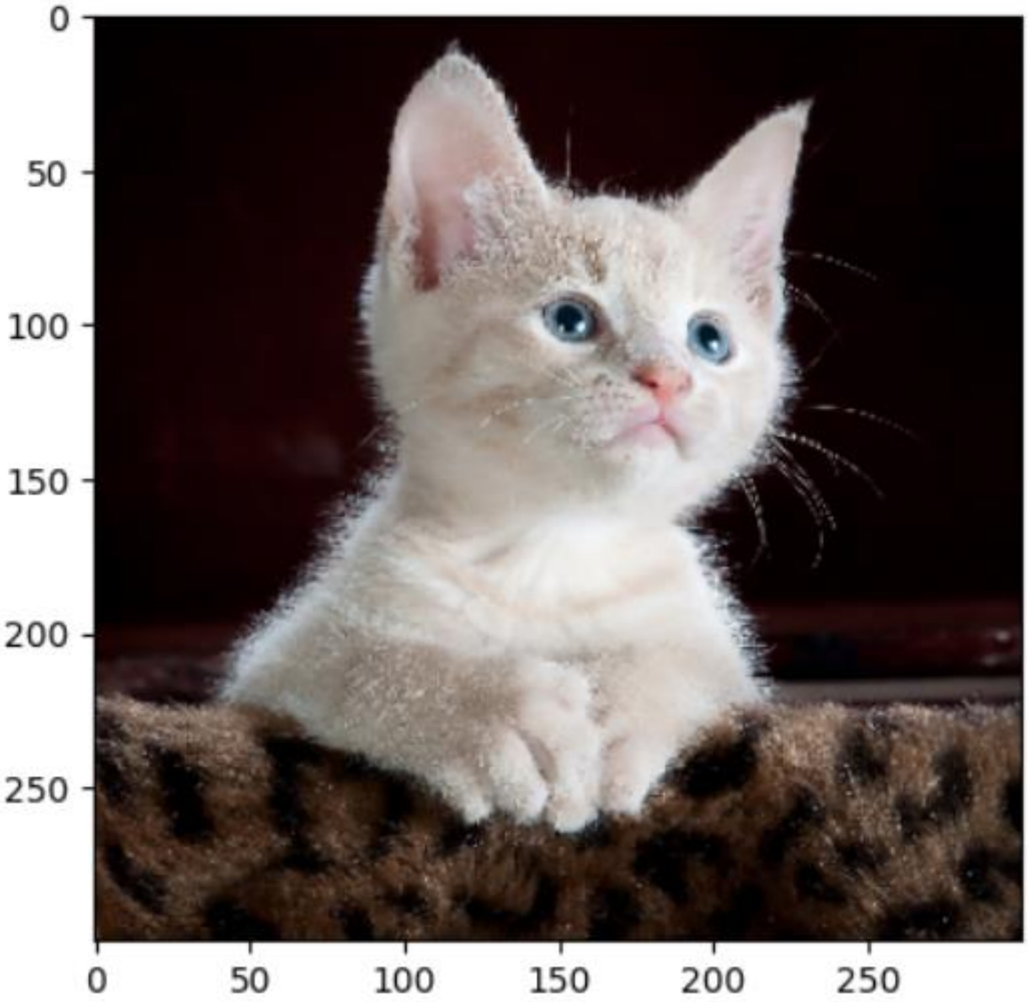
Output Image:



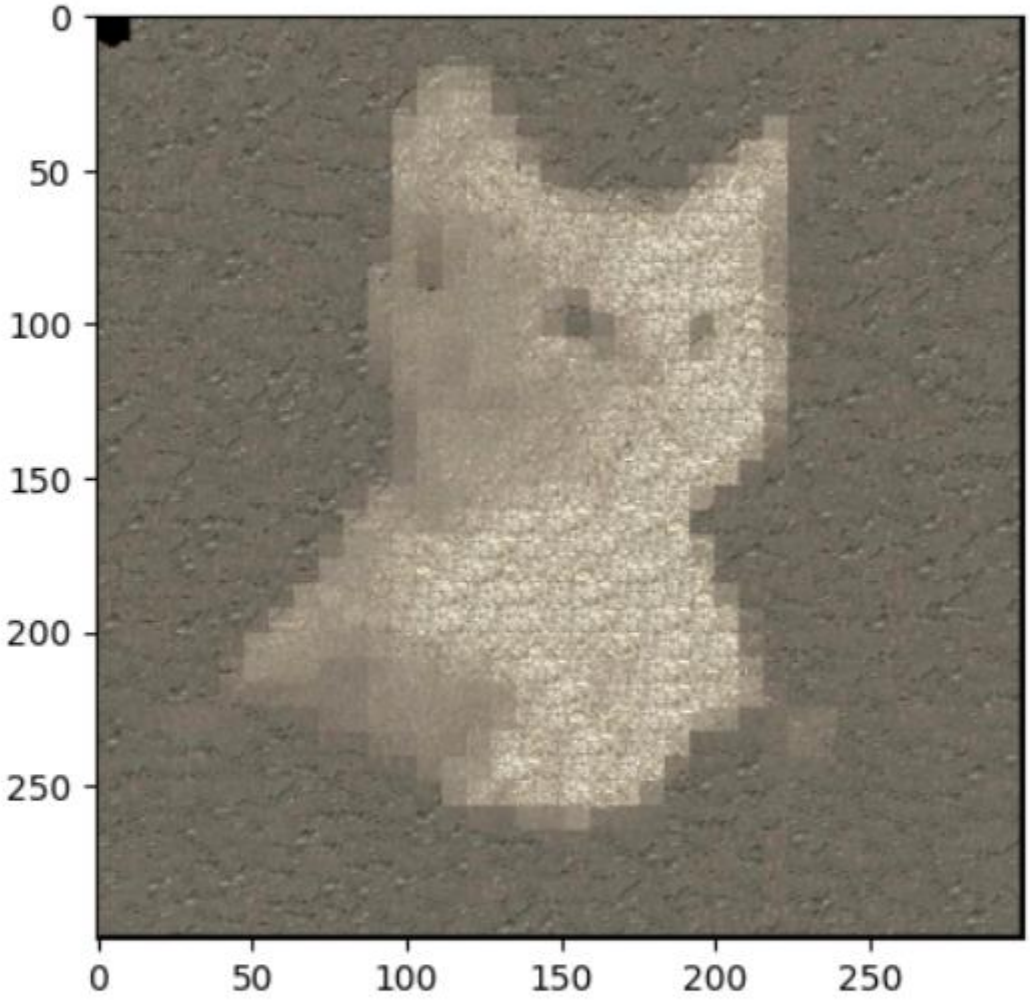
The Second result:
Input texture:



The Target Image:



The Output Image:



6. Quality of results/report

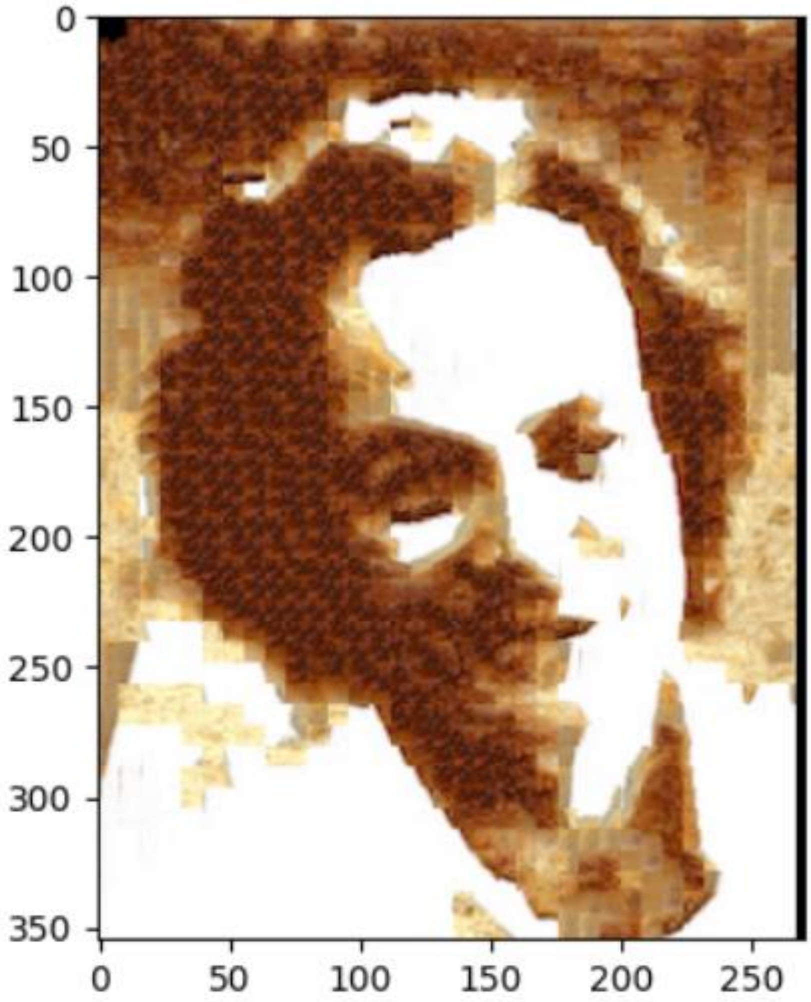
Nothing extra to include (scoring: 0=poor 5=average 10=great).

7. Iterative Texture Transfer (B&W)

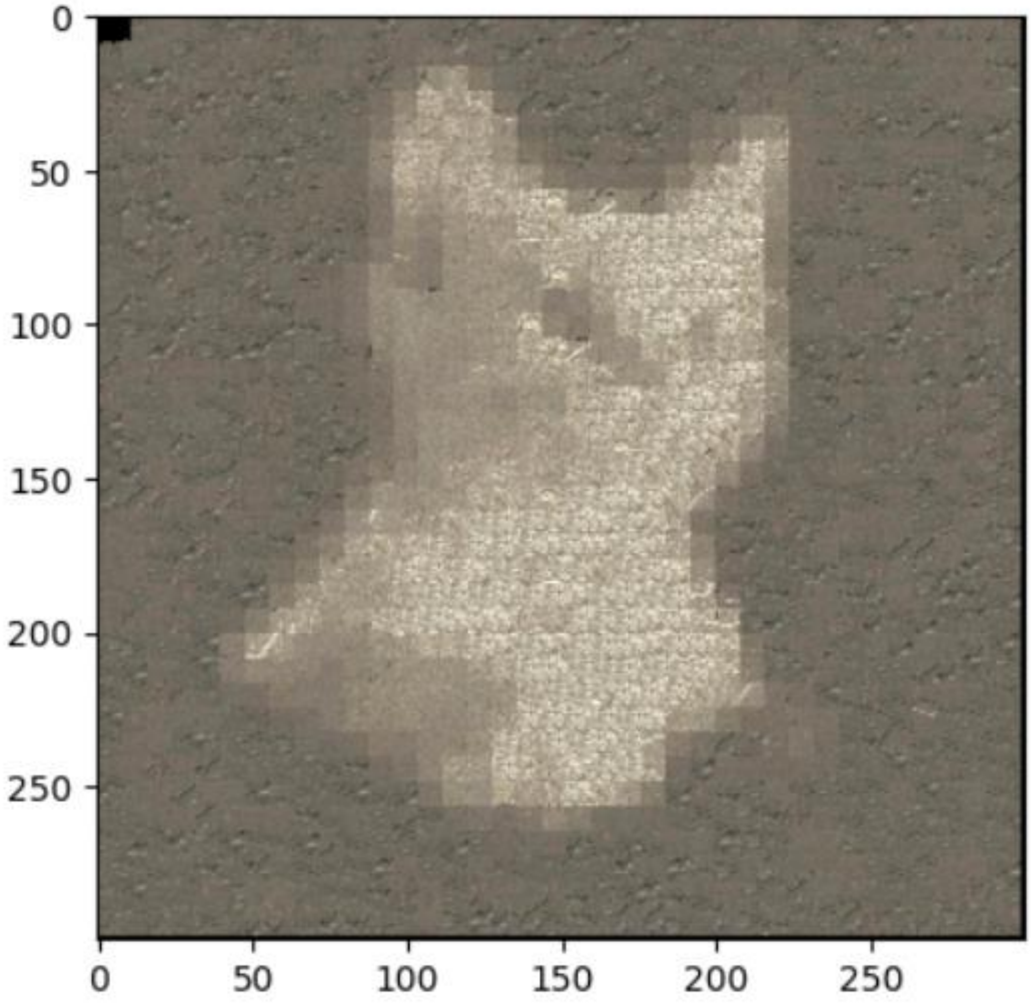
Include

- Describe method
- Results on same images as shown for texture transfer.

The first output:



The Second Output:



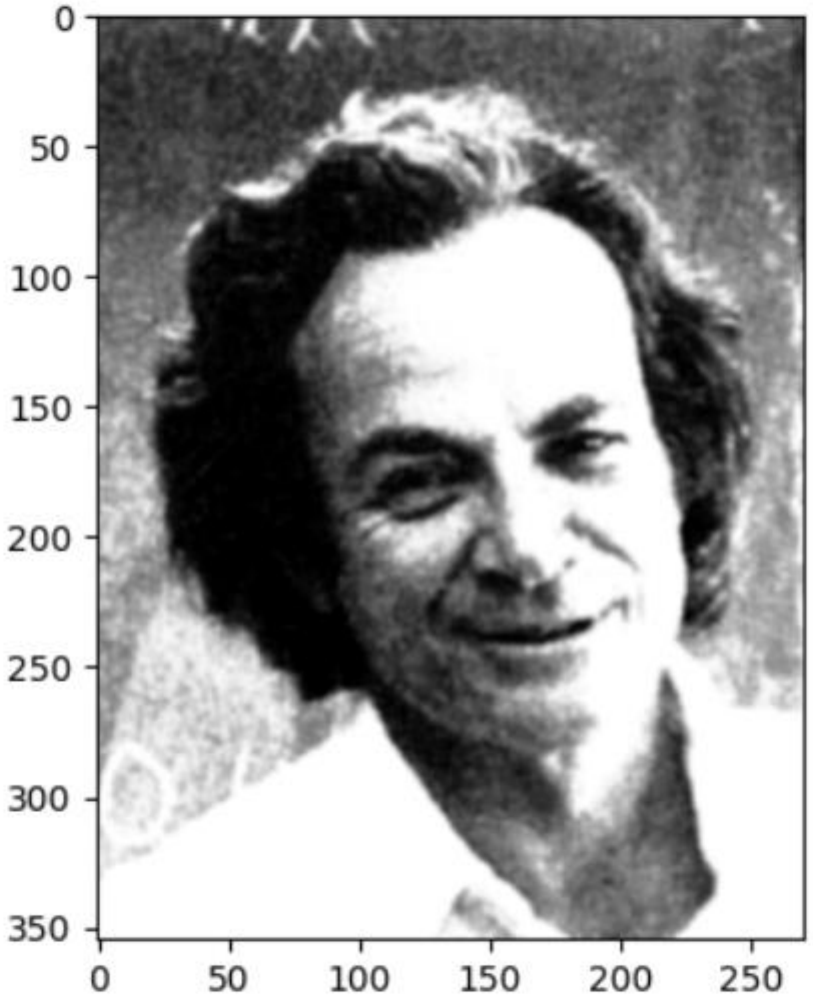
In order to do an iterative transfer texture function, I will use the texture transfer function which I already implemented. The idea is that I will use the output for each iteration as my target image and then I will use the tolerance value to determine whether the current change is too small. If the change is too small, then I will stop the iteration earlier.

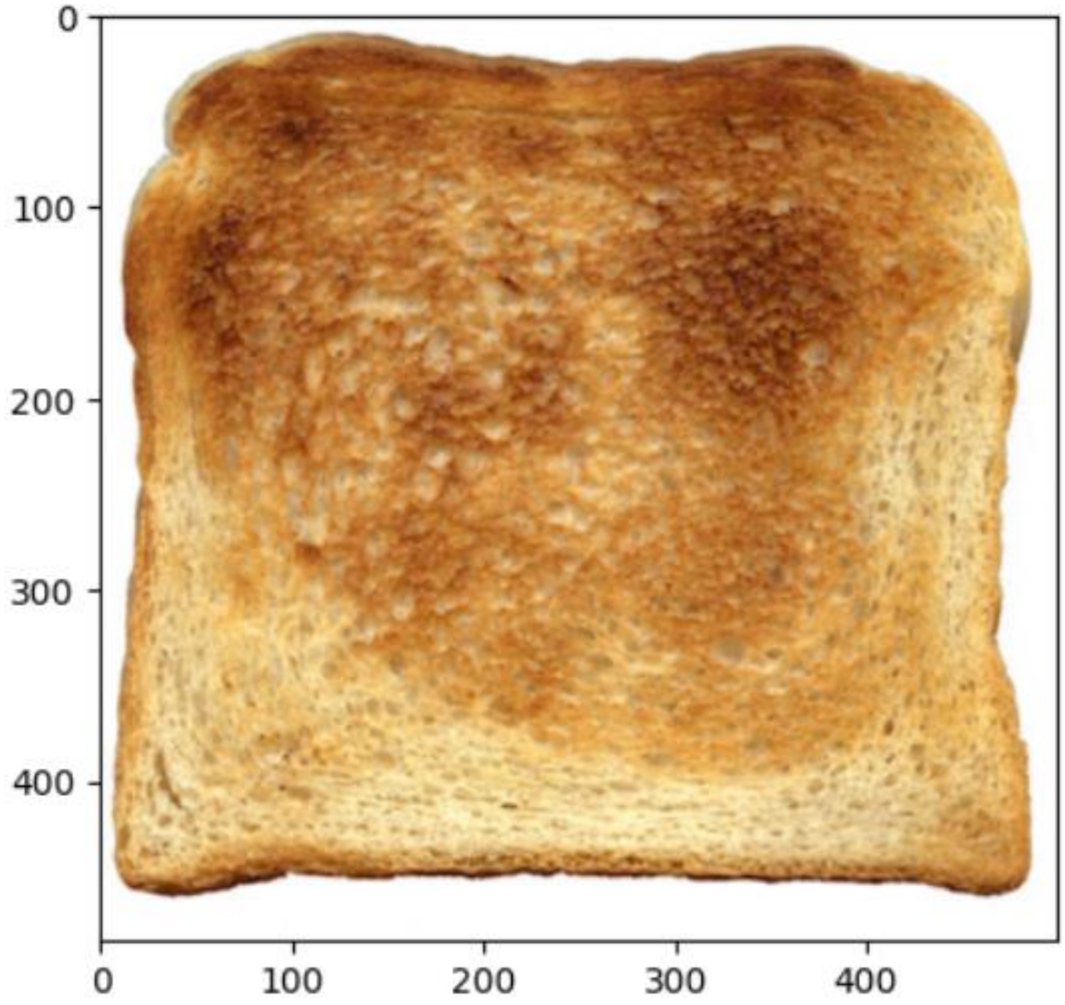
8. Face-in-Toast Image (B&W)

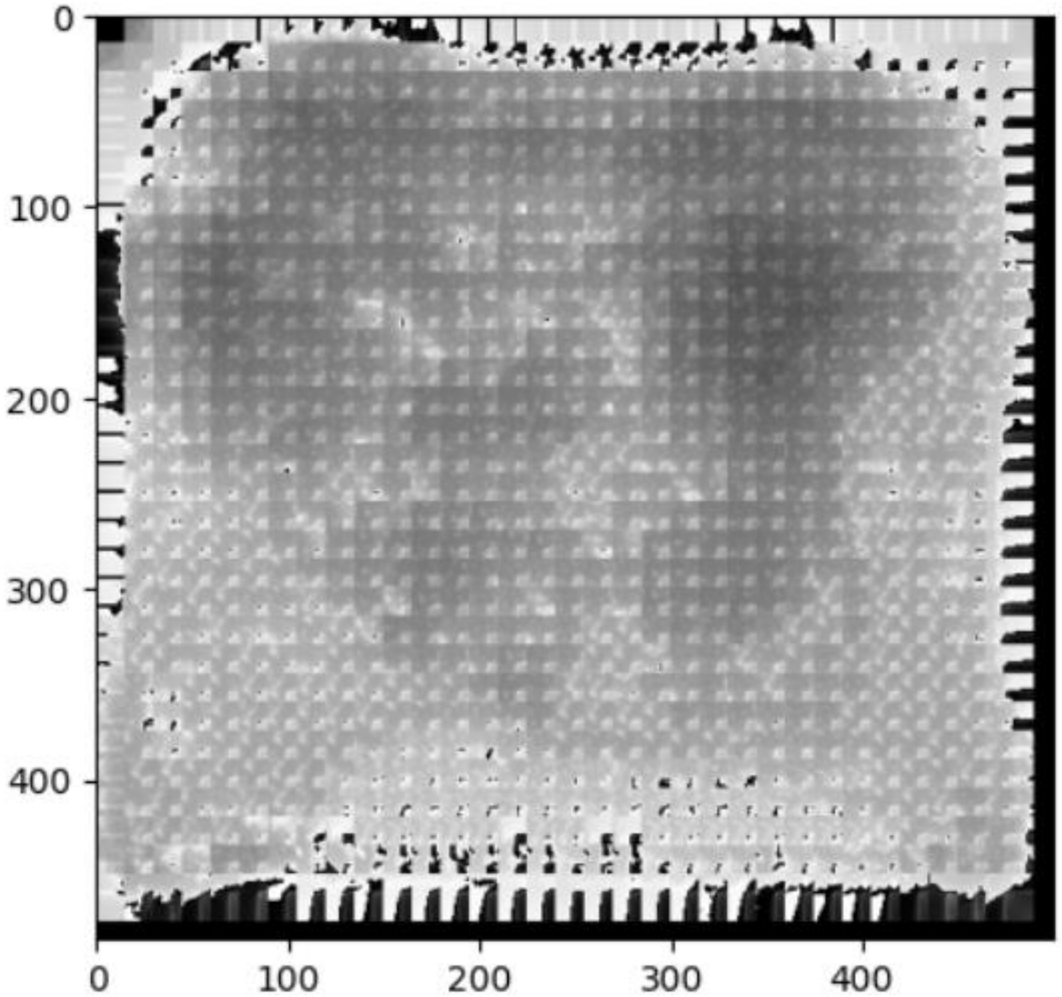
Include

- Describe method
- Show input face image, toast image, and final result

I first built the gaussain_pyramid and the laplacian_pyramid based on the Gaussian one for the template, new_patch, and the mask. Then, blend the laplacian_pyramid and reconstruct the laplacian_pyramid. By this laplacian pyramid, we don’t need to use the diff and cut like we did before. However, it seems that the texture transfer function didn’t work as my expectation and I don’t know why. It just maintains the face image texture and the color, and works on the toast instead of adding the face to the surface of the toast.


0
50
100
150
200
250
300
350
0
50
100
150
200
250


0
100
200
300
400
0
100
200
300
400


0
100
200
300
400
0
100
200
300
400

9. Hole filling w/ priority function (B&W)

Include

- Describe method
- Show result on at least two images (show input with hole and output)

Acknowledgments / Attribution

List any sources for code or images from outside sources