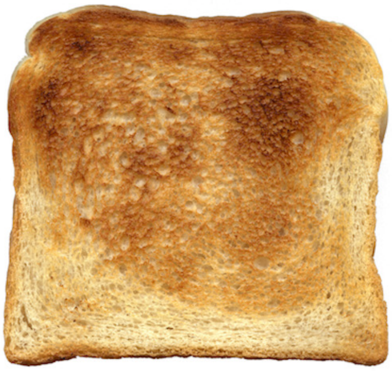
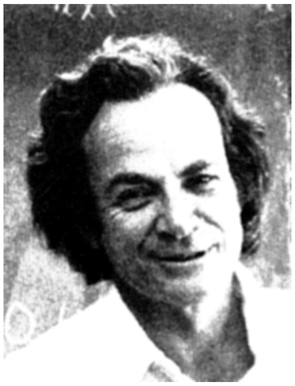
Get Started with Computer Graphics: Face-in-toast [1][2]

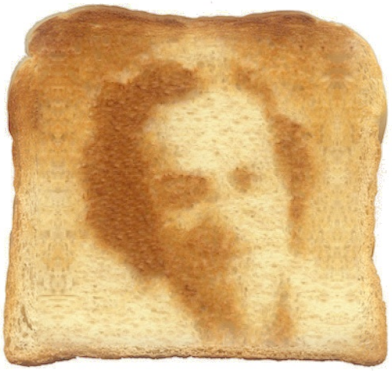
Table of Contents



toast



face



Face-in-toast

Want to put a face on toast? I don't know how to make it happen in kitchen but I can show you how to do it with a computer.

There are just two steps:

1. Transfer the toast texture to a face image.
2. Blend the toast-textured face into a toast image.

Setup

To begin with, we can download the code from here (a link is needed). After unzipping the folder, please install the required packages first by running the following command in terminal.

[1] The original idea and part of the content is from: https://courses.engr.illinois.edu/cs445/sp2020/projects/quilting/ComputationalPhotography_ProjectQuilting.html

[2] The document is written by yuw02176@gmail.com and submitted to Bit Project for evaluation. Please ask Wei for permission before using any part of the document for any other purpose.

```
pip install requirements.txt
```

Now we can open the Jupyter notebook example.ipynb and run the first input block to import the libraries we need.

```
import os
import math
import numpy as np
import cv2
import matplotlib.pyplot as plt
from utils import *
```

No output is good because it means there is no package missing. If there is error, it will say which packages are missing and we need reinstall it.

Texture Transfer

The toast image is already provided in the folder samples. We only need the non-crust part for texture and thus we will crop the image and save only the non-curst part.

```
sample_img_dir = 'samples/toast.jpg'
origin_img = cv2.imread(sample_img_dir)
is_texture = np.zeros_like(origin_img)
is_texture[50:-50, 50:-50, :] = 1
sample_img = origin_img[is_texture > 0].reshape((is_texture.shape[0] - 100, i
s_texture.shape[1] - 100, 3))
```

The original color channel is orded by BGR and the pixel values range from [0, 255]. We want to reorder to RGB and normalize the pixel values to [0, 1].

```
sample_img = cv2.cvtColor(sample_img, cv2.COLOR_BGR2RGB) / 255.0
```

For the face image, we can use the default one or upload one of our own. Here I am showing the command for the default face. For a customized face, we just need to change the filename.

```
target_img_dir = 'samples/feynman.tiff'
target_img = cv2.imread(target_img_dir)
target_img = cv2.cvtColor(target_img, cv2.COLOR_BGR2RGB) / 255.0
```

The texture transfer function is provided. We can just run it. Feel free to play with the parameters and see what happens. For details, you may read the paper[3].

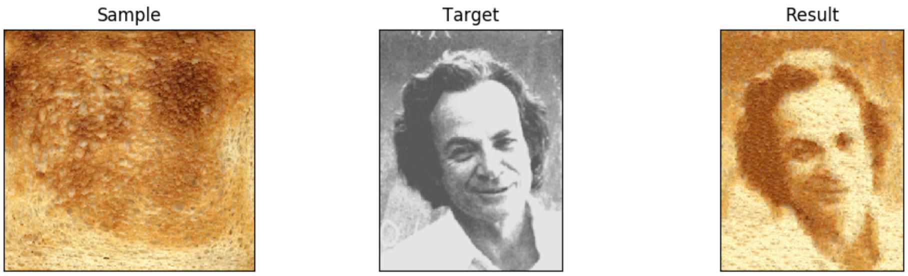
```
res_step_1 = texture_transfer(sample_img, target_img, 12, 5, 0.001, 0.4)
```

Now we can display the result and save it to file.

```
fig, axes = plt.subplots(1, 3)
```

[3] https://people.eecs.berkeley.edu/~efros/research/quilting/quilting.pdf

```
fig.set_size_inches(12, 3)
axes[0].imshow(sample_img)
axes[0].set_title('Sample'), axes[0].set_xticks([]), axes[0].set_yticks([]);
axes[1].imshow(target_img)
axes[1].set_title('Target'), axes[1].set_xticks([]), axes[1].set_yticks([]);
axes[2].imshow(res_step_1)
axes[2].set_title('Result'), axes[2].set_xticks([]), axes[2].set_yticks([]);
if is_save:
    im_to_save = np.flip(res_step_1, axis = 2)/ 1.0 * 255.0
    path = 'results/feynman_in_toast_step_1.jpg'
    cv2.imwrite('results/' + path, im_to_save)
```



If you see an output like the above, congratulations! We can continue to blending.

Blending

Now we want to blend the step 1 result with the toast crust.

We first load the image of a toast with crust.

```
img_A = cv2.cvtColor(origin_img, cv2.COLOR_BGR2RGB) / 255.0
```

Then we load our step 1 result, the image of face of toast texture and pad it to the same size as the image of toast.

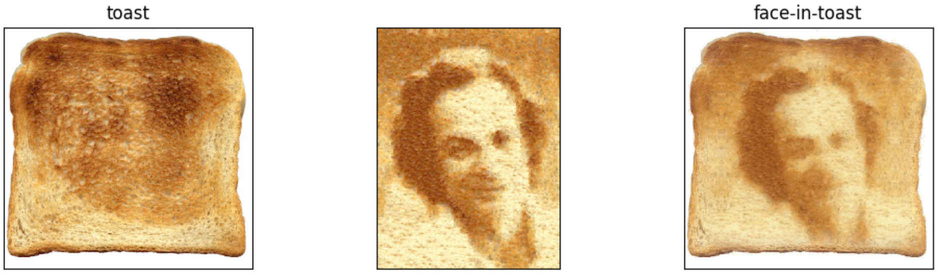
```
img_B = fill_canvas(res_step_1, inside_img, loc_start ='center')
```

The blending function is provided. We can just call it to run. If you want to know more, we are using a method called Laplacian blending. You can learn more details here[4].

```
res_step_2 = blend_images(img_A, img_B, is_texture, 6)
fig, axes = plt.subplots(1, 3)
fig.set_size_inches(12, 3)
axes[0].imshow(img_A)
axes[0].set_title('toast'), axes[0].set_xticks([]), axes[0].set_yticks([]);
axes[1].imshow(res_step_1)
axes[1].set_title(''), axes[1].set_xticks([]), axes[1].set_yticks([]);
axes[2].imshow(res_step_2)
```

[4] http://graphics.cs.cmu.edu/courses/15-463/2005_fall/www/Lectures/Pyramids.pdf

```
axes[2].set_title('face-in-
toast'), axes[2].set_xticks([]), axes[2].set_yticks([]);
if is_save:
    im_to_save = np.flip(res_step_2, axis = 2)/ 1.0 * 255.0
    cv2.imwrite('results/feynman_in_toast_step_2.jpg', im_to_save)
```



Here we go! Feel free to try other faces and surprise people with the “miracle” made by computer science.