Image Processing Example

January 31, 2020

[1]:
```
pip install Pillow
pip install matplotlib
pip install numpy
```

Requirement already satisfied: Pillow in ./env/lib/python3.7/site-packages (7.0.0)
WARNING: You are using pip version 19.2.3, however version 20.0.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.

Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: matplotlib in ./env/lib/python3.7/site-packages (3.1.2)
Requirement already satisfied: kiwisolver>=1.0.1 in ./env/lib/python3.7/site-packages (from matplotlib) (1.1.0)
Requirement already satisfied: cycler>=0.10 in ./env/lib/python3.7/site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in ./env/lib/python3.7/site-packages (from matplotlib) (2.4.6)
Requirement already satisfied: python-dateutil>=2.1 in ./env/lib/python3.7/site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: numpy>=1.11 in ./env/lib/python3.7/site-packages (from matplotlib) (1.18.1)
Requirement already satisfied: setuptools in ./env/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matplotlib) (41.2.0)
Requirement already satisfied: six in ./env/lib/python3.7/site-packages (from cycler>=0.10->matplotlib) (1.14.0)
WARNING: You are using pip version 19.2.3, however version 20.0.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.

You should consider upgrading via the 'pip install --upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: numpy in ./env/lib/python3.7/site-packages (1.18.1)
WARNING: You are using pip version 19.2.3, however version 20.0.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.

```python
from matplotlib.pyplot import imshow
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image

#matplotlib inline
pil_im = Image.open('/Users/tomerweiss/Desktop/lena_color.gif', 'r')
pil_im = pil_im.convert('RGB')
pix = pil_im.load()
fig = plt.figure(figsize=(18, 16), dpi=80, facecolor='w', edgecolor='k')
imshow(np.asarray(pil_im))
```

```
<matplotlib.image.AxesImage at 0x110dfdad0>
```
# do not copy paste code to your homework. we are using a different boundary condition here.

def reduceNoise(image):
    pix = image.load()
    reducedNoiseImage = Image.new('RGB', (image.width, image.height), color='white')
    pixReduced = reducedNoiseImage.load()
    for x in range(1, image.width - 1):
        for y in range(1, image.height - 1):
            # ================
            # the code below is too verbose. can you think of a more elegant formulation?
            top = pix[x,y - 1]
            top_left = pix[x - 1,y - 1]
            top_right = pix[x + 1,y - 1]
            left = pix[x-1, y]
            center = pix[x,y]
            right = pix[x+1, y]
            bottom = pix[x, y+1]
            bottom_left = pix[x-1, y+1]
            bottom_right = pix[x+1, y+1]
            # ================
            averagedPixelRed = int(np.average([center[0], top[0], top_left[0], top_right[0], left[0], bottom[0],
                bottom_left[0], bottom_right[0], right[0]]))
            averagedPixelGreen = int(np.average([center[1], top[1], top_left[1], top_right[1], left[1], bottom[1],
                bottom_left[1], bottom_right[1], right[1]]))
            averagedPixelBlue = int(np.average([center[2],
                top[2], top_left[2], top_right[2], left[2], bottom[2], bottom_left[2],
                bottom_right[2], right[2]]))
            pixReduced[x,y] = (averagedPixelRed, averagedPixelGreen, averagedPixelBlue)
    return reducedNoiseImage

img_noiseless = reduceNoise(pil_im)
fig = plt.figure(figsize=(18, 16), dpi=80, facecolor='w', edgecolor='k')
imshow(np.asarray(img_noiseless))

[4]: <matplotlib.image.AxesImage at 0x1213b3f90>