SamplingQuantization

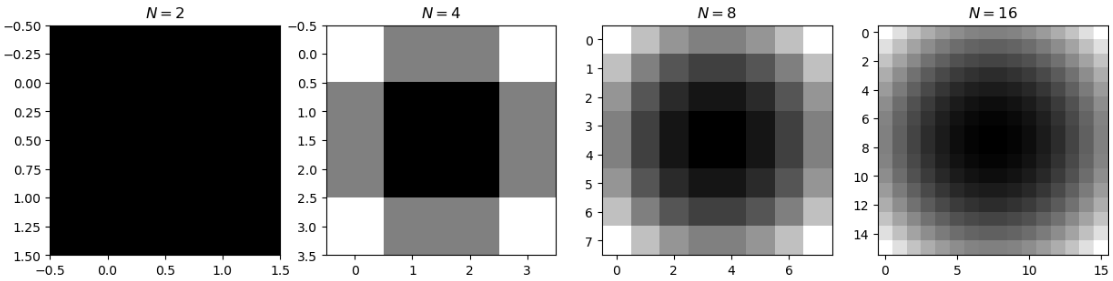
April 1, 2024

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
from skimage.io import imshow, imread
```

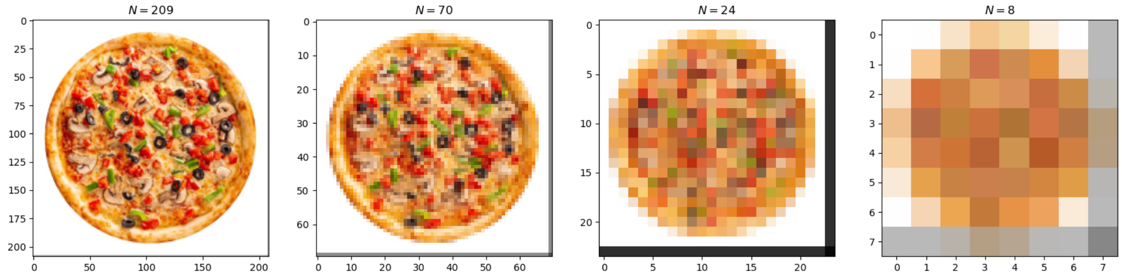
[2]:

```
def circle_image(x, y):
    X, Y = np.meshgrid(x, y)
    return X**2 + Y**2

factors = 2**np.arange(1, 5)
fig, ax = plt.subplots(1, len(factors), figsize=(15, 4))
for i, N in enumerate(factors):
    image = circle_image(np.linspace(-1, 1, num=N), np.linspace(-1, 1, num=N))
    ax[i].imshow(image, cmap='gray')
    ax[i].set_title('$N = {}$'.format(N))
```



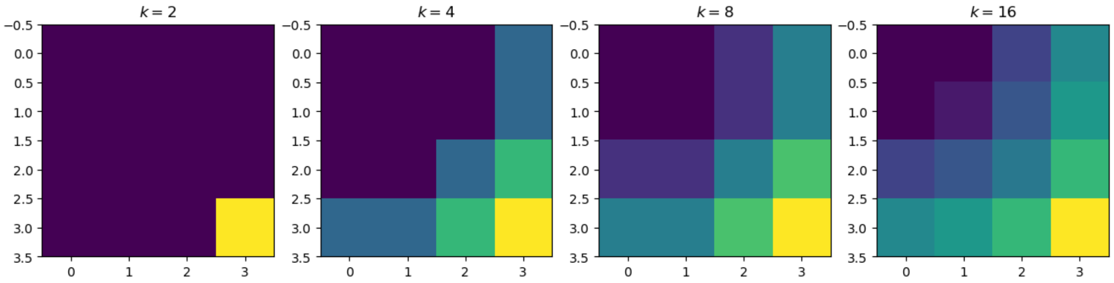
[3]:

```
pizza = imread('/home/nmit/Downloads/pizza.jpg')

from skimage.transform import downscale_local_mean
factors = 3**np.arange(1, 5)
figure, axis = plt.subplots(1, len(factors), figsize=(20, 6))
for factor, ax in zip(factors, axis):
    image = downscale_local_mean(pizza,
                                 factors=(factor, factor, 1)).astype(int)
    ax.imshow(image)
    ax.set_title('$N={}$'.format(image.shape[0]))
```



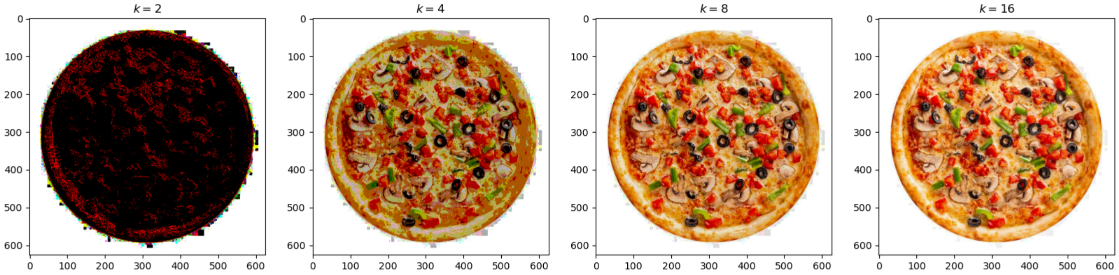
[4]:

```python
def circle_image(x, y):
    X, Y = np.meshgrid(x, y)
    return X**2 + Y**2

factors = 2**np.arange(1, 5)
circ_image = circle_image(np.linspace(0, 1, num=4),
                          np.linspace(0, 1, num=4))
fig, ax = plt.subplots(1, len(factors), figsize=(15, 4))
for i, k in enumerate(factors):
    bins = np.linspace(0, circ_image.max(), k)
    image = np.digitize(circ_image, bins)
    image = np.vectorize(bins.tolist().__getitem__)(image-1)
    ax[i].imshow(image)
    ax[i].set_title('$k = {}$'.format(k))
```



[5]:

```python
factors = 2**np.arange(1, 5)
figure, axis = plt.subplots(1, len(factors), figsize=(20, 6))
for k, ax in zip(factors, axis):
    bins = np.linspace(0, pizza.max(), k)
    image = np.digitize(pizza, bins)
    image = (np.vectorize(bins.tolist().__getitem__)
                         (image-1).astype(int))
    ax.imshow(image)
    ax.set_title('$k = {}$'.format(k))
```


$k = 2$
$k = 4$
$k = 8$
$k = 16$
0
100
200
300
400
500
600
0
100
200
300
400
500
600

[ ]: