Work on slidedoc dont let it run away from me

Get caught up on checkpoints

MP1

February 15, 2022

1 Mini Project 1

1.0.1 Luis Pereda

[127]:
```
# This is for ECE580: Intro to machine learning Spring 2020 in Duke
# This is translated to Python from show_chanWeights.m file provided by Prof. Li by 580 TAs

# import ext libs
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# from scipy.misc import imread   # Make Sure you install the required packages like Pillow and scipy

def imgRead(fileName):
    """
    load the input image into a matrix
    :param fileName: name of the input file
    :return: a matrix of the input image
    Examples: imgIn = imgRead('lena.bmp')
    """
    imgIn = plt.imread(fileName)
    return imgIn


def imgShow(imgOut):
    """
    show the image saved in a matrix
    :param imgOut: a matrix containing the image to show
    :return: None
    """
    imgOut = np.uint8(imgOut)
    plt.figure(figsize=[20,20])
    plt.imshow(imgOut, cmap = 'gray')


def imgRecover(imgIn, blkSize, numSample):
```

```
    """
    Recover the input image from a small size samples
    :param imgIn: input image
    :param blkSize: block size
    :param numSample: how many samples in each block
    :return: recovered image
    """
    ##### Your Implementation here

    return None

"""
if __name__ == '__main__':
    a = imgRead('lena.bmp')
    print(np.shape(a))
    imgShow(a)
    print(a)
"""
```

[127]:
```
"\nif __name__ == '__main__':\n    a = imgRead('lena.bmp')\n
print(np.shape(a))\n    imgShow(a)\n    print(a)\n"
```

[128]:
```
fishing_boat = imgRead("fishing_boat.bmp")
```

[129]:
```
nature = imgRead("nature.bmp")
```

[130]:
```
imgShow(nature)
```

[131]:
```
(fishing_boat[1:3, :4])
```

[131]:
```
array([[177, 176, 177, 178],
       [176, 176, 176, 178]], dtype=uint8)
```

[132]:
```
fishing_boat
```

[132]:
```
array([[176, 177, 178, …, 182, 160, 125],
       [177, 176, 177, …, 170, 131,  88],
       [176, 176, 176, …, 145,  98,  52],
       …,
       [100,  95,  99, …,  11,  14,  16],
       [ 98,  94, 103, …,  10,   8,   3],
       [ 64,  64,  64, …,  71,  71,  71]], dtype=uint8)
```

[133]:
```
# Sample into 8x8 blocks
blocks = []
for i in range(fishing_boat.shape[0] // 8 - 1):
    for j in range(fishing_boat.shape[1] // 8):
```

```
        blocks.append(fishing_boat[(i * 8):((i + 1) * 8), (j * 8):((j + 1) *␣
↪8)])
```

[134]:
```
import random

def sample_block(block, num_samples):
#     block = blocks[1]
    indices = np.linspace(0, np.prod(block.shape) - 1, np.prod(block.shape))
    random.shuffle(indices)
    indices = indices[0:(len(indices) - num_samples)]
    indices = [int(x) for x in indices]
    ret = block.flatten()
    ret[indices] = 0
    ret = ret.reshape([block.shape[0], block.shape[1]])
    return ret
```

[135]:
```
sample_block(blocks[0], 32)
```

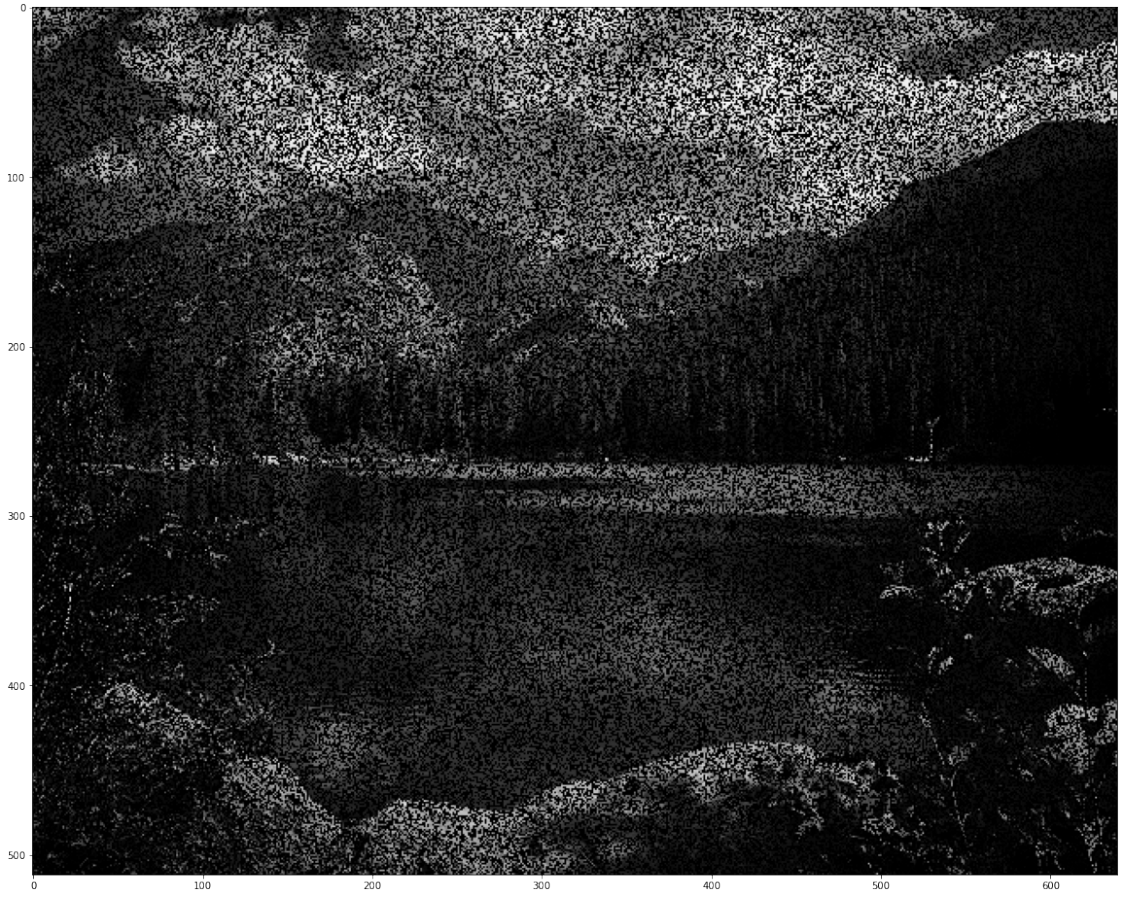
[135]:
```
array([[  0, 177,   0,   0,   0,   0,   0, 179],
       [  0, 176, 177,   0, 178,   0, 178, 177],
       [176, 176,   0, 178,   0,   0,   0,   0],
       [177,   0,   0, 177, 178,   0,   0, 177],
       [  0, 180, 179, 179,   0, 177,   0,   0],
       [  0, 181, 180, 179, 179, 177,   0, 176],
       [  0, 179,   0, 182, 185,   0, 178,   0],
       [178,   0,   0, 181,   0,   0, 182, 181]], dtype=uint8)
```

[136]:
```
blocks[0]
```

[136]:
```
array([[176, 177, 178, 179, 179, 178, 181, 179],
       [177, 176, 177, 178, 178, 178, 178, 177],
       [176, 176, 176, 178, 177, 178, 177, 178],
       [177, 177, 178, 177, 178, 177, 178, 177],
       [180, 180, 179, 179, 178, 177, 182, 176],
       [181, 181, 180, 179, 179, 177, 184, 176],
       [179, 179, 178, 182, 185, 178, 178, 179],
       [178, 178, 179, 181, 183, 180, 182, 181]], dtype=uint8)
```

[138]:
```
imgShow(sample_block(nature, np.prod(nature.shape) // 2))
```

Fix the ordering of blocks after they are sampled (finished after class)


0
100
200
300
400
500
0
100
200
300
400
500
600

[ ]: