MP #1: Peer Feedback #2 (Pre-Feedback Work to Date)

Libo Zhang (lz200)

The structure of my work to date format will follow the recommended project milestones.

Note: Only the code written for **testing** my algorithm will be displayed to help peer review/feedback.

Week 3 & Week 4:

(1) Implement random subset cross-validation to choose the regularization parameter λ.

Note 1: The regularization parameter is denoted as "alpha" in my code because in Python, "lambda" is used to efficiently define a short function so that I avoid using it in case of misunderstanding or unexpected bugs.

Note 2: The print-out shown below is to demonstrate that I do remember that λ may be different for each block!

```
In [30]: # Incremental testing for recovering the small image
         # block size K = 8, sampled pixels S = 50
         recon_img1 = imgRecover(image1, 8, 50)
```

```
The best alpha for current block is 0.100000!

Implement random subset cross-validation to current block
Current block [i, j] is:  15 22
The best alpha for current block is 1.000000!

Implement random subset cross-validation to current block
Current block [i, j] is:  15 23
The best alpha for current block is 0.000001!

Implement random subset cross-validation to current block
Current block [i, j] is:  15 24
The best alpha for current block is 0.100000!

Implement random subset cross-validation to current block
Current block [i, j] is:  16 1
The best alpha for current block is 0.010000!

Implement random subset cross-validation to current block
Current block [i, j] is:  16 2
```

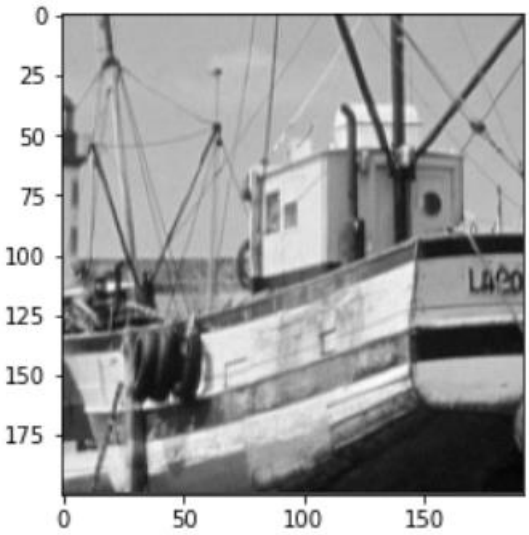
(2) Begin and continue simulations to explore the impact of the proportion of pixels that are sampled (S).

(3) Begin interpreting results (for discussion).

```
# Explore the impact of the proportions
# of pixels that are sampled (S)
re1_img1 = imgRecover(image1, 8, 10)
re2_img1 = imgRecover(image1, 8, 20)
re3_img1 = imgRecover(image1, 8, 30)
re4_img1 = imgRecover(image1, 8, 40)
re5_img1 = imgRecover(image1, 8, 50)
```

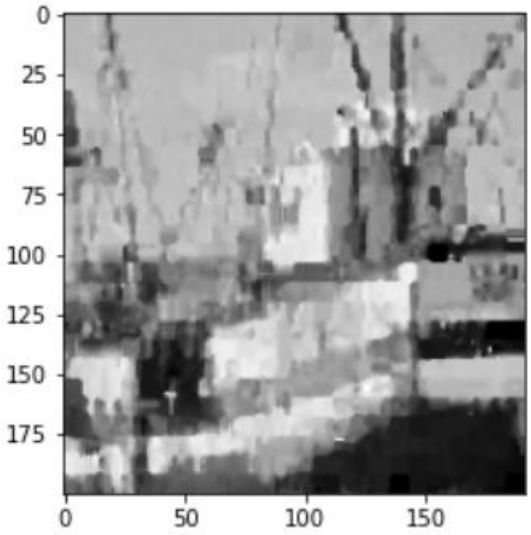
```
# Original image
plt.imshow(image1, cmap = "gray")
```

```
<matplotlib.image.AxesImage at 0x1386e037
```



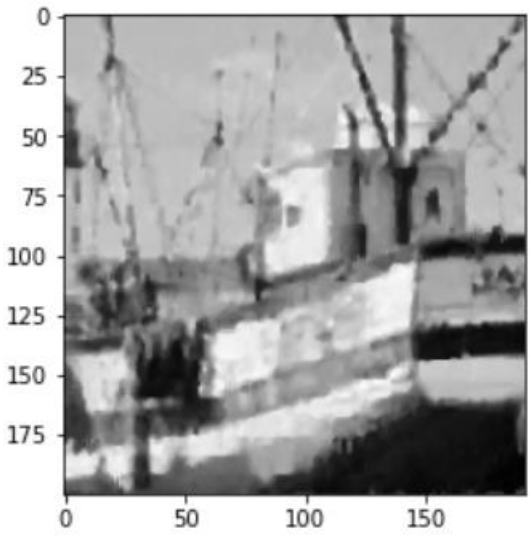

```
# S = 10, recovered image
plt.imshow(re1_img1, cmap = "gray")
```

```
<matplotlib.image.AxesImage at 0x1386e
```



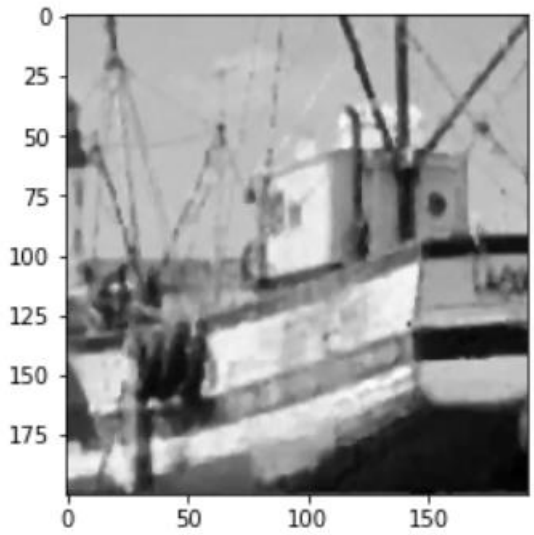

```
# S = 20, recovered image
plt.imshow(re2_img1, cmap = "gray")
```

```
<matplotlib.image.AxesImage at 0x1386e
```



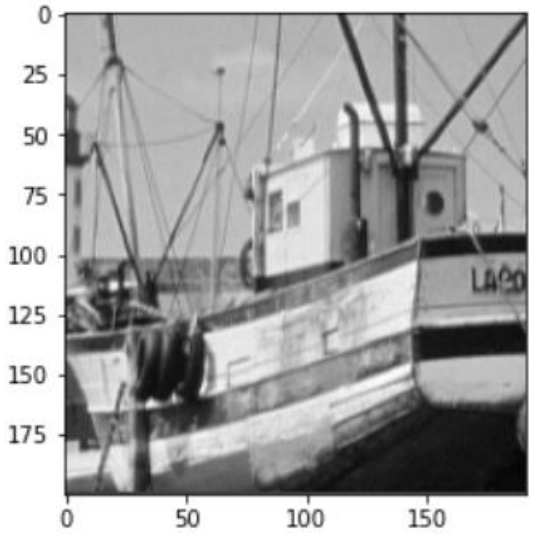

```
# S = 30, recovered image
plt.imshow(re3_img1, cmap = "gray")
```

```
<matplotlib.image.AxesImage at 0x1386e
```



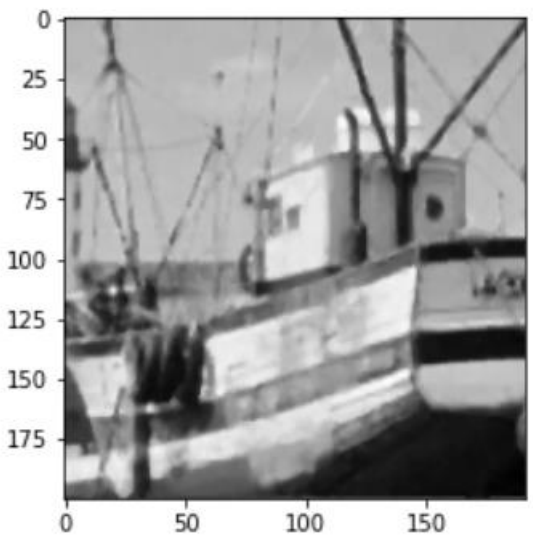
```
# Original image
plt.imshow(image1, cmap = "gray")
```

```
<matplotlib.image.AxesImage at 0x1386e0
```




```
# S = 40, recovered image
plt.imshow(re4_img1, cmap = "gray")
```

```
<matplotlib.image.AxesImage at 0x1386e7
```




```
# S = 50, recovered image
plt.imshow(re5_img1, cmap = "gray")
```

```
<matplotlib.image.AxesImage at 0x1386e
```