COMPUTATIONAL IMAGING

ACTIVITY

04

COMPRESSIVE SENSING FOR IMAGES

SUBMITTED BY:
RENE L.PRINCIPE JR.

SUBMITTED TO:
DR. MARICOR N. SORIANO

Background

Images are 2D signals, and recall that a signal $\boldsymbol{x}$ can be represented by a sparse vector **s** and a basis transformation $\boldsymbol{\Psi}$:

$$\boldsymbol{x} = \boldsymbol{\Psi}\, \boldsymbol{s}.$$

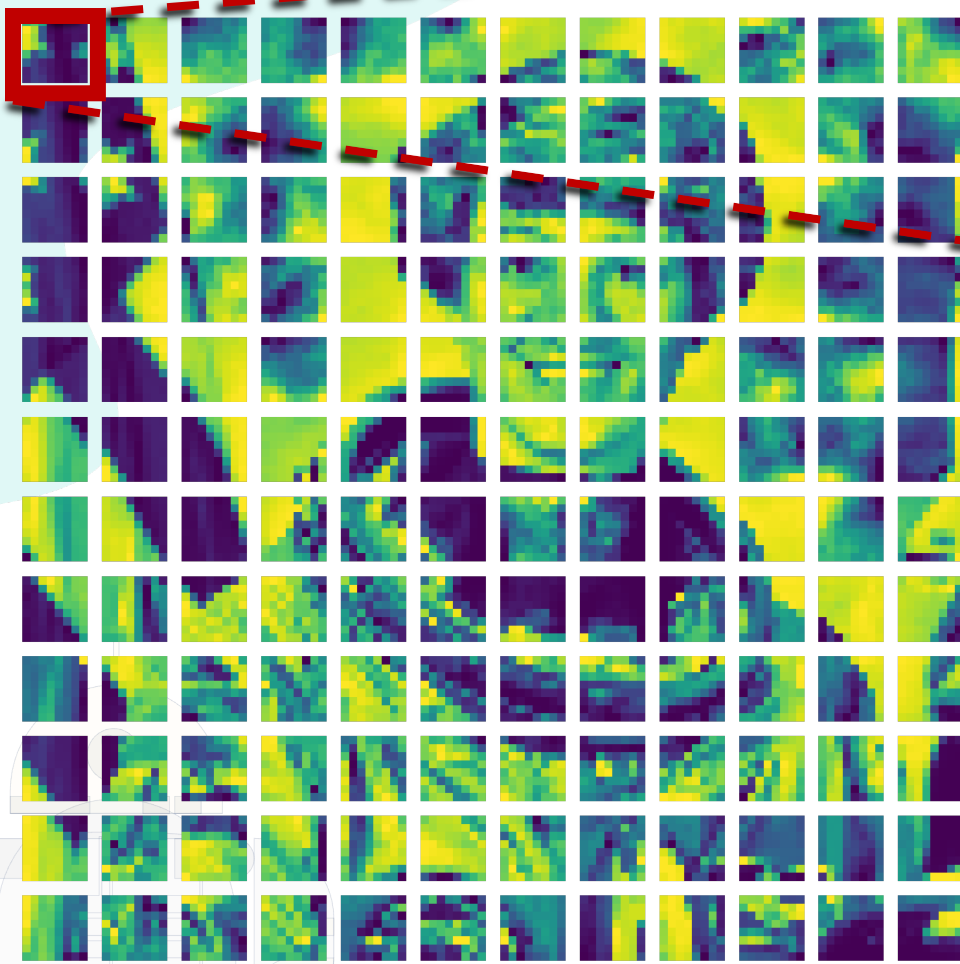
In this activity, we performed random sampling compressive sensing (RS-CS) on image blocks. Measurement matrix $\boldsymbol{C}$ is then again obtained by taking random measurements $\boldsymbol{y},$ that is

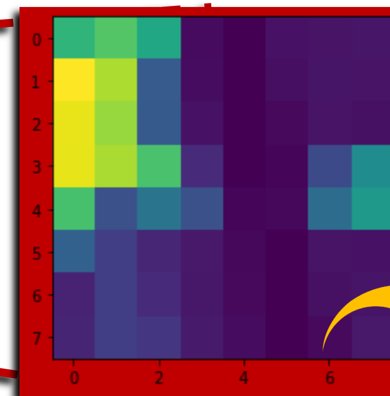
$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} = \boldsymbol{C}\boldsymbol{\Psi}\boldsymbol{s}.$$

The sparse vector can be solved through minimization and here we used Lasso optimization algorithm. Using DCT as a known basis $\boldsymbol{\Psi}$, the original image signal $\boldsymbol{x}$ can be recovered.

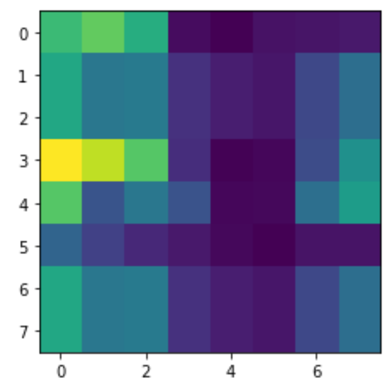
The reconstruction was automated in Python such that effects of increasing sampling rate and varying input image sizes are returned. RD-CS was done on individual 8x8 non-overlapping blocks, before they are reassembled to represent the recovered image. Finally, the accuracy was measured using the Structure Similarity Index (SSIM).

Overview: Block RS-CS





1. RS: 50% sampling rate
2. Lasso optimization
3. DCT reconstruction

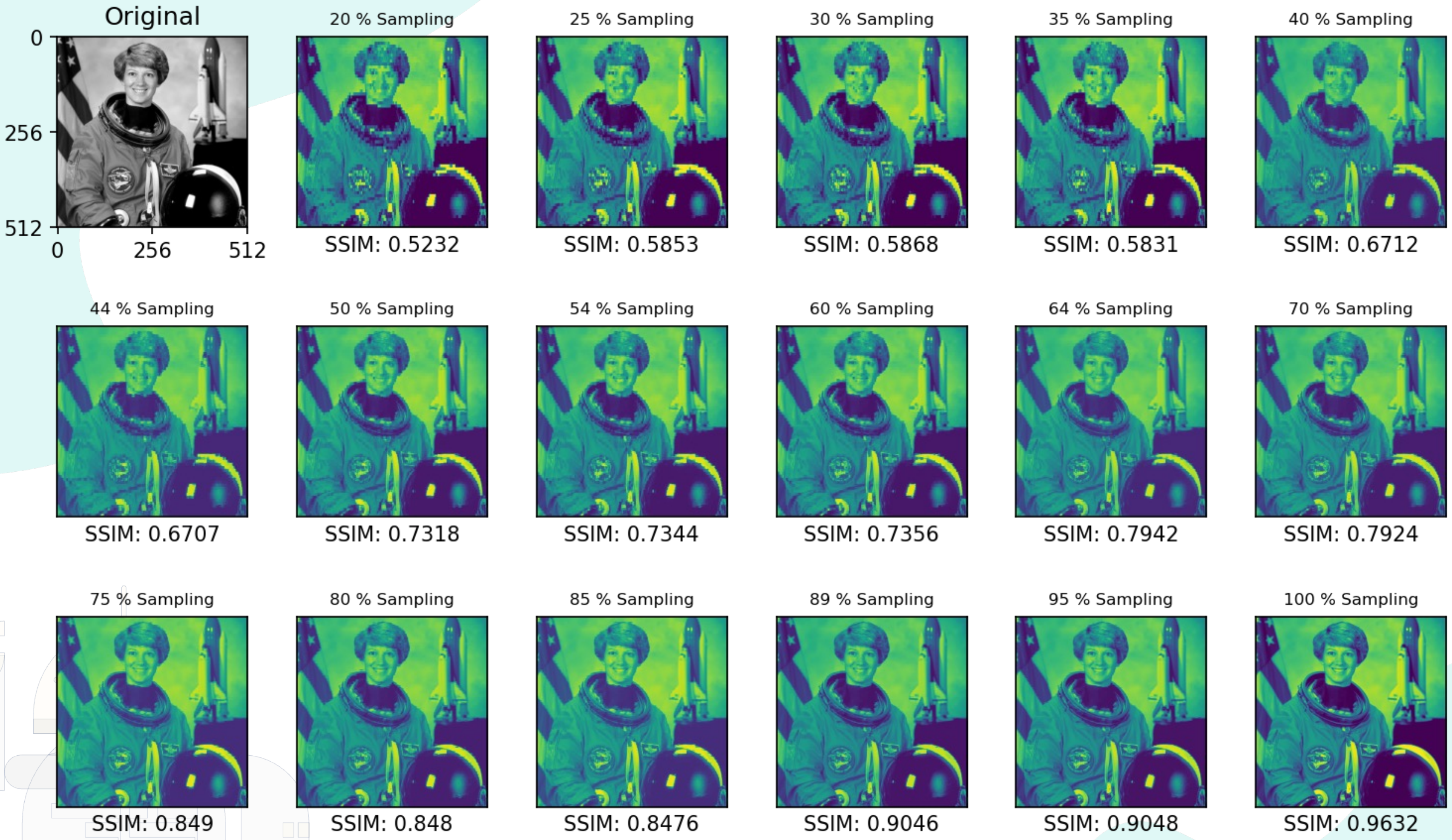


4. Block re-assembly
5. SSIM calculation

RS-CS: 128 X 128 Image

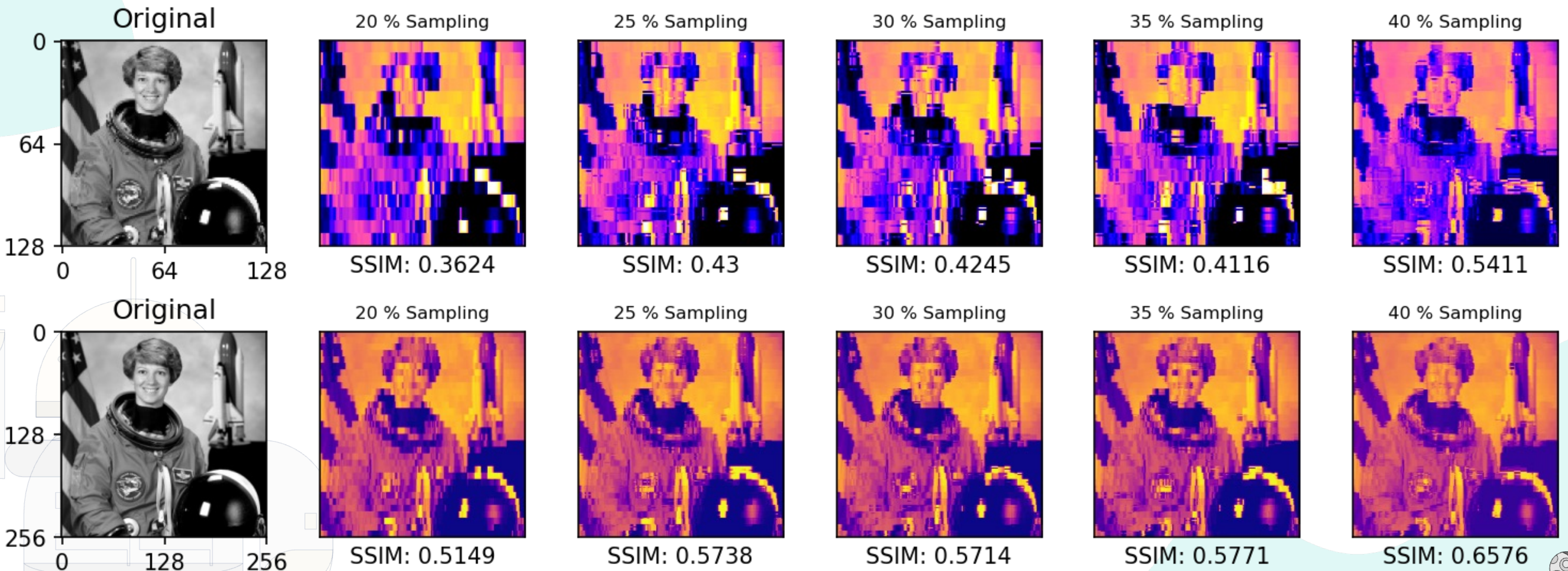
RS-CS : 256 X 256 Image

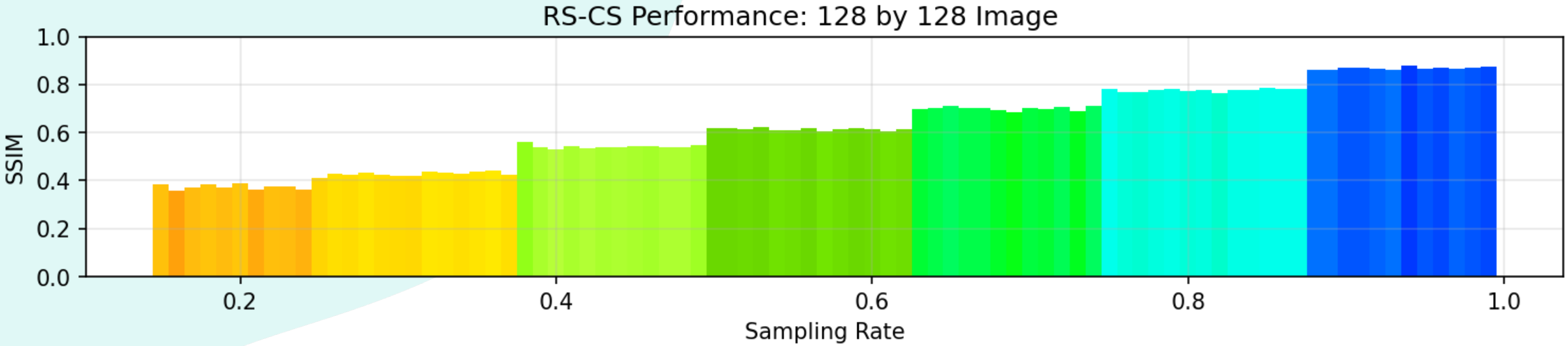
RS-CS : 512 X 512 Image

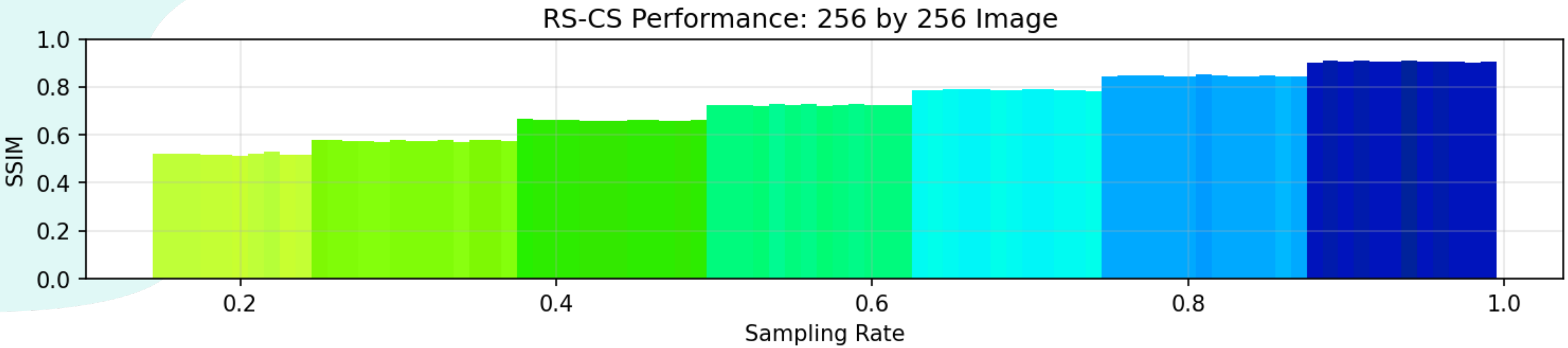


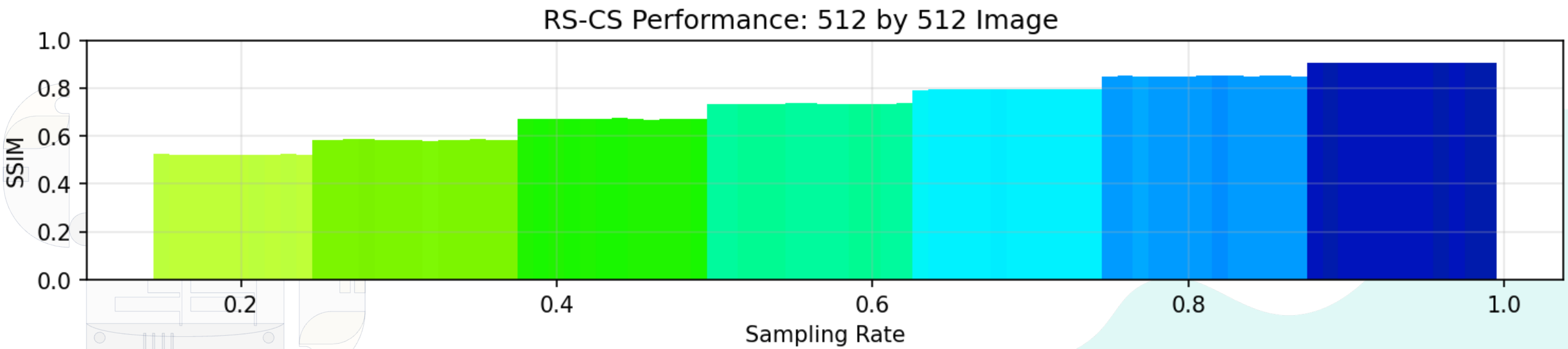
Results

Interestingly, a distinct jump in the reconstruction's SSIM was observed upon increasing the input image's dimension from (128 x 128) to (256 x 256) as shown below. Although the SSIM is increased, one main trade-off is that it took four times as long to perform RS-CS on by increasing the image size to 2^{n+1}. Intuitively, increasing the random sampling rate means better reconstruction. I then performed large-scale analysis to compare RS-CS performances for increasing input image size and increasing sampling rate.




RS-CS Performance: 128 by 128 Image
SSIM
Sampling Rate
0.0
0.2
0.4
0.6
0.8
1.0


RS-CS Performance: 256 by 256 Image
SSIM
Sampling Rate
0.0
0.2
0.4
0.6
0.8
1.0


RS-CS Performance: 512 by 512 Image
SSIM
Sampling Rate
0.0
0.2
0.4
0.6
0.8
1.0

Discussion

Increasing the image size to (512 x 512) didn't return significant improvements in the reconstruction quality. Also, the automated RS-CS couldn't recover images for sampling rates below 15%. On average, the results show that at 20% sampling rate, a 40% similarity can be recovered.

Take note however that the image used is complicated, hence performing RS-CS on simpler objects with very distinct peaks in the frequency space would only need lower sampling rate. As demonstrated last activity, distinct frequency peaks were retained even at 5% sampling rate.

reflection

I put off this activity because I honestly thought this is difficult and overwhelming, however, I reused 90% of the code that I used in implementing compressive sensing for audio signals. The process is the same, it's just that flattening and recompiling the image was carried out, and SSIM measure was used to evaluate the reconstruction quality. Overall, this report showed a sufficient set of results and straightforward implementation and discussion of Random Sampling Compressive Sensing (RS-CS) for images aka 2D signals. It was really interesting and same sentiment as before; I genuinely wish I did this way back so that I could've explored it extensively. Regardless, I have shown a large-scale analysis on increasing image sizes and fine increase in sampling rate to generalize the RS-CS performance.

With that said, I'd give myself a score of **100/100.**

references

- [1] M. Soriano, Physics 305 – Compressive Sensing of Images, (2023).

SOURCE CODE

https://github.com/reneprincipejr/Physics-305/tree/main/Activity%204%20-%20Compressive%20Sensing%20for%20Images