Part 3: Analysis Question

Process to randomly remove ‘x’ percent of pixels from the input image

1. Calculate the number of pixels to remove from the image using the formula: numPixelsToRemove = (width * height * x)/100
2. Initialize a set to store the removed pixel co-ordinates
3. Randomly remove ‘numPixelsToRemove’ pixels i.e set the RGB values of these pixels to zero (make sure that you don’t chose the same pixel coordinates) and add the coordinates to the set

Recompute new values for each removed pixel using valid neighborhood samples

1. Iterate over the removed pixel set
2. For each removed pixel, compute the pixel by averaging over a 3x3 valid neighborhood of pixels (3x3 filter) in the original image centered around the removed pixel

Calculating the reconstruction error

1. Reconstruction error is obtained by computing the sum of squared differences between all the pixels of the original and reconstructed image for each channel and then averaged over all the channels

   Here’s the formula:

```
diffR = abs(inImage.GetRed(x, y) - reconstructedImage.GetRed(x, y));

diffG = abs(inImage.GetGreen(x, y) - reconstructedImage.GetGreen(x, y));

diffB = abs(inImage.GetBlue(x, y) - reconstructedImage.GetBlue(x, y));

Reconstruction_error += (diffR*diffR + diffG*diffG + diffB*diffB)/3;
```

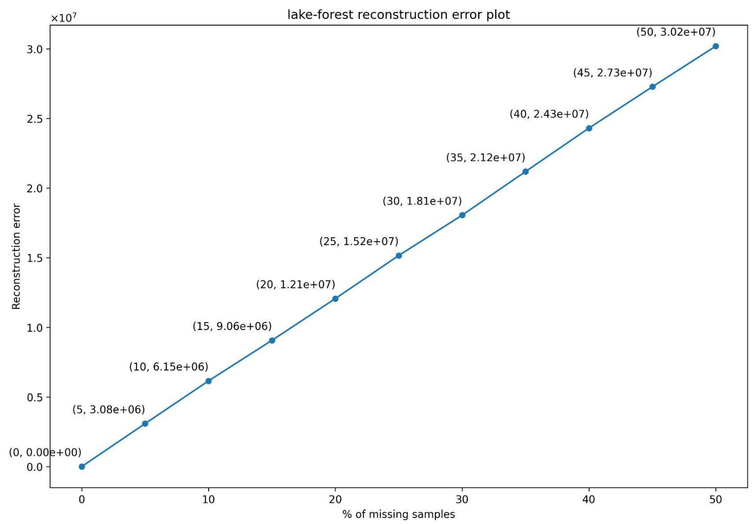
The sampled/noisy image and reconstructed image can be found in the folder /sampled-and-reconstructed-images. The number in filename corresponds to the x value.

Questions:

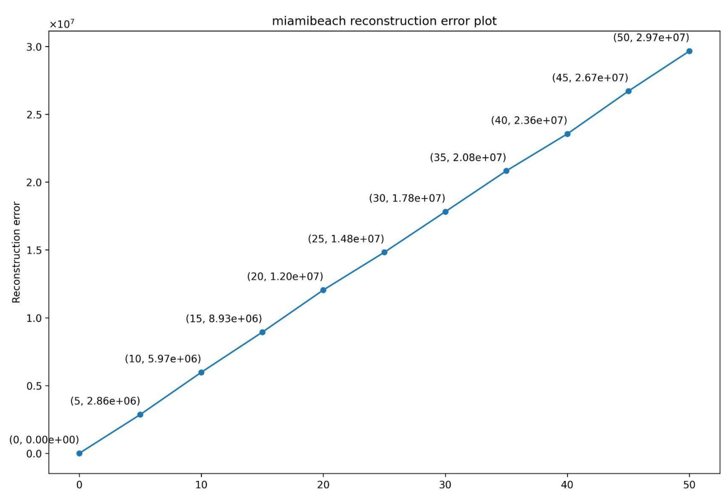
1. For each of the given images, plot a graph for the reconstruction error

   The graph images and the comparison are attached as images, please feel free to view them and skip to page 6

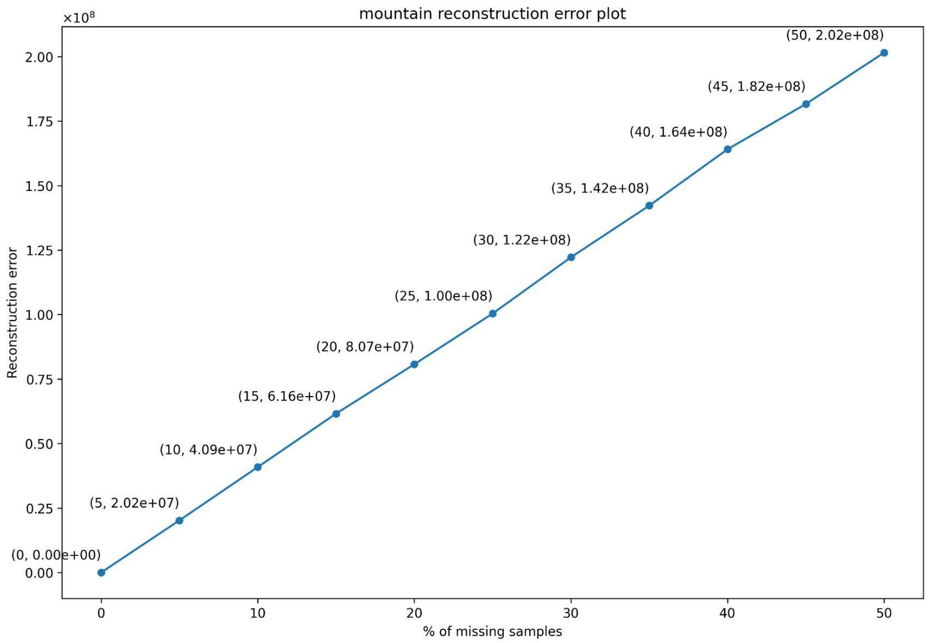
   a. **lake-forest**



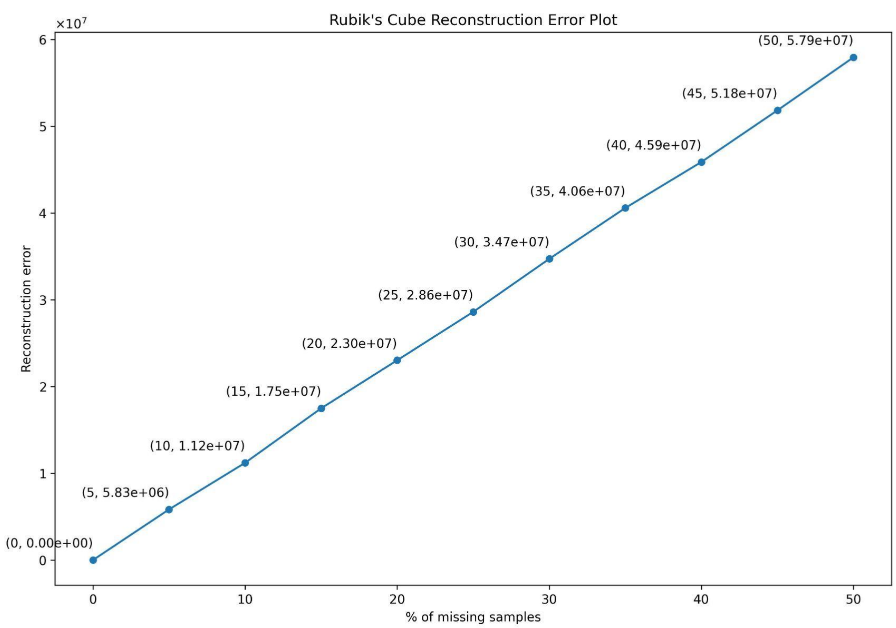
   b. **miamibeach**



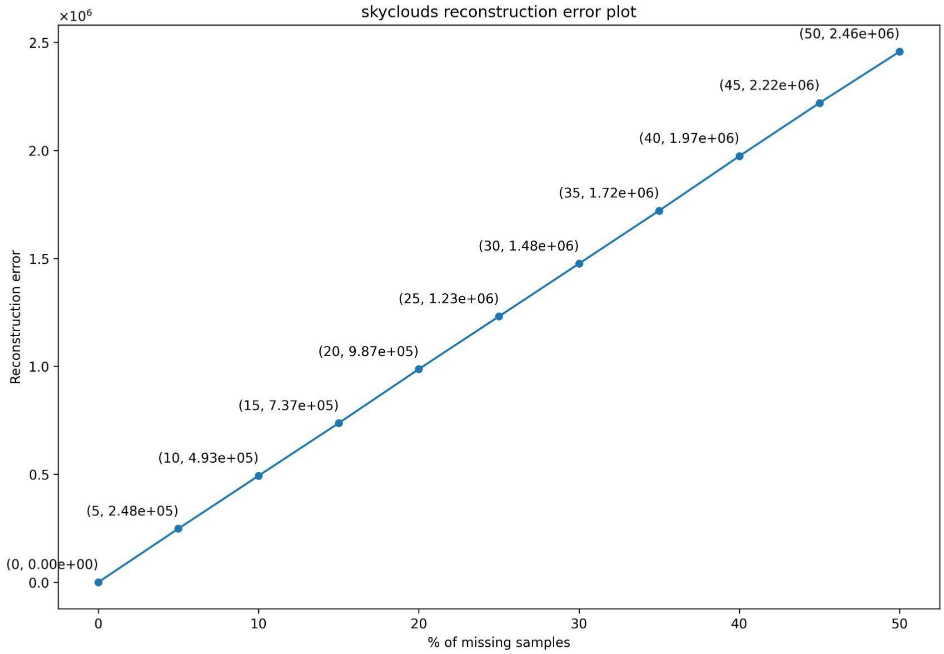
c. Mountain



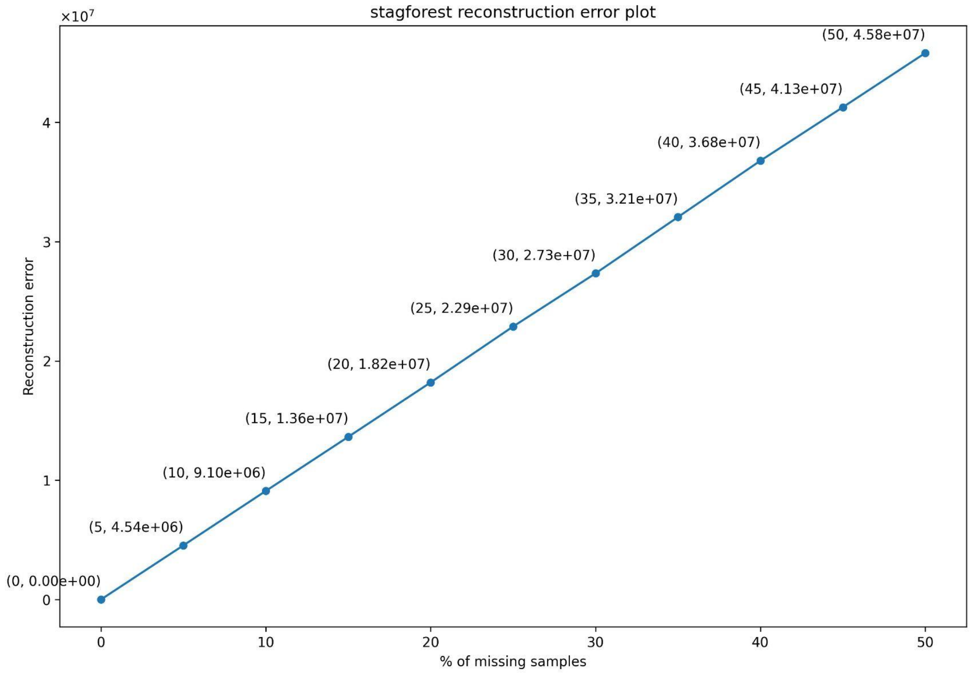
d. Rubixcube



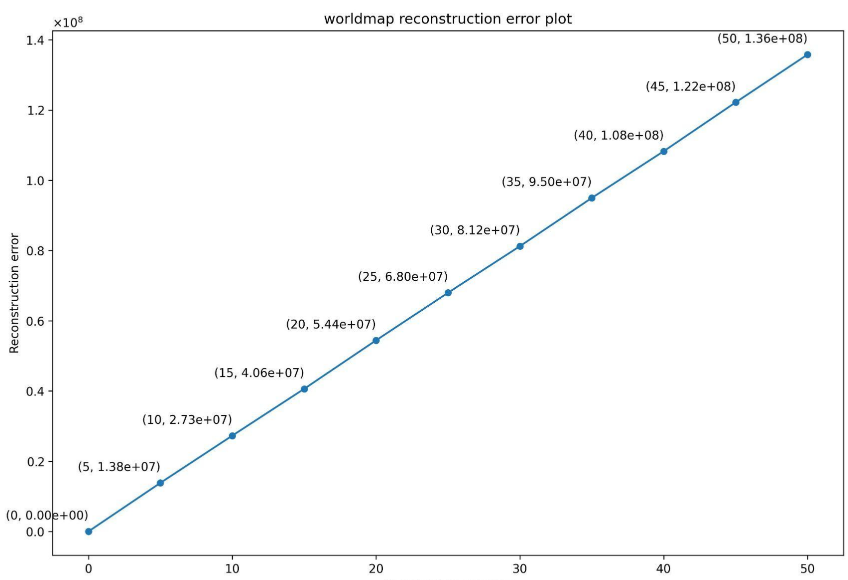
e. skyclouds



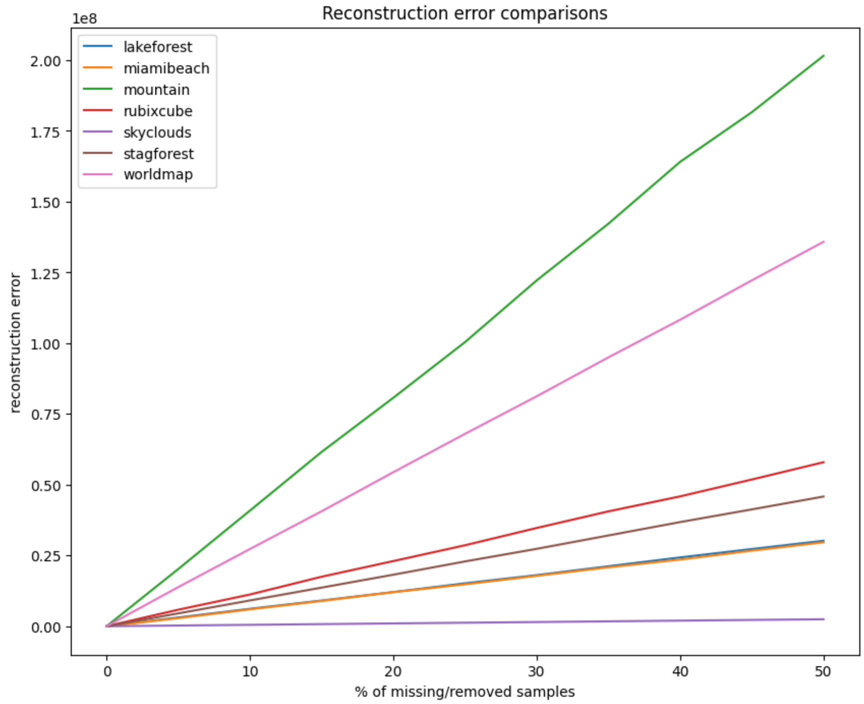
f. Stagforest



g. Worldmap



Here’s a comparison between all the given images



2. Which image has higher errors, which image has lower error? Why are all the plots different?

As observed from the above comparison results, mountain.rgb has the highest reconstruction error while skyclouds.rgb has the lowest error.

Although all plots are linearly increasing as percent of samples are increased, they are different for each image as the **reconstruction errors are different.** For the same interpolation technique (average of 3x3 neighboring pixels), the reconstruction error depends on **several factors** such as **image content, data distribution and complexity**. Images with sharp edges like rubixcube or boundary of a country in the worldmap are more challenging to reconstruct as the fine details are lost during the **reconstruction process.**

3. From your quantitative analysis, can you qualitatively describe which image will have higher error and which image will have lower error.

1. **Images with more fine, sharp patterns, complexity or variations** are likely to have high errors. For example, in our case, the data distribution is highly variable and complex in the forest region of the mountain image leading to high reconstruction error.
2. **The presence of noise** in the original image also impacts the reconstruction error. Noise can make it challenging to estimate missing values correctly. For example, The white

stars (or dots) in the sky scattered around in the mountain image appear to look like a **salt pepper noise** which makes it challenging to estimate missing values correctly.

3. Images with **similar structure** over a large region or in other words **less variant or less distributed have lower errors** as the neighboring pixels would be similar, effectively computing the accurate pixel value (by averaging). And that's why, skyclouds image has low error since it's less variant (mostly blue and even).

Extra Credit

Given an input image and x percent of samples randomly removed, how to predict reconstruction error?

The reconstruction error as described above depends on data distribution or image varies. That hints us towards computing variance for each region (3x3 neighborhood).

Mathematically, the local variance is given by $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$

Where N is the number of pixels in the neighborhood (for example, N = 9 for a 3x3 filter), x_i is the pixel value within the region and μ is the average pixel in that region.

This local variance provides insights as to how noisy or how much detail is present in the region. It quantifies how much pixel values vary within a specific region. This model assumes that local variance reflects the variation in pixel values within each region and the errors in different regions are not correlated. Thus, It can be used to estimate how much the local image characteristics might affect the quality of the reconstructed image when samples are removed.

Mean squared error for each local region i can be computed as:

$MSE = \frac{1}{N} * \sigma_i^2 * \frac{x}{100}$

(x is the percentage of samples removed)

Reconstruction error for the entire image can be estimated as:

Reconstruction error = $\frac{1}{M}\sum_{i=1}^{M} MSE_i$

This error can be compared with the actual reconstruction error obtained!