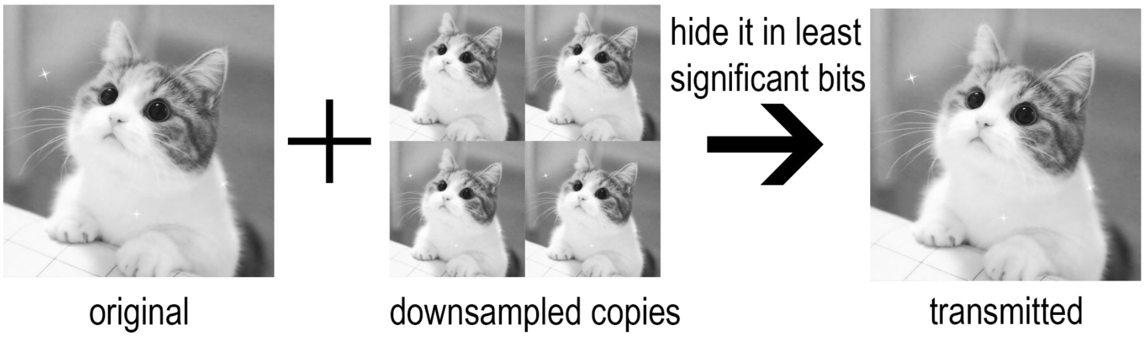
Homework 2: Error Concealment in Images

Imagine you are transmitting an image over a channel but during this transmission, an error has occurred and image data is corrupted. What might be a clever way to hide this problem and show the image more or less similar to the original data? In this homework, you will implement one solution to this task and simulate this process.

Core algorithm is as follows:

1. Create 4 copies of the original image (downsampled half the size) (Downsampling algorithm: select only the odd numbered rows and columns, check PS3 materials) . Concatenate these 4 images into a single one as in the figure.
2. Get the most significant n bits of the downsampled image and hide this data in least significant n bits of the original image.
3. Transmit this new image over a channel (we will just assume this has happened as we are only imagining a situation)
4. An error has occurred. Let's, for example, say 30 rows of the original image is corrupted with random numbers. (You simulate this manually)
5. Select a downsampled copy in an uncorrupted quadrant and extract the hidden data from least significant n bits. Since they are in least significant n positions, don't forget to shift these bits back to their original positions. Now, upsample back to original size using copying each pixel 4 times. (this method of upsampling is in PS3 materials.).
6. Replace corrupted image with the image you got from previous step.



Every student will have a different corrupted row calculated using their student number. Let's say your student number is = 2017400212. Last three numbers are 212. You are going to corrupt 30 rows starting from the row 212 of the original image. You will be given images of size 512x512. If your number is bigger than 220, take ((your_number % 220) + 1).

Use rng(1) command to set the random seed of MATLAB to 1 so that your results are reproducible.

To create the random row, use the rand() function.

Submission

You are given three images cat.png, dog.png and otter.png. Work in **grayscale** and for each of them do the following:

1. Run the above algorithm for your own corrupted row for n = 2,3,4,5.
2. For each of the n values, calculate three different RMSE values.
3. First rmse value is in between original grayscale image and transmitted image (image just after the step 2 of the core algorithm)
4. Second rmse value is in between original grayscale image and corrupted transmitted image (image just after the step 4 of the core algorithm)
5. Third rmse value is in between original grayscale image and recovered image (image just after the step 6 of the core algorithm).

In total you will have 12 values for each image. Put them in a table for each image separately and put these values both **as tables** and **as plots (show how rmse changes as n increases)** in a short report. Your report need not be very detailed or long. Please add final recovered images and a short commentary of the results in your report.

Create a zip file with the name YOUR_STUDENT_NUMBER.zip that contains your MATLAB code and report pdf. If your report pdf exceeds moodle upload limit, you can provide an online link too.