Coding Assignment 1: Image Steganography and Steganalysis

Due: 9/9/2022

As we learned in class, image steganography is the technique of hiding a message within an image. In this assignment, first, you will write code that encodes a message stored in a file called `secret.txt` into an image, as well as decoding the message (which is the reverse process of encoding). Then, you will write code that will detect messages in steganographic images.

Part 1: Image Steganography (80%)

For this section, you first need to design an algorithm for image steganography and implement it. Follow the following steps:

1. Read this article: https://www.geeksforgeeks.org/image-based-steganography-using-python/?ref=lbp
2. Download the python code in the article and play with it (that is, encode and decode images). Alternatively, you can use the code provided to you.
3. Develop a design for an image steganography algorithm of your choosing (that is, decide how to encode and decode a message in an image). Describe your algorithm design in a file called `steganography.txt`. (20%)
4. Implement the `encode` and `decode` methods of your design in python in a file called `steganography.py`. (40%)
5. Quantify how hard is to break your proposed steganographic scheme: how many operations, trials, minutes, hours, days, or years would it take an attacker to break it? What is the reason behind your assumption? (20%)

Part 2: Image Steganalysis (20%)

For this section, you need to design an algorithm for image steganalysis (an algorithm for detecting messages in steganographic images).
Follow the following steps:

1. Read this article (in particular the section on **Detecting Steganography)**: https://www.garykessler.net/library/fsc_stego.html
2. Come up with an algorithm for image steganalysis. For this attack, ~~you have 3 options:~~
    a. **Option 0 (recommended):** You may assume that you know the original carrier image and the steganographic image, and your goal is to retrieve the message.

b. ~~**Option 1 (potential final project):** You may assume that you know the message encoded, but you don't know the encoding algorithm, and your goal is to figure out where the bits of the message are placed in the image. In order words, your goal is first to reverse engineer the encoding algorithm you will use to decode messages from other steganographic images.~~

c. ~~**Option 2 (potential final project)**: You may assume that all you have is the steganographic image, and your goal is to recover the message.~~

Your steganalysis design needs to be "smarter" (or more comprehensive) than your `decode` from Part 1. Describe your design in a file called `steganalysis.txt`. (10%)

3. Implement the `analysis` method of your steganalysis attack in python in a file called `steganalysis.py`. (10%)

Your code from Part 2 will be used against images generated using Part 1 from all the students, plus 25% more steganographic images generated by the instructor. On Wednesday, 9/14 (or later?), as part of a class activity, you will run your steganalysis algorithm against the provided steganographic images to retrieve the messages hidden inside them (Remember to bring your laptops to class!).

Put all your files in a folder, compress the folder and then upload it to Moodle. Your submission needs to include:

1. A file called `steganography.txt` in which you explain your design and the reason why this design makes it harder for an attacker to detect it
2. A file called `steganalysis.txt` in which you explain your design for finding messages in images
3. `steganography.py` with your `encode/decode` code
4. `steganalysis.py` with your `analysis` code