# Final Project - Compression with Constraints: Steganography

# 9/8/2021

**Group member**

Chongwei Shi(21113454)

**Description of the project:**

The natural images are compressible, usually a full-size image of several megabytes can be reduced to kilobytes level easily without lossing too much important information. Such property has been used widely to denoising, deblur, etc. techniques.

The steganography is a topic lying in the cryptography. It is concealing a file, message, image, or video within another file, message, image, or video.

The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny. Plainly visible encrypted messages, no matter how unbreakable they are, arouse interest and may in themselves be incriminating in countries in which encryption is illegal.

Whereas cryptography is the practice of protecting the contents of a message alone, steganography is concerned both with concealing the fact that a secret message is being sent and its contents.

Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. For example, a sender might start with an innocuous image file and adjust the color of every hundredth pixel to correspond to a letter in the alphabet. The change is so subtle that someone who is not specifically looking for it is unlikely to notice the change.

In this project, we deal with a special case: stegranography with images only. So unlike many practical scenes, for instance, encrypt text, document in images, this task might not be able to produce perfect recovery of information.

**Descriptions of method and result:**

In this assignment, I perform the LSB and NN method for steganography and both methods are effective and inspiring.

First, I went through the LSB method and perform it on the example pictures. Since this method is quite simple and the code is friendly-reading, I finished it without difficulties.

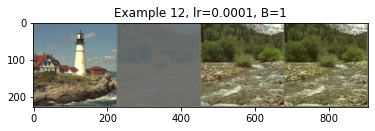
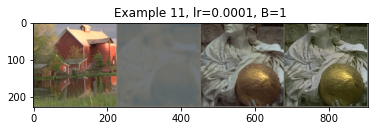
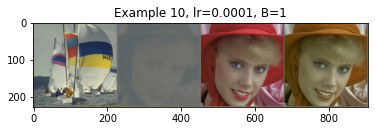
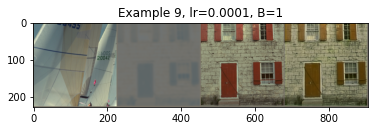
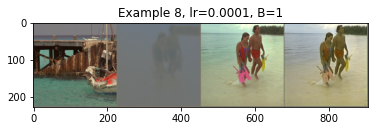
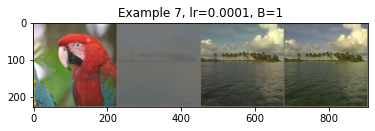
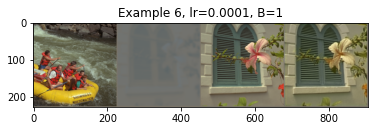
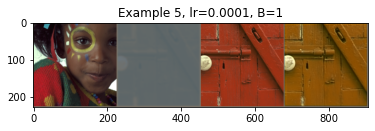
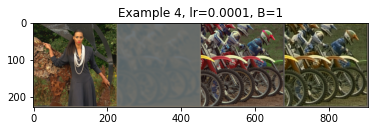
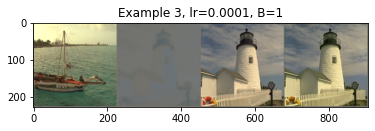
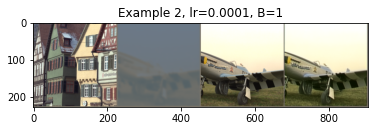
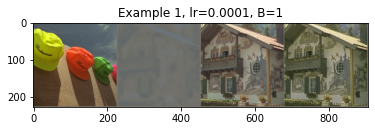
For the second task, I need to build up a neural network first which is quite demanding. Instructed by the given paper and coding scheme, I built the neural network step by step. For this task, I also choose the structure that divide the network into three parts: The preparing network, the hiding network and the revealing network and the Adam is chosen as the optimizer.

During the construction of the network and the relative coding scheme, the most annoying part is to match the data file to the loader of torch.nn. It really took me lots of time. In order to solve the problems caused by the object type occurring during the experiment, I went to the “torch” package and read thoroughly every embedded class to figure out how its functions work. After times of trying, I adjust the file folder to the support the network and finally got satisfying results.

Another difficulty is that the dataset collected from Kaggle is too large so it will take very long time to train the network. Therefore, in my experiment, I use sampling to sample 20% of pictures from each folder. In the end, even though the train data is reduced to around 450 pictures, it still takes 1.5 hour to train the network.

For the testing part, first we use neural network method to test the data.

We randomly select two pictures from the test data so that there will be 12 groups. Some results are as follows:



The average loss of test loss on the dataset is 0.69