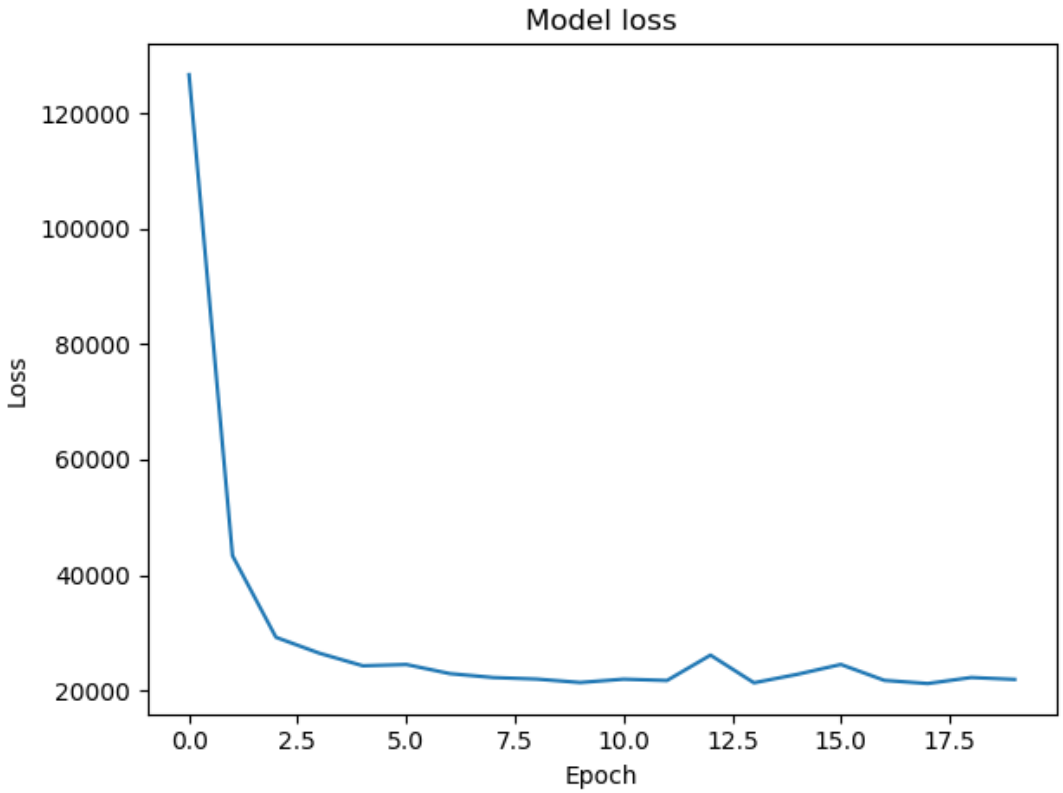
Steganography

First task:

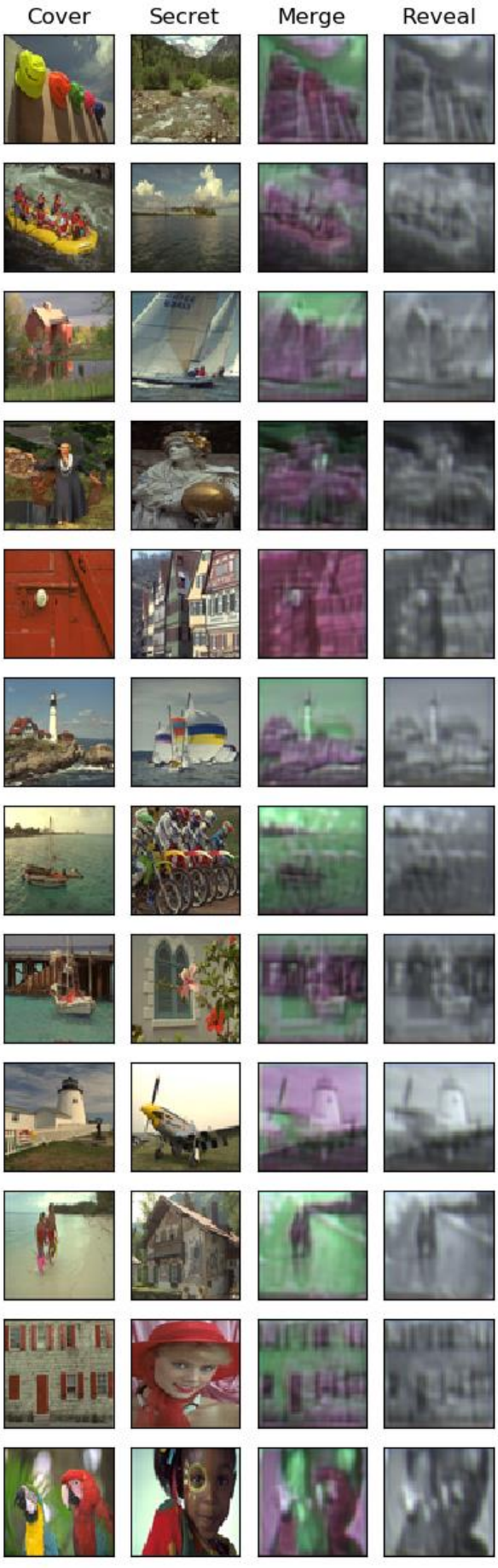
Implement LSB, the images (original and secret) are of size 64x64.

The main code work includes two function for data conversion (from integer to binary; from binary to integer), an encryption function, a decryption function, and the process of displaying the results on the test data set. There are 24 figures in the data set. Select 12 figures out of them randomly as original figures and the other 12 figures as secret figures. The result is shown as follows.

Second task:
Use neural network to approximate D and E. The main code work includes defining the NN model, including a preparation network, a hiding network, and a reveal network. The preparation network and the hiding network comprise the encryption model and the reveal network comprises the decryption model. The network structure is learned from https://github.com/alexandremuzio/deep-steg. I sampled a small portion of 400 images for training. The training process can finally work. However, the performance looks bad. The loss function decreases significantly in the first 20 training epochs shown as follows.



It can be seen that the merge images are close to the revealed images, which means that the decryption network may have bad performance. However, for most of the test images, the secret images are hidden, although the merge images look a little strange. I think the neural network has achieved its basic functionality, while it can be still further improved. The results are shown as follows.


Cover
Secret
Merge
Reveal