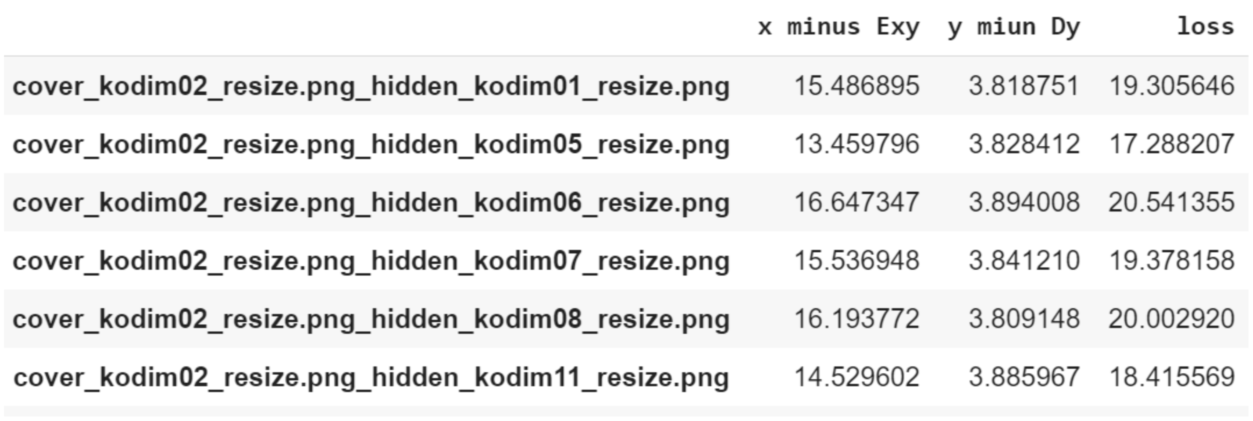
Report for Optional Task: Multi-image compress

Yaoqi Li, Yuxuan Wang, Gufeng Yang, Zarin Tasnim Ohiba

LSB

Overview

Recall the one-to-one LSB. We test our the dataset “Kodak Lossless True Color Image Suite” and resize to 64*64 to boost speed of processing. Based on the test result, the merged image and unmerged image are very identical to cover image and hidden image, which provides convincing approach to achieve steganography. Now we are considering 2 or more images merge steganography.



	x minus Exy	y miun Dy	loss
cover_kodim02_resize.png_hidden_kodim01_resize.png	15.486895	3.818751	19.305646
cover_kodim02_resize.png_hidden_kodim05_resize.png	13.459796	3.828412	17.288207
cover_kodim02_resize.png_hidden_kodim06_resize.png	16.647347	3.894008	20.541355
cover_kodim02_resize.png_hidden_kodim07_resize.png	15.536948	3.841210	19.378158
cover_kodim02_resize.png_hidden_kodim08_resize.png	16.193772	3.809148	20.002920
cover_kodim02_resize.png_hidden_kodim11_resize.png	14.529602	3.885967	18.415569

ave loss for merge image: 14.869128421059846
ave loss for unmerge image: 3.8857086277797603
ave loss for total loss: 18.754837048839605

(Data provided by Gufeng Yang)

Theoretical Analysis

We believe the LSB would not work for multi-image steganography but can only recover last hidden image we merge. The proof is quit simple. Since we only replace the last four digits for cover image, the value of each pixel from cover image will not change significantly no matter how many images we merge. However, when we merge 2nd time, the last 4 digits will be replaced by 2nd hidden image and all the information of 1st hidden will be lost and each pixel value will be “00000000” when we attempt to recover 1st hidden image.

Experiment

First we permute the dataset to 3 groups. Denote the “Cover_image” “hidden_image1” “hidden_image2” For each cover image we will try to cover all image from “hidden_image1”, and cover all image from “hidden_image2” after each time we cover image from “hidden_image1”. So for 24 images, the operations are 8*8*8 = 512

We redefine the loss function as:

$$f(x, y_1, y_2) = \|x - E(x, y_1, y_2)\| + \|y_2 - D(z_2)\| + \|y_1 - D(z_1)\|$$

x: cover image

y_1: first hidden image

y_2: second hidden image

E(x,y_1,y_2), Z_2: merged image

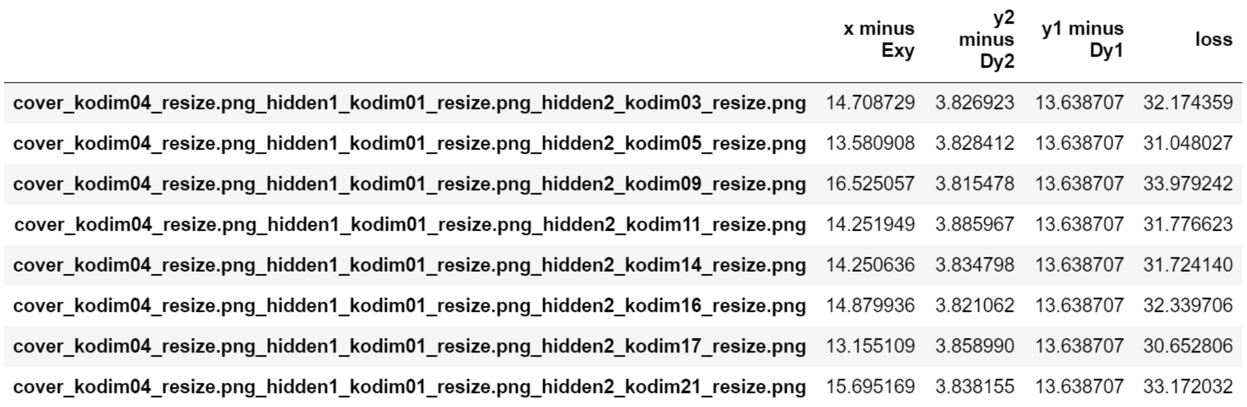
D(Z_2), Z1: first unmerged image

D(Z_2): second unmerged image

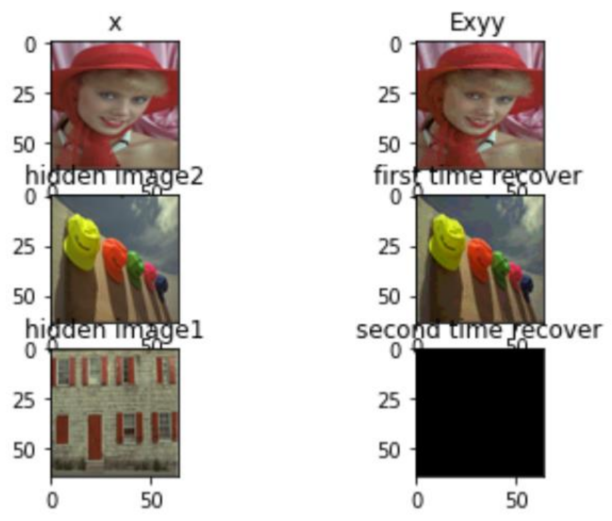
Result

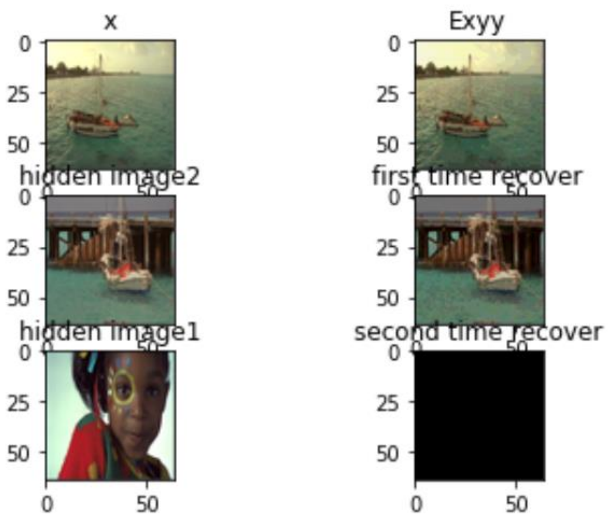
The result is quit same as expect. The quality of merged image which is E(x,y_1y_2) is identical to LSB one-to-one. The first time recover is rationally acceptable but 2nd time recover image is totally black, and loss value is significantly varied.

ave loss for merge image: 14.11166051627266
ave loss for the 1st unmerge image: 3.8387229025195153
ave loss for the 2nd unmerge image: 13.648226607807306
ave loss for total loss: 31.59861002659948



	x minus Exy	y2 minus Dy2	y1 minus Dy1	loss
cover_kodim04_resize.png_hidden1_kodim01_resize.png_hidden2_kodim03_resize.png	14.708729	3.826923	13.638707	32.174359
cover_kodim04_resize.png_hidden1_kodim01_resize.png_hidden2_kodim05_resize.png	13.580908	3.828412	13.638707	31.048027
cover_kodim04_resize.png_hidden1_kodim01_resize.png_hidden2_kodim09_resize.png	16.525057	3.815478	13.638707	33.979242
cover_kodim04_resize.png_hidden1_kodim01_resize.png_hidden2_kodim11_resize.png	14.251949	3.885967	13.638707	31.776623
cover_kodim04_resize.png_hidden1_kodim01_resize.png_hidden2_kodim14_resize.png	14.250636	3.834798	13.638707	31.724140
cover_kodim04_resize.png_hidden1_kodim01_resize.png_hidden2_kodim16_resize.png	14.879936	3.821062	13.638707	32.339706
cover_kodim04_resize.png_hidden1_kodim01_resize.png_hidden2_kodim17_resize.png	13.155109	3.858990	13.638707	30.652806
cover_kodim04_resize.png_hidden1_kodim01_resize.png_hidden2_kodim21_resize.png	15.695169	3.838155	13.638707	33.172032





(Data provided by Gufeng Yang)

Conclusion

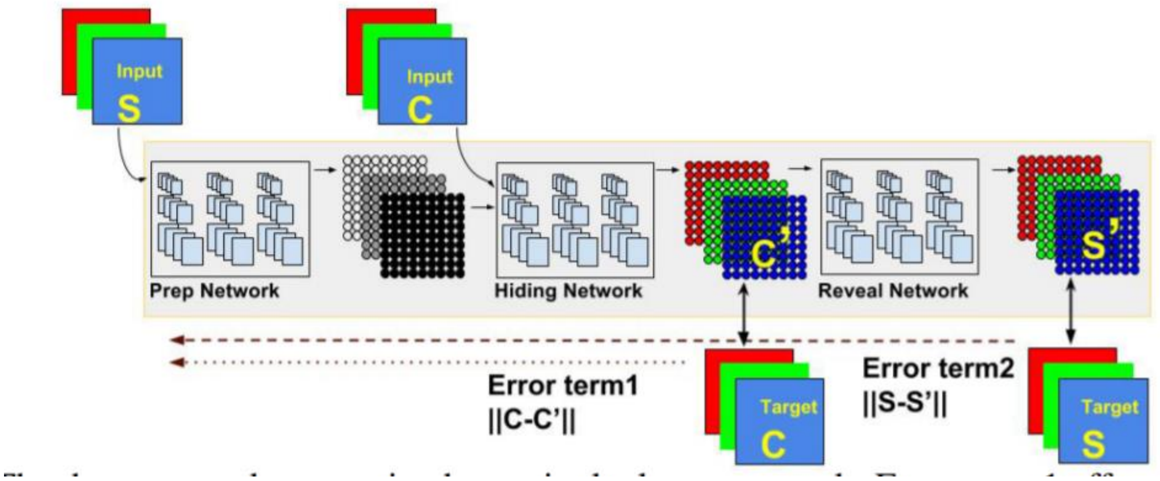
The LSB can achieve good quality of Steganography by one-to-one merge technique but fail to recover all images if we do multiple merging.

Discussion

There are couple approach to achieve multi-image merge. We consider hind images via different channel of color layer of cover image. However, we will lost the entire color layer information of cover image when we try to recover it. And Obviously, the loss value and noticeability will be varied. Another approach is using the bigger size cover photo so we have more space to store the information.

NN

Discussion: Since the time compactness and the coding difficulty. We only provide the hypothesis of the approach. We believe the NN would not recover the image. The key point it the image gradient of natural image is smooth, which means continues and differential for non-edged pixels. The CNN extract the partial information for image where CNN can observe the some statistical pattern of the gradient and even Hessian.



Suppose the S is not natural image, the C will merge with decoded layers of S which comes a image does not preserve the smooth property of gradient. The CNN will engage the difficulty of finding the pattern of the gradient.

One approach to this task is we might try to apply **Gaussian Filter** many times after we got each merged image. The Gaussian filter can reduce the gap in the gradient of image if we are choosing less variance, so we could obtain some the naturally-like image for hiding network processing. However, the gaussian filter will take the risk of losing the information of previous hidden images, so the recover processing would tough.

If we try to form the new problem, we would get the new optimization problem:

$$\min_{c\prime, s_0\prime, s_1\prime \ldots s_n\prime} L(c, s_0, s_1 \ldots s_n) = \|c - c\prime\| + \sum \gamma_i \|s_i - s_i\prime\|$$

Theoretically, the problem would potentially have more multiple optimizers when n is increasing. Meanwhile, even if we successfully to get the optimizer, it does not mean we could obtain least noticeability for each image. For example, for a pixel in covered image [1,1,1] and two different uncovered one [0,0,0] and [0,0,sprt(3)]. Even if they have same value for loss function, it’s hard to tell which one is more similar to the covered.

Therefore, we might need to carefully redefine the loss function(et. Like try first norm, infinity norm) , and apply more tools to play the combinatorics for the solver pipeline to achieve the multi-images merger steganography.