The introduction and simple Implementations of Some Applications About Image Encryption

YuHsuan Wang

***Abstract*—Nowadays, there are many information passing online through internet, including some private data that we don't want to easily share with other strangers. Although sending data by using internet since to be the most common, convenient, and necessary way, we can prevent the condition of easily-accessed and read other's data. Talking about data, there are many ways to store and present data such as text and image. In this article, we are going to further discuss one of the data types, image. Image is a easy way to present data to people watching through eyes, namely, it is easy that others may get the information or data reveal from the image. For example, the x-ray image is a patient's privacy data, and somebody can know that patient's health condition by just getting the information from the x-ray image. Another example is about copyright, others can directly download the image, which is of course made or designed by the author, and use it as if it was owned by them, and they are the original author. In this way they may confuse the user, causing some bad effect on author's copyright. So, the purpose of this article is to explain the method to hide the original image through implementing a project about some basic encryptions of image processing in several different ways. In this article, some results may be shown by a few of pictures that have been process before and after a program written in python language and compiled through "Spyder" IDE (Integrated Development Environment).**

I. Introduction

THE method to prevent image get by strangers easily is usually called "Image Encryption". This project may simply talk about the topics of image encryption, such as how we encrypt data into image, the hidden effectiveness, the applications of processed image, and so on. At the begin of the article, we will talk about the principle of encryptions on image and how we apply the technology actually. After introducing the background of image encryption, there will be several implementations of image encryption through different ways, such as naïve, XOR (Exclusive OR), changing mask, and so on. Not all of the method will be implemented in this project.

II. Encryption

Speaking of image encryption, it's obvious that the principle of how images will be processed into encrypted state, which means the image has been hidden, is important. Before understanding image encryptions, we should know some basic knowledge of image. First, we should know that images are stored in a 2 or 3 dimension-matrix which contains value between 0 and 255. Second, images can be classified into color images (3D matrix) and gray-scale images (2D matrix), color images usually have 3 layers of 2D matrix which represents in [Red, Green, Blue] color layer respectively; while gray-scale images have single layers to represent the gray color depend on the value between white (255) and black (0). Third, there are different ways and formula to convert matrix for special use in each kind of conditions, the one I will use in the project is the formula about how to convert RGB image into gray-scale image. Above three points are the basic knowledge of image, next we are going to talk about how we encrypt the image. Take gray-scale image for example, we can easily create a matrix called mask to modify the value in the image matrix or just directly change the value through special-designed algorithm. Two simple ways are direct addition of mask matrix and Exclusive or (XOR), we will introduce in the project. Through each kind of modifications, we will get different outcome effect. At last, there is still a problem about how we reverse the image into the original one remaining. We will talk about it in the next section.

III. Decryption and Retrieval

In the previous section, I have talked about some ways to encrypt data which we want to hide into another picture. Normally, we can use XOR again to get retrieval image, as for the addition of mask matrix (naïve process), we cannot just subject the mask matrix, there may be some problem. We will discuss int the next section. Sometimes, the picture we made will be pass through internet and this cause some concern to check if the picture or data is the one that has been processed by ourself. In this condition, we can try decryption to see if we can get data from retrieval image by using our mask or algorithm.

IV. Implementations

In this section, I am going to show some results of pictures being processed through my program.

A. Pre-processing and settings of this program

In this project, all the materials of images are from https://pixabay.com/ we aim to deal with a gray-scale image due to its convenience on dealing images. Different from the three dimensions in RGB image, which means three matrices, the gray-scale image is easily to be processed on its only matrix. After processing we will get a gray-scale format, if we want to reverse it back into original RGB format, we can use the formula below (Y represents value in gray scale):

$$Y = 0.299 * R + 0.587 * G + 0.114 * B. \quad (1)$$

$$U = -0.169 * R - 0.331 * G + 0.5 * B + 128. \quad (2)$$

$$Y = 0.5 * R - 0.419 * G + 0.081 * B + 128. \quad (3)$$

$$R = Y + 1.13983 * (V - 128). \quad (4)$$
$$G = Y - 0.39465 * (U - 128) - 0.58060 * (V - 128). \quad (5)$$
$$B = Y + 2.03211 * (U - 128). \quad (6)$$

$$(Y, U, V \in [0,255])$$

So, in the program of this project, I turned the input image into a gray-scale form, we can see the examples below:

Fig. 1. Original input image

Fig. 2. Gray-scale input image

B. Naïve process versus XOR process

The most straight forward or so-called naïve method to encrypt the image into a processed picture or data is to use an easy-designed matrix (mask matrix), such as a pure number matrix, which can add on or be subtracted by the matrix that stored the original input image. Due to the interaction of mask matrix and original matrix, we can get an easily processed matrix, and then we can decrypt the data from processed matrix.

There will be some problem if we use this kind of naïve process. The outcomes of values in the image are out of bound ([0, 255]) has been shown from Fig. 4. To Fig. 5.

Fig. 3. Naïve mask (all 128)

Fig. 4. Naive processed (original image + naïve mask)

Fig. 5. Naïve retrieval (masked image – naïve mask)

The reason that causing the wrong retrieval image is that when we directly use naïve mask, the value modified at the first time has been clipped into the maximum (255) or minimal (0) value boundary. For example, original value is 200, after adding 128 it should become 328, but the value will be clipped into 255, so when it subtracts to the naïve mask, we will get 127 (255-128) instead of 200. The above phenomenon is not so good for decryption, the data may be wrong to distinguish into the original image.

To solve the problem above we can use a technic, XOR. XOR is a logic operand which only presents TRUE when TRUE XOR FALSE. With the characteristic, it is a useful tool to

encrypt the image at the first time of XORing mask, and we can get original value by XORing mask again. There is a example showing below:

Fig. 6. original image XOR naïve mask

Fig. 7. Masked image XOR mask matrix

We can clearly see that the masked image is perfectly decrypted into the original image, so in this experience we know that XOR is a good tool as a mask in the convenience of reversing masked image into original image, it has no over boundary conditions we talked in naïve processing. Since it's a good tool, we will use this technic in all of the below implementations.

C. Use special image or certain picture as a mask

Now, we have a technic to process images, the next thing we can do is to find a better mask. Other than naïve mask, an easy way is to use another picture as the mask, in this way, others may not guess the mask by iterating value of pure number matrix. Below are the implementations of using picture as mask within XOR technic. Compare to the previous masked image, it is not that easy to guess all of the values in the image, except for the people knows which picture is used as the mask.

Fig. 8. Picture mask

Fig. 9. Original image XOR picture mask

Fig. 10. Retrieval image from picture masked image

But there are still some risks because if the picture used for mask can be searched online, we may still reveal the data to strangers. So, we can decrease the risk by designing our own image or watermark as a new mask, in this way the watermark can not only act as the symbol of us but also become a secret key that almost known by ourself only. We will implement watermark mask in the next section.

D. Self-Designed Special Watermark

We can identify them via the special water mark create by ourself in the use of mathematic equations. We can create our own water mark by using mathematic equations for designing

the format of water mark, of course we can also use picture or words as the water mark.

The self-designed water mark is formed by the following functions:

$$| y - x | \text{ or } | x - (w / 2) | \text{ or } | y - (w - x) | \text{ or } | (h - y) - x | \text{ or } | (h - y) - (w - x) | < 5 \quad (1)$$

$$| y - (h / 2) | < 30 \text{ and } | y - (h / 2) | > 10 \text{ and } | x - (w / 2) | < 30 \text{ and } | x - (w / 2) | > 10 \quad (2)$$

$$| y - (h / 2) | \text{ or } | y - (h / 3 | \text{ or } | y - 2 * (h / 3) | < 5 \quad (3)$$

(w : width of the graph; h : height of the graph)

The four inclined lines with width of 5 units are formed by function (1); the four squares with width of 20 is formed by function (2); and the three horizontal lines are formed by function (3).

The self-designed water mark is a method to encrypt graphs with the special picture that only known by the designer, which means only the designer can get the retrieval of encrypted images with XORing the water mark. It is a way to create secret keys for protecting encrypted images. Implementations are shown below:

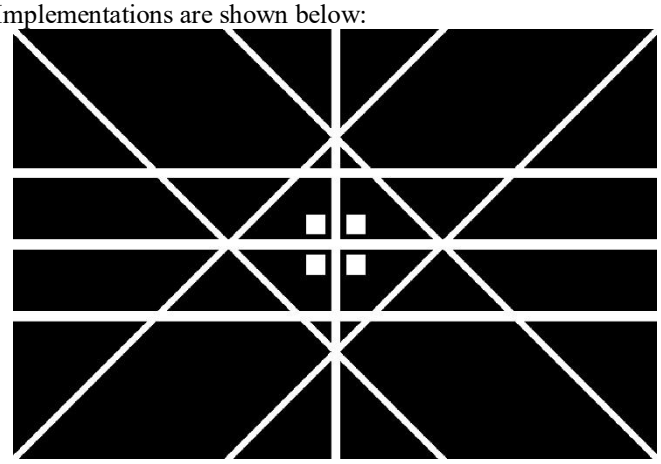

Fig. 11. Watermark mask

Fig. 12. Original image XOR watermark mask

Fig. 13. Retrieval image of watermark masked image

So far, we have found a useful technic to encrypt image and a way to design unique watermark. The next topic I am interesting is why all implementations above are so significance. We truly hide our image and reverse them, but the above ways are too easy to be find out there are some things hiding in the picture, although people may spend time to try the correct mask or key, they will finally get the data if the data is important. This is also a condition we, who owns the private data, do not expect. So, I searched 2 ways to blur the significance of processes on the image encryption. I will demonstrate in the next two section.

E. Use mosaic mask

The first technic is called mosaic, this method is to blur everything, so that people won't figure out it is a special image that encrypts data inside. In this way, the picture may be dressed up as background, and not easy to be found it strange. The mosaic mask can also be designed into big block or small block, here, I designed the mosaic mask in small block.

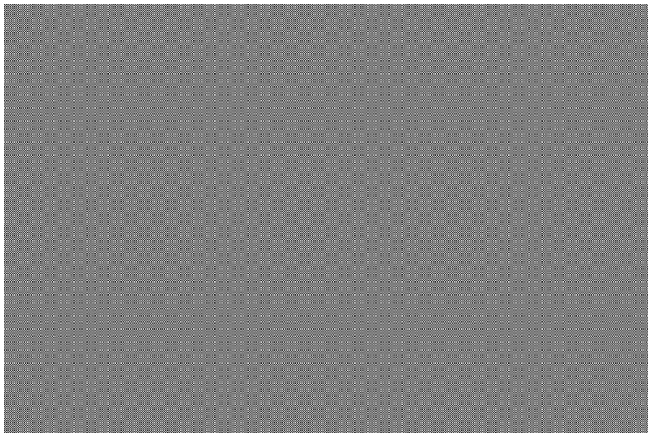

Fig. 14. Mosaic mask

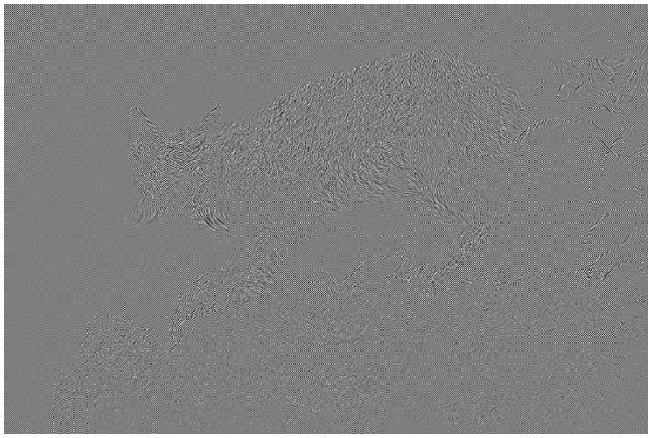

Fig. 15. Original image XOR mosaic mask

Fig. 16. Retrieval image of mosaic masked image

F. XOR process using LSB technic

In this section, we want to try another technic called LSB, Least Significant Bit, it is a common professional terminology in computer science, which means the right hand side bits of a binary number, those bits located at right hand side are called LSB, as for why they are least significance, it is because they are small numbers in that binary number for example, 2^0, 2^1, 2^2, and so on, in the image processing, the biggest number we usually used is 255 which stands for 8 bit unit control. 1, 2, 4 are relatively least significance to 255, and with the characteristic we can hide data by using LSB, it won't change too much for the image value, so that other people cannot figure this image's hidden data easily. They will just view it as the original image if they do not watch it clearly.

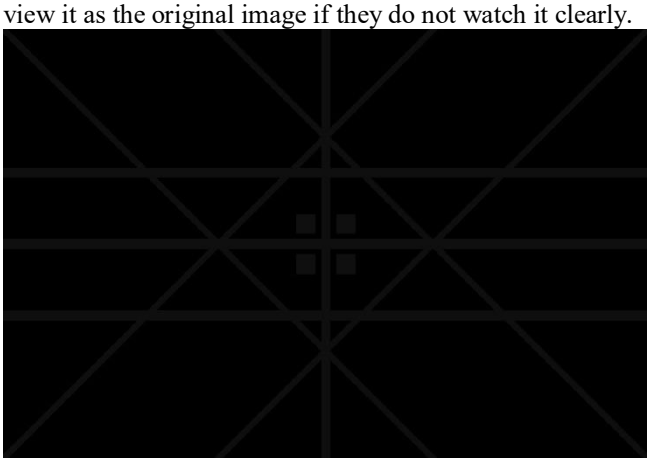

Fig. 17. LSB mask (15, 00001111 in binary)

Fig. 18. Original image XOR LSB mask

Fig. 19. Retrieval image of LSB-masked image

VI. Conclusion

In this final project, I truly searched data and learned a lot from image encryption, the topic is an interesting issue in the fast-changing of technology nowadays. I know there are still a lot of information and topics I didn't investigate into, and I still need to read and study them if I want to know more about this topic in image processing field.

Speaking of the implementations all above, I think there are many other fascinating technics to be figured out. The two most remain topic for me is about the algorithm to encrypt

image and one kind of encryption that can hide text into image in ascii code number form. Besides the topic of image processing, I also practiced my python language when doing this project, I searched many other functions that did not be taught in class. Compare with C, C++ language, python is really a good language that can implement functions in only a few codes or instructions because of its strong library system, and many complicate functions are packed into a single function.

At the end of this article, I want to say image processing is a large field, since I choose image encryption as the topic of this project, I may explore other topics of image processing in the future, trying each different topics can let me not only boost my program capability but also some math, logic in algorithm, or more background knowledge of image processing.

REFERENCES

[1] SRINAYAN, "IMPLEMENTATION OF VISUAL CRYPTOGRAPHY FOR COLOUR IMAGES", GITHUB, NOV , 2018, GITHUB - SRINAYAN/VISUALCRYPTOGRAPHY: IMPLEMENTATION OF VISUAL CRYPTOGRAPHY FOR COLOUR IMAGES

[2] 中央警察大學資訊密碼暨建構實驗室（ICCL）, "數位影像資訊隱藏技術利弊互見," Taiwan, ROC, Nov 2, 2011.

[3] 中央警察大學資訊密碼暨建構實驗室（ICCL）, 社團法人台灣 E 化資安分析管理協會, "視覺密碼學用肉眼解密　資安視覺系統體驗最直覺," Taiwan, ROC, Jan. 25, 2022.

[4] 社團法人台灣 E 化資安分析管理協會 國立屏東大學 AI 資訊安全與多媒體實驗室, "縮圖加密技術保護原始檔　便利瀏覽兼顧隱私防盜," Taiwan, ROC, JUL. 28, 2020

[5] Department of Information Management, National Central University. (2002, Aug. 26). *Young-Chang Hou, Visual cryptography for color images*. [Online]. Available: doi:10.1016/S0031-3203(02)00258-3 (psu.edu)

[6] Faculty of Computing and Informatics, Universiti Malaysia Sabah (UMS), Kota Kinabalu, Sabah 88400, Malaysia, Faculty of Information and Statistics, Guangxi University of Finance and Economics, Nanning, Guangxi 530003, China. (2021, Sep. 8). *WANG JI JUN AND TAN SOO FUN , (Member, IEEE), A New Image Encryption Algorithm Based on Single S-Box and Dynamic Encryption Step*. [Online]. Available: IEEE Xplore Full-Text PDF:

[7] Scientific reports, nature. (2020, Apr. 14) GuodongYe Kaixin Jiao , Xiaoling Huang, Bok-MinGoi & Wun-SheYap. *An image encryption scheme based on public key cryptosystem and quantum logistic map*. [online]. Available: s41598-020-78127-2.pdf

[8] JOURNAL OF COMPUTING, VOLUME 3, ISSUE 4, (APRIL 2011). Abdullah Bamatraf, Rosziati Ibrahim and Mohd. Najib Mohd. Salleh, *A New Digital Watermarking Algorithm Using Combination of Least Significant Bit (LSB) and Inverse Bit* [Online]. Available: Microsoft Word - A New Digital Watermarking Algorithm Using Combination of Least Significant Bit LSB and Inverse Bit (arxiv.org)