### Assignment 07: General Weighted Modulus Reversible Data Hiding (GMWRDH) Algorithm

授課教師：王宗銘

2023/12/10

1. 請撰寫1個**[嵌密**程式](學號-ass07-GMWRDH-MEA.py)。該程式以General Weighted Modulus Reversible Data Hiding (GWMRDH)演算法對單張原始灰階影像(Origin Grayscale Image)嵌入秘密訊息，產生三張嵌密灰階影像(Marked Grayscale Images)。

2. 請撰寫1個**[取密與回復]**程式(學號-ass07-GMWRDH-EXT.py)，練習以General Weighted Modulus Reversible Data Hiding (GWMRDH)演算法對三張嵌密灰階影像(Marked Grayscale Images)擷取秘密訊息後，回復產出單張原始灰階影像(Origin Grayscale Image)。若嵌密前事先位移像素，以防止pixel overflow/underlow，則回復後之影像(restored image)略異於原始影像，此可藉由計算MSE得知差異程度 詳見後續之敘述。

3. 使用或儲存結果之檔案目錄，共7個目錄，說明如下：

origin: 原始影像 (original image)

marked: 嵌密影像 (marked image)

rpatab: 嵌密參數檔 (RPA Table參數檔)

mesmea: 嵌入之祕密訊息 (embedding message)

mesext: 取密獲得之祕密訊息 (extracted message)

restor: 恢復之原始影像 (restored image)

imgres: 嵌密、取密、回復數據(quality result)

3.嵌密所需之參數紀錄於Reversible Pixel Alternation Table (RPA Table)，請讀取位於rpatab目錄下的RPA Table參數檔。此次嵌密提供4個RPA Table參數檔，分別嵌入M=12, 32, 50, 61進制秘密訊息，使用之嵌入權重embedding weigh記載檔案名稱之括號內。

(1). RPA Table參數檔名稱RPA\_3\_12\_(1\_3\_7)\_1.csv

N: no. of pixels in a cluster, N=3

M: M-ary number system, M=12

W: embedding weight, (1, 3, 7)

Z: the maximal pixel variation, Z=1

(2). RPA Table參數檔名稱RPA\_3\_32\_(1\_4\_14)\_2.csv

定義同上。

(3). RPA Table參數檔名稱RPA\_3\_50\_(1\_6\_43)\_2.csv

定義同上。

(4). RPA Table參數檔名稱RPA\_3\_61\_(1\_13\_47)\_3.csv

定義同上。

2. 嵌密時，請讀入RPA Table參數檔，自行建構嵌密所需之RPA Table。以RPA\_3\_12\_(1\_3\_7)\_1.csv為例，rows 1-17為檔案內容，黑線部分為嵌入訊息所需之RPA Table，如下所示。N=3, M=12, Z=1在第1列，W=(1, 3, 7)在第2列，RPA table在第3列以後。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| row 1 | RPA | 3 | 12 | w1 | w2 | w3 | 1 |
| row 2 | Index | d | SE | 1 | 3 | 7 |  |
| row 3 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| row 4 | 1 | 1 | 1 | 1 | 0 | 0 |  |
| row 5 | 2 | 2 | 2 | -1 | 1 | 0 |  |
| row 6 | 3 | 3 | 1 | 0 | 1 | 0 |  |
| row 7 | 4 | 4 | 2 | 0 | -1 | 1 |  |
| row 8 | 5 | 5 | 1 | 0 | 0 | -1 |  |
| row 9 | 6 | 6 | 2 | -1 | 0 | 1 |  |
| row 10 | 7 | 7 | 1 | 0 | 0 | 1 |  |
| row 11 | 8 | 8 | 2 | 0 | 1 | -1 |  |
| row 12 | 9 | 9 | 1 | 0 | -1 | 0 |  |
| row 13 | 10 | 10 | 2 | 1 | -1 | 0 |  |
| row 14 | 11 | 11 | 1 | -1 | 0 | 0 |  |
| row 15 |  | TSE | 16 |  |  |  |  |
| row 16 |  | MSE | 0.444444 |  |  |  |  |
| row 17 |  | PSNR | 51.65263 |  |  |  |  |

3. 嵌密程式: 學號-ass07-GMWRDH-MEA.py

輸入:

(1)位於origin 目錄的原始影像 (4張灰階)。

(2)位於rpatab目錄下的4個RPA Table參數檔，如下

RPA\_3\_12\_(1\_3\_7)\_1.csv

RPA\_3\_32\_(1\_4\_14)\_2.csv

RPA\_3\_50\_(1\_6\_43)\_2.csv

RPA\_3\_61\_(1\_13\_47)\_3.csv

(3)位於mesmea目錄的4個欲嵌入之秘密訊息檔案，如下。

mes\_mea\_1.txt

mes\_mea\_2.txt

mes\_mea\_3.txt

mes\_mea\_4.txt

假設訊息長度夠大，足以嵌入整張影像。請自行先以seed=100, 200, 300, 400 產生對應進制的秘密訊息.txt檔案。例如，若進制M=12，以seed=100產出一系列整數，並將之mod 12，即可產出12進制秘密訊息。訊息之間以空格分開。例如 11 5 4 2 8 9 10……。

輸出:

位於marked目錄下的4組已嵌密影像，每組有3張已嵌密影像，編號分別為I1, I2, I3, I4，共12張已嵌密影像。請自行使用RPA Table 參數檔配對原始影像，原則上1個原始影像使用1個RPA Table參數檔。

(2)輸出之已嵌密影像請加入“mark”與嵌密參數n, M, weight, Z，以茲識別。例如，若原始影像為Kodim07.png，使用RPA\_3\_12\_(1\_3\_7)\_1.csv參數檔，則嵌密影像檔名分別為

Kodim07\_mark\_N3\_M12\_1\_3\_7\_Z1\_I1.png

Kodim07\_mark\_N3\_M12\_1\_3\_7\_Z1\_I2.png

Kodim07\_mark\_N3\_M12\_1\_3\_7\_Z1\_I3.png

4. 取密與回復程式:學號-ass07-GMWRDH-EXT.py程式

輸入：

1. marked目錄下的4組嵌密影像，每組3張嵌密影像，編號分別為I1, I2, I3, I4，共12張嵌密影像。例如

Kodim07\_mark\_N3\_M12\_1\_3\_7\_Z1\_I1.png

Kodim07\_mark\_N3\_M12\_1\_3\_7\_Z1\_I2.png

Kodim07\_mark\_N3\_M12\_1\_3\_7\_Z1\_I3.png

(2) 位於rpatab目錄下的4個RPA Table參數檔，如下

RPA\_3\_12\_(1\_3\_7)\_1.csv

RPA\_3\_32\_(1\_4\_14)\_2.csv

RPA\_3\_50\_(1\_6\_43)\_2.csv

RPA\_3\_61\_(1\_13\_47)\_3.csv

輸出：

(1) 位於mesext目錄下的4個取出訊息檔案，如下

mes\_ext\_1.txt

mes\_ext\_2.txt

mes\_ext\_3.txt

mes\_ext\_4.txt

(2)位於restor目錄下的4個回復影像。請加入 “rest”與RPA Table參數檔資訊，例如

Kodim07\_rest\_N3\_M12\_1\_3\_7\_Z1.png

(3)於imgres 目錄下的嵌密與取密結果csv檔案。請加入 “qualit”與RPA Table參數檔資訊，例如Kodim07\_qualit\_N3\_M12\_1\_3\_7\_Z1.csv。其內容如下：其中

第1列、第2列同於解密參數。

第3列：計算MSE，代表原始影像(Kodim07)與回復影像(Kodim07\_rest\_N3\_M12\_1\_3\_7\_Z1.png) 之差異，以mean square error量化之。請注意，若原始影像有經過pixel overflow/underflow 調整，MSE0。若無任何像素有先調整，則MSE=0。

第4列：計算PSNR。PSRN=10\*log10(255\*255/MSE)。

若MSE=0, PSNR=infinity。PSRN取小數2位，第3位四捨五入。

第5列：計算嵌入量(EC)。，HV代表Kodim07.png影像之水平(H)與垂直(V)解析度。PSRN取整數，小數第1位四捨五入。

第6列：計算嵌入率(ER)。。ER取小數5位，第6位四捨五入。

範例之Kodim07\_qualit\_N3\_M12\_1\_3\_7\_Z1.csv

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| row 1 | RPA | 3 | 12 | w1 | w2 | w3 | 1 |
| row 2 | Index | d | SE | 1 | 3 | 7 |  |
| row 3 | MSE | 0.002 |  |  |  |  |  |
| row 4 | PSNR | 75.13 |  |  |  |  |  |
| row 5 | EC | 1409664 |  |  |  |  |  |
| row 6 | ER | 1.19499 |  |  |  |  |  |

5. 撰寫之程式：

(1)請注意使用課程投影片內建議方法，來解決pixel overflow或pixel underflow之問題。

(2)嵌入後，建議可立即使用課程投影片的Message Extraction數學式，驗證秘密訊息擷取之正確性。

(4)可以使用openCV套件。

(5) python版本>=3.10，請確認程式在IDLE python 64 bit是可執行的。

(6) 請遵守檔案編號原則，以免助教判定繳交格式錯誤，導致錯誤執行，不予評分。

6.提供：作業提供已規劃建構之7個目錄。目錄內之影像隨舉例影像，並非嵌密、取密回復之結果。

7.繳交: 請繳交壓縮檔案，壓縮方式請選zip或rar。

壓縮檔案名稱: 學號-ass07.rar，包含下列2個程式、7個目錄、1個readme.txt

1. 1個嵌密程式(學號-ass07-GMWRDH-MEA.py)，此程式可由Assignment 06修改之。
2. 1個取密與回復程式(學號-ass07-GMWRDH-EXT.py)
3. 7個目錄

origin: 原始影像 (original image)

marked: 嵌密影像 (marked image)

rpatab: 嵌密參數檔 (RPA Table參數檔)

mesmea: 嵌密使用之祕密訊息 (embedding message)

mesext: 取密得出之祕密訊息 (extracted message)

restor: 恢復之原始影像 (restored image)

imgres: 嵌密、取密、回復結果 (embedding, extraction, image quality result)

1. 1個readme.txt，請放在與python程式同目錄層，敘述如何執行python程式，載明是否需要額外的套件。(請提供)