

به نام خدا

پروژه ی 2

محمدرضا علوی – شماره ی دانشجویی: 810100253

بخش اول:

برای پنهان کردن پیام پیام را به معادل بانری آن با کمک مپ ست تبدیل می کنیم و یک رشته از صفر و یک درست می کنیم سپس مقدار هر پیکسل را به باینری تبدیل می کنیم و بیت کم ارزش هر پیکسل را معادل یکی از اعداد رشته قرار می دهیم. برای پیدا کردن پیام همین کار را به صورت برعکس انجام می دهیم.

تمرین 1-3: در نگاه اول خیر زیرا ما مقدار هر پیکسل را در بیشترین حالت یک واحد تغییر می دهیم (بیت کم ارزش را تغییر می دهیم) و این تغییر خاصی در تصویر ایجاد نمی کند.

تمرین 1-5: خیر چون رمزگذاری به مقدار هر پیکسل بستگی دارد و نویز می تواند مقدار پیکسل را عوض کند و اگر فقط مقدار یک پیکسل یک واحد تغییر کند می تواند باعث شود پیام عوض شود.

تمرین 1-6: بله در این رمز گذاری انتروپی تغییر می کند. تصاویر یک میزان نویز مشخص دارند ولی وقتی رمزگذاری می شود نویز جدید و غیر طبیعی اضافه می شود و انتروپی بالا می رود. با استفاده از χ^2 می توان فهمید. با نمودار هیستوگرام می توان فهمید.

بخش دوم:

برای سنتز سیگنال هر شماره را به دست می آوریم سپس با کاکت کردن سیگنال شماره ها سیگنال نهایی را به دست می آوریم. برای آنالیز از سیگنال نهایی به اندازه ی یک شماره نمونه برداری می کنیم و سیگنال شماره ها کورولیشن می گیریم شماره برابر بیشترین کورولیشن است و این کار را برای همه ی نمونه ها انجام می دهیم و عدد به دست می آید.

تمرین 2-2: عدد 810198 است.

بخش سوم:

ابتدا عکس مدار و آی سی را سیاه و سفید می کنیم و عکس آی سی را کپی می کنیم و 180 درجه می چرخانیم. تابع corr1 را این جوری تعریف می کنیم که عکس آی سی را روی عکس مدار حرکت می دهیم و هر دفعه با آن بخش از عکس مدار که روی آن است با corr2 کورولیشن می گیریم و آن را در یک آرایه ذخیره

می کنیم. تابع `corr2` را مطابق با فرمول گفته شده تعریف می کنیم. این کار را یک بار برای خود تصویر و یک بار برای تصویر چرخیده انجام می دهیم. سپس برای نتایج کورولیشن ترشولد 0.9 تعریف می کنیم و ردیف ها و ستون هایی که مقدار های آن که بیشتر از آن باشند را پیدا می کنیم. جایگاه های مستطیل برابر با ردیف ها و ستون های آن ها با طول و عرض عکس آی سی است. روی ردیف های حلقه می زنیم و مستطیل هایی در ردیف و ستون آن ها و با اندازه های طول و عرض عکس آی سی می کشیم.

```
1 - clc;
2 - close all;
3 - clear;
4
5 - image1 = imread ('PCB.jpg');
6 - image2 = imread ('IC.png');
7
8 - image1_gray = rgb2gray (image1);
9 - image2 = rgb2gray (image2);
10
11 - image3 = imrotate (image2, 180);
12
13
14 - result1 = corr1 (image1_gray, image2);
15 - result2 = corr1 (image1_gray, image3);
16
17 - [r, c, a] = size (image2);
18
19 - imshow (image1)
20 - hold on;
21 - [rows, cols] = find (result1 > 0.9);
22 - for i = 1:length (rows)
23 -     rectangle ('Position', [cols(i) rows(i) c r], 'Edgecolor', 'b')
24 - end
25
26 - [rows, cols] = find (result2 > 0.9);
27 - for i = 1:length (rows)
28 -     rectangle ('Position', [cols(i) rows(i) c r], 'Edgecolor', 'b')
29 - end
```

تصویر 1 - کد بخش 3

```

1  function result = corr1 (image1, image2)
2      image1 = double (image1);
3      image2 = double (image2);
4
5      s1 = size (image1);
6      r1 = s1(1);
7      c1 = s1(2);
8
9      s2 = size (image2);
10     r2 = s2(1);
11     c2 = s2(2);
12
13     r3 = r1 - r2 + 1;
14     c3 = c1 - c2 + 1;
15
16     result = zeros (r3, c3);
17
18     for i = 1:r3
19         for j = 1:c3
20             image3 = double (image1 (i:i + r2 - 1, j:j + c2 - 1));
21             result(i,j) = corr2 (image2, image3);
22         end
23     end
24 end

```

تصویر 2 - کد تابع corr1

```

1  function result = corr2 (x, y)
2      result = sum (sum (x.*y)) / sqrt (sum (sum (x.*x)) * sum (sum (y.*y)));
3  end

```

تصویر 3 - عکس کد تابع corr2