بسم الله الرحمن الرحیم

دانشکده مهندسی کامپیوتر

استاد : آقای دکتر ابوالفضل دیانت

- امیرحسین مجتهدی- سید مهدی رضوی

اسفند ۱۴۰۱

فهرست مطالب

فهرست تصاویر

۱ تمرین اول

پیاده‌سازی یک روش نهان‌کاوی به عنوان آشکارسازی بر نهان‌نگاری به روش LSB .
یعنی فرض‌کنید که ما با روش LSB عملیات نهان‌نگاری را انجام دادیم شما باید یک روش نهان‌کاوی به منظور تشخیص آن پیاده‌سازی کنید.

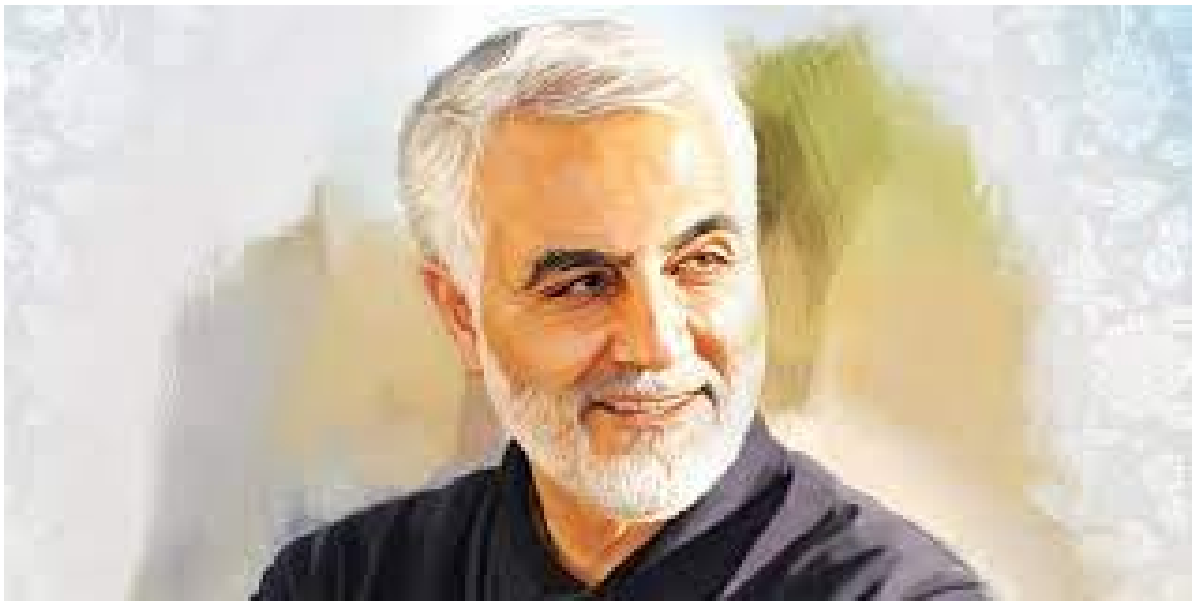
طبق الگوریتم این روش نهان‌نگاری ابتدا تک تک بیت های پیام رمز CipherText را در کم ارزش‌ترین بیت های پیکسل های موجود در عکس قرارمی‌دهیم.
ابتدا باید پیام مدنظر خود را که در زیر آمده‌است را باید در تصویر پنهان کنیم. در کد پایتون زیر این عملیات را با استفاده از عملگرهای بیتی انجام داده ایم.
Shahid Ghasem Soleimani

```
with Image.open("shahid_soleimani.jpeg") as img:
    width, height = img.size
    print(f'width , height : {width , height}')

    for x in range(0, width):
        for y in range(0, height):
            pixel = list(img.getpixel((x, y)))
            for n in range(0 , 3):
                if(index < limit):
                    # print(b_message[index])
                    before = bin(pixel[n])

                    # if index == 0:
                    #     print(pixel[n])

                    pixel[n] = ( (pixel[n] & ~1) | int(b_message[index]) )
                    # print(pixel[n])


                    after = bin(pixel[n])

                    index = index + 1

            img.putpixel((x,y), tuple(pixel))
    img.save("steganoGraphed.jpeg")
```

همانطور که در کد بالا ، که برای پنهان‌سازی پیام می‌باشد ، مشاهده‌می‌کنید : برای جاگذاری بیت‌های پیام به جای کم‌ارزش‌ترین بیت هر پیکسل ، ابتدا می‌بایستی بیت‌آخر هر پیکسل را نابود کنیم.
یا به عبارت بهتر ، بیت آخر هر پیکسل را صفر قرار دهیم. این کار با یک عملیات `AND` منطقی صورت می‌پذیرد.
(`pixel[n] and 1`)
سپس با عملیات `OR` منطقی ، بیت پیام را در جایگاه مدنظر قرارخواهیم داد.
و همینطور بالعکس ، برای آشکارسازی این پیام نهفته‌شده می‌بایستی که ابتدا تمام بیت‌های کم‌ارزش را استخراج کنیم.
سپس تا زمانی که به رشته `Tamam` برخورد نکرده‌ایم ، این مجموعه رشته باینری را به کداسکی متناظر تبدیل میکنیم.
نکته مهم در اینجا این است که طبیعتا تعداد پیکسل‌های تصویر ما بسیار بیشتر از تعداد بیت‌های پیام ما خواهد بود.
برای این منظور باید به گیرنده، با یک پیام مشخص ،انتهای پیام خود را نشان‌دهیم.
این پیام در این تمرین `Tamam` خواهد بود.

```
with Image.open("steganoGraphed.jpeg") as img2 :
  width , height = img2.size

  for x in range(0 , width):
    for y in range(0 , height):
        pixel2 = list(img2.getpixel((x, y)))

        for n in range(0 , 3):
            # if x < limit:
              data2 += str(pixel2[n] & 1)
              # x = x + 1
              # print(pixel[n] - pixel2[n])


for i in range(0 , len(data2) , 8):
  if tmp[-5 : ] == 'Tamam':
    print('Breaked Breaked')
    break

  letter = data2[i : i + 8]

  # print(f'word : {chr(int(str(letter), 2))}')
  tmp += chr(int(str(letter), 2))

print(f'tmp : {tmp[0 : len(tmp) - 5]}')
```

شکل ۱: پنهان‌کردن و آشکارکردن پیام Shahid Ghasem Soleimani



شکل ۲: تصویری که بر روی آن steganography را انجام داده‌ایم

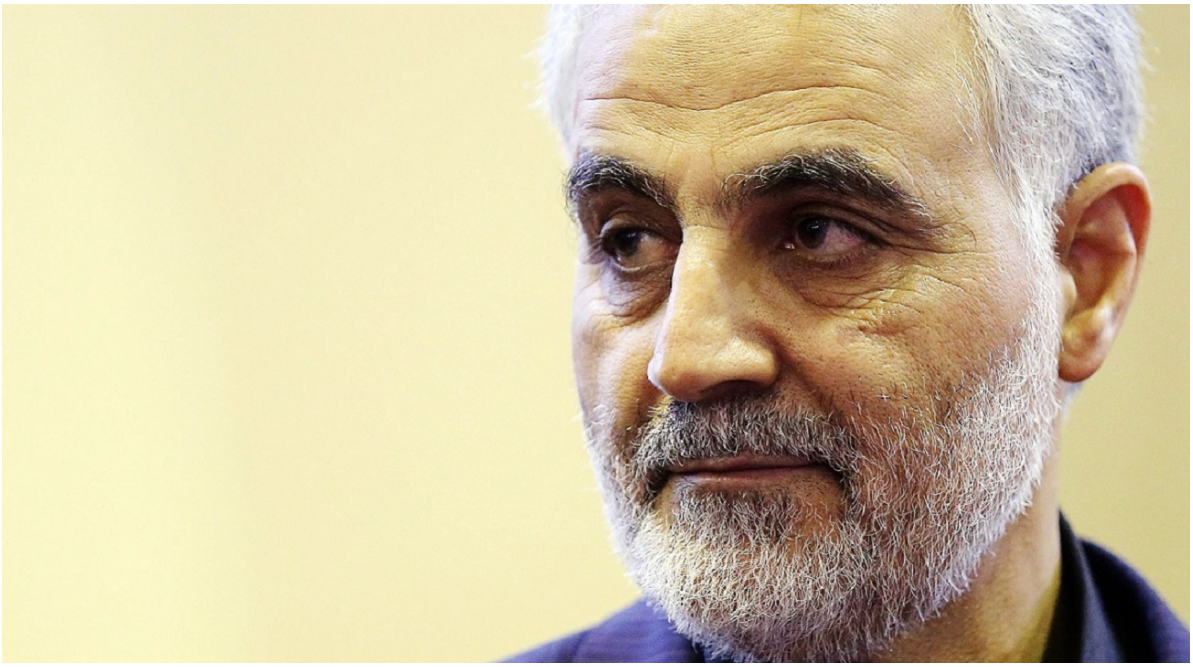
۲ تمرین دوم

پیاده‌سازی یک روش نشان‌گذاری از نوع Visible و شکننده.
البته بدیهی است که ابتدا باید تحقیق کنید و یک روش نشان‌گذاری از نوع Visible و شکننده را پیدا کنید و سپس آن را پیاده سازی کنید.
پیاده‌سازی باید به زبان‌های C++ یا Python باشد.

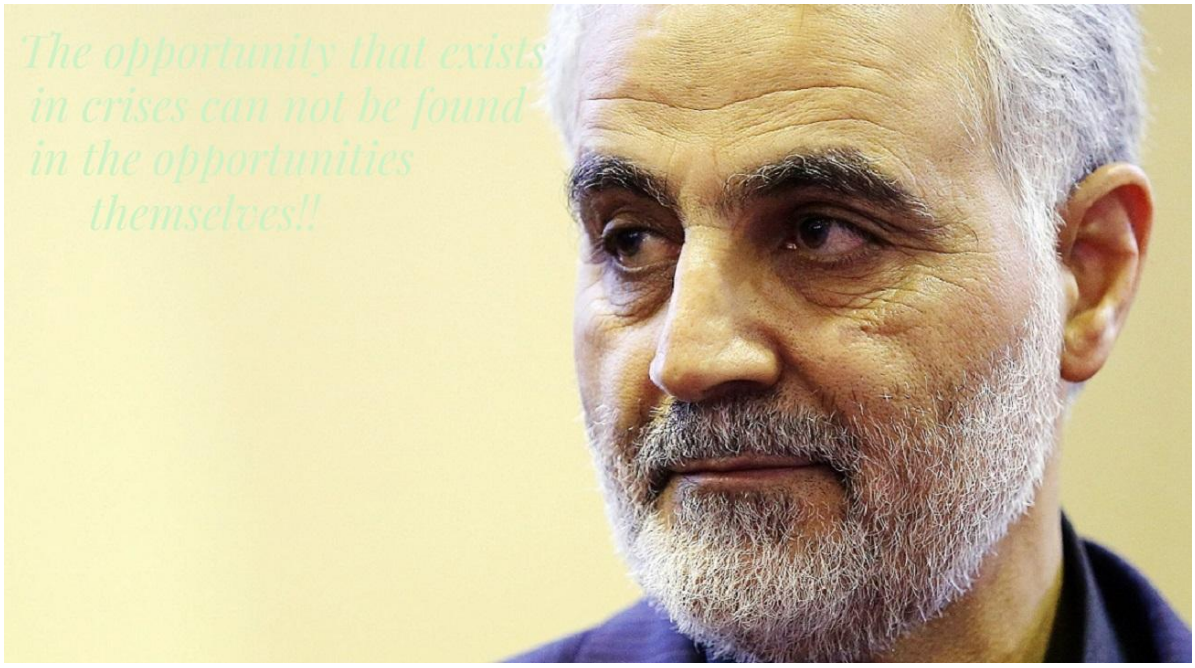
ما برای ایجاد یک watermark بر روی تصویر شهید حاج قاسم سلیمانی از کتابخانه معروف پردازش تصویر پایتون استفاده کردیم.
ما از کتابخانه PIL برای نوشتن متن مدنظر برای watermark شدن استفاده خواهیم کرد.
جمله‌ی مدنظرما برای انتقال به وسیله این تصویر

The opportunity that exists
in crises can not be found
in the opportunities
themselves !! می‌باشد.

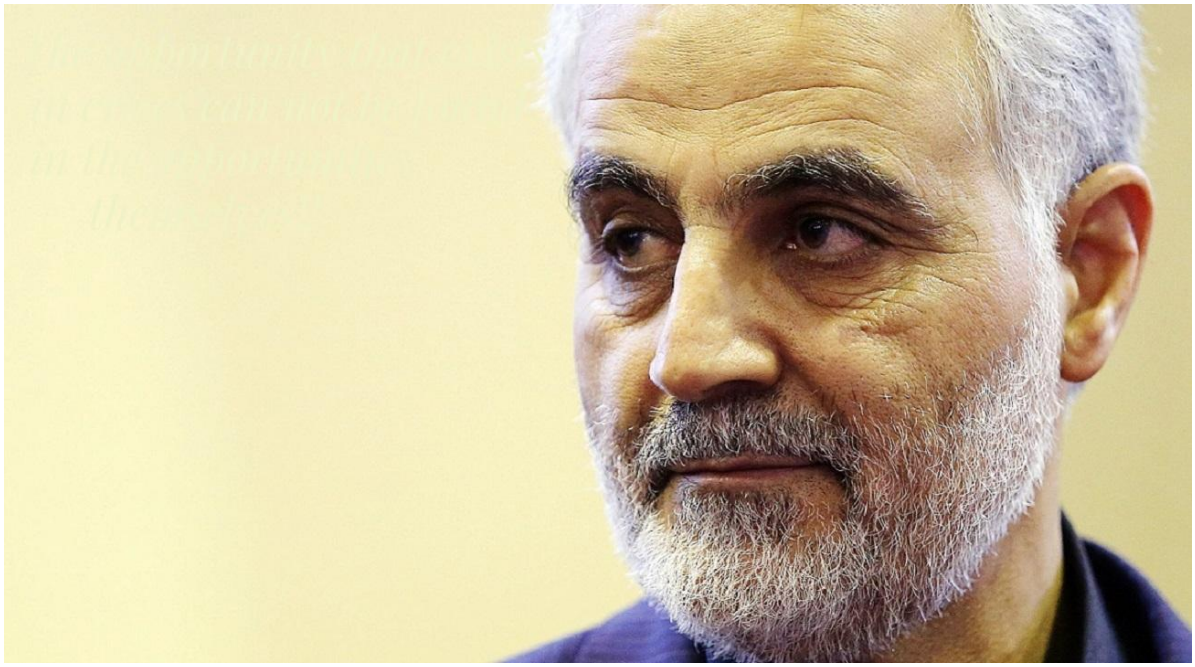
که ترجمه جمله معروف ایشان مبنی بر فرصتی که در بحران‌ها هست در خود فرصت‌ها نیست.



شکل ۳: تصویر بدون watermark



شکل ۴: تصویر با شفافیت بیشتر watermark



شکل ۵: تصویر با شفافیت کمتر watermark

کد مربوط به این watermark ها را در زیر مشاهده خواهید کرد.

```
from PIL import Image, ImageFont, ImageDraw

my_image = Image.open("sardar2.jpg")

title_font = ImageFont.truetype('Playfair_Display/IranNastaliq.ttf', 100)
title_font2 =
    ImageFont.truetype('Playfair_Display/PlayfairDisplay-Italic-VariableFont_wght.ttf'
    , 50)

title_text2 = 'The opportunity that exists \n in crises can not be found \n in the
    opportunities \n      themselves!!'

image_editable = ImageDraw.Draw(my_image)
image_editable.text((15,15), title_text2, (207,243,199), font=title_font2)

my_image.save("result.jpg")
```