

Curso 2020/21

La Magia de Ocultar Mensajes



Ismael Herrera Gutiérrez

Francisco Javier Álvarez Jiménez

Jorge Miguel Tenorio Rodríguez

Manuel Nogues Casas

1º GS Desarrollo de Aplicaciones

Multiplataforma

3/12/2020

Índice

Materiales Consultados.....	2
Esquema de Investigación.....	3
Lista de Conocimientos Obtenidos.....	7

Materiales Consultados

<https://stackoverflow.com/es/q/5131991>

En esta página consultamos la forma de convertir un mensaje de texto en binario para poder insertarlo en la imagen posteriormente.

<https://recursospython.com/guias-y-manuales/anadir-texto-imagen-pillow/>

Esta página nos ayudó a generar el código para insertar el texto ya en binario dentro de la imagen en concreto para así esconderla dentro de esta.

<https://parzibyte.me/blog/2018/04/06/esteganografia-python-imagenes-lsb-introduccion/>

Aquí consultamos como obtener el mensaje guardado dentro de una foto, y además encontramos información interesante con la que mejorar los códigos anteriores.

<https://es.stackoverflow.com/questions/349957/convertir-una-imagen-a-escala-de-grises-pero-conservando-los-canales-python>

Por último, en este blog de preguntas encontramos justo lo que necesitábamos para pasar la foto en cuestión a una escala de grises. Nos sirvió de gran ayuda.

Esquema de Investigación

En primer lugar, importamos las librerías necesarias para que funcione el programa:

```
import cv2
import numpy as np
import types
```

Una vez hecho esto implementamos funciones para cada apartado del ejercicio.

Después definimos la función para pasar el mensaje a binario:

```
def mensaje_a_binario(mensaje_1):
    if type(mensaje_1) == str:
        return ''.join([format(ord(i), "08b") for i in mensaje_1])
    elif type(mensaje_1) == bytes or type(mensaje_1) == np.ndarray:
        return [format(i, "08b") for i in mensaje_1]
    elif type(mensaje_1) == int or type(mensaje_1) == np.uint8:
        return format(mensaje_1, "08b")
    else:
        raise TypeError("Tipo de dato no soportado")
```

Una vez hecho esto, implementamos funciones para guardar el mensaje en la foto en cuestión y para obtener dicho mensaje que servirán como soporte:

```
def esconder_datos(imagen, mensaje_secreto):
    n_bytes = imagen.shape[0] * imagen.shape[1] * 3 // 8

    if len(mensaje_secreto) > n_bytes:
        raise ValueError("La imagen no es válida")

    mensaje_secreto += "#####"

    indice_datos = 0
    binary_secret_msg = mensaje_a_binario(mensaje_secreto)

    data_len = len(binary_secret_msg)
    for valores in imagen:
        for pixel in valores:
            r, g, b = mensaje_a_binario(pixel)
            if indice_datos < data_len:
                pixel[0] = int(r[-1]) + binary_secret_msg[indice_datos, 2]
                indice_datos += 1
            if indice_datos < data_len:
                pixel[1] = int(g[-1]) + binary_secret_msg[indice_datos, 2]
                indice_datos += 1
            if indice_datos < data_len:
                pixel[2] = int(b[-1]) + binary_secret_msg[indice_datos, 2]
                indice_datos += 1
            if indice_datos >= data_len:
                break
    return imagen
```

A continuación, enseñamos los datos de la imagen

```
def enseñar_datos(imagen):
    datos_binarios = ""
    for valores in imagen:
        for pixel in valores:
            r, g, b = mensaje_a_binario(pixel)
            datos_binarios += r[-1]
            datos_binarios += g[-1]
            datos_binarios += b[-1]
    todos_los_bytes = [datos_binarios[i: i + 8] for i in range(0, len(datos_binarios), 8)]
    datos_decodeados = ""
    for byte in todos_los_bytes:
        datos_decodeados += chr(int(byte, 2))
        if datos_decodeados[-5:] == "####":
            break
    return datos_decodeados[:-5]
```

Posterior a esto, definimos la función que obtiene el texto a guardar que ingresa el usuario y se guarda en la foto.

```
def Ingresar_Mensaje():
    nombre_imagen = "proyimag1T.png"
    imagen = cv2.imread(".\proyimag1T.png")
    tamaño = imagen.shape
    altura = tamaño[0]
    anchura = tamaño[1]

    print(f"{nombre_imagen} tiene de {anchura} de ancho y de {altura} de alto.")
    cv2.imshow('Imagen original', imagen)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
    print()
    Mensaje = input("Introduzca el mensaje de texto a ocultar: ")
    if (len(Mensaje) == 0):
        raise ValueError('No hay ningún mensaje')
    print()

    archivo = "proyimod1T.png"
    imagen_encodeada = esconder_datos(imagen, Mensaje)
    cv2.imwrite(archivo, imagen_encodeada)
    print("Insertando texto en la imagen...")
    print()
    print(f"El fichero {nombre_imagen} es diferente a {archivo}")
    print()
    imagen2 = cv2.imread(".\proyimod1T.png")
    cv2.imshow('Con texto oculto', imagen2)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

Añadiremos otra para recuperar dicho mensaje, usando funciones anteriores como soporte:

```
def Extraer_Mensaje():  
    archivo = "proyimod1T.png"  
    datos_imagen = cv2.imread(archivo, 1)  
    print("El fichero de la imagen con texto oculto se llama: " + archivo)  
    print()  
    text = enseñar_datos(datos_imagen)  
    return text
```

Ahora se realizan las funciones para pasar la foto a escala de grises.

```
def Escala_Grises():  
    archivo = 'proyimag1T.png'  
    print("El fichero de la imagen se llama: " + archivo)  
    imagen = cv2.imread(archivo)  
    imagen_gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)  
    imagen_gris = cv2.cvtColor(imagen_gris, cv2.COLOR_GRAY2RGB)  
    cv2.imwrite('proyimgr1T.png', imagen_gris)  
    cv2.imshow('Escala de grises', imagen_gris)  
    cv2.waitKey(0)  
    cv2.destroyAllWindows()
```

Y crearemos además la función del menú para salir del programa:

```
def Salir():  
    print("Gracias por usar el programa!")  
    print()  
    print("Tenga un buen día!")
```

Por último, creamos el bucle que nos mantiene dentro del programa hasta que el usuario desee salir de él, para ello se crea un menú:

```
def Menu():
    while True:
        print("\033[1m\033[4mAPLICACIÓN ESTEGANO\033[0m\033[0m".center(100, " ")), end="\n")
        print()
        a = input(
            "1. Insertar mensaje oculto en una imagen \n2. Extraer mensaje oculto de una imagen \n3. Convertir la imagen a escala de grises \n4. Salir \nOpción: ")
        usuario = int(a)
        if usuario == 1:
            print()
            print("OPCIÓN: Insertar mensaje oculto en una imagen")
            print()
            Ingresar_Mensaje()

        elif usuario == 2:
            print()
            print("OPCIÓN: Extraer mensaje de una imagen")
            print()
            print("Extrayendo el texto de la imagen")
            print()
            print()
            print("El texto oculto es: " + Extraer_Mensaje())
            print()

        elif usuario == 3:
            print("\nOPCIÓN: Convertir la imagen a escala de grises")
            print()
            Escala_Grises()
            print()
            print("Convirtiendo la imagen a escala de grises...")
            print()

        elif usuario == 4:
            print()
            Salir()
            break
        else:
            print()
            print("Opción no válida")
            print()

Menu()
```

Looks like you're using NumPy
Would you like to turn scientific mode on?

Lista de Conocimientos Obtenidos

- Pasar a binario una cadena de texto
- Insertar una cadena de texto en una imagen para ocultarlo
- Como mostrar una imagen en una ventana con un nombre concreto
- Obtener el texto guardado en una imagen que está oculto
- Pasar una foto a escala de grises a partir de código con Python
- Trabajo en equipo y compenetración con los compañeros
- Administración del tiempo del que se dispone para la realización de los trabajos dentro de los plazos de entrega