

CC5508: Procesamiento y Análisis de Imágenes

Tarea 1: Esteganografía

Profs. José M. Saavedra

Agosto 2019

1. Objetivo

El objetivo de esta tarea es familiarizarse con la manipulación de imágenes a través de algoritmos básicos de ocultamiento de datos (esteganografía).

2. Descripción

Se llama **Esteganografía** a la práctica de encubrir datos (texto, imagen, videos, etc) dentro de un archivo. Particularmente, nosotros estamos interesados en la esteganografía aplicada sobre imágenes. En este sentido, la tarea consiste en ocultar un texto determinado dentro de una imagen. Para este fin, cada símbolo del texto se codifica (e.g. ASCII) y el código se guarda en los N bits menos significativos de los píxeles de la imagen. Para no alterar visiblemente la imagen destino, N toma un valor pequeño, siendo 1 y 2 los valores generalmente utilizados. ¿Qué pasaría con $N > 2$?

En esta tarea se pide implementar un programa en *Python* que pueda ser ejecutado por la consola de entrada (entrada estándar). El programa debe ser nombrado *tarea_1.py* y debe permitir recibir los siguientes argumentos:

- **--encode**, indica que la tarea es codificar un texto de entrada dentro de una imagen. La imagen resultante tiene el mismo nombre que el de la imagen de entrada más el prefijo *out*. La extensión es la misma que la de la imagen de entrada.
- **--decode**, indica que la tarea es decodificar una imagen, en este caso se mostrará el texto oculto.

- **--nbits** <N>, indica el número de bits menos significativos a ser usado. N toma un valor entre 1 y 8, inclusive. Este argumento es usado en el modo *encode*.
- **--image** <image filename>, indica la imagen sobre la que se codificará un texto. Pueden suponer que la imagen está en escala de grises.
- **--text** <text filename>, indica el archivo que contiene el texto a codificar. Este argumento es usado en el modo *encode*.

Un ejemplo para codificar el texto de *texto.txt* en *imagen.jpg* usando los 2 bits menos significativos será:

```
>>python tarea_1.py --encode --image imagen.jpg
                        --text texto.txt --nbits 2
```

y para decodificar tendremos:

```
>>python tarea_1.py --decode --image imagen.jpg
```

Deben notar que para la decodificación, no es necesario indicar la cantidad de bits. En este sentido, al codificar un texto deberán, además, guardar en la misma imagen el número de bits utilizados, de modo que puedan recuperarlo directamente de la imagen durante el proceso de decodificación.

Para facilitar el parseo de los argumentos de entrada se recomienda ocupar el modulo **argparse** de python.

Además, el programa debe permitir recibir una imagen en escala de grises o una imagen a color. Así, el programa debe ser capaz de determinar el tipo de imagen antes de proceder a codificar.

Deberán entregar por u-cursos el código fuente junto con un informe (tipo paper) que describa el trabajo realizado junto con ejemplos de uso y un análisis de los resultados. Para el análisis deben, además, determinar el número de bits que pueden ocupar sin que la imagen pierda calidad visual.

3. Esquema de Informe

1. **Abstract o Resumen:** es el resumen del trabajo.
2. **Introducción:** se describe el problema y el contexto. Aquí deben definir el término Esteganografía y sus aplicaciones en imágenes. (10 %)

3. **Desarrollo:** se describe el diseño e implementación del programa. (40 %)
4. **Resultados Experimentales y Discusión:** se debe presentar los resultados y hacer un análisis de los mismos. (40 %). En este punto se debe analizar la cantidad de bits que se puede ocupar para agregar información a la imagen sin perder calidad visual.
5. **Conclusiones** (10 %)

4. Restricciones y Condiciones

1. NO se aceptan tareas sin informe.
2. NO hay atrasos.
3. La tarea es individual.
4. Poner mucho esfuerzo en la redacción del informe.
5. La implementación se realizará en Python.

5. Entrega

La entrega se realiza por u-cursos hasta el lunes 19 de agosto, 2019, 23:50 hrs. Se debe incluir:

1. Código fuente (en Python)
2. Informe