Text In Image

Project 1
Assigned: February 2, 2017
Due: April 6, 2017

BACKGROUND

Steganography is the study of hiding data (text, images, audio, etc.) within data (text, images, audio, etc.). Some uses of steganography are exchanging hidden messages, watermarking, or exfiltrate data from a network. There are many ways to do steganography.

A physical example of this is when a soldier, Jeremiah Denton, spelled "torture" by blinking his eyes in morse code during an interview to inform the world that POWs were being tortured.

A digital example of this is Fabien A.P. Peticolas' mp3stego. As the name suggest, this stores hidden data inside mp3 audio files without altering the audio representation.

PROJECT INFO

Students are required to implement a program that hides text inside of images.

In order to hide the text in an image students need to think of the image as a series of pixels, each pixel as an RGB value, and each RGB value as three sets of eight bits. Students are required to hide the data inside the least significant bit of each RGB value, to minimize the changes in the visual representation of the images.

Students must embed the data in each pixel moving from the bottom right to the top left.

Students must use the first 11 pixels on the bottom right to embed the text length. The remaining pixels are used to store the text.

Data must be stored inside each RGB value in the least significant bit. Output to a lossless filetype.

REQUIREMENTS

Your project must include all the following features:

- Text embedded inside images
- Data embedded from the bottom right to the top left
- Converts text to binary
- Converts text length to binary
- Use the bottom right 11 pixels to hide the text length
- Use the remaining pixels to hide the text
- Replace the least significant bit of the RGB values
- Gracefully exit if supplied data will not fit in the image
- Consumes jpeg image
- Exports png image
- Extract text length from image
- Extract text from image

RECOMMENDATIONS

- Use Python 3
- Use Python's Pillow (https://python-pillow.org/)

WHAT TO TURN IN ON TITANIUM

- Your documented source code (Link to github or zip file)
- Image(s) embedded with your source code in it
- README file with the following:
    - Name
    - Brief description of your application architecture (1-3 sentences)
    - Instructions on how to execute your application

HOW GRADES ARE CALCULATED

Artifact	Grade (100 total points)
Documented Source Code	50 points
Images with embedded source code	20 points
README	20 points
Coding Style	10 points
Total	100 points