Task 2

Heuristic Steganographic Embeddings and the χ^2 Detector

Christian Riess (christian.riess@fau.de)

Let us try out LSB embeddings with and without additional Hamming Coding. As a detector, let us use our χ^2 test.

1 Steganographic LSB embedding

Write a (simplified) steganographic embedding system. As plaintext, you can grab an arbitrary text from a webpage. To encrypt the text, you can use a simple solution that approximately produces a random sequence of bits. For example, you can use symmetric DES[1] (however, note that this is insecure) or, e.g., AES.

Allow two modes for your stegosystem:

- Naive embedding with specified payload (or distortion). Use every pixel for embedding to achieve a payload of 1 bpp, and embed in every k-th pixel for a smaller payload (real stego systems would use a proper visual mask or, better, a distortion function, but let us skip this).
- Hamming-encoded embedding with specified payload or distortion[2]
- Decode your messages to ensure that the embedding works.

As in the previous exercise, please feel free to reuse the UCID images in `/proj/ciptmp/sichries/` in the computer science CIP pool.

2 Detecting LSB Embedding with a χ^2 Test

Implement the χ^2 test. Note that the test function itself is provided in python, so this task boils down to feeding the implemented χ^2 test with the proper input.

- Detect the naive embedding with a payload of 1. Double-check that your detector does with high probability not detect a natural (non-stego) image.
- Decrease the payload. When does detection with this test not work anymore?

[1] Code samples are, e.g., here: https://www.tutorialspoint.com/cryptography_with_python/cryptography_with_python_quick_guide.htm

[2] The construction of Hamming codes with different code word lengths is not difficult, but requires some bit-fiddling, see, e.g., https://www.gaussianwaves.com/2008/05/hamming-codes-how-it-works/