Writeup

For this project, we made a class, ImageBits that takes a number of customizable parameters in its constructor and then reads in and saves bits from an image according to those parameters.

These parameters include:

1. path (image's path)
2. bit_pattern (which bit to extract, starting from the right)
3. combine (saves each bit up to bit_pattern instead of just bit_pattern)
4. bits (the bits it has to extract can be passed as a parameter to avoid extra work)
5. rotation (how many degrees the image should be rotated before extraction)
6. channel (whether the class should only extract bits in a specific channel and if so, which one)
7. swap (which order channels are read in. i.e. RGB or BGR)
8. reversed (whether the program is going Right to Left or Left to Right)
9. diagonal (whether we are only checking pixels in the diagonal)

Additionally, this class is extended with HiddenImage and HiddenText which use their class field self.bits to decipher messages.

An outer level script in `main.py` is used to get all different kinds of bit extractions from all of the images and save them to `/found_bits`. It is also used to loop through all the bit extractions analyze them at numerous locations (1, 2, 3, 1000, 1001, etc.).

We also tried using an English dictionary library to sift through all the characters in each bitstring, but we didn't end up getting very far with that approach. There is a lot of junk in those files. All of our different trials are up on our GitHub in branches other than the master. Our answers really aren't indicative of all the work we put in and we hope you keep that in mind when grading!