



## Hiding one AI generated picture inside another

### Description

Steganography is the art of concealing one image inside another. I discussed the basics of the technique in a post: [Hiding one surface inside another](#). It involves *bit-shifting*.

In an 8-bit image each pixel is represented by an integer in RGB (red, green and blue). With 256 shades of RGB that provides over 16 million colours.

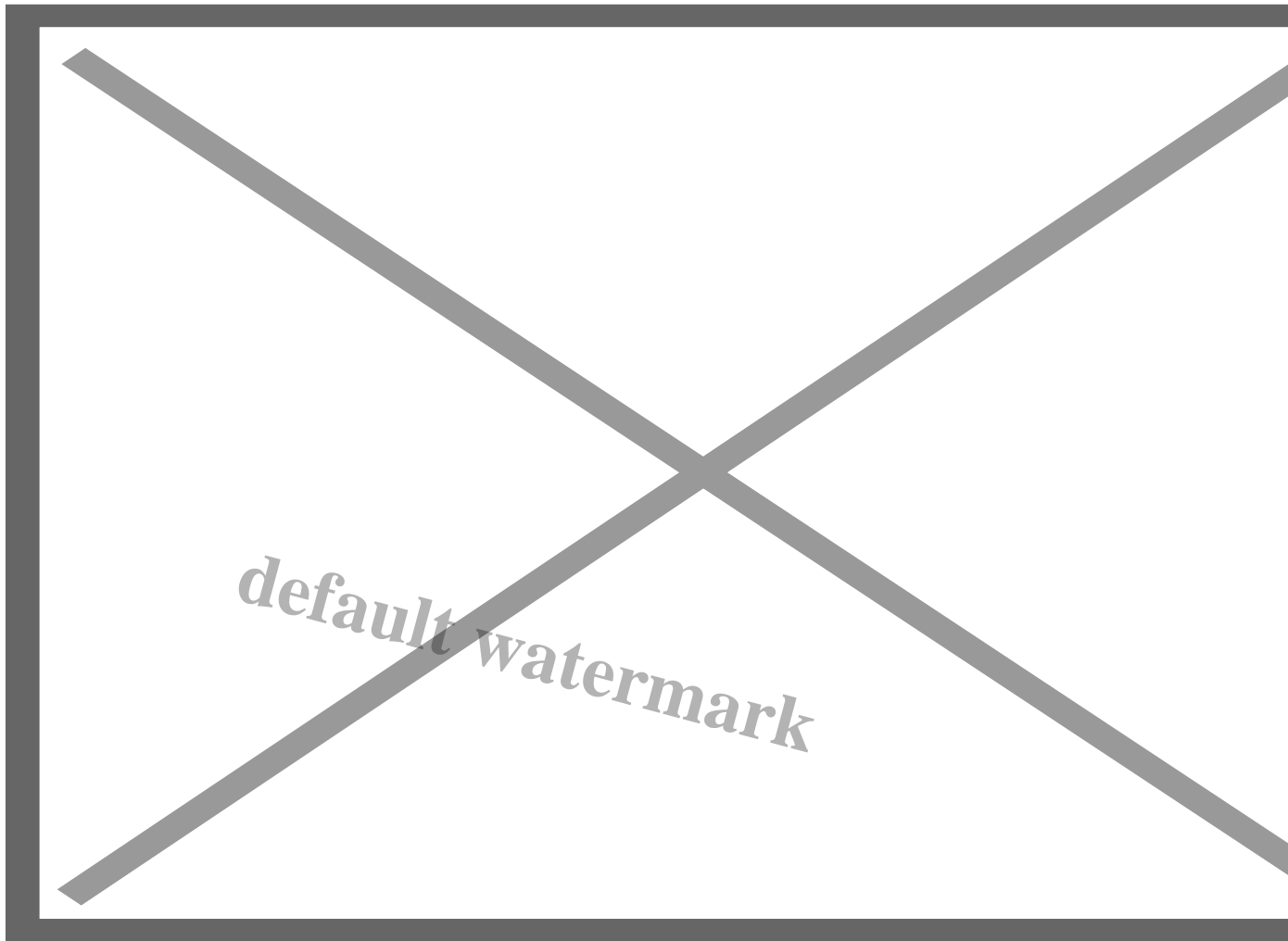
Considering the RGB pixel values in binary, the first four bits are known as the most significant bits (MSBs). The image would look ok with just those bits. That would produce an image at colour resolution of 4 bits, i.e. just over 4,000 colours.

The last four bits of an 8-bit pixel can be relatively noisy. These could get scrambled without affecting the clarity of the main colour variation in the image. These least significant bits (LSBs) can also contain hidden information, secret code, or another image.

### AI generated images

ChatGPT4 will not yet generate steganographic images, but it can help write code in Python to combine images in this way. I produced two AI images and with ChatGPT4's help wrote code to strip the last 4 bits from the pixel values of each image. Then the program appended the MSB of the secret image to the MSB of the host image. When displayed, the secret image is virtually invisible.

To make the secret image visible we simply strip away the MSBs of the 8-bit composite to reveal it. To make the reveal a bit more interesting we can slide the LSBs along the 8 bit composite one bit at a time. That produces the gradual reveal in the images below.



## Watermarking

Steganography is relevant in AI-generated images not least as we think of copyright infringement, protection of cryptoart and [NFTs](#) and appropriation of protected images in AI training. The secret image can be a watermark, or a series of numbers. A secret image, string or message can occupy fewer than the last 4 bits, or occupy any of the bit positions in the 8-bit string.

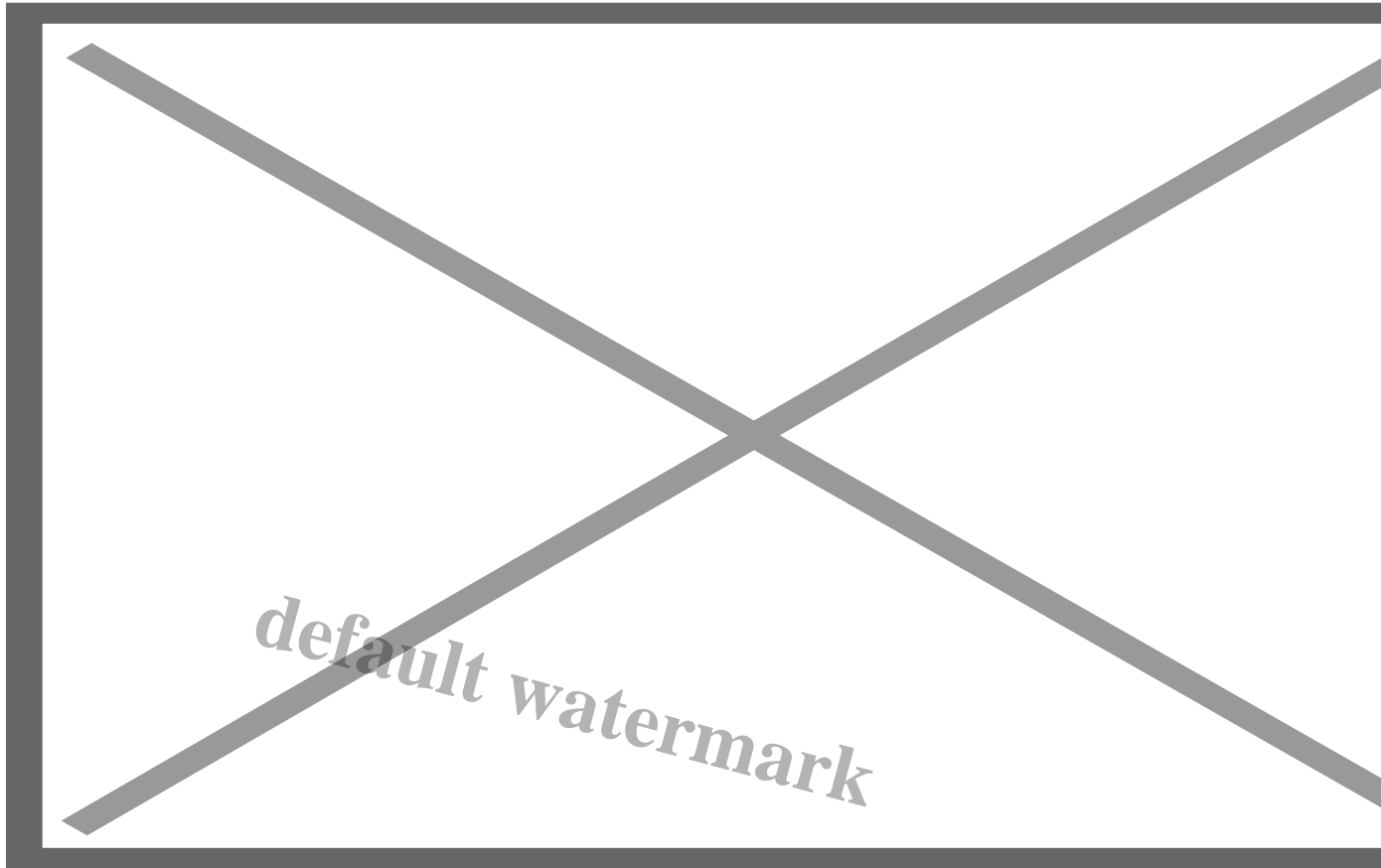
## Fun with bits

James Stanley has a good interactive web tool for experimenting with steganographic bit placements and bit shifts at <https://incoherency.co.uk/image-steganography/>. Here are two of the outcomes from an eccentric bit shift.



## Note

Hereâ??s a transition between two photographs taken from the top of the Burj Khalifa, in Dubai, April 2022.



Here's a secret self portrait, Bermeo, Costa Vasca, Spain August 2019.



Printout showing the progress of the least significant bits (LSBs) for a single pixel as they progress to the front of the bit sequence. The numbers are the integer equivalents of the bit strings.

```
recovery started
Recovered as partially_recovered_secret.png; shift = -1
X: 400 Y: 300
R: 32 00100000
G: 64 01000000
B: 80 01010000
Recovered as partially_recovered_secret.png; shift = 0
X: 400 Y: 300
R: 38 00100110
G: 71 01000111
B: 88 01011000
Recovered as partially_recovered_secret.png; shift = 1
X: 400 Y: 300
R: 44 00101100
G: 78 01001110
B: 81 01010001
Recovered as partially_recovered_secret.png; shift = 2
X: 400 Y: 300
R: 26 00011010
G: 92 01011100
B: 97 01100001
Recovered as partially_recovered_secret.png; shift = 3
X: 400 Y: 300
R: 50 00110010
G: 60 00111100
B: 69 01000101
Recovered as partially_recovered_secret.png; shift = 4
X: 400 Y: 300
R: 98 01100010
G: 116 01110100
B: 133 10000101
Recovered as partially_recovered_secret.png; shift = 5
X: 400 Y: 300
R: 96 01100000
G: 112 01110000
B: 128 10000000
```

default watermark

## Category

1. Artificial Intelligence

## Tags

1. steganography

## Date Created

November 18, 2023

## Author

rcoyne99