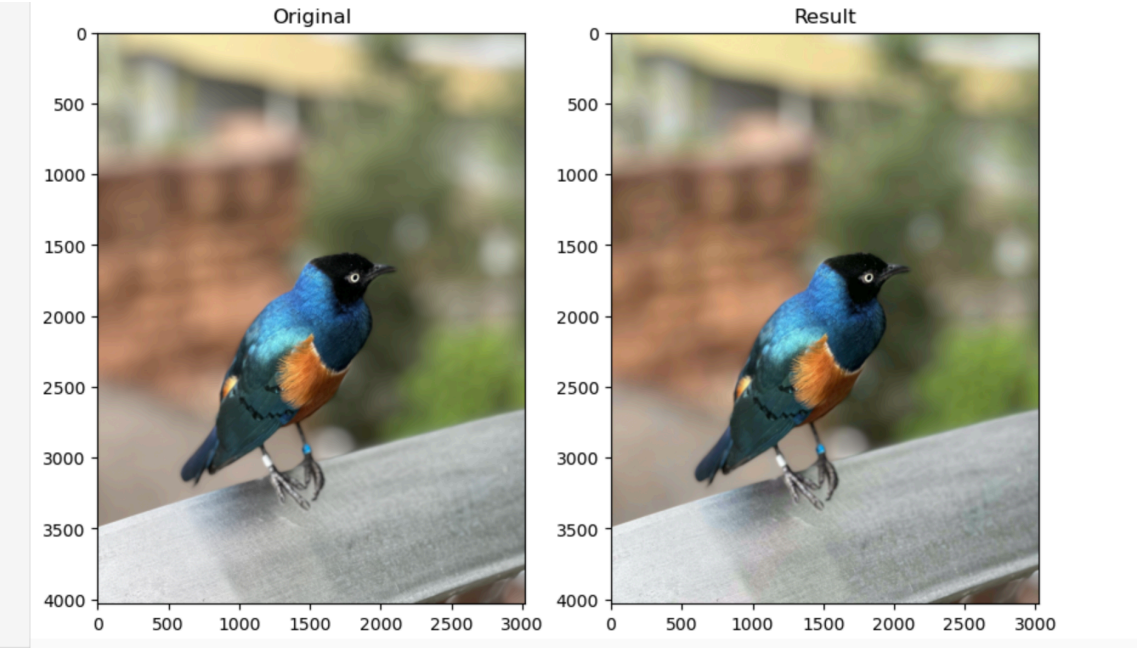
Process Flow:

1. Encoding:
    a. Read image, convert to YCbCr domain.
    b. Read audio and fft.
    c. Keep only positive frequencies and filter high-frequency components.
    d. Round result of fft, convert to --> int16 --> binary, and split in group of 2 digits.
    e. Encode real part to Cb, and imaginary part to Cr, replacing the last 2 bits of pixel values with fft results.
    f. Reconstruct the image by merging the channels.
2. Decode:
    a. Read image, split to YCbCr.
    b. Get the real and imaginary array from the last 2 bits of Cb Cr channels.
    c. Merge results to get complex numbers.
    d. Reverse the array to get inverse frequencies, merge and get full fft result.
    e. Perform ifft and get audio back.

Result:



This method is valid since:

1. I observed that both quantization tables for the image are arrays of all ones, i.e. lossless compression when converted using JPEG, so safe to encode messages to pixels.
2. The max and min of the fft result are within [-2^15, 2^15-1], so safe to use int16.
3. The message is encoded into Cb Cr channels since human eyes are less sensitive to Cb Cr changes