Assignment 1: LSB steganography

Task 1. LSB extraction and re-insertion

1.1. Read a color image and convert it to grayscale
I = imread(‘filename’), G = rgb2gray(I);

1.2. Extract the k=8 bit-planes of theimage
For each pixel I(i,j)of the image, convert it to binary (dec2bin, de2bin)
Store each bit of the binary value in the corresponding matrix

1.3. Display each bit-plane of theimage
figure, imshow(bitplane(:,:,k) , []);
figure, subplot(a,b,c), imshow(bitplane(:,:,k))

1.4. Reconstruct the grayscale image
For each pixel of each bit-plane bitplane(i,j,k), convert the binary value to decimal and store it into a new matrix (bin2dec, bi2dec)

1.5. Save the modified image
imwrite(G_mod, ‘format’, parameters);

Task 2. Embedding a message into the LSB

2.1. Read a color image and convert it to grayscale
I = imread(‘filename’), G = rgb2gray(I);

2.2. Extract the kbit-planes of the image
bitget(G,k);

2.3. Convert a string into a binary message of length L
repmat(s,M,N), dec2bin()

2.4. Insert the message into random positions of the LSB
rand(‘state’,key), randperm()
payload = log2(length(s))-length(G(:));

2.5. Reconstruct the image and save it

2.6. Extract the LSB of the modified (stego)image
Compute the L positions as in embedding by using same key
Compare with the input message: xor(message_or, message_rec)

2.7. Compare the cover image x and the modified stego image y
Pixel-wise difference:
diff =abs(x-y);
Distortion:
d(x,y) = sum(sum((x-y).^2)));
Peak Signal-to-Noise Ratio (PSNR):
[M,N] = size(x);
mse = sum(sum((x-y).^2))/ (M*N);
p = 20* log10(255) - 10*log10(mse);
Structural SIMilarity index (SSIM):
[ssim, map] = SSIM(double(x), double(y));

Task 3. Embedding of binary image into LSB

3.1. Read a color cover image of size [M,N]

3.2. Read a grayscale secret image of size [M,N] and make it binary

```
S_bw = im2bw(S); or    S_bw = S>128;
```

3.3. Extract the channel (e.g. green) where the message is embedded

3.4. Get the LSB plane of the green channel

```
LSB_green = bitget(green,1);
```

3.5. Replace the LSB with the secret image

```
LSB_green = S_bw;
```

3.6. Reconstruct the green channel green_mod

3.7. Reconstruct the stego color image and save it in TIFF format

```
stego = cat(3, I(:,:,1), green_mod, I(:,:,3) );
imwrite(I_stego, ‘stego.tiff’,’tiff’,’Compression’,’none’);
```

3.8. Compare each channel of the cover image x and the modified stego image y

Pixel-wise difference

Distortion

Peak Signal-to-Noise Ratio (PSNR)

Structural SIMilarity index (SSIM)

Color difference: delta = sqrt((Lx-Ly).^2+(Ax-Ay).^2+(Bx-By).^2);

3.9. Repeat the embedding procedure on a different channel (red, blue) and compute again the above metrics. Indicate the difference

3.10. Modify the algorithm in such a way that the secret image is scrambled according to a key (see Task 2)

3.11. Modify the algorithm so that a smaller image can be embedded

3.12. Recover the hidden image

Extract the LSB of the modified (stego) image

Compare with the input image

```
xor(message_or, message_rec)
```

Task 4. Embedding of binary image into LSB

4.1. Explore the effects of various processing on LSB stego images

4.2. Verify that the secret message (image) is inside the stego image

4.3. Crop an arbitrary portion of the image and verify what happens to the LSB message. What if the message was scrambled at embedding?

```
I_crop = imcrop(I,[x,y,h,w]);
```

4.4. Add some noise to the stego image and verify what happens to the LSB message

```
I_noise = imnoise(I_stego, ‘noise_type’, params);
```

4.5. Compress the image with JPEG and verify what happens to the LSB message

```
imwrite(I_stego, ‘stego.jpg’ ‘jpeg’,  ‘Quality’,100);
I_jpeg = imread(‘stego.jpg’);
```

4.6. How to achieve robustness to JPEG compression? LSB in DCT domain

Task 5. Embedding of binary image into LSB

5.1. LSB embedding leaves traces into image histogram

5.2. Read the color stego image from Task 3, extract the green channel hosting the LSB secret message

5.3. Compute the pixel histogram of the original green channel and of the stego green channel

```
h_or = hist(green_or, 256);
h_stego = hist(green_stego, 256);
```

5.4. Plot the histograms and observe their differences

```
figure, bar( h_or );
figure, bar( h_stego );
```

5.5. LSB revealed by LSB enhancement

```
figure, imshow(255*bitget(I_stego,1));
```