**CM0669 Machine Learning and Computer Vision**

**Lab 9** Ethical and Legal issues in Computer Vision,Techniques for security

**1. Image watermarking**

You are required to implement a multi-bit watermarking system which operates in the wavelet transform domain. The system embeds a binary string of one byte (8 bits) into an image and saves the watermarked image to a file. Use for example the following binary string: 01011101.

A Matlab code should be written for the embedding stage and another code should be written for the extraction of the watermark. The embedding system should first read a picture and a key, which is saved in a file named key\_file.mat, containing the locations of the coefficients that will hold the watermark bits. These coefficients are located in the approximation sub-band of the third level of decomposition. Use the Haar wavelet.

Obviously, 8 coefficients will be used to hide 8 bits. Each coefficient is first divided by 30 **(i.e. quantized)** and then approximated to the closest integer. The resulting coefficient is then used to embed a watermark bit. If the watermark bit is 0, the coefficient is incremented by 1 only if it is odd. Otherwise, the coefficient is kept unchanged. Likewise, if the watermark bit is 1, the coefficient is incremented by 1 only if it is even. Otherwise, the coefficient should be kept unchanged. After the modification of the coefficients, they should be multiplied by 30 (**i.e.** **de-quantized**). Finally, the inverse wavelet transform is applied to reconstruct the watermarked image in the pixel domain. The code should save the watermarked picture in a file.

The watermark extraction code should read the watermarked image, perform a similar wavelet decomposition and read the key to extract the watermark bits from the right coefficients. To this end, the coefficients are first **quantized** and **rounded** to the closest integer. Then, the watermark bit is 0 if the coefficient is even. Otherwise, the watermark bit is 1.

**Question**:

1- Apply some image processing manipulations (e.g. noise, histogram equalization, sharpening, average filtering) to the watermarked image and then extract the watermark. Analyse the robustness of the watermarking system.

**Hints**: use the function ‘mod’ to check if a number is even or odd.

**2. Perceptual image hashing**

You are required to implement a perceptual image hashing system which operates in the discrete cosine transform domain (DCT). The system extracts a binary sequence and saves it to a file. Four pictures are provided for the experiments.

A Matlab code should be written for the extraction process and another code should be written for the decision on the similarity of two pictures.

The hash extraction system should first read a picture and divide it into non-overlapping blocks of size 32×32. Each block is then transformed in the DCT domain. The first two coefficients in the horizontal and vertical directions (i.e. at coordinates (1,2) and (2,1)) of each block constitute a vector of coefficients that will be used to get the binary hash. The hash bits take 1 if the corresponding coefficients are positive and 0 otherwise.

To decide whether two pictures are similar (one is derived from the other), a Matlab code has to be implemented as follows.

a- Read two hashes corresponding each to a test image.

b- Calculate the normalised Manhattan Distance.

c- If the distance is greater than 0.2, the system decides that the two images are dissimilar and displays a message. Otherwise, the two images are similar and an appropriate message will be displayed.

**Questions**:

1- Use the four test pictures to evaluate the performance of the hashing system in terms of discriminability.

2- Apply some image processing manipulations (e.g. noise, histogram equalization, sharpening, average filtering) to the original images and then extract the hash. Analyse the robustness of the hashing system.