Université de Genève

Multimedia Security and Privacy
14x016

TP 4 : Watermarking Detection

Author: Deniz Sungurtekin

E-mail: Deniz.Sungurtekin@etu.unige.ch

May 2021

1 Introduction

In this work, we will embed a watermark with uniform distributed values {-1,1} inside a host image. Then we will simulate an attack by adding an Additive White Gaussian Noise with 0 mean and a standard deviation of 1. After a first analysis of the obtained images, we will do a non-blind watermark detection by estimating the watermark knowing the host image and determine the linear correlation between the original watermark and the estimated one. Afterwards, we will do a blind watermark detection using the maximum likelihood estimate where we will do the detection without using directly the original host image. Finally, we will compare the two obtained linear correlation.

2 Watermark Embedding and Channel modeling

First, we will read a gray scale image x which will be our host image and generate a matrix of the same size with uniform distributed values {-1,1}. We will randomly sample from matrix w' with density $\theta = 0.5$. In other words, after generating w', we will set to 0 N = N_1 * N_2 /2 pixels where N_1 and N_2 are the dimensions of our host image. This manipulation allows us to easily embed the watermarking by a simple matrix addition (y = x + w, we will not have a watermark on the pixels where we add 0). Then, we generate our Additive White Gaussian Noise matrix z and add it to obtain our attacked image v (v = y + z = x + w + z).

Here we can observe our three images:

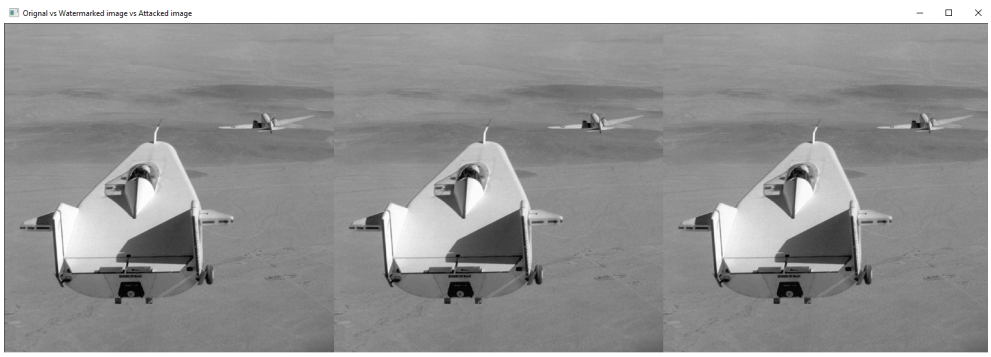

Figure 1: Original VS WaterMarked Vs Attacked

Our initial image is made of value in range 0 to 255, so obviously when we add a watermark with value -1 or 1 in half of the pixels it's almost impossible to see a difference with our eyes. Moreover, we add a Gaussian noise with mean 0 and standard deviation of 1 which generate value in range [-3,3] (3 and -3 in the very worst case) so the differences is very hard to perceive. So, as we can observe the three images seems to be the same but they are statically different.

3 Non-blind detection and Blind watermark detection using the Maximum Likelihood estimate

First, we just estimate the watermark from the attacked image v using the original image x by simply subtracting the matrix v and x and compute the linear correlation between the original watermark and the new obtained watermark. Before discussing about the obtained value, we will first compute the blind watermark and the corresponding linear correlation to compare them.

Now, we will blindly estimate the watermark. We estimate the host image by computing the local mean with overlapping blocks of the marked image v. In order to make sure to keep the same size for the local mean image, we do a zero padding on the right and bottom side of the marked image. The size of the padding will depend on the size of the used blocks to compute the local means. We do this to remove the white Gaussian noise and the watermark in the marked image. Because the noise and the watermark have a mean of 0, the more we take samples in each block the more we will remove the noise and the watermark. That's why we can assume that this local mean is a good approximation of x, the original image. In this work I used 3x3 window shape to obtain the local mean image.

Here are the two obtained linear correlations:

```
print(linear_corr)
print(linear_corr_blind)
```

```
0.9995208205997793
0.8567045950829456
```

Figure 2: Non-Blind VS Blind approximation

The detection of the watermark is made by the computation of this correlation, the more this value is big (in our case, the maximum value is 1 because of the chosen uniform value), the more we are sure that there is a watermark. For the non-blind correlation, we have almost the maximum value of 1. For each element, the watermarking is either 1 or -1 which squared value is 1, so when the estimated watermark is equal to the original watermark, we will have a linear correlation of 1. However, our correlation is not equal to 1, this is explained by the fact that there is also a Gaussian noise in the estimated watermark, so the value can be slightly over 1 or under 1. In regards to the blind estimation, we observe less good result, of course removing the watermark and the noise by doing a local mean on the marked image isn't perfect, so the host original image isn't perfectly recovered. Moreover, there is still the Gaussian noise on the marked image, so when we do our subtraction, we don't have exactly our original watermark.