RAPPORT

Rémy AUDA

Table des matières

Compilation:

First, go to the root directory of project. After, run the following command line :

./setup.sh

```
auda@auda-VirtualBox:~/Documents/Image_Forensics$ ./setup.sh
```

This bash could install you automatically all the requirements that you need.

Next, you have the choice. Either, you can run all demo of different algorithms or you can run each algorithm separately.

To run all demo of different algorithms, execute this following command line to the root directory of project :

./runAll.sh

```
auda@auda-VirtualBox:~/Documents/Image_Forensics$ ./runAll.sh
```

To run all demo of each algorithm separately, execute this following command line to the root directory of project :

./runELA

```
auda@auda-VirtualBox:~/Documents/Image_Forensics$ ./runELA.sh
```

./runCopyMoveDetection

```
auda@auda-VirtualBox:~/Documents/Image_Forensics$ ./runCopyMoveDetection.sh
```

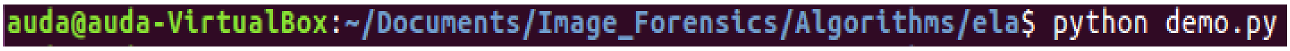
./runPerceptualHash

```
auda@auda-VirtualBox:~/Documents/Image_Forensics$ ./runPerceptualHash.sh
```

Of course, you can also run the demo manually with following command :

To execute ELA algorithm, go to the directory Algorithms/ela and execute the following command :

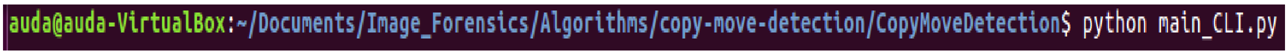
python demo.py



```
auda@auda-VirtualBox:~/Documents/Image_Forensics/Algorithms/ela$ python demo.py
```

To execute Copy-Move-Detection algorithm, go to the directory Algorithms/copy-move-detection/CopyMoveDetection and execute the following command :

python main_CLI.py



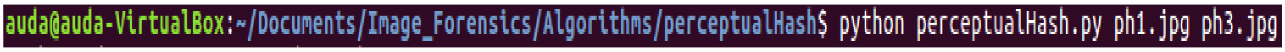
```
auda@auda-VirtualBox:~/Documents/Image_Forensics/Algorithms/copy-move-detection/CopyMoveDetection$ python main_CLI.py
```

To execute perceptual Hash algorithm, go to the directory Algorithms/perceptualHash and execute the following command :

python perceptualHash.py <image1> <image2>

For example :

python perceptualHash.py ph1.jpg ph3.jpg



```
auda@auda-VirtualBox:~/Documents/Image_Forensics/Algorithms/perceptualHash$ python perceptualHash.py ph1.jpg ph3.jpg
```

1. ELA:

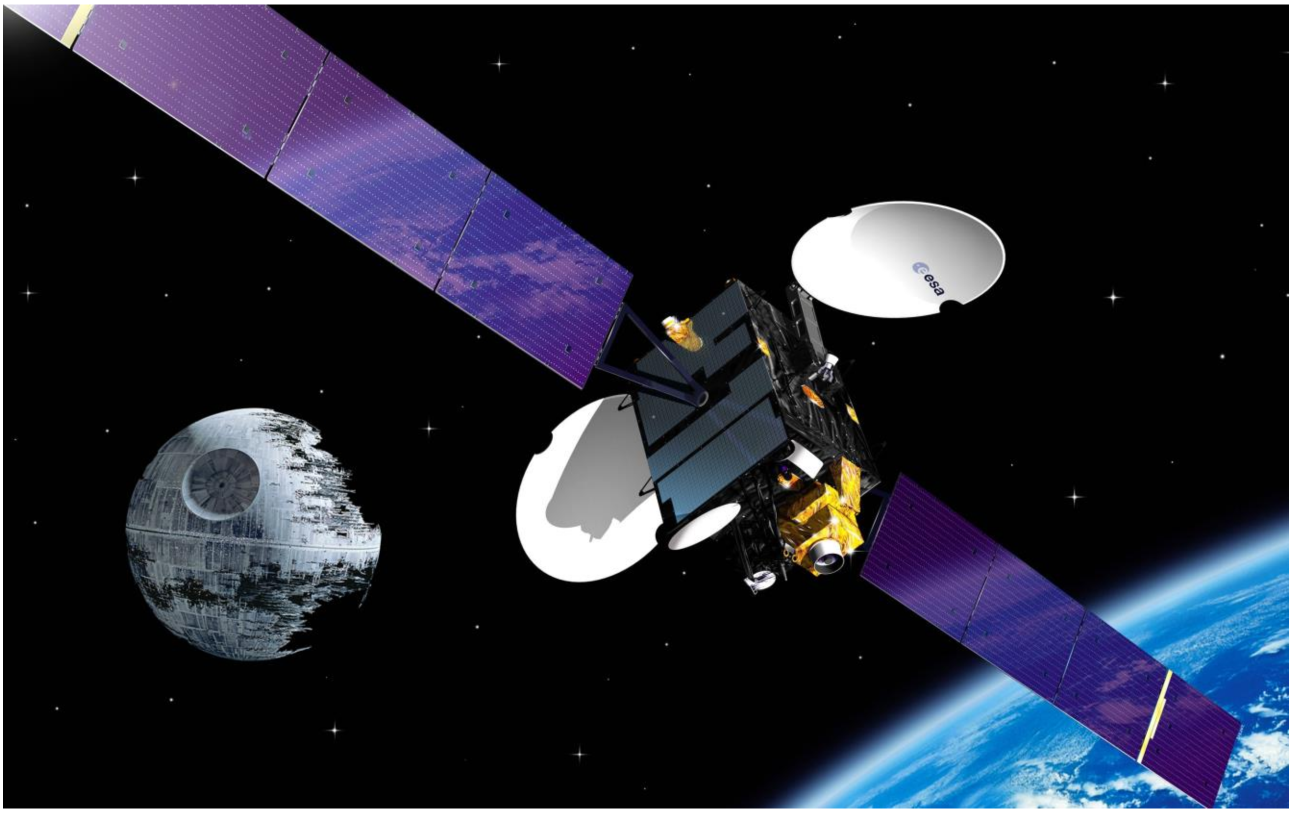
1.1-Description:

This algorithm is an implementation of the « Error Level Analysis algorithm ».

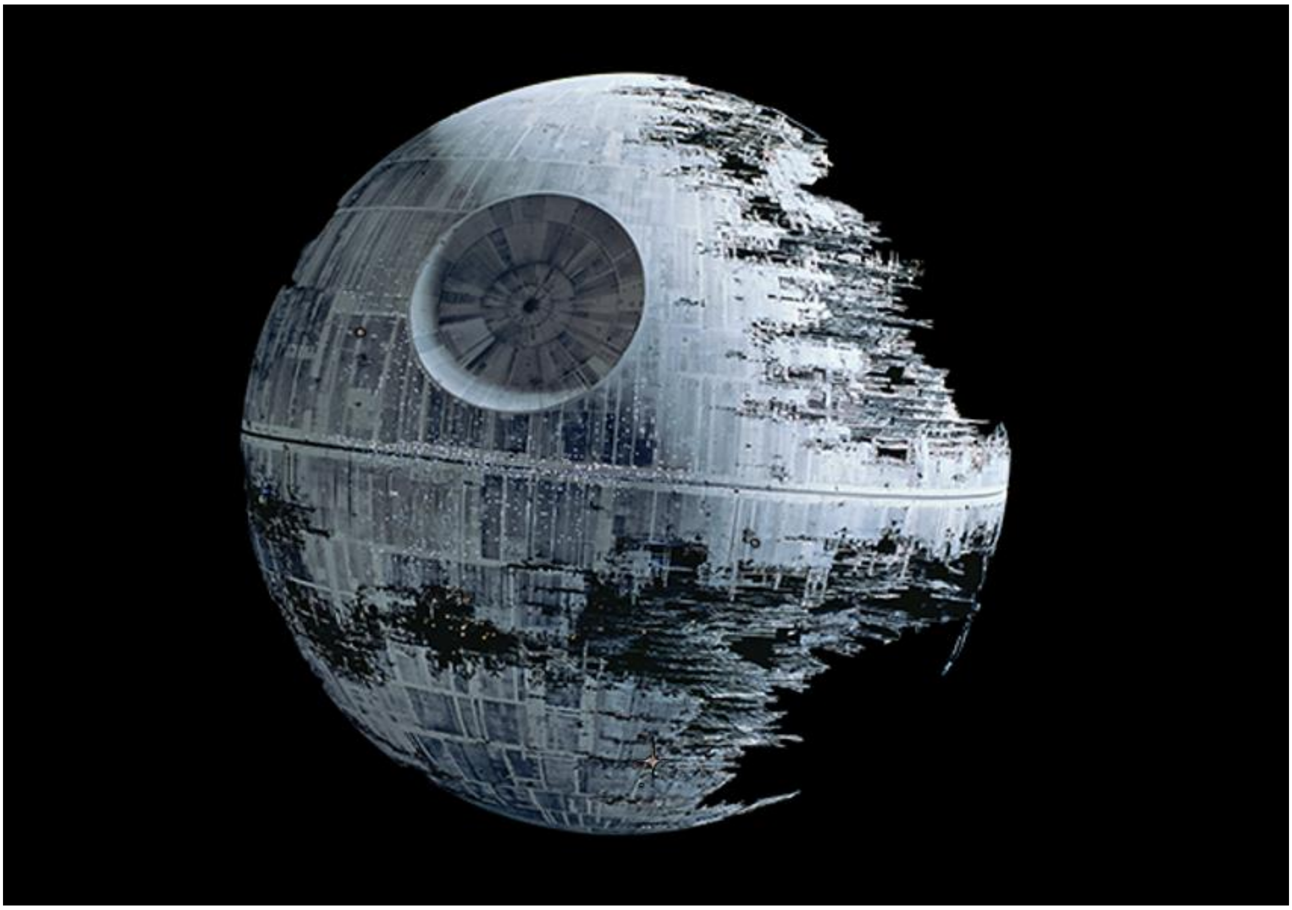
Thus, the algorithm find the different between two images.

Original image :

Forgered image :



The algorithm find the different between the image :



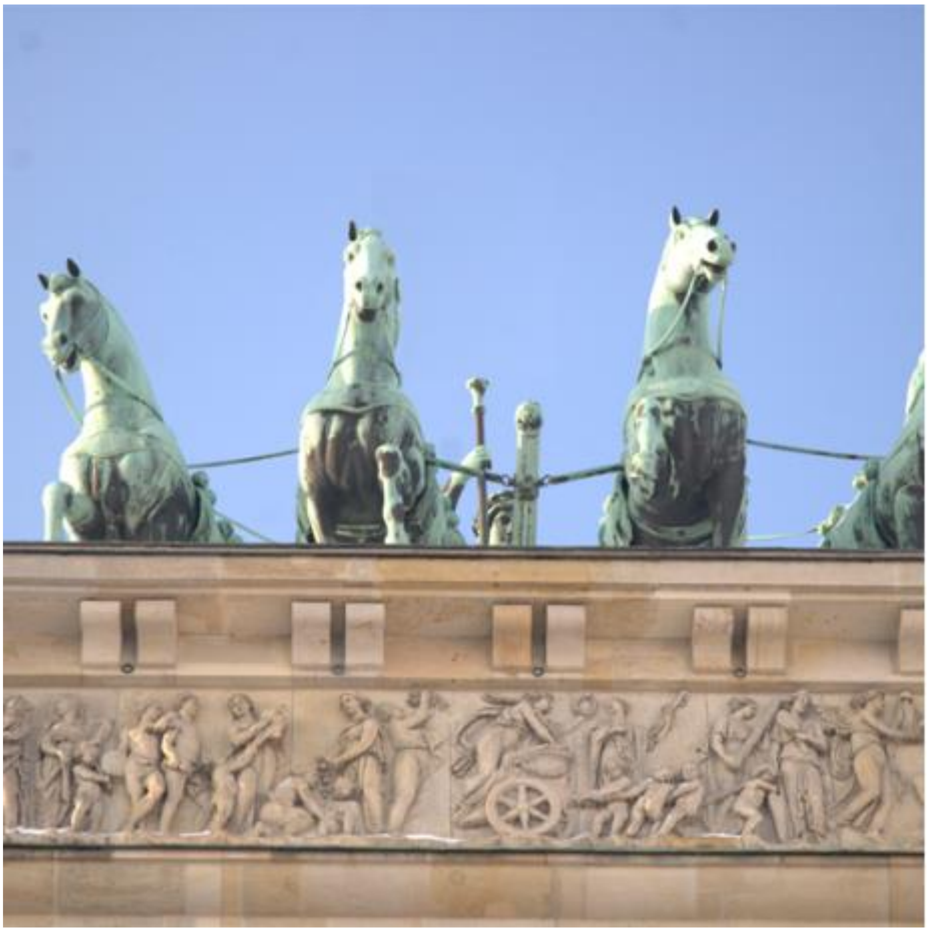
2. Copy-Move-Detection:

This algorithm is an implementation of the « Copy-move-Detection ».

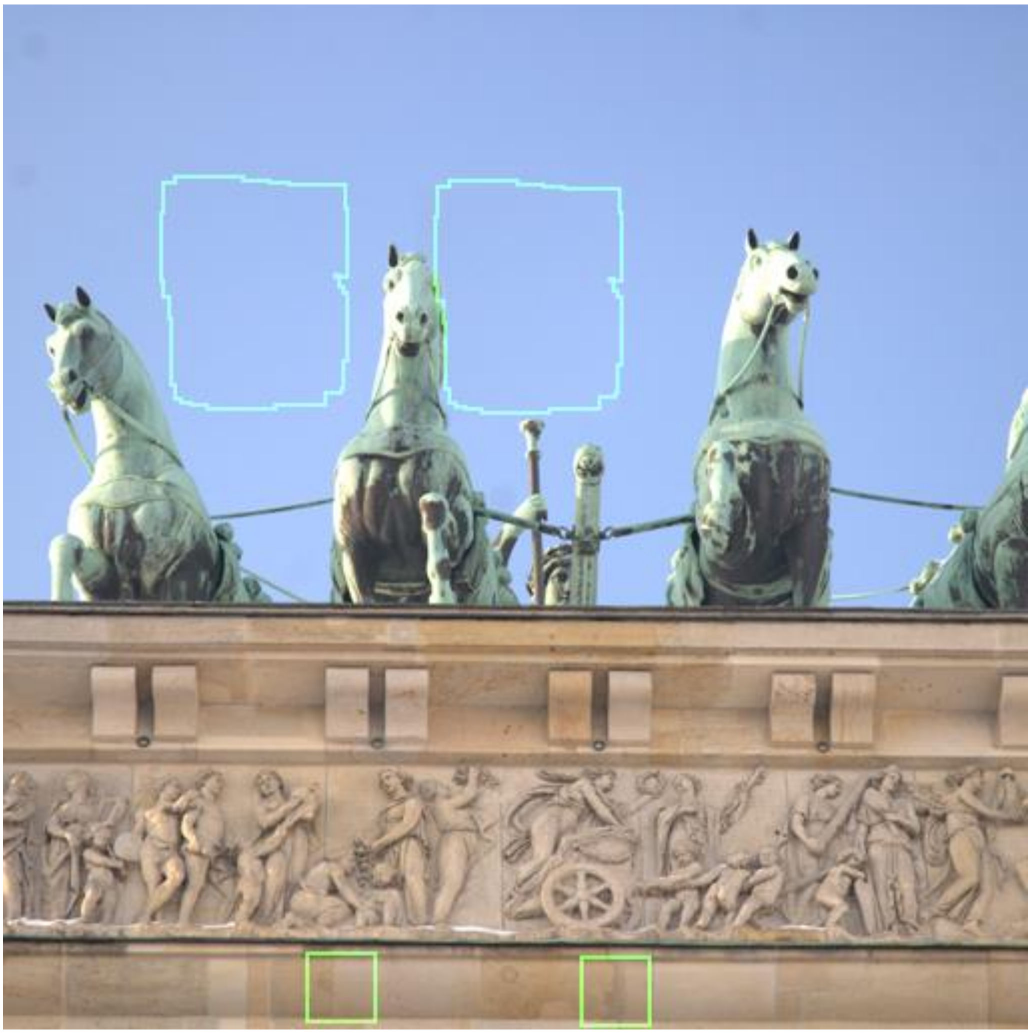
Thus, this python script is an implementation to detect a copy-move manipulation attack on digital image based on Overlapping Blocks.

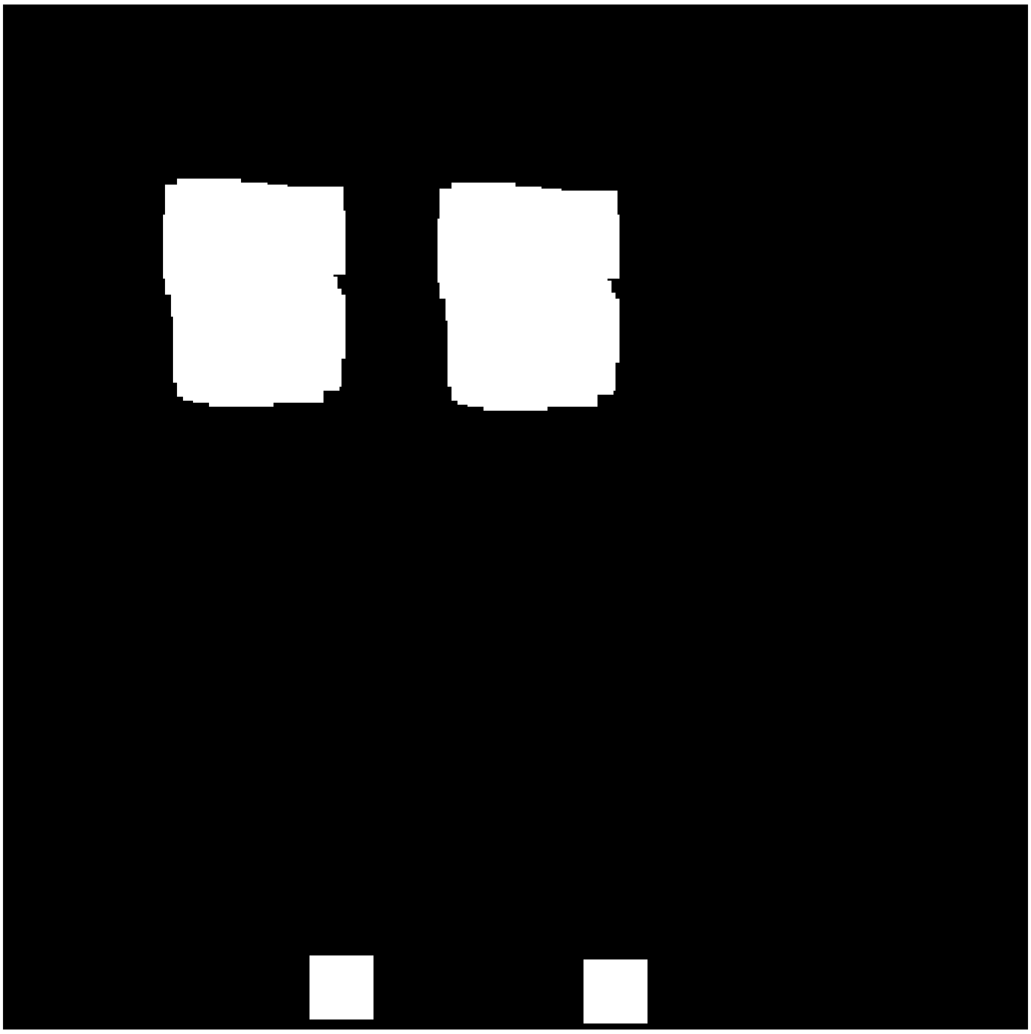
Original image :

Forgered image :



Result :





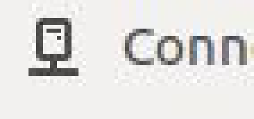
3. Perceptual Hash:

This algorithm is an implementation of the « perceptual Hash ». This implementation of perceptual hash is based on Neal Krawetz's dHash algorithm

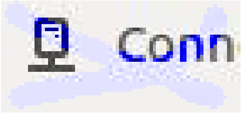
Thus, this python script is an implementation to generates a "difference hash" for a given image. Also, we can compute the bit delta between two images.

First example :

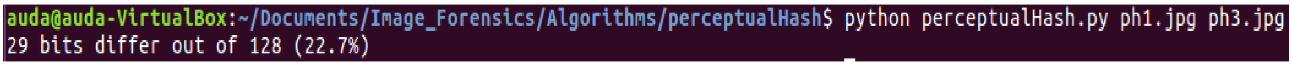
First image :



Second image :



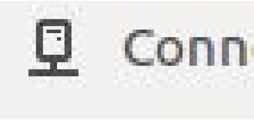
Result :



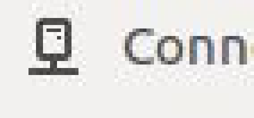
```
auda@auda-VirtualBox:~/Documents/Image_Forensics/Algorithms/perceptualHash$ python perceptualHash.py ph1.jpg ph3.jpg
29 bits differ out of 128 (22.7%)
```

Second example :

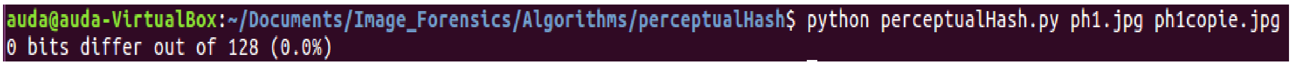
First image :



Second image :



Result :



```
auda@auda-VirtualBox:~/Documents/Image_Forensics/Algorithms/perceptualHash$ python perceptualHash.py ph1.jpg ph1copie.jpg
0 bits differ out of 128 (0.0%)
```