Automated Intelligence Gathering Through Comparison of JPEG Images and their Thumbnails

Nicolas Bodin, Clément Coddet, Olivier Fatou and Eric Filiol
ESIEA, (C + V)^{O} Laboratory, Laval, France
nicolas.bodin@esiea.fr
coddet@et.esiea.fr
fatou@et.esiea.fr
eric.filiol@esiea.fr

Abstract: JPEG images may often contain many metadata thanks to the EXIF standard, including a thumbnail of the image. Whenever editing and modifying a JPEG image, the software does not always modify the new thumbnail in the image accordingly and as a consequence everybody can access the unchanged thumbnail and see which modifications have been made. Our work consists in the development of algorithms for an automatic comparison of a JPEG image and its thumbnail in order to decide whether they are different or not. Then thanks to a web crawler we have developed from scratch, we have applied these techniques to gather intelligence over the Internet and over the Darkweb.

Keywords: metadata, JPEG, image comparison, privacy, thumbnail, digital forensic, intelligence

1. Introduction

Detecting changes between two images is of widespread interest due to a large number of applications in various areas including surveillance, medical diagnosis and treatment (Bosc, HeitzArmspach, Namer et al 2003), and driver assistance systems (Fang, Chen and Fuh, 2003). In all these fields, the images came from the same camera and therefore have the same properties and features (file format, size and quality). The core principle is generally to compare images having the same characteristics. On the contrary, the goal of this article is to present a method to detect modifications between images with different characteristics. We will address the particular case of the comparison of a JPEG image with its thumbnail. This case is specific to JPEG images and other images file format indicated in the EXIF standard (EXIF, 2010) which defines the methods to store data related to the image as metadata, including a thumbnail of the image. We choose to focus on the JPEG format (JPEG,1994) since it is the most common one used for images on the internet, due to its ability to compress an image without damaging the image visually. The interest of such a study is mainly for digital forensics and the automation of the analysis on a huge amount of images. Indeed, a few examples (Schwartz, 2003) show that whenever a JPEG image is edited, its thumbnail can remain unchanged, and therefore reveals the original picture and the information that the owner potentially wants to hide. This project also aims to make peoples aware of the problems linked to metadata with respect to their privacy. Indeed, the information contained in metadata can be more or less sensitive and private (Castoro, 2012) (Rodewig, 2012).

This study stands out from previous studies insofar the set of elements to compare is extremely heterogeneous as the collected images can be of various size and quality. All these parameters are not controllable by users. Moreover, the goal is simply to know whether there is a difference or not between the two scenes represented by the image and its thumbnail. Finally, in order to compare two images, it is imperative that both of the images have the same dimensions. It is therefore necessary to either zoom-in the thumbnail or zoom-out the image, but in both cases there will be deterioration either of the image or its thumbnail and the two compared images will necessarily be different. It is then necessary to use a different method to measure the differences between two images while neglecting those introduced by the size adaptation. This comparison process can be decomposed in three classic steps: pre-processing, processing and post-processing. Pre-processing is the adaptation of the dimensions of the image and its thumbnail, the processing is the measure of the difference between the two images, and the post-processing is the decisional part which clusters the images with respect to chosen attributes or features.

The problem consists in clustering a set of images having various properties (size, quality, color-spaces, etc) into two sets: the set of images which represent the same scene as their thumbnail, and the set of images which represent a different scene than their thumbnail. To do so, two measures of differences have been defined. These methods are adaptable by varying some parameters. Then, once a significant difference is measured, it is necessary to read these values and to cluster the images accordindly. The differences observed can be of various nature and appeared in several ways while impacting more or less the image: adding a watermark, blurring an area, hiding an area with a constant color, rotating the image, cropping the image, changing colors, and changing all the represented scene… Finally, a last part consists in sorting the images detected as different according to the nature of the difference.

After a brief state of the art, a first part will present the methods used for the comparison and the calculus of the difference between an image and its thumbnail, before presenting the method used for the clustering step. A last part will explain and describe the results of a large scale application of these methods over the Internet and the Darkweb during campaigns we have performed for the French government.

1.1 State of the art

As explained previously, there are a lot of works about the detection of changes between images, but most of them need raw and unchanged images. These researches are conducted in specific goals such as video surveillance or medical diagnosis (Radke, Andra, Al-Kofahi et al, 2005). Moreover, the modification detection methods developed in those works are based on the analysis of high-quality images whereas our project is mostly based on amateur images. Some of these methods are based on the analysis of pixels and others on the comparison of "objects" in the image (Hussain, Chen, Wei et al, 2013). Others methods aims to detect if an image was edited – for forensic analysis – by analyzing the compression rate in the image, but these methods can only detect the changes made on the image, but the original *cliché* cannot be retrieved (Krawetz, 2007).

A similar work as ours was realized in 2004 by (Murdoch and Dornseif, 2004). At this date, the only trace available is the PDF used for the presentation of their work during the Chaos Computer Congress the same year. The tools and methods used by Murdoch and Dornseif were – to our knowledge – not published or are not available. This project aims to see if the results are consistent with the ones obtained by Murdoch and Dornseif, even more than 10 years after.

2. Comparison of an image and its thumbnail

The main difficulty in the comparison of an image and its thumbnail is that we compare two different images which can represent the same scene. In order to compare these two images, it is necessary that they both have the same dimensions. A pre-processing step is therefore needed to adapt the size of the images. The ideal would be to recreate the thumbnail from the image. However, the EXIF standard does not specify which algorithms to use and which quality factor to consider. The thumbnail is created with algorithms which are specifics to the camera manufacturer or to the software editor. Moreover, it is not possible to recreate the thumbnail when only knowing the quality to use, because other parameters are set by the device (or the software) which created the image. Consequently, two options are possible: to zoom in the thumbnail or to zoom out the image. The choice of the method for the pre-processing depends mainly on the comparison method used and the nature of the differences we are looking for. The problem is that this pre-processing step does not produce two exactly identical images – the zooming will deteriorate the image – and it is the role of the comparison methods to be able to neglect the differences introduces by the pre-processing and to detect the potential modifications. The extraction of the thumbnail is made with the software "exiftool", developed by (Harvey, 2016). The adaptation of the dimensions is made by the comparison program with the OpenCV library. The next parts aim to introduce two measures for the difference between an image and its thumbnail.

2.1 Pixel-based change detection

An instinctive approach consists in comparing the image pixel per pixel and to compute the Manhattan distance between each pixel and its corresponding pixel of the resized thumbnail. Then the difference between the two images is simply the sum of the inter-pixel distances. Therefore, the greater the difference between the image and its thumbnail, the more images are different. Hence they are likely to represent two different scenes. Practically, the image is simply considered as a matrix, each of its entries is representing a pixel of the image. The comparison then consists in computing the difference between two matrices, and to sum each element of this difference. Let $I = \{i(k)\}$ and $T = \{t(k)\}$ represent the image of $m \times n$ pixels and its resized thumbnail respectively, where $i(k)$ is the k^th^ pixel of the image ($k \in [\![1 : m \times n]\!]$). Let p represent the dimension of the color space of the image (typically, $p = 3$ for color images and $p = 1$for grayscale images. The difference Δ between the two images is calculated according the formula: $\Delta = \sum_k |i(k) - t(k)|$ where$|i(k) - t(k)|$represent the Manhattan distance in $\mathbb{N}^p$.

This difference is then normalized in order to limit the dependences between the result and the size of the image. This step is essential, because unlike previous works which compare images of the same size, we compare here images of any size, and the results must depend only from the scene represented and be the more independent as possible of the characteristics (size, quality, color space …) for a better and easier interpretation of the results. As a consequence, to normalize the result, the difference Δ is divided by the maximum distance possible for same size images. This maximum distance Δ_{max} is computed from a fictive comparison between a white image and a black image to have always a maximum distance between each pixel.

However, this method suffers from some drawbacks due to the pre-processing step. Indeed, the size adaptation necessarily introduces some differences. As the goal of the application is to detect small changes (for example when a watermark is added), it is necessary to zoom-in the thumbnail to avoid detail losses that we could result from zooming-out the image. However, as the zoom factor can be important (the zoom ratios can vary from approximately 100 to 0,5), border effects are common and consequently introduce differences between the two images which are not scene differences but will be detected such as. This initial comparison method evolved to neglect zooming effects and focus on scene differences. Note that various zoom algorithms were tested (nearest neighbor, bilinear, bi-cubic and Lanczos over 8x8 neighborhood), but their influence on the final result is low, and border effects will always be present.

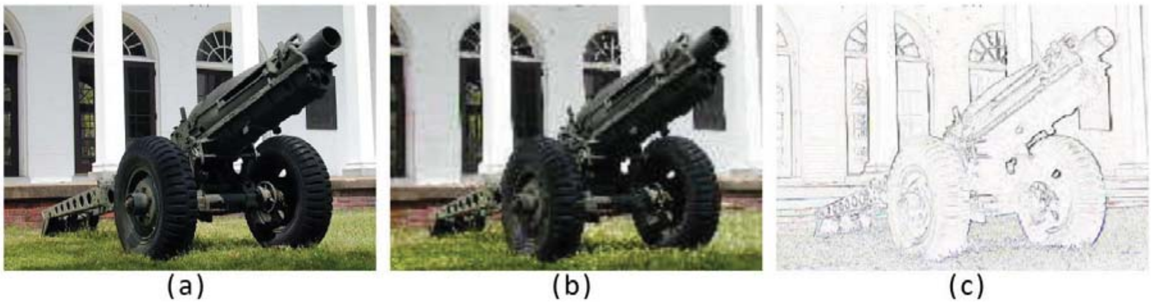

Figure 1: Example of side effect due to the preprocessing

Figure 1 represents the original image (a) and its thumbnail resized to the same dimensions as the image using linear zoom (b). Image (c) is the subtraction of the two previous ones. The colors of the third image are inverted for visualization purposes. Dark regions represent high-difference area. Even if the two images represent the same scene, the comparison of these images will reveal unwanted differences.

Several improvements have been explored in order to neglect small area (some pixels) differences (so not very visible) and to promote big area differences (more visible, so more likely to be scene differences) on the one hand. On the other hand we neglected low differences: two pixels with a low difference are less interesting than two pixels with a high difference because they do not represent the same color.

A first method to neglect the impact of the pre-processing consists in using a threshold to decide whether two pixels are different or not. This threshold is arbitrarily decided and allows to neglect soft pixel differences, which cumulated on the image and finally can have a significant impact on the final difference. This threshold is about 0.5% of the maximum difference between two pixels. Thereby, color-similar pixels will be considered as identical. This method is also combined to a weighting method. Each difference between two pixels is weighted to promote large area differences and neglect small ones. The weighting is a function of the number of adjacent pixels which are detected as different, and the difference of the compared pixel is multiplied by the square value of its number of different neighbors.

A second method consists in dividing the image in blocks and to compute the differences blockwise. The difference block per block is then computed on the same principle as the pixel per pixel comparison explained previously. This block per block method is then combined with a threshold and a block weighting, which allows neglecting the border effect as long as the size of the block is of the same order of magnitude of the size of the border effects.

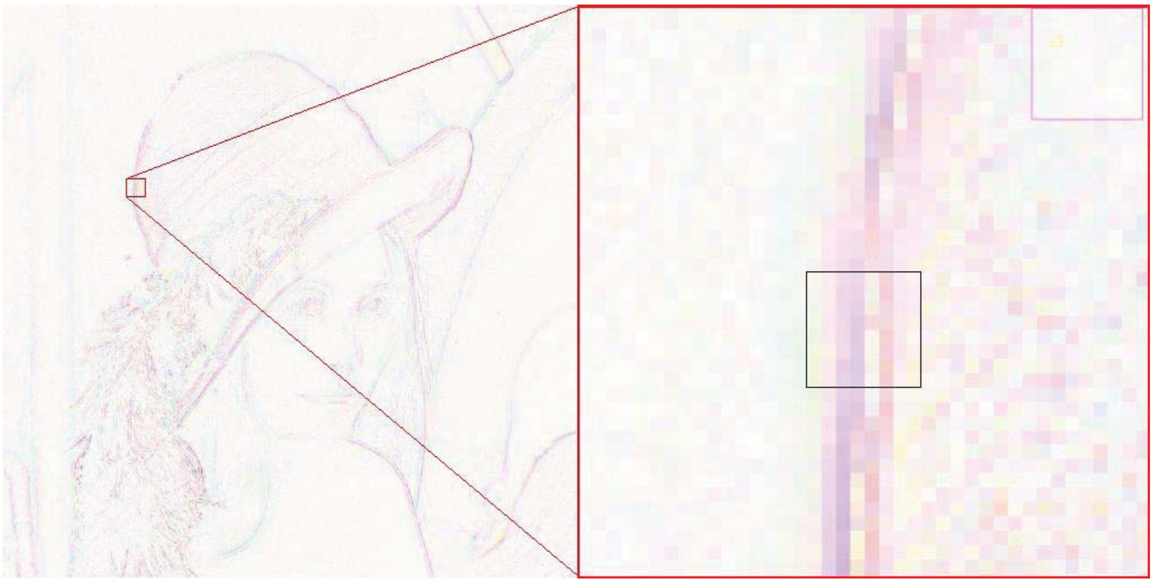

Figure 2: Example of typical block size

The left part of Figure 2 is the difference between an image and its thumbnail when resized. The right part is a zoom over a particular area to show the typical size (in pixels) of an edge due to the pre-processing. The black square represent the typical size of a block.

2.2 DCT based change detection

As explained in the introduction, this project is specific to images which contain a thumbnail. According to the EXIF standard, they are JPEG and TIFF file formats. As JPEG is far more common on the Internet than TIFF, a JPEG specific method has consequently been developed.

JPEG images differentiate from other images format (like PNG for example), because they do not map the pixels, but the variations from a pixel to another.

The key point in JPEG compression is that it is the variation of intensity from a pixel to another one which are stored actually. Therefore, a change of the image (typically a black shape on the face of a subject) will introduce a constant color area, so having low frequencies. When the image will be compared to its thumbnail, the modified area (with few high frequencies) will be compared to the unchanged one (with high frequencies), and will reveal the modification area.

The comparison method is consequently based on quantified DCT blocks which contain the frequency information of the image. However, unlike pixel per pixel comparison, it is no longer possible to zoom in the thumbnail for the comparison. Indeed, zooming in the thumbnail introduces a blur, and consequently distorts the frequencies of the DCT blocks, hence the result of the comparison. That is why the image is resized to the dimensions of the thumbnail in order to avoid the blur and keep a maximum of high frequencies in the DCT blocks. Moreover, as the quantization step causes data losses, the resized image must be saved with the same quality parameters as the thumbnail. Even if only the quality factor can be considered, the quantization tables are copied from the thumbnail to save the resized image because it can be hard to guess the exact quality factor used of an image and the tables can vary from a manufacturer or software to another (the JPEG standard defines recommended but not mandatory tables). Using the same quantization tables also avoid potential information losses if the resized image and the thumbnail do not have exactly the same tables, assuming that the table corresponding to the lowest quality is used. Consequently, to resize the image, the quantization parameters of the thumbnail are copied, the image is decompressed, resized and compressed using the previously copied quantization parameters.

The comparison is composed of two steps. The first one consists in the comparison of each block of the resized image with the corresponding ones in the thumbnail, as we did in the pixel per pixel comparison. The same formula is used on each block, but with DCT coefficients instead of pixels. We then have a block differences matrix. The second step aims to determine whether the image and its thumbnail are different or not. To do so, two approaches are then possible. The first one consists in computing the sum of all the coefficients of the block difference matrix, and to normalize the result to make it independent of the image size. This method was implemented with the same improvements as those presented in the pixel per pixel comparison. The aim is to compensate the problems linked to the size adaptation of the images. For that purpose we the use threshold, and weighting techniques. An in-block weighting can be used in order to promote differences on low-frequency coefficients (which are easily noticeable differences) and neglect the others. But this method is not very versatile because some differences will be transparent. For example, blurring a part of the image mainly impacts high frequency coefficients and has quite no effect on the others ones. Moreover, it is pointless to use weighting to promote high frequency coefficients because they represent information that is not easily noticeable and the corresponding coefficients are often reduced to 0 by the quantization step and the use of a low quality factor. Another approach focuses on the variations of differences from a block to another one. Indeed, the transition from an unchanged area to a modified one (and vice versa) will produce a high variation of the block difference. Therefore, by reading the block differences matrix line by line, we can consider it as a signal. We then have just to detect the high variations on this signal.

However, this method suffers from a major drawback: a large part of the images have a low quality thumbnail (under 50), and the image quality is consequently deteriorated for the comparison, distorting by the way the potential modification. Furthermore, a low quality can imply more noise on the signal (so more high variations) and generate false positive results. This problem also impacts the statistical analysis of the signal as the statistical series will have a high standard deviation, and the statistical measures are thus not very relevant.

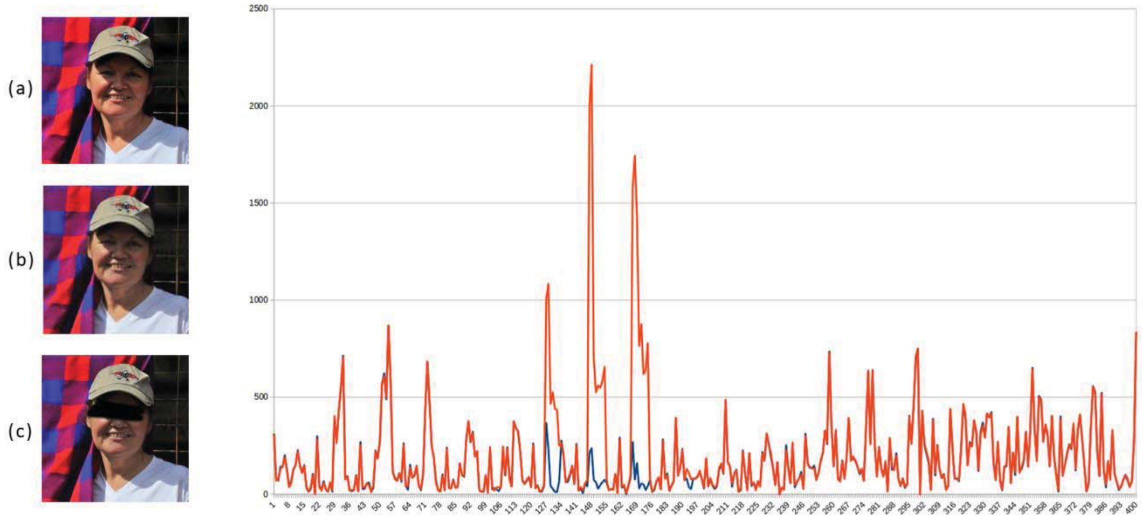

Figure 3: Image and its signal – first example (image from 123RF)

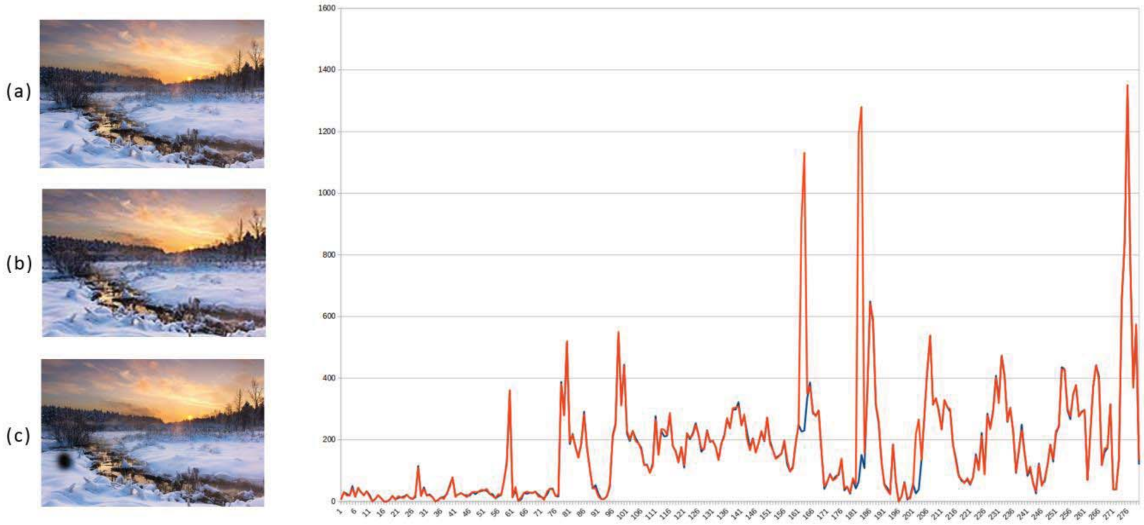

Figure 4: Image and its signal – second example (image from DeviantArt)

Figure 3 represents the signals extracted from the comparison of an image (a) and its thumbnail (b) (in blue), and from the same image with a modification (c) and the original thumbnail (b) (in red). We can easily see the high variations on the graph, due to the modification. However, Figure 4 shows that it is not always that obvious with the last high variation which is common to the two comparisons.

2.3 Image clustering

The previous parts introduced two different methods to measure the difference between an image and its thumbnail. The use of thresholds leads to different measures, and the goal is to find the threshold values which lead to the optimal results. However, the measures which have presented previously, describe the difference between an image and its thumbnail only, but this value needs be analyzed to decide whether the image and its thumbnail represent the same scene or not. This is a classical problem of data clustering, where the goal is, from several measures, to determine to which group an image belongs: the clean group with images which represent the same scene as their thumbnail, or the unclean group.

Two tests sets of respectively 494 and 582 images were available for the tests: a set with images only from the clean group, and the other set with images from the unclean group. These images were manually sorted and mainly gathered from Deviantart and Tumblr. These two sources present the advantage to be easily crawled and to contain images from various natures, from professional images to selfies, including computer-made images and other digital drawings. Because only a small number of images were representing a different scene as their thumbnail, we were forced to create a part of the images for the unclean set. We tried to vary a maximum the nature of the introduced modifications.

A k-means clustering (and similar methods) was first considered, but the statistical properties of the two sets are too similar to have an efficient clustering method. Indeed, even if the average difference of the two sets is relatively apart, the standard deviation of the two sets are too important, and, as a consequence, the joint area of the two distributions is important too.

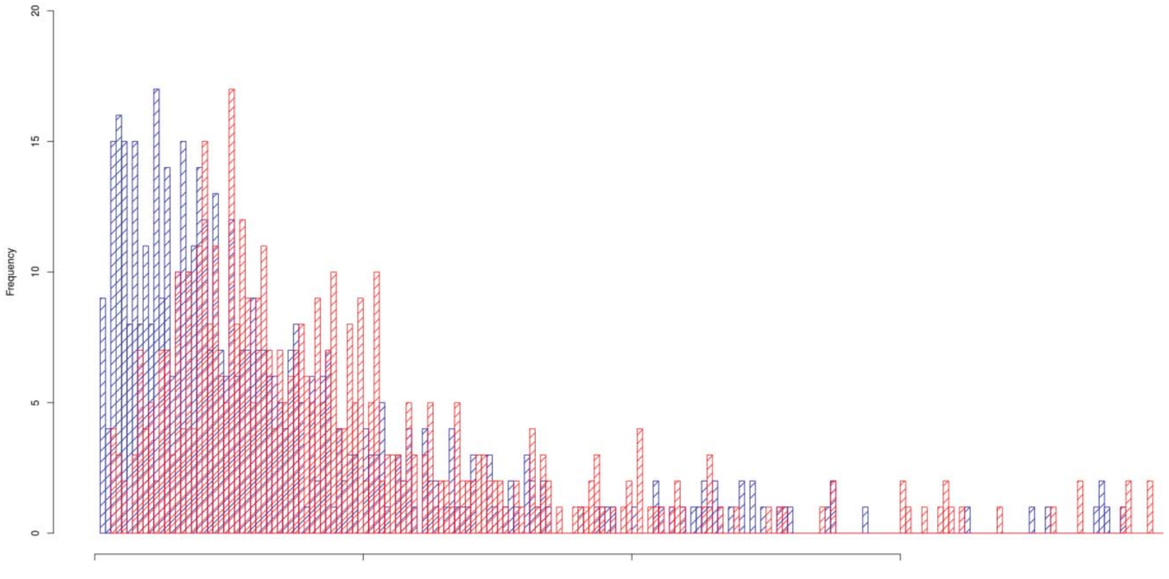

Figure 5: Distribution of differences for pixel based comparison

Figure 5 shows the distribution of differences of the two tests sets of images using DCT comparison. In red, the clean set and in blue the other set. The two distributions are too close to use the k-means method for the clustering.

Finally, the selected method consists in using of a fixed threshold with the pixel per pixel comparison. The optimal threshold is obtained with a dichotomy, and leads to a sensibility of 91% with 14% of false positives. However, these results can be maladjusted to an operational context. Indeed, the obtained results shows that only 10% of the gathered images contain a thumbnail and 10% of these images represent a different scene as their thumbnail. Thus, even with a false positive rate of 14%, during the analysis of a large number of images, the major part of the images detected as different will be false positives, due to the rareness of image different from their thumbnail. This problem is accentuated by the fact that every type of differences is searched, from the watermark to the complete change of scene, so the threshold must be able to detect these changes. If some changes are not desired to be detected (for examples watermarks), it is possible to adapt the threshold to highly decrease the number of false positives since watermarks have a low incidence on the difference measured between the image and its thumbnail.

The various results obtained show the complexity to obtain better results with the proposed tools, mainly because of the heterogeneity of the images. Images came from various sources, are produced by various software and cameras, and therefore have various properties (size, quality …). The size of the images can vary with a factor of 100, the quality parameter is generally between 50 and 100, and even with the same quality factor, and the quantization tables can be different from an image to another one. Thumbnails do not have a size limit or a quality or creation method recommended by the EXIF standard. It is indeed common to see a thumbnail with a quality bellow 50. All these parameters cannot be controlled by the user, and the preprocessing step needs to deteriorate the quality of the image (or the thumbnail) for the comparison, which inevitably led to a loss of information. Moreover, as the goal of this project is to analyze a large number of images, the use of complex zoom algorithms is excluded for time reasons.

In order to improve the comparison efficiency, a classification function which separates the images according to the nature of their differences has been implemented. This function aims to detect rotations, crops, color changes, colored stripes and heavy differences between an image and its thumbnail. The goal of this classification is to facilitate the viewing of the results. If the program detects a heavy difference, a set of controls for the classification is engaged. The first type of difference checked is rotations, then the algorithm looks for black or white stripes at the top and bottom – or on both sides – of an image but not on the other, and finally for color changes.

For the rotation detection, the image is simply rotated 90 degrees, and the difference is computed again, three times. If the difference once indicates that the rotated image is identical as the thumbnail, the image is tagged as a rotation and sorted in consequence. The detection of stripes checks if the three first and last rows of an image are only composed of black pixels – or only white ones – and that we find a stripe on the thumbnail but not on the image – or the opposite. If the result is negative, the image is turned through 90 degrees and the operation is repeated. The detection of color changes is done by converting the image and the thumbnail in gray-scale images and applying histogram equalization. Then the two images are compared again to check whether they are identical or not. Each detection step is executed only if the previous result is negative. The order of steps is not insignificant: in our sets, we found more rotations than colored stripes and more colored stripes than color variations.

The previously described methods were mainly tested on images from Deviantart and Tumblr. These two sites were used because the images can easily and quickly be gathered, and they have a high probability to contain metadata. Indeed, some websites can change the posted images, for example by deleting all the metadata including the thumbnail (e.g. Facebook). The images are collected with a web crawler which maps the website and analyzes the source code of every HTML web page found to search for some particular regular expressions which match JPEG images.

The nature of the visited websites for a larger-scale test also imposed to anonymize the connection to the target website. To do so, the TOR network has been used. In order to collect the images more quickly and without being banned from the website because of a potential huge number of requests, multiples instances of TOR have been used and the image gathering process is distributed through these instances. Thus, each TOR instance is parameterized to generate its own routes, and therefore to have a potential unique exit node and the intensive crawl is hidden. Moreover, as the crawler is based on the analysis of HTML pages, it is possible to analyze images from .onion websites.

3. Large scale tests

Once the calibration for the image analysis has been completed, some larger scale tests were run on various websites. The nature of the analyzed targets involved some minor changes in the collection of data. Indeed, as JPEG images can be extracted from PDF files, the crawler also download every PDF document found. Moreover, the project does not only compare an image to its thumbnail. It also reads all metadata to search important and useful information – for a forensics point of view – especially GPS coordinates, camera maker and model. It is not rare to find this information on regular websites.

We consequently collected images from various websites (news, social networks, galleries …) to have enough content for validation tests for the classification. On traditional websites, more than 90,000 images were collected. Among all these images, 9,098 of them contained thumbnails. Among all these images with thumbnails, 1,535 are different from their thumbnails. 507 images contain stripes, 178 are rotations and 20 are color changes. These results concur with the ones obtained by Murdoch and Dornseif. It is important to note that the results can vary a lot from a website to another as they can have different policies regarding to metadata.

Then, the scans were oriented to Islamist and mafia websites for digital forensics purposes. The visited websites are either HTTP/HTTPS websites or .ONION ones (darkweb). The obtained results are less interesting as the ones obtained on 'traditional' websites because the users of these sites are aware of the problems related to metadata, and often use tools to delete this information. We therefore rarely find images with metadata, and fewer images with thumbnails. More than 11 websites were analyzed and 32,682 images collected. Only 751 of these images contained thumbnails and 13 of these images were different from their thumbnail. Figures 6 and 7 are some example of the collected images which are different to their thumbnails – (a) is the image and (b) the thumbnail.

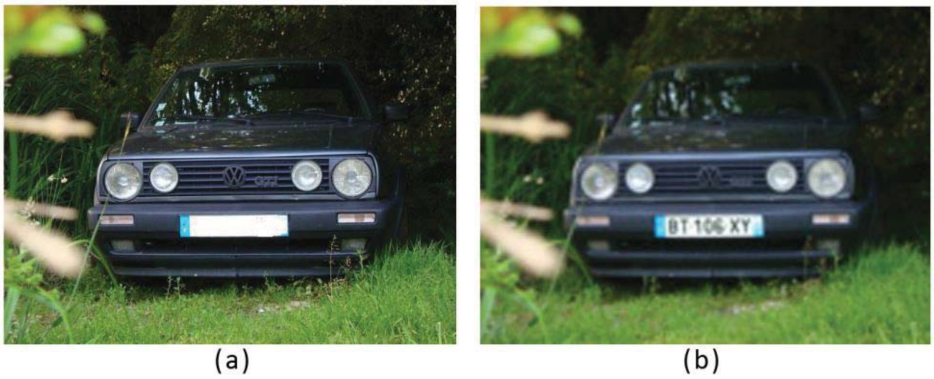

Figure 6: Example of readable license plate on the thumbnail (image from hiboox.com)

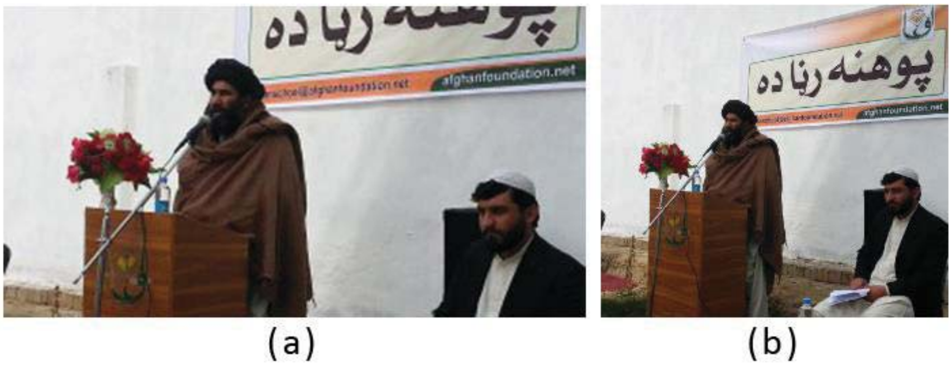

Figure 7: Example of readable text on the thumbnail (image from Islamic website)

Some PDF files were also gathered. Among these 50 files, 736 images were extracted, but none of them contained thumbnails. However, the metadata indicated the platform and the software used to edit this file.

The previously described methods were successfully applied to the scan of a list of targets, and some useful information – like GPS location or part of images that were hidden in the thumbnail – were found. The purpose of this project can also be easily adapted to penetration testing, thanks to the information an image can reveal through its thumbnail or its metadata.

4. Conclusion

The importance of metadata is well established, and JPEG images can be a wealth of information. In this paper, we have explained how a pixel-based comparison could detect a difference between an image and its thumbnail. Other methods based on the edges present in an image also shown their efficiency. However, the low quality and the small size of thumbnails makes the task harder since images must be preprocessed, and information is inevitably lost.

The large scale tests show that some peoples are well aware of the potential data leak metadata represent, and know how to protect themselves. Although up so far no images with interesting information in the thumbnail were found on Islamic or mafia websites, we found numerous images with GPS location or camera model and serial number. On regular websites, it is easier to find images which are different from their thumbnail or which contains metadata, since peoples are less cautious and aware. We therefore find images of cars with hidden license plate on the image, but readable one on the thumbnail.

Some software are seems aware of this problem and compute a new thumbnail when the image is edited, like “Photoshop” or “The Gimp”. However, some others, like PhotoFiltre – versions above 7 were not tested – simply keep the thumbnail as it is.

The next step of the study will be to improve the clustering by developing new methods. The use of Markov chains to guess the supposed value of the next block may lead to a new measure of the difference. Moreover, provided we have enough measures of different nature, the use of a linear classifier like a Support Vector Machine could improve the clustering.

References

Bosc, Heitz, Armspach et al, (2003) “Automatic change detection in multimodal serial MRI: Application to multiple sclerosis lesion evolution”, Neuroimage, Vol 20, pp 643–656.

Castoro, R. (2012) “We Are with John McAfee Right Now, Suckers”, [online], Vice magazine, December, http://www.vice.com/read/we-are-with-john-mcafee-right-now-suckers.

Fang, Chen and Fuh (2003) “Automatic change detection of driving environments in a vision-based driver assistance system”, IEEE Trans. Neural Netw., vol. 14, no. 3, May, pp. 646–657.

Harvey P. (2016), 'Exiftool', [software].

Hussain M., Chen D., Cheng A. et al. (2013), “Change detection from remotely sensed images: From pixel-based to object-based approaches”, ISPRS Journal of Photogrammetryand Remote Sensing, Vol 80, pp 91-106.

JIETA (2012) “Exchangeable image file format for digital still cameras: Exif version 2.3” CP-3451C.

JPEG (1994) “Information technology – Digital compression and coding of continuous-tone still images”, ISO/IEC 10918.

Krawetz (2007) '*A Picture's Worth: Digital Image Analysis and Forensics*', Hacker Factor Solutions, August 2007.

Murdoch and Dornseif (2004) “Hidden data in Internet published documents”, University of Cambridge and University of technology Aachen, December, pp. 35.

Radke, Andra, Al-Kofahi et al (2005) “Image Change Detection Algorithms: A Systematic Survey”, Rensselaer Polytechnic Institute, New York.

Rodewig C. (2012) “Geotagging poses security risks”, [online], The United State Army, March, https://www.army.mil/article/75165/Geotagging_poses_security_risks/.

Schwartz Catherine (2003), [online], https://en.wikipedia.org/Catherine_Schwartz#Personal_life.