Forensics Project

Tool for fake/edited/photoshopped image detection

Submitted By:

Sejal JAIN

Table of Content

Introduction:

Today the ease of access to techniques and tools for editing images has led to numerous edited images in the digital domain. Detection of edited images is a very crucial step in courts and forensic studies. Numerous techniques are available and the research is still ongoing in the lines of detecting forged images. In this project I have developed two tools to detect whether an image is edited or not, usable on a linux environment. The first tool is focussed on images where the original is not available to compare. In this tool along with the image metadata, the image content is checked for copy move edits. In the second tool, I have made an effort in detecting whether an image is a modified version of the original by calculating a distance and show which area was edited.

Implementation:

The implementation of the tool comprises of techniques based on cloning, jpeg compression, and, image matching and image exif metadata. The idea of cloning is taken from recommended Hany Farid's survey paper[1]. In this technique the area in the picture is copied and positioned to the different area in the image[2]. There are many techniques to implement jpeg compression but the one that is used in this tool is Error Level Analysis(ELA). When we re-save an image, it is done at 95% compression and then we evaluate the difference by comparing it with original image. Image analysis can be done based on eyes, reflections and image's metadata. In order to get the information on any image first thing that comes in mind is Exif metadata of the image. This data contains all kind of information likes the timestamp of the image, software used to photoshop or edit the image and if the image is original, it contains the model of the phone, camera.

But, exif data can be easily manipulated so it is very hard to rely on this information.

Project Plan and Execution:

While designing the tool, I was primarily focused on detecting the forgery for single images. So the technique I used was based on exif metadata. I installed PIL library(Python Imaging Library) in order to support the image file formats. To detect if image is fake or not I compared different images that includes photoshopped, edited and original images in order to see the difference between their metadata. For every image there is different values for 'DateTimeOriginal' and 'software' keys. Then my code[3] was able to differentiate between the fake and original images. But sometimes exif data may also fails in depicting if image is original or not because exif data can can be manipulated.
As exif data cannot be trusted I tried cloning technique[2]. This technique is based on copy-move forgery detection(CMFD)[6]. In this algorithm it first computes Discrete Cosine Transform(DCT) of each 16*16 sliding window in the image. Each of the coefficients is then quantized with an extended JPEG quantization matrix. After quantizing each blocks, they are inserted as rows into matrix and then sorted. As the adjacent rows matching the matrix are identified then for each match a shift vector is computed. There is a counter kept to check the numbers of times shift vector is computed and if shift vector is greater for some specified threshold, that particular block corresponding to that shift vector are viewed in the original image which may possibly be the forged region. Higher is the threshold of shift vector, more blocks of the image are copied and that region is marked as forged region. Detection of forge region also depends on parameter called quality factor that weights the quantization matrix. Lower is the quality factor, the less likely it is to have false matches.[7]

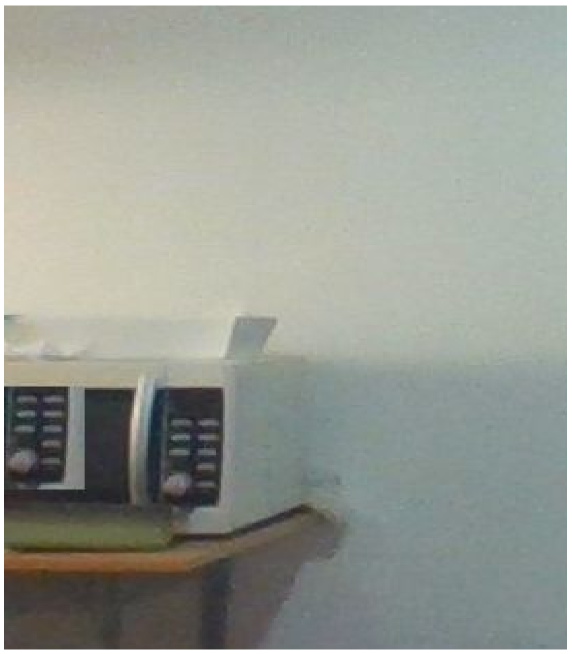
After implementing the tool for single images, I wanted to try other approaches to see how it works out when we compare two images. Compressing the image in jpeg is very lossy as many pixels are altered. In ELA[4], the image lost its originality because by each resaving of image introduces some error which is not linear. Like if we have an image that is 89% original and when we resave the image than it is equivalent to one time save of 70%. ELA work as by willingly resaving the image by some percentage and then evaluating the difference between the two images(original and edited). When there is no change in the picture then the pixels reach to its local minima for error based on the percentage level and when there is large amount of change then the pixels are not at their local minima. This algorithm was nicely working on photoshopped images instead of slightly edited images.

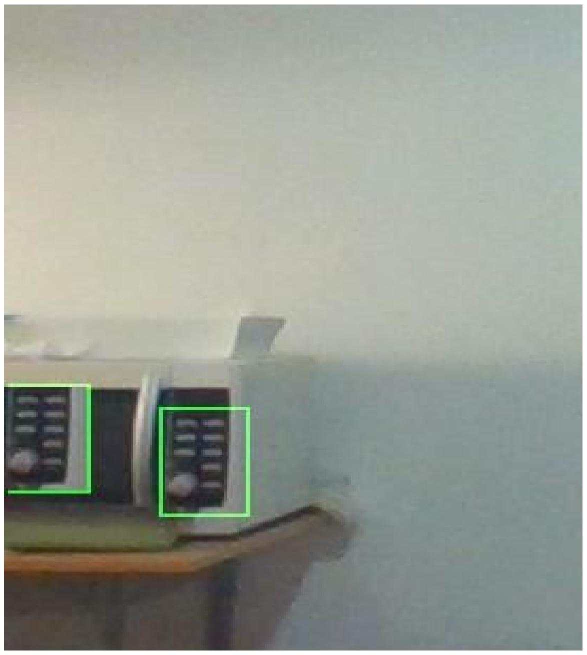
Difficulties faced:

- In the exif data implementation, the problem was it does not detect all the softwares used to edit/photoshop the images. For some of the softwares it does not even view the exif data.
- ELA method was not working for all kind of edited images. I assumed that image should be edited more than 20%, then only it will be able to detect.
- I also tried to implement a technique using Principal component analysis but the success rate of detecting image was nearly 30% and the algorithm but not at all trustworthy.
- Many techniques are mentioned in Hany Farid’s survey paper. I tried to implement Double Jpeg technique but the algorithms were not clear enough to implement this technique on the images. I was firstly focusing on providing input as single image but the algorithms were very complicate to implement and also the implementation was not available.

Experiment Results :

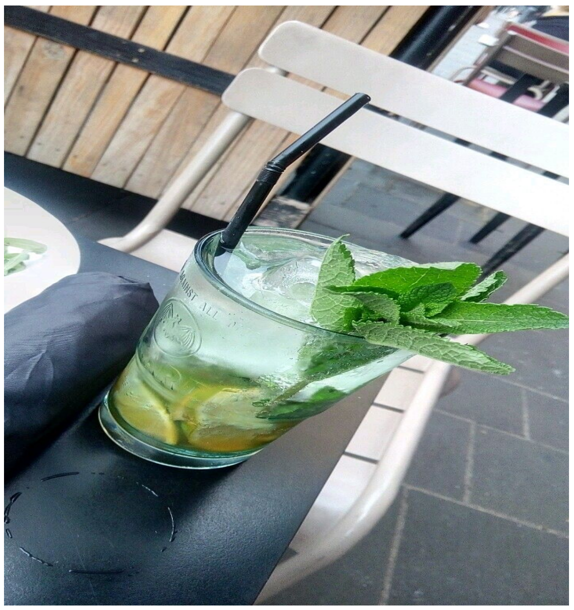
Using ImageForensics tool for single image:

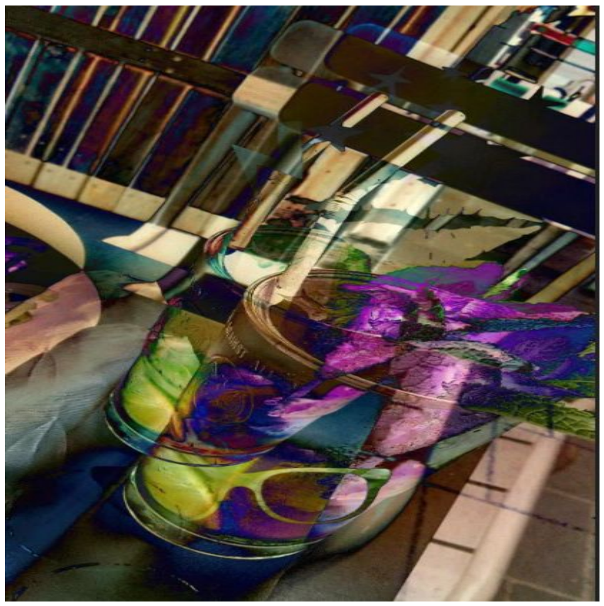




Input image Output Image

Using ela tool for two images:





Input image Output Image

Conclusion :

I was able to implement the algorithm to detect the fake images. But sometimes it does not work as expected. Also the copy-move detection implementation takes a very long time for large images. The future work can be to optimize the code so that it can work perfectly on any kind of image with a higher success rate. Along with the performance improvement, we also need to implement more techniques and keep it updated for building a better tool for detecting the forged images.

References:

[1] https://ieeexplore.ieee.org/document/4806202/?arnumber=4806202

[2]https://github.com/BradySheehan/copy_move_forgery_detection/blob/master/paper/final_paper.pdf

[3] https://stackoverflow.com/questions/4764932

[4]http://blackhat.com/presentations/bh-dc-08/Krawetz/Whitepaper/bh-dc-08-krawetz-WP.pdf

[5] https://gist.github.com/ewencp/3356622

[6] http://www.ws.binghamton.edu/fridrich/Research/copymove.pdf

[7] https://github.com/rahmatnazali/image-copy-move-detection