INDIAN INSTITUTE OF TECHNOLOGY GANDHINAGAR

EE-605, DIGITAL IMAGE PROCESSING

PROJECT REPORT

Name:
NIDHIN HARILAL

Roll Number:
17110092

Contents

Passive detection of doctored JPEG image via block artifact grid extraction

Nidhin Harilal
IIT Gandhinagar
nidhin.harilal@iitgn.ac.in

1 INTRODUCTION

With the advent of image processing softwares, editing images have become very easy. Even though, these softwares can be utilized for fascinating applications but, this technique also brings some legal crisis. Because of the ease of editing images, it has become very hard to discriminate actual images with one that has been edited or tampered. The threat of images being fake has become increasingly apparent in our lives.

1.1 Image Forensics

The techniques for detection of image forgery can be broadly divided into 2 categories:

(i) Active Protection Techniques

(ii) Passive Detection Techniques

Active protection techniques include methods such as embedding some watermarks or signatures into original unaltered pictures as soon as they are clicked, proposed by [5]. So, If pictures were tampered with then the embedded watermark will get altered too. Since, most of the current image capturing devices do not implement this method, therefore, active techniques becomes ineffective. Passive Detection techniques, on the other hand do a high order statistical analysis on the image to find out the forgery in the image.

In this project, I am analyzing a Passive technique based algorithm proposed by [1] for detection of doctored JPEG Image. Their algorithm is based on Block Artifact Grids (BAG) which are considered a defect of JPEG compression.

1.2 Block Artifact Grids (BAG)

During lossy JPEG compressing, the involved blocking process would introduce some horizontal and vertical breaks into image. These breaks are periodic and uniformly form an 8×8 grid, which are known as block artifact grid (BAG).

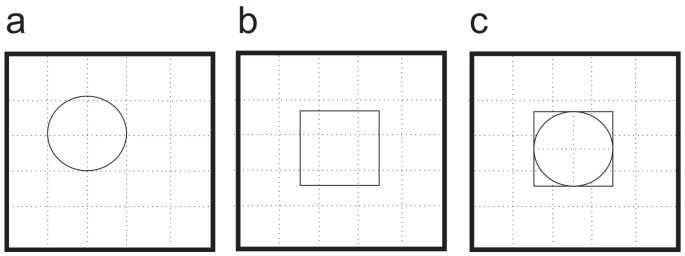


Figure 1 An Example[1] of BAG mismatch: (a), (b) are original images, (c) is the synthetic image.

When common methods of Image forgery such as Cropping, Painting, Copy-Pasting are done, then these bag gets deformed and are misplaced. The idea behind the algorithm is that to extract these BAGs which would lead to finding the doctored areas or regions in the image. **Figure.1** shows one such example of how copy-pasting leads to a BAG mismatch. All the further analysis has been done with the help of open-source MATLAB *Image-Forensice*[6] toolkit.

2 BAG EXTRACTION ALGORITHM

Since, JPEG Block artifact grid is formed by the horizontal lines and vertical break lines with a periodicity of 8 at both directions. Thus, the approach is to extract the weak horizontal and vertical edges with periodicity of 8 separately, and then combine them as the BAGs.

Figure 2 A sample Image S on which BAG extraction algorithm will be performed.

In the following sections, the raw image which is shown in **Figure.2**, denoted as S would be used to find the BAGs, the weak horizontal edges that are extracted as E_h, the weak vertical edges as E_v. Then final lines of BAG with periodicity 8 are represents as G_h and G_v for vertical ones.

2.1 Extracting Weak edges

Firstly, an absolute second-order difference is calculated to extract the weak horizontal and vertical edges. Consider $S(y,x)$ to be pixels in the doubtful image S and $d(y,x)$ the elements in absolute second-order difference D, which can be calculated with the formula:

$$d(y,x) = |2S(y,x) - S(y-1,x) - S(y+1,x)| \qquad \text{(For Horizontal weak edges)}$$

$$d(y,x) = |2S(y,x) - S(y,x-1) - S(y,x+1)| \qquad \text{(For Vertical weak edges)}$$

Generally, block artifacts are much smaller than 50. So, all differentials larger than 50 are ignored. After that, In order to equalize amplitudes the image, a local median is reduced from each elements. For the **below equation**, the function $Mid[A\{\}]$ returns the median of set $A\{\}$. and $e_h(y,x)$ are elements in image E_h.

$$e_s(y,x) = \sum_{i=x-16}^{16} d(y,i) \qquad \text{(For Horizontal weak edges)}$$

$$e_s(y,x) = e_s(y,x) - Mid[e_s(y,x)|y-16 \leq i \leq y+16]$$

In the similar way as above, we extract the weak vertical edges E_v

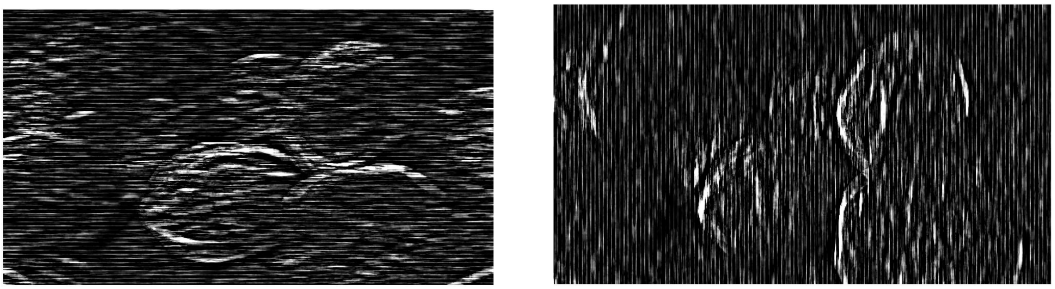


Figure 3 *Left Image* shows the extracted E_h and *Right Image* shows the extracted E_v

Figure.3 shows what happens when the above described equations are applied to an image. Observe here that apart from the regular BAG lines, some other random edges are also present due to the strong object boundary. This shows that we need to further process these edges.

2.2 Enhancing edges and Extracting lines

As observed from the **Figure.3** that the second-order difference causes some few random strong lines due to the influence of the object boundaries. For extraction of clear BAGs, these are considered as *noise*. To further enhance the BAGs from these noise, the weak horizontal edge image E_h and weak vertical edge image E_v are further periodical median filtered with the below formula:

$$G_h(y,x) = Mid[\{e(i,x)|i = y-16, y-8, y+8, y+16\}] \quad \text{(For Horizontal lines)}$$

$$G_v(y,x) = Mid[\{e(y,j)|j = x-16, x-8, x+8, x+16\}] \quad \text{(For Vertical lines)}$$

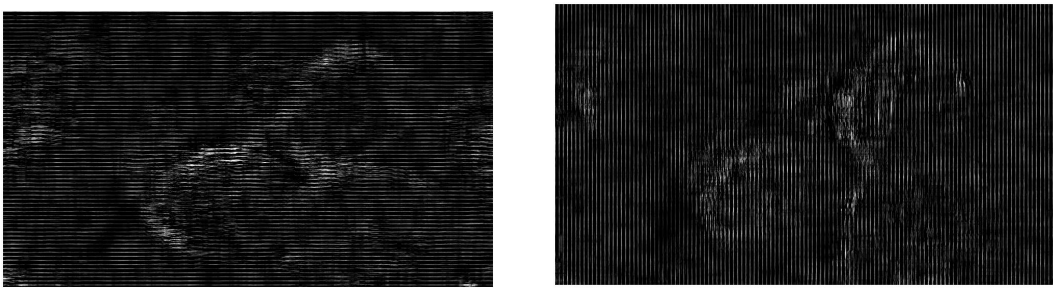


Figure 4 *Left Image* shows the extracted G_h and *Right Image* shows the extracted G_v

Figure.4 shows the enhanced and noise removed generated image from the weak extracted edges when above described formula is applied. In the above formula, five elements with spacing 8 are used in the median filter. Thus, the strong BAGs and weak BAGs are smoothed, and strong burst lines without period 8 are removed. Ideally, The more number of elements are used in the median filter, the extracted BAGs gets better but, on the downside, more doctored area may be missed.

2.3 Obtaining the final BAG image

In order to to finally obtain the required BAG image by the above extracted horizontal line image G_h and vertical line image G_v, both these are added:

$$G(y, x) = G_h(y, x) + G_v(y, x)$$

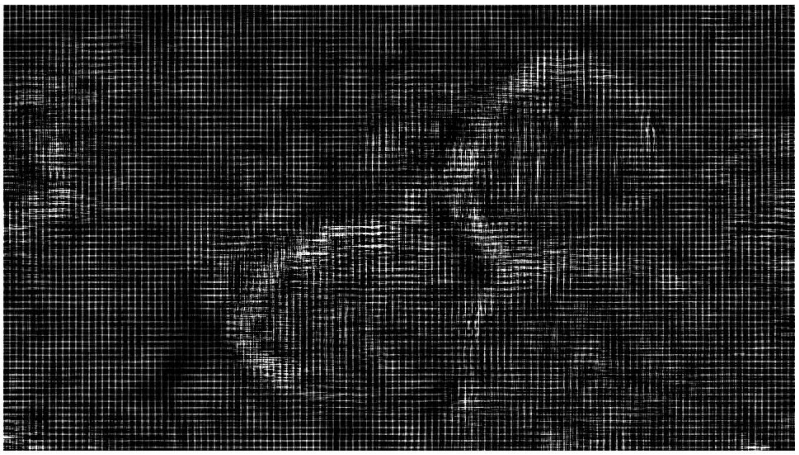


Figure 5 This shows the Final extracted BAG from Image S

3 BAG Extraction VS Quality Factor (QF)

This experiment is to analyze the quality of BAGs extracted versus the Quality Factor(QF) on which JPEG image has been compressed. **Figure.6** shows the comparison of BAGs of the same image with Quality Factor. The image used for comparison is the same as that used by the work[1]. The image has been compressed to JPEG with different Quality Factors using an online tool (`www.imgonline.com`). The results obtained in **Figure.6** are similar to the results by this work[1].

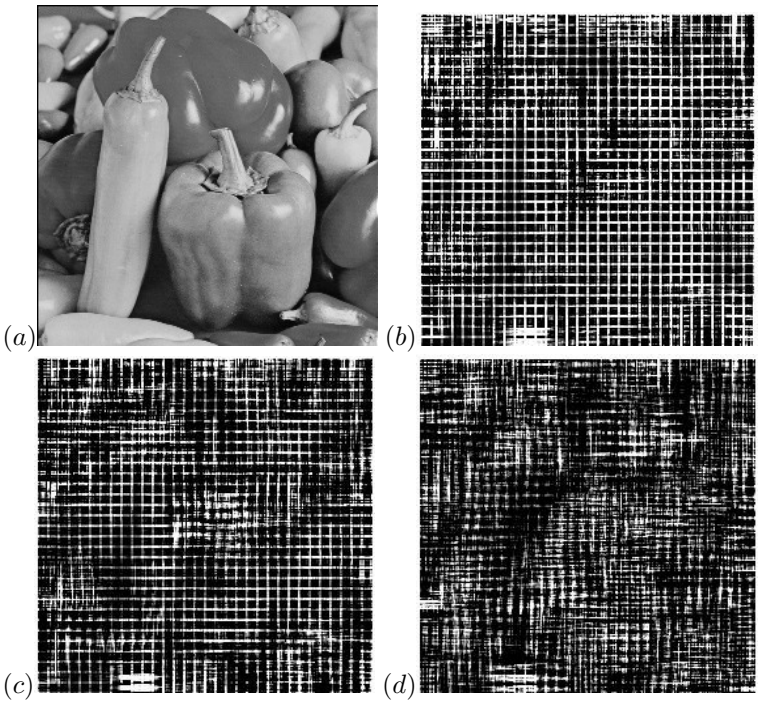


Figure 6 Comparison of extracted BAG images for different JPEG compression QF: (a) original image capsicum, (b) extracted BAG image (QF = 50), (c) extracted BAG image (QF = 75) and (d) extracted BAG image (QF = 85)

It can be clearly observed from **Figure.6** that with increasing QF the quality of BAGs extracted decreases. Therefore, the proposed algorithm for BAG extraction works well for images with low QF values of JPEG compression.

4 IMAGE FORENSIC VIA BAG EXTRACTION

4.1 Detection of Image cropping

The technique of Image cropping is to conceal or reduce the dimensions of the image. The problem here is that, with the help of cropping people may conceal the object to show only the information that they want.

BAGs here can play a very significant role in detecting such cases. Generally, a person who does the cropping doesn't take care of the BAGs involved in the image. Therefore, there is a very high chance that image was not cropped properly along the BAG lines. This produces an incomplete BAG lines around the sides.

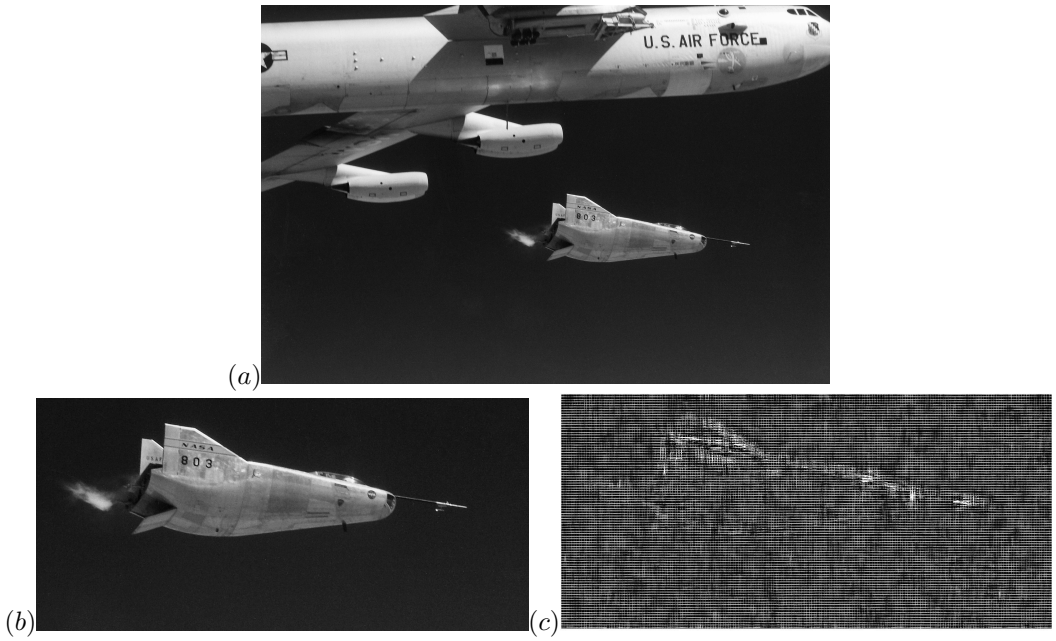


Figure 7 Detection of cropped image: (a) original image, (b) cropped image and (c) extracted BAG image.

Figure.7 shows the detection of cropped image. The image used here is the same as the image that this work[1] has used in the detection of cropping section. This image has been scraped from the NASA webpage[4]. The forgery is that smaller plane has been cropped out and compressed in JPEG using *MSPaint.exe*.

The (c) part in **Figure.7** shows the extracted BAGs of the cropped image. If these BAGs are looked closely, then it can be observed that the blocks found in the *bottom* part of it are incomplete. This means that image has been cropped from in-between the block, hence, detecting the forgery of cropping. If compared to the work[1], they have obtained very less number of BAGs compared to the above obtained images. One reason may be that the image which I have used has been directly downloaded from the NASA with the highest resolution and the work[1] may be using a lower resolution image.

4.2 Detection of Painted Image

The technique of Image painting is to conceal or hide certain components of the image. The problem here is that, with the help of painting people may conceal the object to show only the information that they want.

The basic idea is that the region of the image that is painted is uncompressed. So, there will be no BAGs extracted from the painted region. However, a legitimate natural images always conosist small noises, and BAGs exist after JPEG compressoion, even for a dark background. So if there are large blanks observed in the extracted BAG image, then we can conclude that these region is doctored.

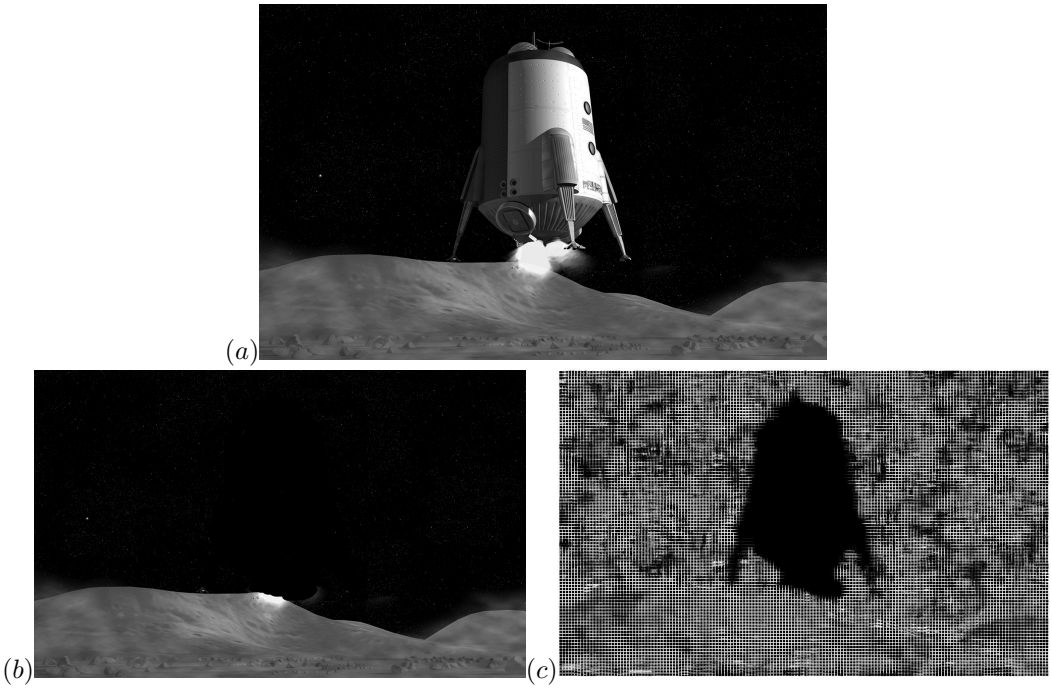


Figure 8 Detection of painted image: (a) original image, (b) Painted image and (c) extracted BAG image.

Figure.8 shows the detection of painted image. The image used here is the same as the image that this work[1] has used in the detection of painted image section. This image has been again taken from the NASA webpage[3]. The forgery present here can be seen in part (b) as the region of rover has been painted with black such that it vanishes. This image has been painted compressed to JPEG using *MSPaint.exe*.

The (c) part in **Figure.8** shows the extracted BAGs of the painted image. It can be clearly observed that the region where rover was present (which was basically painted) does not contain any BAGs, hence, detecting the forgery of painting. If compared to the work[1], the results are very similar.

4.3 Detecting of copy–pasted image which copied from a JPEG image

The technique of Copy–paste is a very common used method for editing an image. The idea here is that when a slice of some another or even the same image is copied and pasted, then

the matching of BAGs are hardly considered. So if the extracted BAG image has mismatching, then, it can be treated as possible clue of forgery. Since, checking of BAG mismatching is very subjective and not a very good metric to predict the forgery. This work[1] has introduced a marking procedure to mark BAG mismatching areas routinely.

Since, it is evident that JPEG happens on a periodic 8×8 blocks. So, If there is a BAG mismatch case then, the lines inside this 8×8 block must be locate at abnormal positions, i.e, there will be BAG lines in the 6×6 center-matrix. In-order to detect such regions, this work[1] has used the following formula:

$$b = Max\{\sum_{i=2}^{7} a(i,x)|2 \leq x \leq 7\} - Min\{\sum_{i=2}^{7} a(i,x)|x = 1,8\}$$

$$+ Max\{\sum_{i=2}^{7} a(y,i)|2 \leq y \leq 7\} - Min\{\sum_{i=2}^{7} a(y,i)|x = 1,8\}$$

where b denoted the updated BAG locations of the doubtful image, $Max[A\{\}]$ and $Min[A\{\}]$ returns the maximum or minimum value of set $A\{\}$.

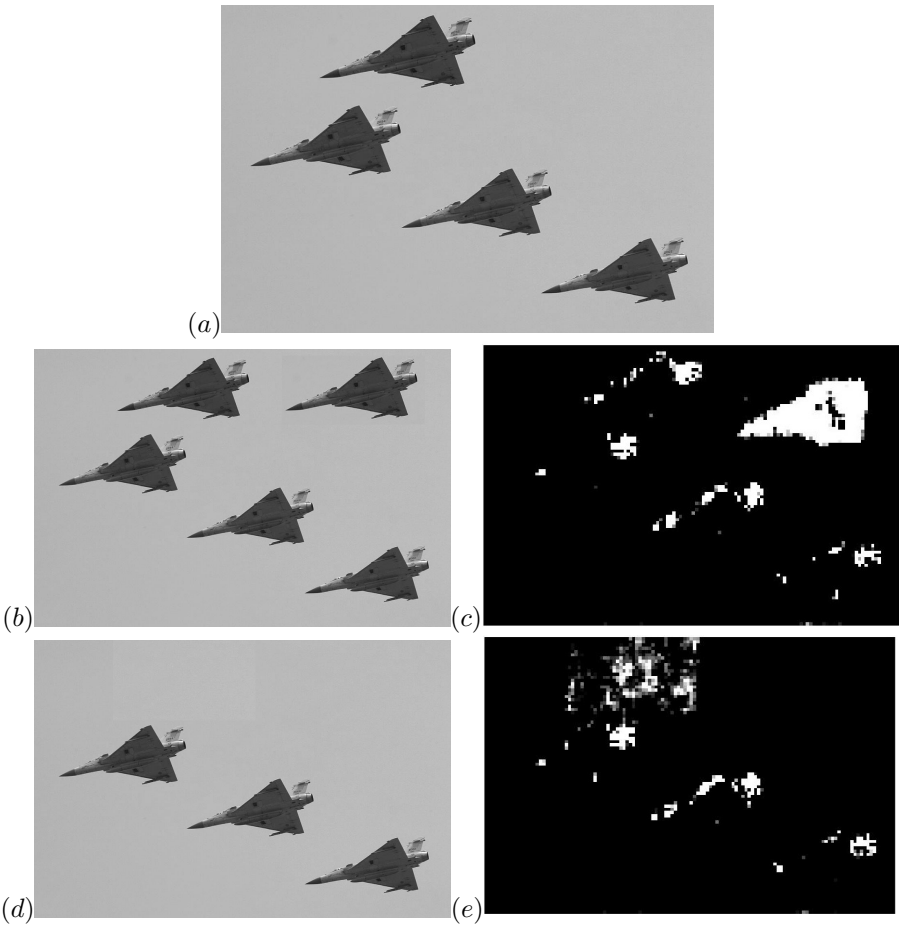


Figure 9 Detection of copy–pasted image: (a) original image, (b) doctored image with one airplane interpolated, (c) marked BAG image of (b), (d) doctored image with one airplane concealed and (e) marked BAG image of (d).

Figure.10 shows the doctored and the corresponding marked image. The image used here is the same as the image that this work[1] has used in the detection of copy-paste detection section. This image has been taken from the NASA webpage[2]. The forgery present here can be seen in part (b), one jet plane has been copy and pasted to the right side. And in part (c), one jet plane has been concealed off by copy pasting a slice of the sky to the plane. This image has been edited and compressed to JPEG using *MSPaint.exe*.

The (c) part in **Figure.10** shows the extracted BAGs of the painted image. It can be clearly observed that the region where rover was present (which was basically painted) does not contain any BAGs, hence, detecting the forgery of painting. If compared to the work[1], the results are very similar. In part c, it can be observed a large region of the jet stands-out. However, in part (e), even though the concealed region is getting white but there is also a little effect of other non-altered jets. The results obtained here for the part (c) are very similar to this work[1] and for part (e), there is a slight noticeable difference.

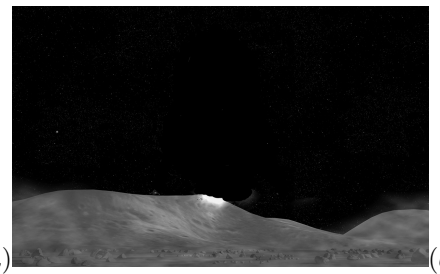
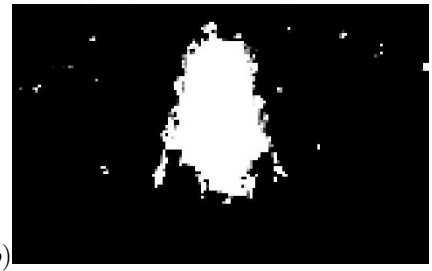
(a)  (b) 

Figure 10 Examples of marked BAG images: (b) marked image of Fig. (a)

Figure. 10 shows the marked image of the painted image which was used in section 4.2. It can be observed that the marking algorithm clearly segments the forged region.

5 SHORTCOMINGS AND ROOM FOR IMPROVEMENT

From the above sections, it can be observed that the proposed work[1] is very efficient in finding out the doctored JPEG Image. However, there are some key limitation that I found out during the analysis of this algorithm which are as follows:

- **Multiple JPEG Compression:** One of the limitations of this algorithm is about how much it is consistent with multiple JPEG compression. Consider an Image with a forged region, at the first compression, there would be a high chance of BAG mismatch. However if that same image is compressed again then new BAGs will get formed which will nullify the affect of the previous mismatched BAGs.

  After experiment with images compressed multiple times with different QFs, one of the important results can be observed in **Figure. 11**

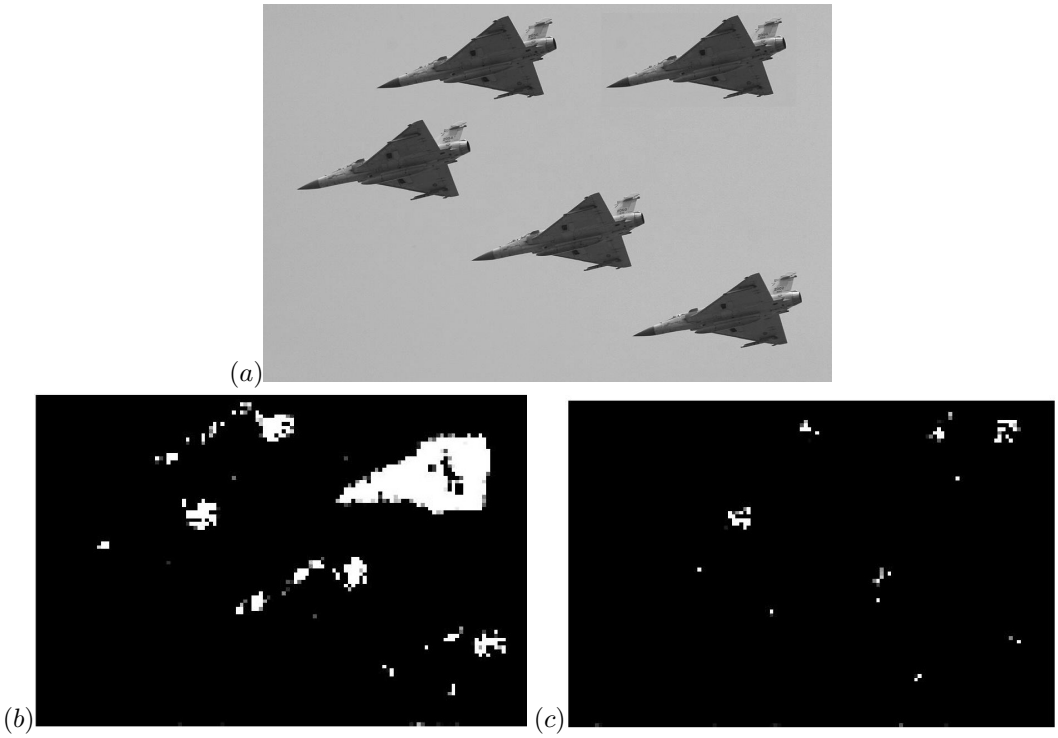


Figure 11 Multiple JPEG Compression marked image: (a) original Forged image with single compression, (b) BAG Marking of image with single compression(QF=75), (c) BAG Marking of image with double compression(QF=65 each time)

  It can be observed that after a double compression with $QF = 65$ each time, the effect of BAGs mismatch reduced significantly.

- **Careful Cropping** Since, the BAGs are periodic with a period of 8×8 blocks, after generating the BAG image, we could actually make a careful crop so that the crop coincides with the BAG boundary. With little patience, I was able to get such a cropped Image with matching boundaries as that of BAGs. **Figure.13** shows the BAG image of that carefully cropped image.

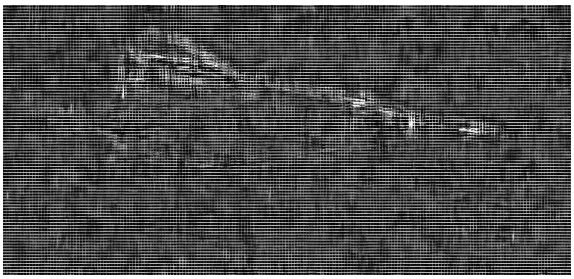


Figure 12 Careful Cropping: A carefully cropped image with matching BAG around the borders

- **Adding carefully choosen Noises** The idea behind this is similar to the Adversarial Attacks found to fool the Convolutional networks. By adding some carefully chosen noises to the doctored region, it migh be possible to fool the BAG algorithm to convince that there is no Tampering present in the image.
- **Efficiency** As observed, most of the computation time by the algorithm is spent on applying multiple median filters. If somehow, we are able to cut down the number of median filters or are able to design some special low computation cost filters which does the same job as these multiple filters do, then we would be able to cut down the computation cost significantly.

6 CONCLUSION

In this, I analyzed one of the Passive based technique for Image Forensics. This work utilized the Block Artifact Grids(BAG) which are the faults that happen due to JPEG Compression. This work has proposed an algorithm to extract these BAGs. Some of the experimental demonstrations in previous sections show the effectiveness of the proposed method to detect different JPEG image Forgery methods, such as cropping, painting, and copy–paste. Also, some of the key issues were addressed which showed the cases where the algorithm failed to detect the forgery. Further work can be done to address these issues and to optimize the current algorithm with a low computation cost.

References

[1] Weihai Li, Yuan Yuan and Nenghai Yu. “Passive detection of doctored JPEG image via block artifact grid extraction”. In: *Signal Processing* 89.9 (2009), pp. 1821–1829. ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2009.03.025. URL: http://www.sciencedirect.com/science/article/pii/S0165168409001315.

[2] NASA. *Air Show: 4 Dassault Mirage 2000*. URL: https://www.nasa.gov/multimedia/imagegallery/index.html.

[3] NASA. *Concept Moon Rover preparing to launch from Moon*. URL: https://www.nasa.gov/multimedia/imagegallery/index.html.

[4] NASA. *US Air Force two-plane side-by-side Image*. URL: https://www.nasa.gov/sites/default/files/images/328866main_EC71-2774_full.jpg.

[5] Q. Sun T.T. Ng S.F. Chang. “Blind detection of photomontage using higher order statistics”. In: *proceedings of IEEE International Symposium on Circuits and Systems, Vancouver, Canada* (2004), pp. 688–691.

[6] Markos Zampoglou, Symeon Papadopoulos and Yiannis Kompatsiaris. “A Large-Scale Evaluation of Splicing Localization Algorithms for Web Images”. In: *Multimedia Tools and Applications* (). DOI: 10.1007/s11042-016-3795-2.

7 Appendix(a)

7.1 Enlarged Version of the Cropped Image Detection from Figure.7

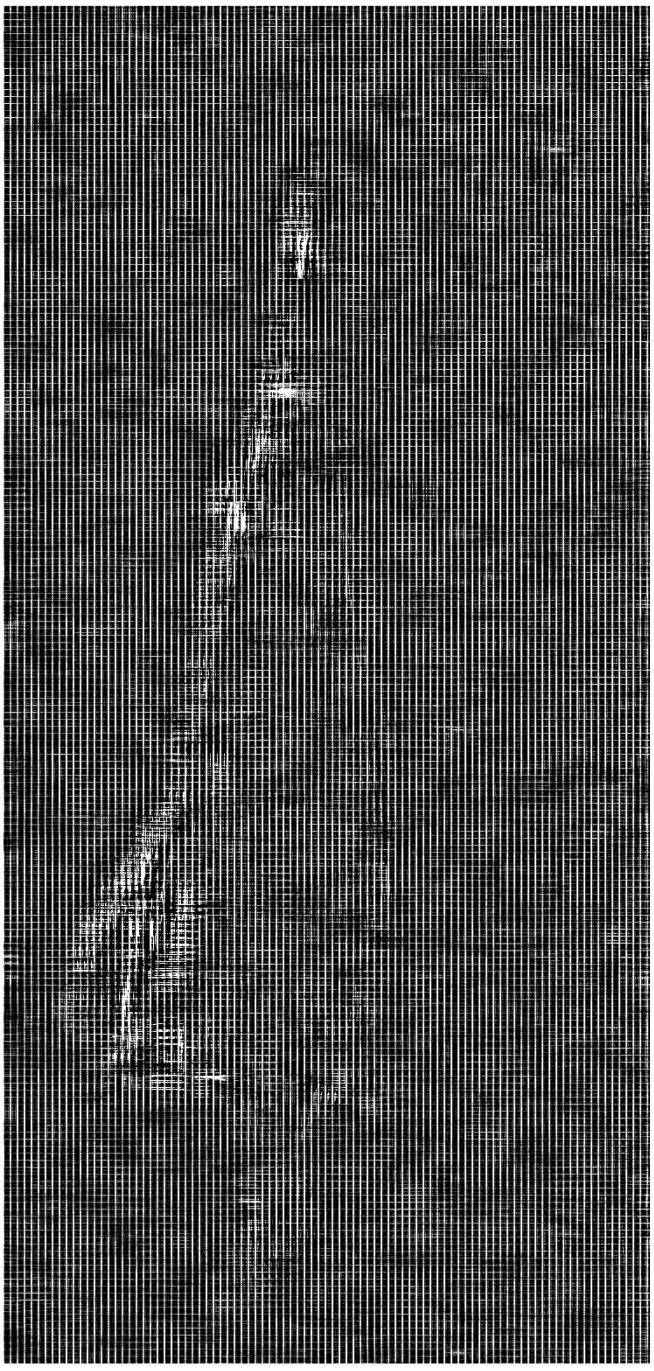


Figure 13 Careful Cropping: A carefully cropped image with matching BAG around the borders

8 Appendix(b)

This contains all the results and Figures of the work[1] in case for the comparison.

8.1 Quality of BAGs vs QF

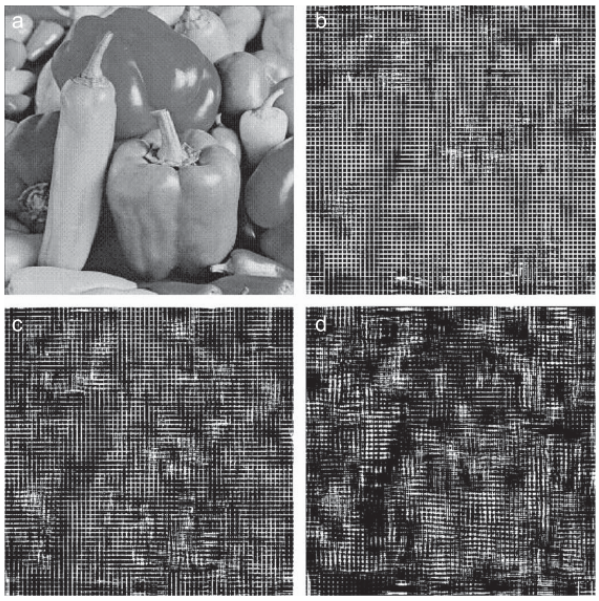


Figure 14 Comparison of extracted BAG images for different JPEG compression QF: (a) original image capsicum, (b) extracted BAG image (QF = 50), (c) extracted BAG image (QF = 75) and (d) extracted BAG image (QF = 85)

8.2 Detecting Cropped Image

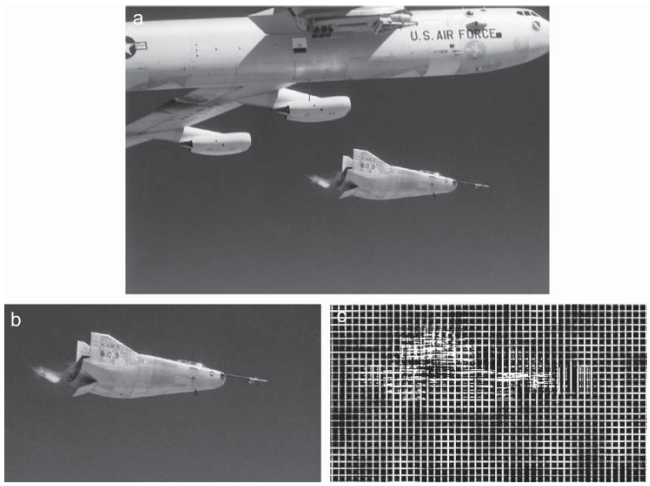


Figure 15 Detection of cropped image: (a) original image, (b) cropped image and (c) extracted BAG image.

8.3 Detecting Painted Image

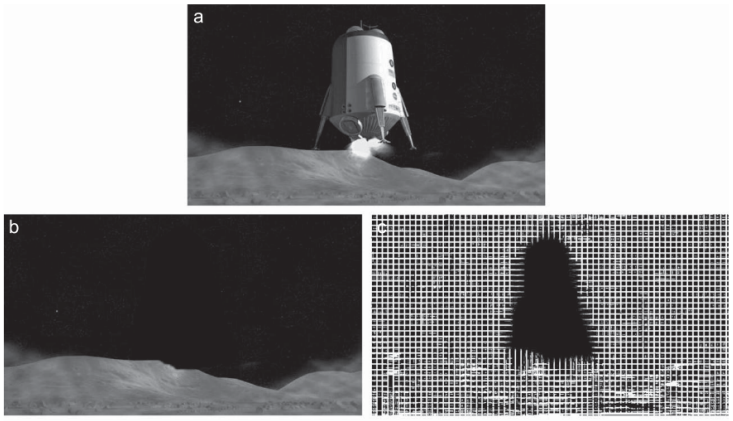


Figure 16 Detection of painted image: (a) original image, (b) doctored image by filling pour black, (c) extracted BAG image from (b)

8.4 Marking Images for Copy-Pasting

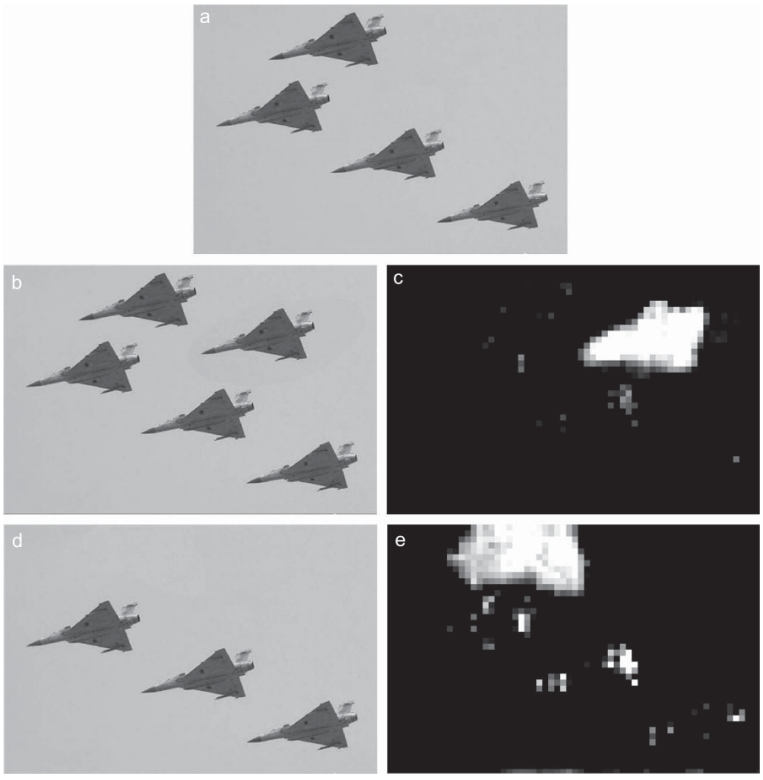


Figure 17 Detection of copy–pasted image: (a) original image, (b) doctored image with one airplane interpolated, (c) marked BAG image of (b), (d) doctored image with one airplane concealed and (e) marked BAG image of (d).