

# ECE3246 Security and Cryptography Assignment

Lew Zhi Yong, Ng Chin Kit

February 25, 2016

## 1 Discussion of Side Chosen

For a long time, photographs have been generally acknowledged as the most reliable and credible expressive media which are extensively used as proves of evidences in diverse realms including forensic investigations, medical imaging, scientific research as well as insurance claims, just to name a few. In conjunction with the rapid technology advancements, digital images have successfully substitute the functionality of conventional photographs in almost every existing application of photographs for various fields. For the sake of integrity in such fields, the authenticity of digital images is undoubtedly of utmost essential. However, due to the increasing availability of sophisticated and professional image editing and processing software tools such as Adobe Photoshop, PhotoPlus, Pixelmator and Corel Paint Shop, the practices of digital images tampering has become increasingly easier and common, thereby brings about severe threats to the credibility of digital images to be still accepted as proves of evidences to the extent that separating a tampered image from the genuine one has appeared to be rather difficult by just looking at it. In view of this worrying and up-set situation, we decided to stand by the light side in this image forensics war assignment with the aims to carry out research to identify several existing digital image tamper detection techniques and subsequently experimenting those techniques applicability in detecting the traces of image tampering on several tampered images so as to verify the integrity of digital images.

## 2 Description of Tampered Images



Figure 1: Faked Missile Test Image – Image Forgery IM1

Figure 2.1 shows an image of Iranian missile test. The original photo is shown on the right side whereas the altered photo is shown on the left side.

This tampered image (on the left) was originated from Iran’s state media publication in year 2008 and was widely used on the front pages of many major newspapers as well as many other major news websites [1]. The tampered image shows that four missiles appear to launch from a desert launch pad. On the next day after the image was published, the original image (on the right) without the fourth missile firing was distributed by the Associated Press news agency [2], thereby disclosing the fact that the four-missile version of the image has been tampered as analysts claimed that the second missile from the right appeared to be edited with the smoke trails and dust clouds of the projectile being cloned from other missiles which had successfully took off. In other word, the image has undergone copy-move forgery. This image is discovered in the photo gallery available on the official website of Fourandsix Technologies, Inc. (<http://www.fourandsix.com/photo-tampering-history/>) [3]. The gallery has gathered some of the most controversial or notorious tampered images throughout the history of image forensics. This image is selected as one of the sample tampered images that would be experimented later using few approaches of image tampering detection as it has aroused intense controversy on the Internet particularly on several major news websites including BBC news and New York Times news during the time it was disclosed with the caption stated that Iran has been denounced of tampering a faked missile test image possibly for the purpose to exaggerate its military capabilities.

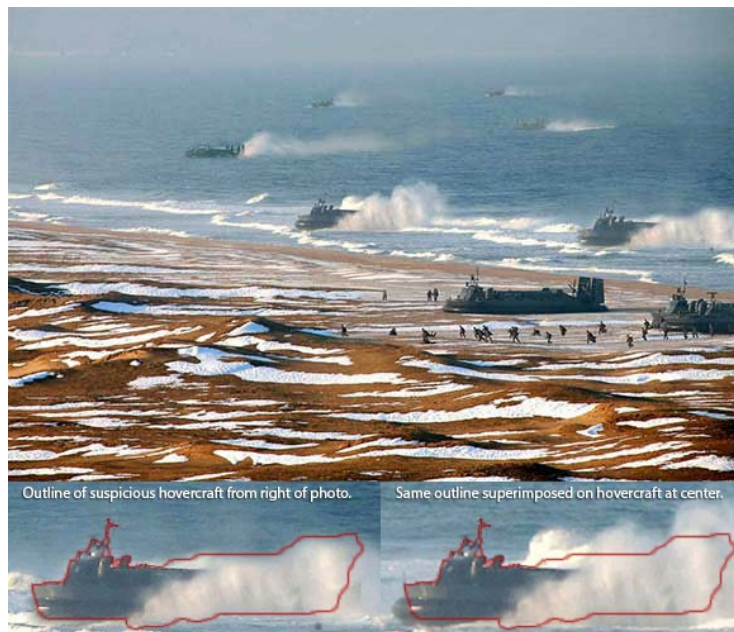


Figure 2: Faked North Korean Hovercraft-Landing Photo – Image Forgery IM2

Figure 2.2 shows an image of North Korean hovercraft-landing during a military exercise conducted along North Korea’s east coast. This tampered image was released by the North Korea’s official Korean Central News Agency (KCNA) on 26 March 2013 and reported to show evidences of image tampering by photo editor Alan Taylor at The Atlantic magazine [4]. The image was found

to have been tampered with at least two or three hovercrafts appear to be the digital clones of each other. Referring to the image, two hovercrafts nearest to the shore are apparently just a single hovercraft whereby one of them has been copied and pasted to give the other hovercraft. Another two hovercrafts which located furthest away from the shore are also suspected to have undergo the similar tampering process of copy-move forgery. In addition, although the leftmost hovercraft does not appear to be cloned, but the slight halo and soft edges of its surrounding also make it suspicious of being tampered. Similarly, this image is also discovered in the photo gallery available on the official website of Fourand-six Technologies, Inc. (<http://www.fourandsix.com/photo-tampering-history/>) [3] which has gathered some of the most controversial or notorious tampered images throughout the history of image forensics. This image is selected as another sample tampered image that would be experimented later using few approaches of image tampering detection as it has demonstrated the squander of image tampering by the North Korean government to make the scene of the military exercise look more threatening.



Figure 3: Forged Helicopter Shark Photo – Image Forgery IM3

Figure 2.3 shows an image of a breaching shark attacking military personnel climbing a suspended ladder of a Special Forces helicopter. The tampered image (on the left side) has gone viral via an e-mail in the year 2001 written with the caption: “AND YOU THINK YOUR HAVING A BAD DAY AT WORK !!”, along with a claim that it was chosen to be National Geographic’s “Photo of the Year” [5]. This forged helicopter shark photo has been debunked the moment National Geographic officially repudiated the genuineness of photo as well as the award and declared that the photo was a hoax [5]. In fact, this tampered image has undergone image splicing forgery whereby the base image (top right image of Figure 2.3) showing a US Air Force helicopter taken in San Francisco has been laterally inverted and spliced with the photo of a breaching shark taken in South Africa (bottom right image of Figure 2.3) [6]. This image is obtained from a page entitle ”10 Most Famous Doctored Photos” available on a blog named Oddee that gathered oddities, weird stuffs and strange things of the world [6]. This image is selected as a sample tampered image to be experimented with few approaches of image tampering detection as it has provided a typical example on how image tampering can be misused in creating a hoax which gives a false impression to the public and delivered untruth statement that would

bring obsession to other party on top of the tampered image.



Figure 4: The Fairy Pools, Isle of Skye, Scotland – Image Forgery IM4

Figure 4 shows an image of purple trees grows wild surrounds the banks of a river in Scotland. The original photo is shown on the right side whereas the altered photo is shown on the left side. This tampered image started circulating online and it was often labeled as "Isle of Sype, Scotland" or "Fairy Pools, Scotland" in social media like Pinterest, Twitter and etc since October 2013. In fact, this photo was taken at the "Shotover River" in New Zealand and all the trees are actually normal shade of green. It was being manipulated with photo editor by using color filter. This image has been proved as a fake photo as analysts claimed that they able to spot some original green color of the trees from the purple vegetation that being edited. [7]



This image is discovered from a page entitled "10 Viral Photos That Probably Fooled You" (<http://indie88.com/10-viral-photos-that-probably-fooled-you/>). [8] This image is selected as a sample tampered image to be experimented with few approaches of image tampering detection as it also demonstrated on how image tampering can tweak an image and gives fake information to the public.

Figure 5 shows an image of a kind of Japanese fruit that looks similarly with

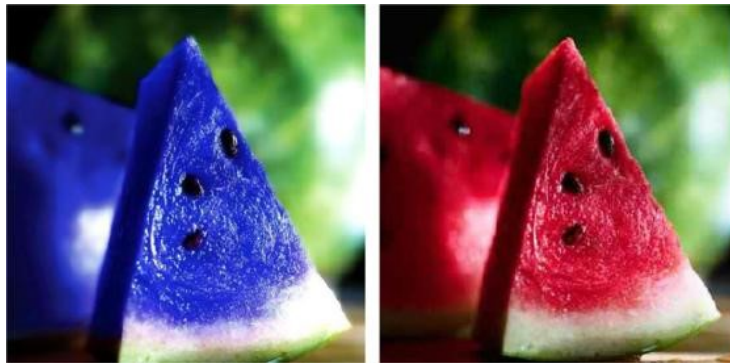


Figure 5: Blue Melon Fruit – Image Forgery IM5

watermelon which is blue in color. The original photo is shown on the right side whereas the modified photo is shown on the left side. This tampered image has gone viral on Web since May 2011 written with the caption: "Moonmelon (scientifically knows as asidus). This fruit grows in some parts of Japan and is known for its vibrant blue colour. This fruit's party trick is that it can switch flavours after you eat it. Everything sour will taste sweet, everything salty will taste bitter, and it gives water a strong orange-like taste! Bucket list fruit!" However, there is no such kind of melon exist in the world and the "asidus" is not a scientific term for any fruits. [9] In fact, it is a normal image of a watermelon slice that has been edited with digital color effects. This image is discovered from a page entitled "Is the "Moonmelon" Real?" (<http://urbanlegends.about.com/od/foodandbeverages/ss/Moonmelon.htm>). [10] This image is selected as a sample tampered image to be experimented with few approaches of image tampering detection as it also demonstrated on how image tampering can be misused in creating a hoax to misleading the public to believe something that is not real.

### 3 Description of Approaches for Digital Image Tamper Detection

#### 3.1 Forensically, Photo Forensics for the Web

Forensically is a browser-based digital image forensics tool (DIFT) developed by Jonas Wagner, a software engineer from Zurich, Switzerland with the main purpose of the tool being the detection of copy-move forgery in digital images [11]. Figure 6 shows the user interface of Forensically. By just clicking on the Open File button in the main menu on top, users can open an image file that they wish to analyze using Forensically. The selected image is processed in the browser without being uploaded to any Cloud or server in order to offer privacy of images. In addition, there is also a Help button which directs the users to a page containing video tutorial as well as summaries of the tools available in Forensically so that people would know how to make use of those several tools for digital image tamper detection. Suiting the main purpose of Forensically by the time it was developed, Clone Detection would be the main feature of

this DIFT. Other than that, there are also some other useful features being available such as Noise Analysis, Level Sweep as well as Meta Data Extraction. The Clone Detection feature, as indicated by its name, is used to detect digital

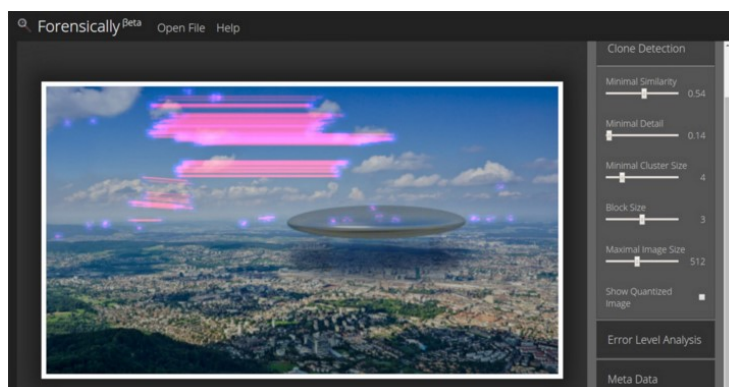


Figure 6: User interface of Forensically

image tampering in term of Copy-Move forgery whereby a region of an image is copied and then pasted in the same image. This clone detector shows the result of clone detection by highlighting possible copied region within an image. The algorithm that is used for the creation of this feature is by traversing a window across the entire image and using all pixels in each window as a key that is stored inside a table. Whenever a similar key is found to match a particular key inside the table, then a clone is detected. A sequence of optimization procedures including compression, filtering and clustering are being done following the basic algorithm mentioned earlier on in order to make the matching of key to be more fuzzy thereby increasing the accuracy of clone detection [11]. There are several parameters available under this feature that can be manipulated by users in order to obtain the optimum and precise detection results. The functionality of some crucial parameters are summarized as follows:

- **Minimal Similarity:** determines the degree of similarity of a cloned pixels with respect to the original one. Decreasing this parameter will give more results on clone detection [12].
- **Minimal Detail:** determines the minimum amount of details in a block for it to be considered when detecting clones. More results would be obtained for small value of this parameter [12].
- **Minimal Cluster Size:** determines the minimum number of clones within a similar region for them to be included in the result of clone detection. The detection results shown under a high value of this parameter would indicate high possibility of copy-move forgery for the particular region of image [12].

Other parameters such as Blocksize, Maximal Image Size define the size of block and the maximum resolution of the image used for clone detection respectively [12]. Another feature under Forensically that is used in this assignment for detecting image tampering is the Level Sweep. The main purpose of this tool

is to make the edges of regions in an image that suffer from particularly copy-move forgery to become more evident by strengthening the contrast of certain brightness levels [12]. The Sweep parameter available for manipulation in this tool allows the users to sweep across the histogram of an image being inspected, thereby identifying discontinuities within the image which may indicate possible tampering have been done to the image. The other two parameters namely Width (indicating slice width of the histogram) and Opacity (determines opacity of sweep layer) are usually kept constant.

## **3.2 Image Forgery Detection using MATLAB Source Code**

There are four MATLAB source codes for image forgery detection being used in this image forensics war assignment as one of the several approaches to determine that the images described in Section 2 have been tampered. These MATLAB source codes correspond to the four image tampering detection techniques which are based on Luminance and HSV (Hue-Saturation-Value) levels of digital image as well as Custom High-Pass Filtering and JPEG Block method.

### **3.2.1 Image Tampering Detection using Luminance levels Technique**

Luminance is an indication about the perceived brightness levels of an image. For two images that are captured using different cameras or under different environments with different lighting, the luminance levels of the images would generally be different from each other. The Luminance levels technique makes use of the different in image luminance to carry out image tampering detection [13] by inspecting any discrepancies in luminance levels of a tampered image resulted from image splicing forgery whereby new image is created by copying a region of one or more image and pasting it to another image. The algorithm used in the MATLAB code for this technique is such that a color or grayscale image is converted to a binary image based on an appropriate threshold value carefully selected with respect to the image being analyzed [13]. Any pixel whose luminance level less than or equal to the chosen threshold would be assigned as black, otherwise it is assigned as white. By examining the resultant binary image for regions with odd and unnatural luminance levels, the tampering of image can thus be detected. The source code of Luminance levels technique is included in Appendix A under Section 8.

### **3.2.2 Image Tampering Detection using HSV levels Technique**

HSV stands for Hue-Saturation-Value, is a color scheme that expresses color in a similar way as how humans perceive color. Hue is the dominant color of a color sample, saturation is the purity of the color whereas value is the brightness or intensity of the color. Similar to the case of luminance levels, the color and brightness of a tampered image would be slightly different for the regions of the image which are copied and pasted from other image sources. The HSV levels technique makes use of the discrepancies and anomalies of HSV color attributes in order to achieve image tampering detection [13]. The algorithm used in the MATLAB code for this technique is such that a color image is converted to HSV color space and the resultant image is observed carefully to identify any abnormal pattern of color distribution that would imply the occurrence of image

tampering. The source code of HSV levels technique is included in Appendix B under Section 8.

### 3.2.3 Image Tampering Detection using Custom High-Pass Filtering Technique

Several image filtering technique based on Sobel, Marr, Roberts' and Prewitt masks have been discussed and analyzed by Lukas as an approach for image tampering detection. Indeed, these filtering techniques do help in uncovering some anomalies present on a tampered image which are inconspicuous to human inspection under normal condition by providing an alternative way to examine the image. However, these techniques have their limitation in detection of image tampering. As such, the High-Pass Filtering technique presents an alternative that utilizes a custom mask for detecting image tampering as well as providing further validation of the occurrence of image tampering [13].

$$\begin{bmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{bmatrix}$$

## Custom Convolution Mask

Figure 7: Custom Convolution Mask used in High-Pass Filtering Technique

Figure 7 shows the convolution mask used in the High-Pass Filtering technique. By performing filtering operation on an image using this masks, all regions in the image which are relatively similar to one another are effectively removed, leaving only those regions which are evidently different. These distinct regions are normally identified as prominent edges and may possibly include the irregular edges introduced as a result of image tampering. The High-Pass Filtering technique makes use of the existence of anomalies such as double edges or abnormal edges pattern to attain image tampering detection [13]. The algorithm used in the MATLAB code for this technique is such that a color image is first converted into grayscale image and then followed by the convolution filtering process. The resultant output image which is generally too dark for inspection is inverted to ease the analysis works. By paying close attention to regions of image with double edges or irregular edges pattern, the presence of image tampering can then be detected. The source code of Custom High-Pass Filtering technique is included in Appendix C under section 8.



### 3.2.4 Image Tampering Detection using JPEG Block Technique

An image that is compressed using JPEG compression standard would be broken up into disjoint  $8 \times 8$  blocks which provide a “fingerprint” to the image [13]. For an image that has been JPEG compressed, a higher Quality Factor would implied that the image is less compressed and therefore the image quality is higher than those compressed using low Quality Factor. Image forgery created through image splicing of two or more JPEG images would generally introduce discrepancies in the statistical information of the resultant image since the JPEG images that are used in creating the forged image are having high chances to have undergone JPEG compression with varying values of Quality Factor. The JPEG Block technique makes use of such anomalies in the tampered image to achieve image tampering detection through analyzing the  $8 \times 8$  blocks “fingerprint” of the image [13]. Any distortion in the  $8 \times 8$  blocks pattern of an image would explicitly indicate that it has been tampered previously. The comprehensive algorithm used in the MATLAB code for this technique is shown in Figure 8.

```

Block_Analysis (image, t)
  divide image into disjoint  $8 \times 8$  compression blocks (i, j)
  for each  $8 \times 8$  JPEG compression block (i, j) within bounds
     $R(i, j) = |A - B - C + D|$  where A = pixel value ( $8*i, 8*j$ ), B = pixel
      value ( $8*i, [8*j] + 1$ ), C = pixel value ( $[8*i] + 1, 8*j$ ), D =
      pixel value ( $[8*i] + 1, [8*j] + 1$ )
    for each  $8 \times 8$  JPEG compression block (i, j) within bounds
       $D_{right}(i, j) = |R(i, j) - R(i, j + 1)|$ 
       $D_{bottom}(i, j) = |R(i, j) - R(i + 1, j)|$ 
    for each  $8 \times 8$  JPEG compression block (i, j) within bounds
      if ( $D_{right}(i, j) \geq t$ ) OR ( $D_{bottom}(i, j) \geq t$ )
        set all pixel values in (i, j) to white
      else
        set all pixel values in (i, j) to black
  end Block_Analysis

```

Figure 8: Algorithm of JPEG Block Technique

Before the image is processed using the algorithm, it will be first converted into a grayscale image if it is originally a color image. Basically, the algorithm divides the image into disjoint  $8 \times 8$  blocks using JPEG compression standard. A variable named  $R(i, j)$  would then be calculated to act as a representation of the level of pixel variability for each of the  $8 \times 8$  block with its 3 neighboring blocks. Two parameters named  $D_{right}(i, j)$  and  $D_{bottom}(i, j)$  are subsequently computed based on the  $R(i, j)$  value of current block with those of the nearest right neighboring and bottom neighboring blocks respectively. These two parameters are used to form two conditional expressions with a user-defined threshold value and the entire 64 pixels in each block are set to either white or black with respect to the outcome of the conditional expressions in order to output a resultant binary image. By observing the binary image output for the

regions marked with white pixels, suspected tampered area could therefore be identified using this technique. In general, the white pixels pattern would emphasis more precisely on tampered area within the image if a higher threshold value is used. The source code of JPEG Block technique is included in Appendix D under Section 8.

### 3.3 JPEGsnoop

JPEGsnoop is a very powerful open source program which allows user to do JPEG image decoding and analyze the details of an image. [14] Figure 9 shows the user interface of JPEGsnoop. User can open a JPEG image with “File -> Open...” or simply drag-and-drop the image file onto the JPEGsnoop window. Then, it will start analysis and decode the image automatically.

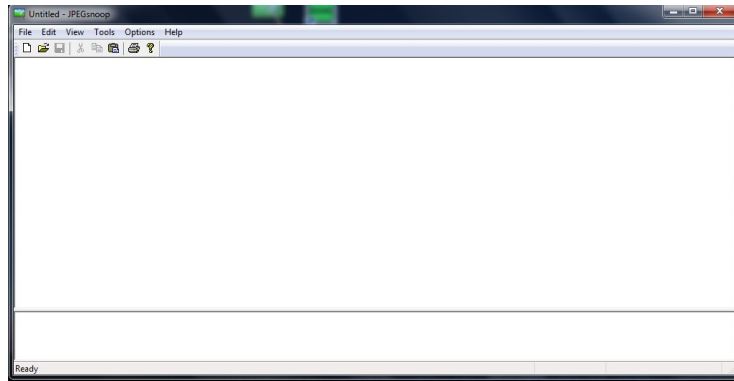


Figure 9: User interface of JPEGsnoop

JPEGsnoop helps the user to identify the image quality, detect error in the corrupted image and determine the various settings such as EXIF metadata that used in the digital camera to take the image. [14] Most importantly, this program able to examine if an image has been edited by using photo editor. JPEGsnoop contains a huge internal database of compression signatures; it will compare the image with the signatures in the database and show the camera and software used to generate the image when a match is found. If the signature of photo editor is recognized, there is high possibility that the image is not an original image. Figure 10 shows the example of signatures in the “Searching Compression Signatures” section after the user open an image in JPEGsnoop.

There are 4 possible classes to evaluate the image:

Class	Assessment
Class 1	Image is processed/edited
Class 2	Image has high probability of being processed/edited
Class 3	Image has high probability of being original
Class 4	Uncertain if processed or original

[15] However, this program is not able to detect a JPEG image that extracted from video files and it will determine these images as “edited” in most cases. Furthermore, in-camera editing and some other factors also demonstrate

```

*** Searching Compression Signatures ***
Signature:          01180AF3DE63318228A86409EF4013DD
Signature (Rotated): 01180AF3DE63318228A86409EF4013DD
File Offset:       0 bytes
Chroma subsampling: 1x1
EXIF Make/Model:   NONE
EXIF Makernotes:   NONE
EXIF Software:     OK   [Adobe Photoshop Elements 2.0]

Searching Compression Signatures: (3347 built-in, 0 user(*) )

-----
EXIF.Make / Software      EXIF.Model      Quality      Subsamp Match?
-----
SW :[Adobe Photoshop    ]                        [Save As 08  ]

NOTE: Photoshop IRB detected
NOTE: EXIF Software field recognized as from editor
Based on the analysis of compression characteristics and EXIF metadata:

ASSESSMENT: Class 1 - Image is processed/edited

```

Figure 10: Example signatures of camera and software in “Searching Compression Signatures” section

that this program cannot provide sufficient evidence to prove the image is an original image. Thus, the result generated by this program cannot absolutely use to determine the authenticity and take it as a proof for legal studies. We have to experiment the tampered images with few more approaches to prove its authenticity.

### 3.4 FotoForensics

FotoForensics is a website sponsored by Hacker Factor that allows user to photo forensics. Hacker Factor has recreated this service after Pete Ringwood decided retire the “errorlevelanalysis.com” website in 2012. [16] User able to examine an image has been tampered or not by using FotoForensics. It supports Error Level Analysis (ELA) which helps to identify which area of the image has been undergoes different compression. Figure 11 shows the user interface of FotoForensics. By just passing the image URL or clicking on the Choose File button, user can upload and open an image file that they wish to analyze using FotoForensics. Furthermore, there is a Tutorials button on the top right side which will teach the user how to use this website.

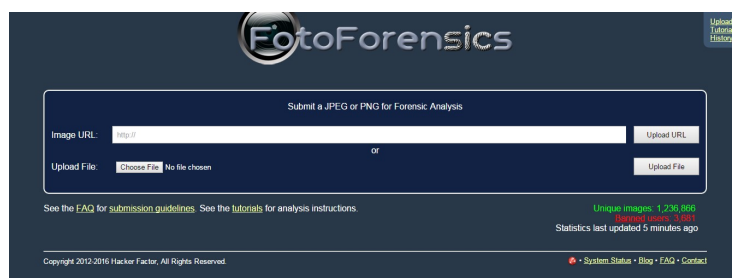



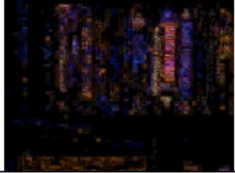

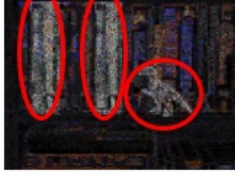


Figure 11: User interface of FotoForensics

An original image is expected to have high ELA values and the entire image supposed to be in same compression level. The ELA values will become lower with each resave. This indicates that the high frequencies and fine details of the image are lost during every resave. If some parts of the image show high different quality level compare with the rest, it likely indicates as modified image.

In other words, an original image will have a lot white color in the ELA over the entire image. With every resave, the ELA will show darker color. [17] If there is any modification was added, the modified area will have higher ELA values and its color is brighter than rest. Example 1 demonstrate how to evaluate the ELA. [17]

Example 1 of Evaluation of ELA

	Image	ELA	Description
Original image			Noise like, white color covered entire image in ELA.
Resaved one time			The color became darker in ELA after resaved.
Modified image			The red circles indicate the regions that have been modified. These regions have higher ELA values from the rest.

## 4 Experimental Results

### 4.1 Screen Capture of the results of Clone Detection and Level Sweep Tool in Forensically



Figure 12: Image Forgery IM1: Output of Clone Detection Tool



Figure 13: Image Forgery IM1: Output of Level Sweep Tool



Figure 14: Image Forgery IM2: Output of Clone Detection Tool

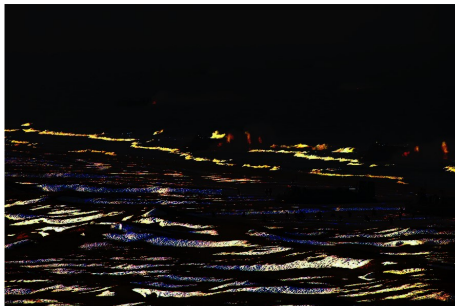


Figure 15: Image Forgery IM2: Output of Level Sweep Tool



Figure 16: Image Forgery IM3: Output of Clone Detection Tool

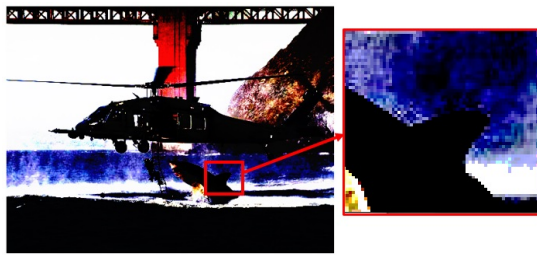


Figure 17: Image Forgery IM3: Output of Level Sweep Tool

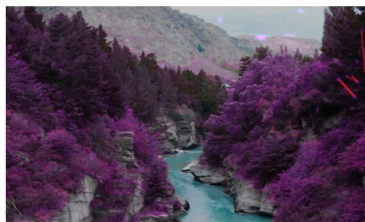


Figure 18: Image Forgery IM4: Output of Clone Detection Tool



Figure 19: Image Forgery IM4: Output of Level Sweep Tool

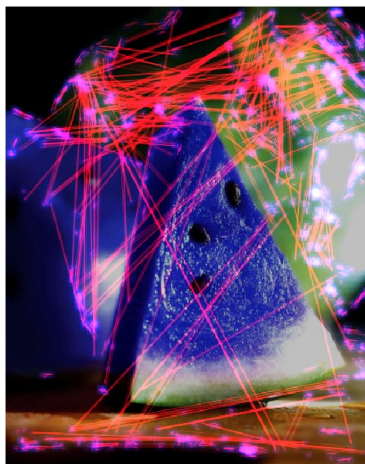


Figure 20: Image Forgery IM5: Output of Clone Detection Tool

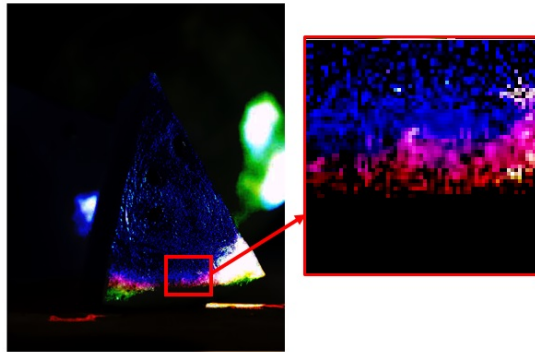


Figure 21: Image Forgery IM5: Output of Level Sweep Tool

#### 4.2 Screen Capture of the results of Image Forgery Detection using MATLAB source code



Figure 22: Image Forgery IM1: Luminance Level Technique with threshold 0.5

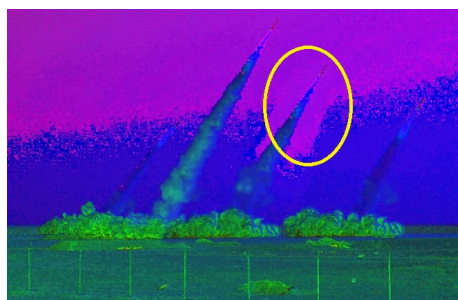


Figure 23: Image Forgery IM1: HSV Level Technique

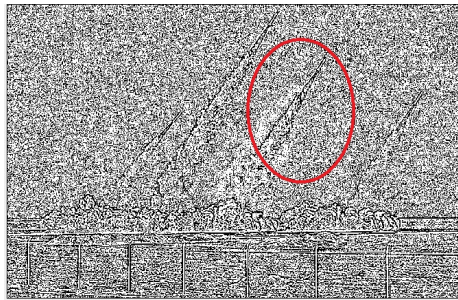


Figure 24: Image Forgery IM1: Custom Filtering Technique

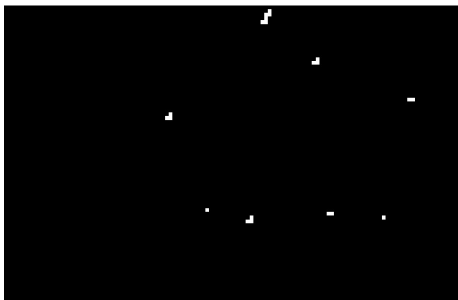


Figure 25: Image Forgery IM1: JPEG Block Technique with threshold 25

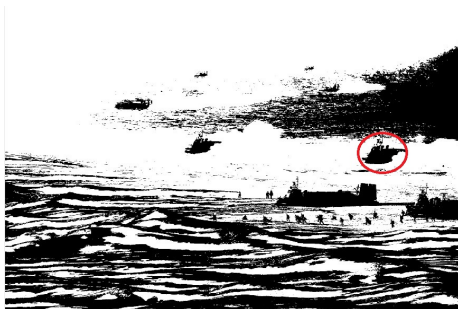


Figure 26: Image Forgery IM2: Luminance Level Technique with threshold 0.5

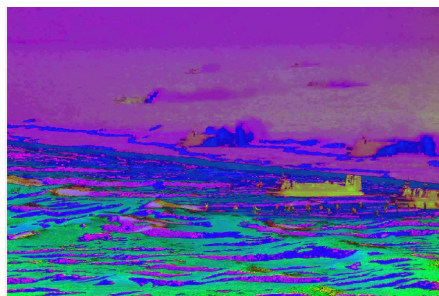


Figure 27: Image Forgery IM2: HSV Level Technique



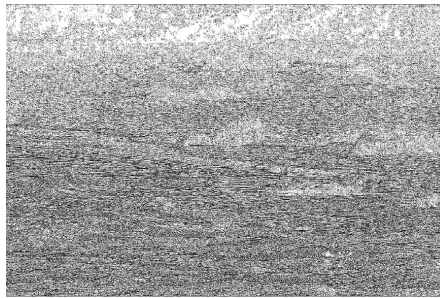


Figure 28: Image Forgery IM2: Custom Filtering Technique

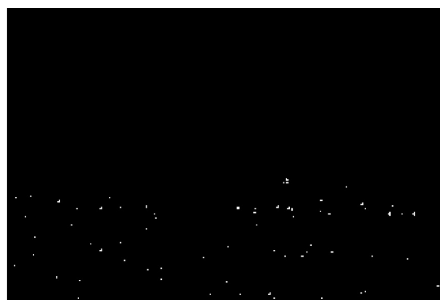


Figure 29: Image Forgery IM2: JPEG Block Technique with threshold 50

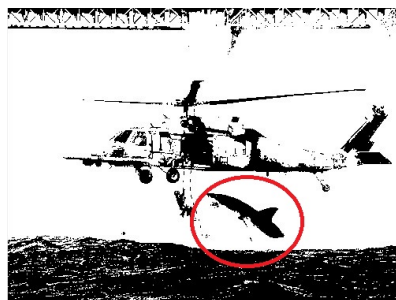


Figure 30: Image Forgery IM3: Luminance Level Technique with threshold 0.2

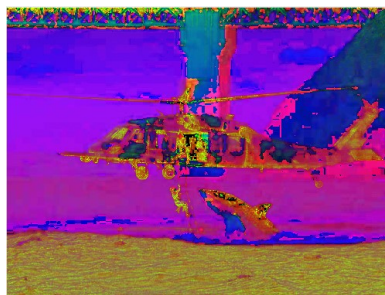


Figure 31: Image Forgery IM3: HSV Level Technique

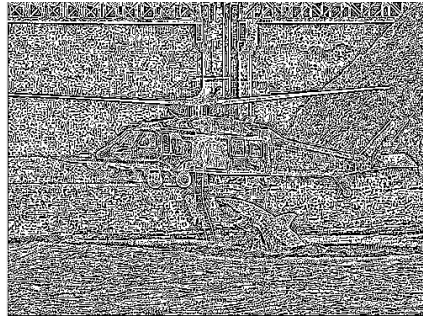


Figure 32: Image Forgery IM3: Custom Filtering Technique



Figure 33: Image Forgery IM3: JPEG Block Technique with threshold 60

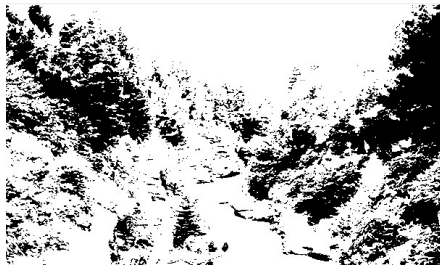


Figure 34: Image Forgery IM4: Luminance Level Technique with threshold 0.2

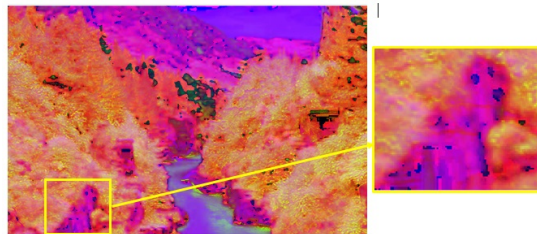


Figure 35: Image Forgery IM4: HSV Level Technique



Figure 36: Image Forgery IM4: Custom Filtering Technique



Figure 37: Image Forgery IM4: JPEG Block Technique with threshold 50



Figure 38: Image Forgery IM5: Luminance Level Technique with threshold 0.15

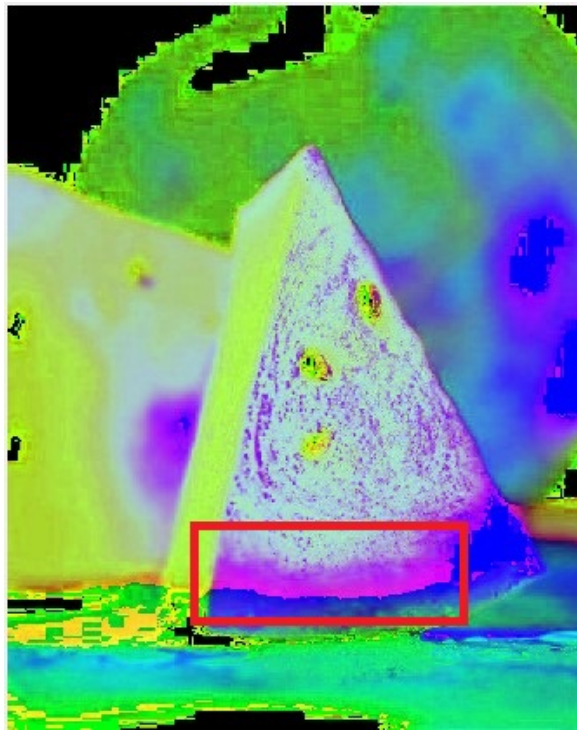


Figure 39: Image Forgery IM5:HSV Level Technique

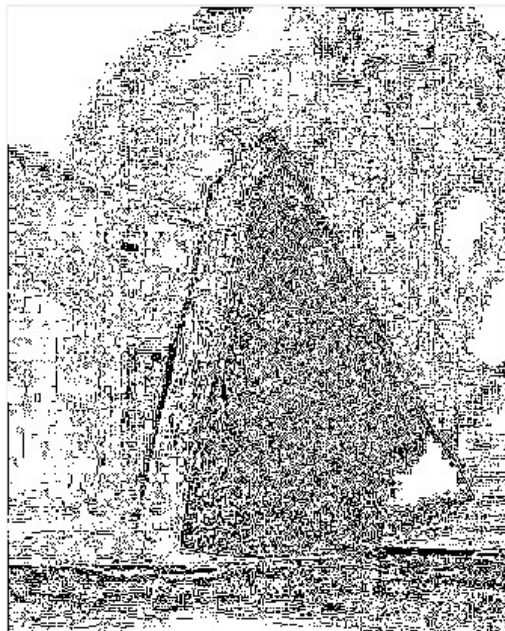


Figure 40: Image Forgery IM5:Custom Filtering Technique

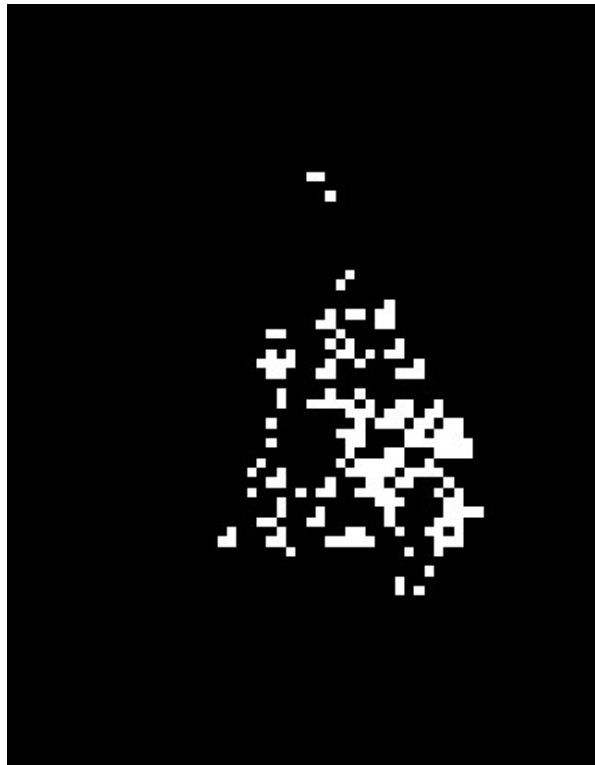


Figure 41: Image Forgery IM5:JPEG Block Technique with threshold 70

### 4.3 Screen Capture of the results of Image Forgery Detection using JPEGsnoop

```

*** Searching Compression Signatures ***

Signature:          0141C51BBA9713ADB2DCCCFBA5E8F51C
Signature (Rotated): 0141C51BBA9713ADB2DCCCFBA5E8F51C
File Offset:       0 bytes
Chroma subsampling: 1x1
EXIF Make/Model:   NONE
EXIF Makernotes:   NONE
EXIF Software:     OK   [Adobe Photoshop CS6 (Windows)]

Searching Compression Signatures: (3347 built-in, 0 user(*) )

      EXIF.Make / Software      EXIF.Model      Quality      Subsamp Match?
      -----
NOTE: Photoshop IRB detected
NOTE: EXIF Software field recognized as from editor
Based on the analysis of compression characteristics and EXIF metadata:

ASSESSMENT: Class 1 - Image is processed/edited

```

Figure 42: Image Forgery IM1

\*\*\* Searching Compression Signatures \*\*\*

Signature: 013E5A347BEB5C2FD641B1432B342192  
 Signature (Rotated): 013E5A347BEB5C2FD641B1432B342192  
 File Offset: 0 bytes  
 Chroma subsampling: 1x1  
 EXIF Make/Model: NONE  
 EXIF Makernotes: NONE  
 EXIF Software: NONE

Searching Compression Signatures: (3347 built-in, 0 user(\*) )

EXIF.Make / Software	EXIF.Model	Quality	Subsamp Match?
SW :[Adobe Photoshop	]	[Save For Web 075]	

Based on the analysis of compression characteristics and EXIF metadata:

ASSESSMENT: Class 1 - Image is processed/edited

Figure 43: Image Forgery IM2

\*\*\* Searching Compression Signatures \*\*\*

Signature: 013BA18D5561625796E986FDBC09F846  
 Signature (Rotated): 01AC57E12793DFA7C46C704625C5AF0F  
 File Offset: 0 bytes  
 Chroma subsampling: 1x1  
 EXIF Make/Model: NONE  
 EXIF Makernotes: NONE  
 EXIF Software: NONE

Searching Compression Signatures: (3347 built-in, 0 user(\*) )

EXIF.Make / Software	EXIF.Model	Quality	Subsamp Match?
CAM:[???	]	[Treo 680	] No
CAM:[Canon	]	[Canon PowerShot Pro1	] No
CAM:[NIKON	]	[E2500	] No
CAM:[NIKON	]	[E3100	] No
CAM:[NIKON	]	[E4500	] No
CAM:[NIKON	]	[E5000	] No
CAM:[NIKON	]	[E5700	] No
CAM:[NIKON	]	[E775	] No
CAM:[NIKON	]	[E885	] No
CAM:[OLYMPUS OPTICAL CO.,LTD	]	[C3040Z	] No
CAM:[PENTAX	]	[PENTAX Optio 550	] No
CAM:[Research In Motion	]	[BlackBerry 9530	] No
CAM:[SEIKO EPSON CORP.	]	[PhotoPC 3000Z	] No
CAM:[SONY	]	[DSC-H7	] No
CAM:[SONY	]	[DSC-H9	] No
CAM:[SONY	]	[DSC-S90	] No
CAM:[SONY	]	[DSC-W1	] No
CAM:[SONY	]	[SONY	] No
SW :[ACDSee	]	[	] No
SW :[FixFoto	]	[fine	] No
SW :[IJG Library	]	[090	] No

The following IJG-based editors also match this signature:

SW :[GIMP	]	[090	] No
SW :[IrfanView	]	[090	] No
SW :[idImager	]	[090	] No
SW :[FastStone Image Viewer	]	[090	] No
SW :[NeatImage	]	[090	] No
SW :[Paint.NET	]	[090	] No
SW :[Photomatix	]	[090	] No
SW :[XnView	]	[090	] No

Based on the analysis of compression characteristics and EXIF metadata:

ASSESSMENT: Class 1 - Image is processed/edited

Figure 44: Image Forgery IM3

\*\*\* Searching Compression Signatures \*\*\*

Signature: 013BA18D5561625796E986FDBC09F846  
 Signature (Rotated): 01ACS7E12793DFA7C46C704625C5AF0F  
 File Offset: 0 bytes  
 Chroma subsampling: 2x2  
 EXIF Make/Model: NONE  
 EXIF Makernotes: NONE  
 EXIF Software: NONE

Searching Compression Signatures: (3347 built-in, 0 user(\*) )

EXIF.Make / Software	EXIF.Model	Quality	Subsamp Match?
CAM:[???	] [Treo 680	] [	] Yes
CAM:[Canon	] [Canon PowerShot Pro1	] [fine	] No
CAM:[NIKON	] [E2500	] [FINE	] No
CAM:[NIKON	] [E3100	] [FINE	] No
CAM:[NIKON	] [E4500	] [FINE	] No
CAM:[NIKON	] [E5000	] [FINE	] No
CAM:[NIKON	] [E5700	] [FINE	] No
CAM:[NIKON	] [E775	] [FINE	] No
CAM:[NIKON	] [E885	] [FINE	] No
CAM:[OLYMPUS OPTICAL CO.,LTD	] [C3040Z	] [	] No
CAM:[PENTAX	] [PENTAX Optio 550	] [	] No
CAM:[Research In Motion	] [BlackBerry 9530	] [Superfine	] Yes
CAM:[SEIKO EPSON CORP.	] [PhotoPC 3000Z	] [	] No
CAM:[SONY	] [DSC-H7	] [	] No
CAM:[SONY	] [DSC-H9	] [	] No
SW : [FixFoto	] [	] [fine	] ]
SW : [IJG Library	] [	] [090	] ]
SW : [ZoomBrowser EX	] [	] [high	] ]

The following IJG-based editors also match this signature:

SW : [GIMP	] [	] [090	] ]
SW : [IrfanView	] [	] [090	] ]
SW : [IdImager	] [	] [090	] ]
SW : [FastStone Image Viewer	] [	] [090	] ]
SW : [NeatImage	] [	] [090	] ]
SW : [Paint.NET	] [	] [090	] ]
SW : [Photomatix	] [	] [090	] ]
SW : [XnView	] [	] [090	] ]

NOTE: JFIF COMMENT field is known software

Based on the analysis of compression characteristics and EXIF metadata:

ASSESSMENT: Class 1 - Image is processed/edited

Figure 45: Image Forgery IM4

\*\*\* Searching Compression Signatures \*\*\*

Signature: 0182230692721ADF5DCBFB56F747490C  
 Signature (Rotated): 01DD92C0CD7077A88C49139F2F15908D  
 File Offset: 0 bytes  
 Chroma subsampling: 1x2  
 EXIF Make/Model: NONE  
 EXIF Makernotes: NONE  
 EXIF Software: NONE

Searching Compression Signatures: (3347 built-in, 0 user(\*) )

EXIF.Make / Software	EXIF.Model	Quality	Subsamp Match?
CAM:[PENTAX	] [PENTAX Optio 550	] [	] No
SW : [Apple ImageIO.framework	] [	] [049	] ]
SW : [IJG Library	] [	] [079	] ]

The following IJG-based editors also match this signature:

SW : [GIMP	] [	] [079	] ]
SW : [IrfanView	] [	] [079	] ]
SW : [IdImager	] [	] [079	] ]
SW : [FastStone Image Viewer	] [	] [079	] ]
SW : [NeatImage	] [	] [079	] ]
SW : [Paint.NET	] [	] [079	] ]
SW : [Photomatix	] [	] [079	] ]
SW : [XnView	] [	] [079	] ]

Based on the analysis of compression characteristics and EXIF metadata:

ASSESSMENT: Class 1 - Image is processed/edited

Figure 46: Image Forgery IM5

#### 4.4 Screen Capture of the results of Image Forgery Detection using FotoForensics



Figure 47: Image Forgery IM1



Figure 48: Image Forgery IM2



Figure 49: Image Forgery IM3





Figure 50: Image Forgery IM4

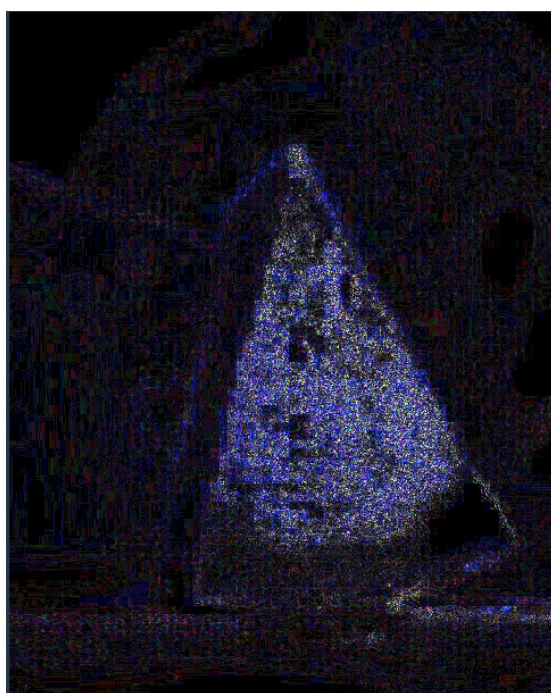


Figure 51: Image Forgery IM5

Image Forgery	Inconclusive	Possible	Definitive
IM1			X
IM2			X
IM3	X		
IM4	X		
IM5	X		

Table 1: Summary of Results of Signs of Tampering obtained using Clone Detection Tool.

## 5 Analysis and Discussion of Results

### 5.0.1 Analysis and Discussion of Results obtained using Forensically online sites

As tabulated in Table 1, the Clone Detection tool in Forensically confirms that image forgery IM1 and IM2 has been tampered with by highlighting the regions

that has been cloned. In IM1, the dust clouds of the third missile is obviously a clone from that of the fourth missile whereas the two hovercrafts nearest to the shore in IM2 are also shown as clones of each other. Nonetheless, this Clone Detection tool fails to give any conclusive signs of image tampering for the remaining 3 images under test. These inconclusive results are due to the fact that Clone Detection tool is primarily meant for detecting Copy-Move image forgery as demonstrated in IM1 and IM2. Since IM3 is an example of image splicing forgery and IM4 and IM5 are examples of image retouching forgery, the tool therefore suffered from difficulties to validate the tampering that have been done to these images.

Image Forgery	Inconclusive	Possible	Definitive
IM1			X
IM2	X		
IM3		X	
IM4	X		
IM5		X	

Table 2: Summary of Results of Signs of Tampering obtained using Level Sweep Tool.

From Table 2, it can be seen that the Level Sweep tool in Forensically can only provide definitive signs of tampering for image forgery IM1 where there is an obvious halo effect being observed around the projectile of third missile from the left. Image forgery IM3 and IM5 show possible signs of tampering when tested using the Level Sweep tool. In IM3, a blurring effect around the shark fin gives suspicion that the breaching shark could be copied from somewhere else and blended into the image whereas the blurring effect between the pulp and the peel of the watermelon in IM5 also gives trivial sign that the image may have been manipulated. There is no decisive signs of tampering observed in image forgery IM2 and IM4 when tested with the Level Sweep tool. The reason of obtaining such results may be due to the fact the color patterns in IM2 are actually originated from the authentic regions within the same image and the brightness level of the tampered area could have been explicitly adjusted to achieve higher quality of the forged image. As for IM4, since the image consists of many soft edges distributed randomly over the entire image with its brightness contrast being changed progressively rather than abruptly, the Level Sweep may have difficulties to emphasize on the tampered areas.

### 5.0.2 Analysis and Discussion of Results obtained using MATLAB source codes

Referring to table 3, two of the five image forgery under test generate outcomes that result in inconclusive signs of tampering when Luminance Level technique is used with certain threshold value as indicated in section 4. In image forgery IM1 and IM4, the tampered regions exhibit luminance levels which are very similar to that of their surrounding areas, hence the Luminance Level technique being used here might not be applicable for detecting tampering in these kinds of image forgeries. On the other hand, the Luminance Level technique is able to provide signs of possible image tampering in the remaining three image forgeries. Other tamper detection techniques are required to further justify the suspicion of

Image Forgery	Inconclusive	Possible	Definitive
IM1	X		
IM2		X	
IM3		X	
IM4	X		
IM5		X	

Table 3: Summary of Results of Signs of Tampering obtained using Luminance Level Technique.

Image Forgery	Inconclusive	Possible	Definitive
IM1			X
IM2	X		
IM3	X		
IM4		X	
IM5			X

Table 4: Summary of Results of Signs of Tampering obtained using HSV Level Technique.

image tampering. The unnatural luminance levels of the circled hovercraft and the breaching shark in IM2 and IM3 respectively give doubt on the authenticity of the particular area. As for IM5, the uneven luminance patterns present in the image also signify possible tampering have been done to it. As a side note, this technique tends to fall short in detecting image forgery done by a skilled personnel since one can actually adjust the luminance levels of a particular tampered region so that it can blend in well into the forged image.

As shown in table 4, the results of using HSV Level technique to detect image tampering in image forgery IM2 and IM3 show that there is no definitive signs of tampering in the two images. The reason of obtaining such results is because the HSV color patterns of the tampered regions in these images are actually originated from either an authentic area within the image itself or an alternative area with similar color-space as the image. Applying the HSV Level technique to IM4 returns result that raises suspicion of image tampering as the magnified version of the image region highlighted in square box shows abnormal HSV color patterns as compared to its surrounding. Conclusive signs of tampering can be obtained for image forgery IM1 and IM5 using the HSV Level technique. The results of this technique uncovers an obvious discoloration in the circled region of IM1 as well as the abrupt change in HSV color pattern in the highlighted square box region in IM5. Overall, the HSV Level technique is shown to be well suited for detecting image tampering which involves discrepancies in HSV color patterns.

As depicted in table 5, the Custom Filtering technique only provides definitive signs of image tampering for image forgery IM1 with a faded halo being identified in the circled region of IM1. Performing this technique on the remaining four test images do not return any conclusive signs of tampering. No irregular edges can be identified in the results of the four image forgeries due to the uniform edges patterns in all the four images.

As shown in table 6, conclusive signs of tampering can be obtained for image

Image Forgery	Inconclusive	Possible	Definitive
IM1			X
IM2	X		
IM3	X		
IM4	X		
IM5	X		

Table 5: Summary of Results of Signs of Tampering obtained using Custom Filtering Technique.

Image Forgery	Inconclusive	Possible	Definitive
IM1			X
IM2	X		
IM3		X	
IM4	X		
IM5			X

Table 6: Summary of Results of Signs of Tampering obtained using JPEG Block Technique.

forgery IM1 and IM5 using the JPEG Block technique. Using an appropriate threshold value of 25 and 70 respectively for generating the result of IM1 and IM5, the white blocks in the output effectively identified the tampered region in both images. Possible signs of tampering is detected for image forgery IM3 using the JPEG Block technique where the white blocks mostly concentrated around the center of the image with the actual tampered area being included as well. For image forgery IM2 and IM4, the JPEG Block technique failed to detect any decisive signs of image tampering probably because the faked hovercraft in IM2 is simply too small for the technique to produce conclusive results and the tampering in image forgery IM4 is strictly dealt with color pattern discrepancies.

### 5.0.3 Analysis and Discussion of Results obtained using JPEGsnoop

Image Forgery	Inconclusive	Possible	Definitive
IM1	X		
IM2			X
IM3		X	
IM4			X
IM5			X

Table 7: Summary of Results of Signs of Tampering obtained using JPEGsnoop.

As illustrated in table 7, conclusive signs of tampering can be obtained for image forgery IM2, IM3, IM4 and IM5 using the JPEGsnoop application. According to the result we get from the “Searching Compression Signatures” section, JPEGsnoop not able to show any matched signatures of camera that generate IM1 but its metadata elements matched with the Photoshop. For IM2, its signature only matched with Photoshop. In contrast, the images matched with many signatures of cameras and photo editors in IM3 and IM4. The

assessment of JPEGsnoop classified all these images in Class 1 which means that the images have been edited.

#### 5.0.4 Analysis and Discussion of Results obtained using FotoForensics

Image Forgery	Inconclusive	Possible	Definitive
IM1			X
IM2			X
IM3		X	
IM4			X
IM5			X

Table 8: Summary of Results of Signs of Tampering obtained using FotoForensics

As shown in table 5.8, conclusive signs of tampering can be obtained for image forgery IM1, IM2, IM4 and IM5 using the FotoForensics. In ELA, an image that covered by white colors can classify as an original image because it has high ELA values. For IM1, we noticed that the background is almost in dark blue colors. We only able to tell that this image has been resaved but unable to examine which part of the image has been edited. In the case of IM2, since the image is undergoes tampering process of copy-move forgery, it is not able to detect the altered part in ELA. We observed that the image has low ELA values. This means the image has been resaved and not an original photo. We can see that the image is almost covered by white colors at the first sight in IM3. There is a chance this image is original. However, when we look at the details and compare edge to edge and surfaces to surfaces, we noticed that the helicopter was touched up. It has stronger ELA values than the background. Although it is not significant, but the background is in darker color. For IM4, we able to see that the image is covered by a lot noise like white colors. However, we noticed that there is significant different error level on the top right corner. It has lower ELA values and shows more black colors from the rest. Thus, we infer this image has been edited. We can see that there are whiter colors and higher ELA values in the center area of the image in IM5. This means that the area in the center of the image is at a different quality level than the rest.

## 6 Task Distribution

**Group member: NG CHIN KIT - 1122701243**

Tasks assigned:

- Search for 3 famous tampered images and give clear description of the images
- Search and describe 2 approaches for digital image tamper detection – (Forensically and MATLAB source codes)
- Carry out image tampering detection using the 2 approaches found and collect the results of tampering detection

- Analyze and discuss the experimental results obtained from the tamper detection process

Parts written:

- 1.0 Discussion of Side Chosen
- 2.0 Description of Tampered Image
- 3.0 Description of Approaches for Digital Image Tamper Detection (3.1 – 3.2)
- 4.0 Experimental Results (4.1 – 4.2)
- 5.0 Analysis and Discussion of Results (5.1 – 5.2)
- 6.0 Task Distribution

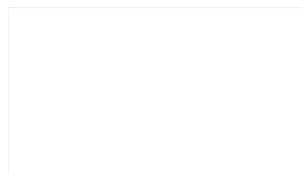
**Group member: LEW ZHI YONG – 1121117284**

Tasks assigned:

- Search for 2 famous tampered images and give clear description of the images
- Search and describe 2 approaches for digital image tamper detection - (FotoForensics and JPEGsnoop)
- Carry out image tampering detection using the 2 approaches found and collect the results of tampering detection
- Analyze and discuss the experimental results obtained from the tamper detection process

Parts written:

- 2.0 Description of Tampered Image
- 3.0 Description of Approaches for Digital Image Tamper Detection (3.3 – 3.4)
- Experimental Results (4.3 – 4.4)
- Analysis and Discussions of Results (5.3 – 5.4)
- References



## 7 References

- [1] Mike Nizza and Patrick J. Lyons. (2008, July 10). In an Iranian Image, a Missile Too Many. Retrieved from: [http://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many/?\\_r=1](http://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many/?_r=1)
- [2] Iran ‘faked missile test image’. (2008, July 10). Retrieved from: [http://news.bbc.co.uk/2/hi/middle\\_east/7500917.stm](http://news.bbc.co.uk/2/hi/middle_east/7500917.stm)
- [3] Fourandsix Technologies, Incorporated. (2014) Photo Tampering throughout History. Retrieved from: <http://www.fourandsix.com/photo-tampering-history/>
- [4] Alan Taylor. (2013, March 26). Is This North Korean Hovercraft-Landing Photo Faked? Retrieved from: <http://www.theatlantic.com/photo/2013/03/is-this-north-korean-hovercraft-landing-photo-faked/100480/>
- [5] Stentor Danielson and David Braun. (2005, March 8). Shark “Photo of the Year” Is E-Mail Hoax. Retrieved from: [http://news.nationalgeographic.com/news/2002/08/0815\\_020815\\_photooftheyear.html](http://news.nationalgeographic.com/news/2002/08/0815_020815_photooftheyear.html)
- [6] Grace Murano. (2009, September 9). 10 Most Famous Doctored Photos. Retrieved from: [http://www.oddee.com/item\\_96803.aspx](http://www.oddee.com/item_96803.aspx)
- [7] Jacqueline Boss. (2012 September 13). Pinterest’s Viral Purple Trees Debunked from: <http://www.escapenormal.com/2012/09/13/pinterests-viral-purple-trees-debunked/>
- [8] Brent Albrecht. (2015 May 1) 10 Viral Photos That Probably Fooled You from: <http://indie88.com/10-viral-photos-that-probably-fooled-you/>
- [9] Maria Vultaggio. (2014 January 14) Is ‘Moon Melon’ Real? Blue WaterMelon ‘Asidus’ Fruit Goes Viral on Pinterest and Twitter from: <http://www.ibtimes.com/moon-melon-real-blue-watermelon-asidus-fruit-goes-viral-pinterest-twitter-1540132>
- [10] David Emery. (2016 January 17) Is the “Moonmelon” Real? from: <http://urbanlegends.about.com/od/foodandbeverages/ss/Moonmelon.htm>
- [11] Jonas Wagner. (2015, August 16). Forensically, Photo Forensics for the Web. Retrieved from: <https://29a.ch/2015/08/16/forensically-photo-forensics-for-the-web>
- [12] Jonas Wagner. (2015, August 16). About Forensically. Retrieved from: <https://29a.ch/photo-forensics/#help>
- [13] Jonathan R. Sturak. (2004, December). Forensic Analysis of Digital Image Tampering.
- [14] Calvin Hass. (2015) JPEGsnoop 1.7.5 – JPEG File Decoding Utility from: <http://www.impulseadventure.com/photo/jpeg-snoop.html>
- [15] Calvin Hass. (2009) JPEGsnoop – Identifying Edited Photos from: <http://www.impulseadventure.com/photo/jpeg-snoop-identify-edited-photos.html>
- [16] Inforesec Institute. (2013 October 25) Photo Forensics: Detect Photoshop Manipulation with Error Level Analysis
- [17] Hacker Factor (2012-2016) Evaluating ELA from: <http://fotoforensics.com/tutorial-ela.php>
- [18] StopFake. (2014 July 29). 13 online tools that help to verify the authenticity of a photo from: <http://www.stopfake.org/en/13-online-tools-that-help-to-verify-the-authenticity-of-a-photo/>