ISLAMIC UNIVERSITY OF TECHNOLOGY

An Efficient Circular Block Approach for Copy-Move Forgery Detection

Authors:

Md. Sirajus Salekin (094405)

Rafsanjany Kushol (094404)

Supervisor:

Md. Hasanul Kabir, PhD

Assistant Professor

Department of Computer Science and Engineering

A thesis submitted to the Department of CSE
in partial fulfilment of the requirements for the degree of
B.Sc. Engineering in CSE

Academic Year: 2012-2013

A Subsidiary Organ of the Organization of Islamic Cooperation
Dhaka, Bangladesh

October 2013

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and investigation carried out by Md. Sirajus Salekin and Rafsanjany Kushol under the supervision of Dr. Md. Hasanul Kabir in the Department of Computer Science and Engineering (CSE), IUT, Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Md. Sirajus Salekin
Student ID - 094405

Rafsanjnay Kushol
Student ID - 094404

Supervisor:

Md. Hasanul Kabir, PhD
Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Abstract

Due to availability of powerful retouching or editing software tools now-a-days it is very easy to tamper any type of digital images. That's why it has been very common to add or remove anything from an original image which causes the lead of digital image forgery. Several types of digital forgery may be happened but copy-move forgery is difficult to detect by our naked-eyes. Copy-move forgery is a special type of digital forgery in which a part of the image is copied and pasted somewhere else in the image with the intent to cover an important image feature. To detect this type of forgery we need a robust detection method which ensures the correct detection even if the image is noised, compressed, scaled, rotated, flipped etc. Several methods has already been proposed but none methods are suitable for all kinds of robustness. Some methods are showing good performance to approximate match or noised or compressed but fails towards scaled or rotated. So still no method can ensure the detection of the forged image in any types of challenges with compatible time performance. We have gone through the existing methods and found out their limitations. We have also analyzed their comparison to find out their comparative performance. In this thesis paper, we propose an efficient method for detecting the copy-move forgery using circular block extraction and calculating the mean, contrast of the image which is robust in rotation, flipping, JPEG compression, blurring, noise etc. The main success of the proposed method is the robustness in JPEG compression, blurred image and rotation in any angels. Basically for using circular block approach we are getting rotation invariant detection method. Another thing is that we are comparing the blocks up to nth consecutive blocks for which our method can show better performance for JPEG compression with low quality factor and also applicable for blurring. The performance of the proposed method has been evaluated with different challenges like Gaussian noise, Gaussian blurring, rotation, flipping, JPEG compression etc. as well as the existing methods. For dataset we consider the benchmark image set so that we can get the real strength of our proposed method.

Contents

List of Figures

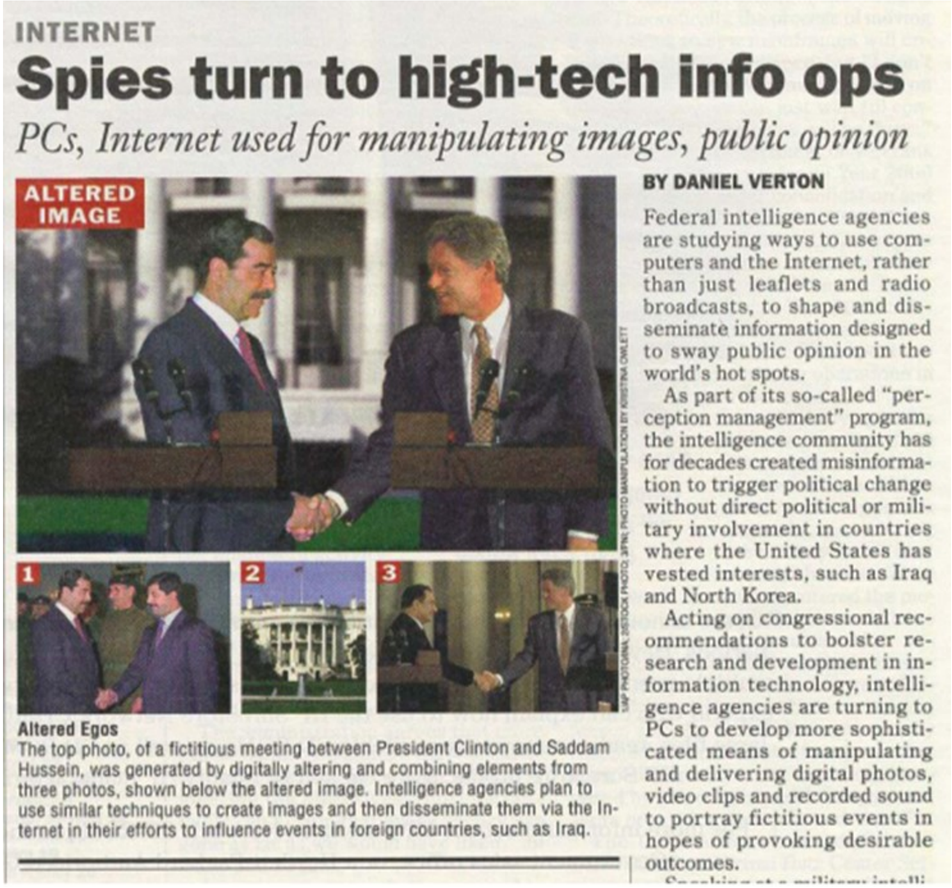
List of Tables

Chapter 1

Introduction

1.1 Overview

The intention of digital forgery is to change the image's original information by adding segment or removing segment. Due to today's powerful retouching software tools it has been very prominent to create the forged image. Detecting digital image forgery is a challenging task in the field of crime, journalism etc. Fake photos many a times are used to publicize in magazines or newspapers. Now a days, many cases are noted in regard to the defaming business as well as political opponents by using fake photographs and videos. This makes, it very essential to know about the integrity of the photos and frames in the video clippings so as to detect the truth. Several types of forgery can be happened such as image splicing or tempering or copy-moved etc. Image splicing is a form of photographic manipulation in which there is digital splicing of two or more images into a single composite image. Image tempering means manipulation of image to achieve a specific result by contrast changing, creating illuminations, color changing, blurring etc. Copy-move forgery is special type of forgery where a segment is copied from a specific image and pasted it within that image. Copy-move forgery is mostly used for creating forged image because the image is forged in such a way so that we cannot detect the forged image by our naked-eyes. So we need a method for detecting the forged image as we cannot do it by our eyes. But the proposed methods are not robust in any types of challenges. That's why still it is a hot field in digital image processing. In the figure: 1.1 we can see a sample of digital image forgery. In the bottom line there are three images which are original. But from those three images the upper image is created which is forged and where we can see that Saddam Hossain is shaking his hand with Bill Clinton in front of the white house. For this purpose the image of white house is scaled and blurred.


ALTERED IMAGE
1
2
3
1)AP PHOTO/RNA; 2)ISTOCK PHOTO; 3)FNI; PHOTO MANIPULATION BY KRISTINA OWLETT

INTERNET

Spies turn to high-tech info ops

PCs, Internet used for manipulating images, public opinion

Altered Egos
The top photo, of a fictitious meeting between President Clinton and Saddam Hussein, was generated by digitally altering and combining elements from three photos, shown below the altered image. Intelligence agencies plan to use similar techniques to create images and then disseminate them via the Internet in their efforts to influence events in foreign countries, such as Iraq.

BY DANIEL VERTON

Federal intelligence agencies are studying ways to use computers and the Internet, rather than just leaflets and radio broadcasts, to shape and disseminate information designed to sway public opinion in the world's hot spots.

As part of its so-called "perception management" program, the intelligence community has for decades created misinformation to trigger political change without direct political or military involvement in countries where the United States has vested interests, such as Iraq and North Korea.

Acting on congressional recommendations to bolster research and development in information technology, intelligence agencies are turning to PCs to develop more sophisticated means of manipulating and delivering digital photos, video clips and recorded sound to portray fictitious events in hopes of provoking desirable outcomes.

Speaking at a military intelli

Figure 1.1: Example of Forged Image

1.2 Problem Statement

As copy-move forgery is difficult to detect comparing to tempering or splicing so it is used mostly. But a digital image forgery can be happened in any ways. So we need to detect all kinds of forgery. But matter is researchers have already found that it is quite impossible to develop any method which can detect all kinds of digital image forgery. That's why we need to focus on a specific type of forgery. And our focus is on copy-move forgery. In copy-move forgery only one image is there but any segment of the image is copied and pasted so that forged image can enhance any image information or reduce image information. In figure: 1.2 we can see an example of copy-move forgery. There is forest and two trucks in the original image. But in the forged image there is only one truck. Another truck is removed by copy of green forest trees.

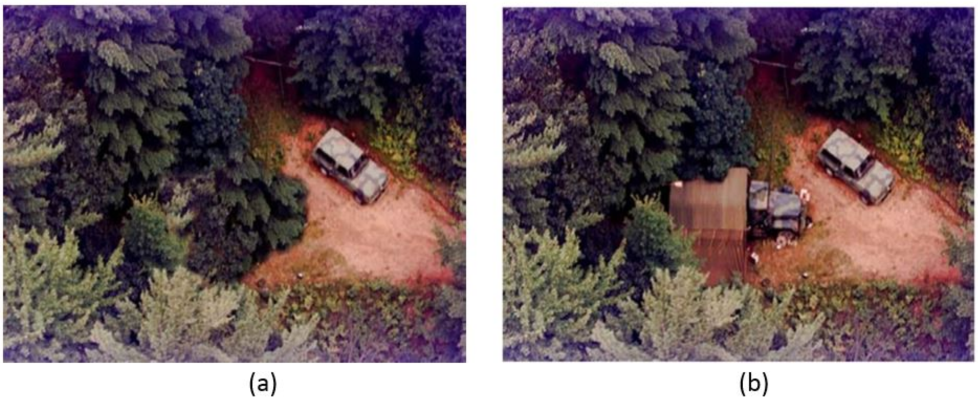


FIGURE 1.2: Copy-move Forgery (a) forged image (b) original image

1.3 Research Challenges

In copy-move forgery detection process there are some challenging situations. These all kinds of situation has not been overcome yet in the existing methods. The challenges are like below.

- Noisy Image: The forged image may be noisy. In that case the detection method will not get proper image information. When we try to compare the region then we will get some wrong information.

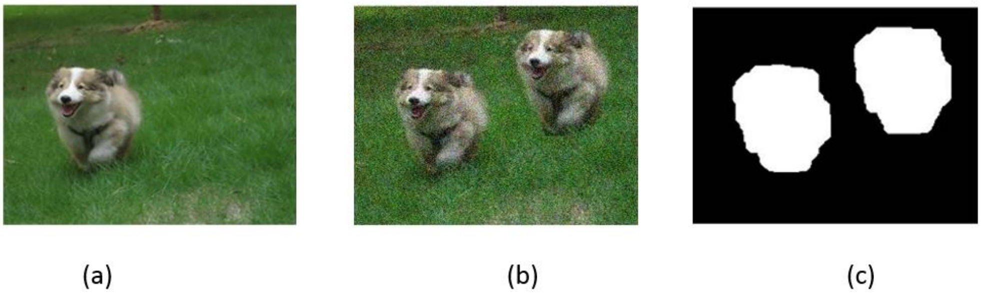


FIGURE 1.3: Noisy (a) original image (b) forged image (c) ground truth

- JPEG Compression: If the image is compressed after forged then a lot image information is lost. As a result we will not get enough information. It may happened that original image segment region has lost some information then it will be very difficult to detect the forged portion of the image.
- Flat or Uniform Region: It may be happened that image has some flat or uniform region suck as blue sky or river. In that case flat region may lead the false positive result. So we have to develop method so that it can reduces false positive result in case of that types of situation.

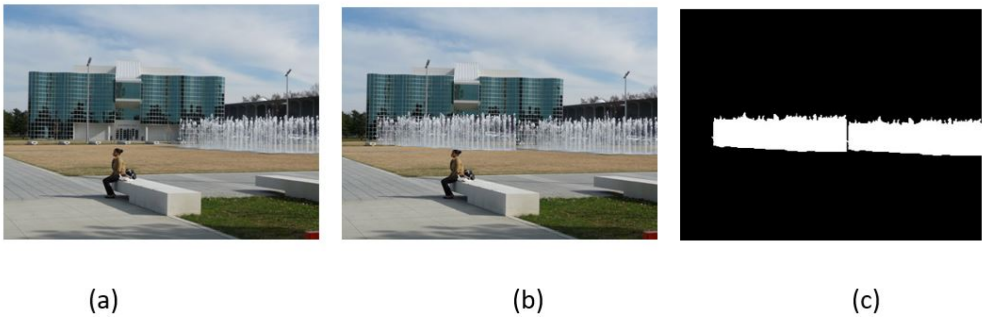


Figure 1.4: Flat Region (a) original image (b) forged image (c) ground truth

- Exact or approximate copy: The copied and pasted segment may not be exact same always. It may be happened that sometime using retouching tools the pasted segment is slightly changed. In that case it is very difficult to get the similarity which cause difficulty in detecting the forged image. So besides exact match we have to consider approximate match also.
- Scaling the pasted segment: When the copied image is pasted then if it scaled before pasting then the original block will not be the same like forged. In that case the detection method will have to face trouble.

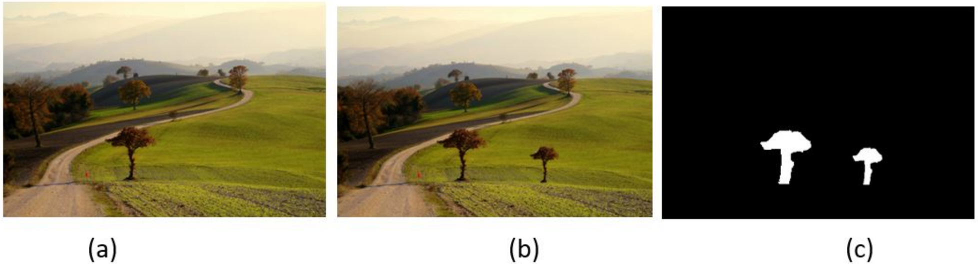


Figure 1.5: Scaling (a) original image (b) forged image (c) ground truth

- Rotating the pasted segment: Besides the scaling if the pasted segment is rotated then when we make comparison with the original block then we will not get the similarity exact pixel positions. So detection will be difficult.

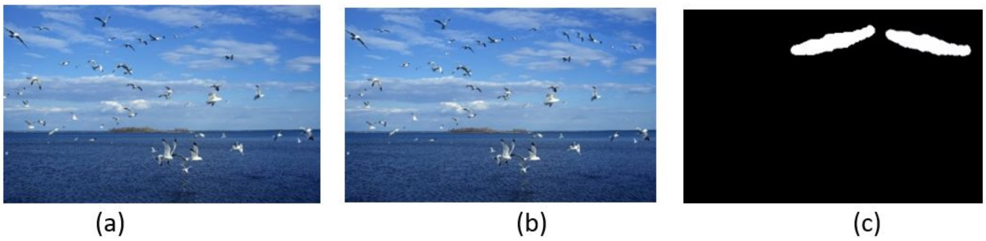


Figure 1.6: Rotation (a) original image (b) forged image (c) ground truth

- Flipping the pasted segment: So far we have considered about scaling and rotation but the segment may be flipped also. In that case the problem arises as like rotation.

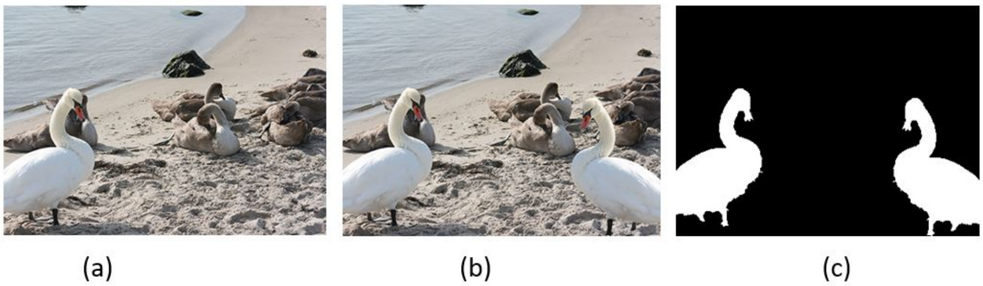


FIGURE 1.7: Flipping (a) original image (b) forged image (c) ground truth

1.4 Thesis Objectives

As it appears that among the different types of digital image forgery copy-move forgery is mostly used because it is forged as like the image is original. So people generally are used to follow this type of forgery for creating forged image. Most of the existing method failed to successfully detect the forged image in different challenging situation. So we want to develop a detection method for detecting the copy-move forgery. Already we have discussed about the research challenges in this field. And also we claimed that no existing method can ensure different situations. Some methods are working well with approximate match but fail to detect in noisy or compressed situation. Even if some methods are working with compressed image but they are valid up to a specific compressed factor. Some methods are suitable for rotation but it only for 90, 180, 270 degrees. They are not suitable for flipping or other types of rotations. And the complexity of the methods is comparatively high. So still there are a lot of scopes to develop the detection method. We want to develop a method which will ensures all types of challenges as well as less complexity. So our thesis aimed at proposing a method which can detect copy-move forgery in case of different challenges like

- Gaussian blurring
- Gaussian noise
- Rotation
- Flipping

- JPEG compression

1.5 Thesis Contributions

We have proposed a detection method for copy-move forgery which is robust comparing to others. The major contributions of our thesis are summarized as follow:

- We have presented a circular block extraction approach of detection method for copy-move forgery which is robust against different challenges of this fields like Gaussian Nosie, Gaussian Blurring, any angel rotation, flipping, JPEG compression etc.
- We have investigated the existing methods for identifying their weakness and tried to resolve it using our proposed method.
- Our proposed method describes a circular block extraction approach for which our method is rotation invariant.
- In the comparison step we have compared up to nth consecutive blocks instead of two consecutive blocks which ensures better detection process for JPEG compression with low quality factor as well as blurring also.
- For experimental results we have implemented our proposed method and figured out the performance using benchmark datasets.
- Comparative analysis of our method with others have also done for getting the comparative performance of our proposed method.

1.6 Organization of Thesis

The rest of the thesis will be organized as follows: in Chapter 2 we present the literature review of existing methods and their performance as well as limitations for the detection process. In Chapter 3, we propose our detection method for copy-move forgery. There we discuss about the overall idea of our proposed method and step by step implementation process. In Chapter 4, experimental set up, experimental result and performance analysis of our proposed method with various challenges are shown. Besides with other methods a comparative analysis is also shown. Finally, in Chapter 5, we conclude our thesis contributions and shows the future scopes for further developing the proposed method.

Chapter 2

Literature Review

2.1 Forgery Detection

Digital images in the modern world play very important role in areas like forensic investigation, insurance processing, surveillance systems, intelligence services, medical imaging and journalism. But the basic requirement to believe what we see is that the images should be authentic. With the advancement of technology and availability of fast computing resources, it is not very difficult to manipulate or forge the digital images. The availability of some software tools makes the problem more menacing. Despite this there is no method available to detect all types of tampering with accuracy. Before coming to the discussion of forgery detection techniques; it is necessary to know about the different types of tampering done with digital images. There are many ways to categorize the image tampering based on different points of view. Generally, we can say that the most often performed operations in image tampering are:

- Deleting or hiding a region in the image
- Adding a new object into the image
- Misrepresenting the image information

Copy move image tampering is one of the frequently used techniques to hide or manipulate the content of the image. Some part of the same image or some other image is pasted on another part of image. To detect the region of some other image statistical methods may work but if the region pasted belongs to the same image then it's quite difficult to detect this forgery. Many methods have been suggested to detect this type of forgery detection.

2.2 Related Works

Although many papers have been published suggesting different detection techniques, the challenges which are faced have not been overcome yet. Every algorithm has some lacking and limitations. For this reason even today it is still a prominent research topic in Image Processing field. For detecting copy-move forgery a number of methods have been proposed already. Most of the methods use square block extraction and different ways for feature extraction such as DCT [1], PCA [2], SVD [3] etc. which takes much time for detection. Another type of approach is sub-blocking the blocks and then feature extraction from the sub-blocks [4–6]. For getting the rotation invariant features some method uses some rotation invariant approach using SIFT [7], LBP [8] etc. Some method also uses circular block extraction [9] for rotation invariant. But these methods could not ensure the other challenges like low SNR or small sized tempered image. Finally, each method uses some sorting process like lexicographical sorting, k-d tree sorting, radix sorting etc. But all methods have some limitations which could not overcome all types of challenges.

2.2.1 Feature Extraction Approach

From the existing works we can get the idea of different approach for detecting the copy-move forged image. J. Fridrich et al. [1] suggested one of the earliest method which was overlapping square block extraction and using Discrete Cosine Transformation (DCT) co-efficient feature extraction process is done. Too much false matching are shown and for flat region this method cannot perform well. Popescu et al. [2] proposed another way using block matching approach and for feature extraction method they suggested Principal Component Analysis (PCA). This method solves the previous problem by reducing feature vector but can detect approximate match with additive noise up to 30db and JEPG image compression factor up to 65, not possible for rotation scaling or flipping. Li et al. [3] tried to reduce the dimension of the feature by using Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD). Although the authors claim high accuracy in the presence of compression (JPEG), but proper robustness against scaling and rotation cannot be found.

2.2.2 Sub-Blocking Approach

Another type of approach is sub-block extraction. Different types of sub-block methods have been proposed for feature extraction which are comparatively faster than the others. Using RGB value and sub-block Lue et al. [4] proposed a seven feature extraction method which works only for color image. Lin et al. [5] proposed sub block method extracting nine features which works only for JPEG compression, noise and some fixed angles. Vivek Kumar Singh and R.C. Tripathi [6] proposed another method from sub-block and Discrete Wavelet Transform (DWT) which extracts nine features. But these have limitations for rotation, scaling, JPEG compression etc.

2.2.3 Rotation and Scale Approach

For getting the rotation invariant or scale invariant feature some methods use Local Binary Pattern (LBP) or Scale Invariant Feature Transform (SIFT). Hailing Huang et al. [7] proposed SIFT which focused on scale invariant feature. But for low SNR and small type of image they give very poor result. Leida Li et al.[8] used LBP focusing on rotation invariant feature. But it cannot detect the forged segment if it is rotated by geometric random angles. Some method also introduce circular block extraction for getting rotation invariant feature. Junwen Wang et al. [9] used Gaussian Pyramid Decomposition and circular block extraction for rotation invariant features. But this does not give any performance for scale invariant feature.

2.2.4 Circular Block Extraction Approach

Some method focused only particular side such as Weihai Li et al. [10] proposed a method using DCT Grid and BAG (Block Artifact Grid) which works only for JPEG images and it is robust for JPEG Compression. The method works even when the copied area does not belong to same image. Shuiming Ye et al. [11] also proposed a method using DCT co-efficient and Blocking Artifact Measure (BAM) which works only for JPEG compression. For improving time complexity M.Sridevi et al. [12] proposed a parallel method using JAVA thread and radix sort for faster calculation.

2.3 Overall Detection Process

After extracting features by any of methods finally, each method uses some sorting process like lexicographical sorting, k-d tree sorting, radix sorting etc. So the overall approach for detecting the forged image is to block feature extraction and sort the features of the block and figure out the duplicate regions. All methods have some limitations which could not overcome all types of challenges. Comparative analysis of these methods can be found in some survey papers [13–15]. We can get the idea of the overall detection process method from the figure: 2.1. Here in the figure: 2.1 we can see that at first image is preprocessed and block extraction technique is applied. Then feature extraction process is done according to respective detection method. Then similar block matching of the image portion is searched. Finally, desired detection area is detected as forgery.

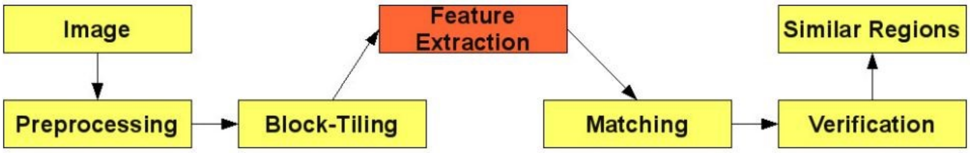


FIGURE 2.1: Overall detection process of copy-move forgery detection

Chapter 3

Proposed Method

3.1 Skeleton of Proposed Method

Our proposed method for detecting the copy-move forgery follows some steps which are shown in the figure. Our method is very simple but robust against the challenges. At first, we extract the overlapping block features and from that blocks we extract our desired reduced features following the proposed method. Finally, sorting the feature vectors and comparison with others, we get the similarity of the duplicated regions. Some threshold values are introduced in several steps for avoiding the false positive result.

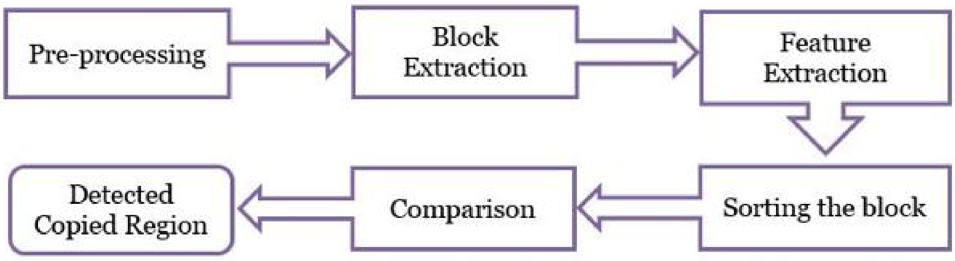


Figure 3.1: Overall detection process of copy-move forgery detection

3.2 Pre-Processing

For our detection method we need a gray scale image. So if we have colored image then we have to convert it into a gray scale image. For that conversion we can just follow the following conversion equation. It may happen that image is suffering from some kind of random noise. In that case, some noise reduction process can be applied for reducing the noise. For instance, if we find some salt and pepper noise then salt and pepper noise reduction procedure can be applied.

$$I = 0.299R + 0.587G + 0.114G$$

It may happen that image is suffering from some kind of random noise. In that case, some noise reduction process can be applied for reducing the noise. For instance, if we find some salt and pepper noise then salt and pepper noise reduction procedure can be applied

3.3 Block Extraction

In our proposed method, we extract circular overlapping block features for getting the rotation invariant features. At first we divide the input image into overlapping blocks of b*b pixels. Each block contains three concentric Circles which have different radius like the figure. The largest circle is named as *circle 1*, the smaller one as *circle 2* and the smallest one as *circle 3*. Thus if our image size is $m * n$ then we have $(m - b + 1) * (n - b + 1)$ blocks in the whole image.

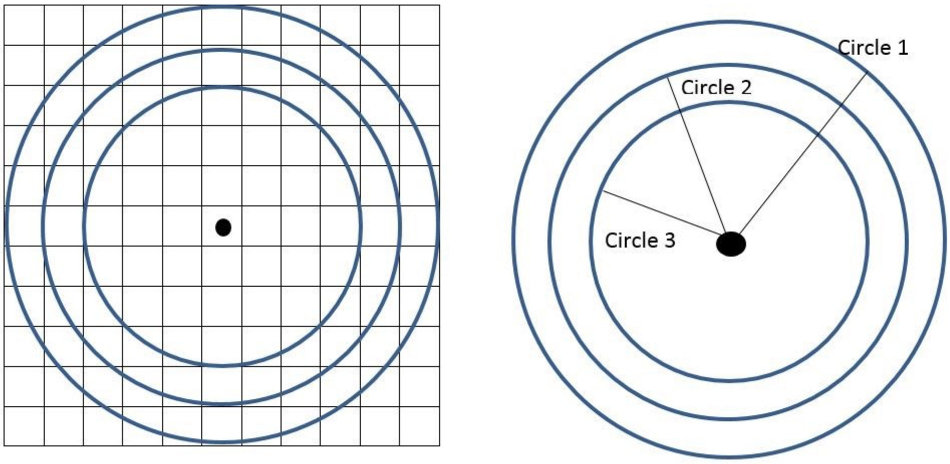


FIGURE 3.2: Overall detection process of copy-move forgery detection

3.4 Feature Extraction

Now for each block extraction we have to reduce the features. For our proposed method we will make a feature vector. For each block B_i ,feature vector

$$V_i = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12})$$

is computed and saved in an array. For each block, 12 features will be calculated where feature f_i will be

- $f_i = \mu(\text{circle i})$ where $i = 1, 2, 3$
- $f_{3+i} = \frac{\sum(I_i(x,y)-\mu_i)^2}{\mu_i}$ where $i = 1, 2, 3$
- $f_{6+i} = \frac{f_i}{f_1+f_2+f_3}$ where $i = 1, 2, 3$
- $f_{6+i} = \frac{f_i}{f_4+f_5+f_6}$ where $i = 4, 5, 6$

3.5 Sorting the Feature Vector and Comparison

After getting all feature vector for each block we get the final value of a two dimensional array where each row is representing each block and its features. Now before making comparison we will sort the two dimensional array. Each row will be lexicographically sorted using Radix Sort method. Now for finding out the similarity between two blocks we will compare each block with it's up to nth consecutive blocks in the sorted rows. For information loss or some post-processing it may happen that our similar block is just go far from the immediate position. So we are going for up to nth consecutive rows. Value of n can be any number. If it is high then we will get finer result but time complexity will be high.
For finding the similarity we use some threshold values. Two blocks are said to be matched, if it passes some threshold values. In our method we use three threshold values which are distance threshold D, frequency threshold F and features threshold F_t. If the distance between two blocks are less than distance threshold D, then they are discarded from the matched list. Distance can be calculated by using following equation.

$$D(V_i, V_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

We are doing these to avoid the uniform flat regions and to ensure that our copied region is keeping aloof for a minimum distance. Then if it passes the distance threshold we can go for frequency threshold F. Number of frequency of the copied region is counted and for detecting duplicated regions. As we are taking very small blocks so for detecting region we have to go for a larger region. Small block increases the false positive result. So we are using this frequency threshold F to get a larger region which are maintaining some distance. Finally, for features matching we are using the feature threshold F_t. and we will get the similar rows from the two dimensional array which lead to duplicated regions. As we have taken the

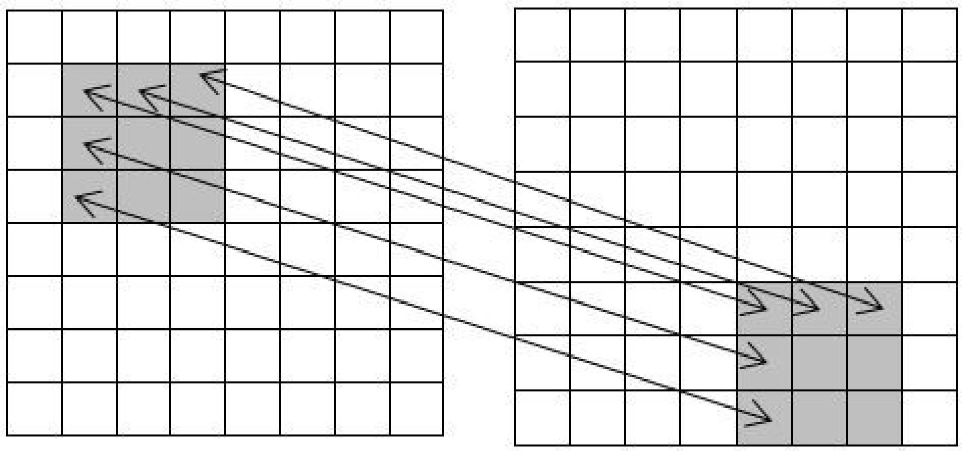


FIGURE 3.3: Overall detection process of copy-move forgery detection

circular block features so we will get rotation invariant features. But if rotated forged region presents then we cannot use the frequency threshold F. Because the distance of the block will not be same. So in that case frequency threshold F=0. As we cannot use this threshold so we will be introduced with some false positive result for rotated forged region.

Chapter 4

Experimental Result and Performance Analysis

4.1 Data Set and Experimental Setup

In our experiments, we have implemented the detection method in Matlab 2010 and all the simulations are performed on a personal computer of 2.13GHz processors with 2 GB main memory. Most of the test images are collected from a benchmark image dataset from the Internet and then copy move forgery is done with the help of Photoshop to those images. We have also implemented five existing methods and run those methods for each images. As there are several challenges in Copy-Move forgery like Rotation, Flipping, Adding noise, JPEG compression, Gaussian blurring etc. Our first section of dataset contains only forged images without any modification and their corresponding ground truth (actual output result). Then simulation was done for those images to get the output result. Our second section of the dataset contains noisy forged images and their corresponding ground truth. For checking the performance of noisy images we have added additive Gaussian noise with Matlab program. We have checked for different noise ratios (40db, 35db, 30db, 25db, 20db) and evaluate their performance for our own detection method as well as implemented existing methods for comparison. Our third section of the dataset contains blurred images made by Matlab program. Blurring was done with different standard deviation (1, 2, 3, 4, 5, 6, 7, 8) and the window size was 5*5. After that performance was evaluated for those images with our detection method as well as others implemented methods. First row of figure: 12 shows one blurred image from our dataset. Our fourth section of the dataset contains rotated and flipped images. Rotation and flipping was done with the help of Photoshop for different angle values like 45, 90, 180, 270 etc. Evaluation was performed to those images to get the output result with our detection method. Our final and fifth section of the dataset contains compressed images with different quality factor. We have compressed the images with different quality factors (30, 40, 50, 60, 70, 80, 90) using FILEminimizer Pictures Software and checked their performance with our detection method as well as implemented existing methods for making comparative analysis.

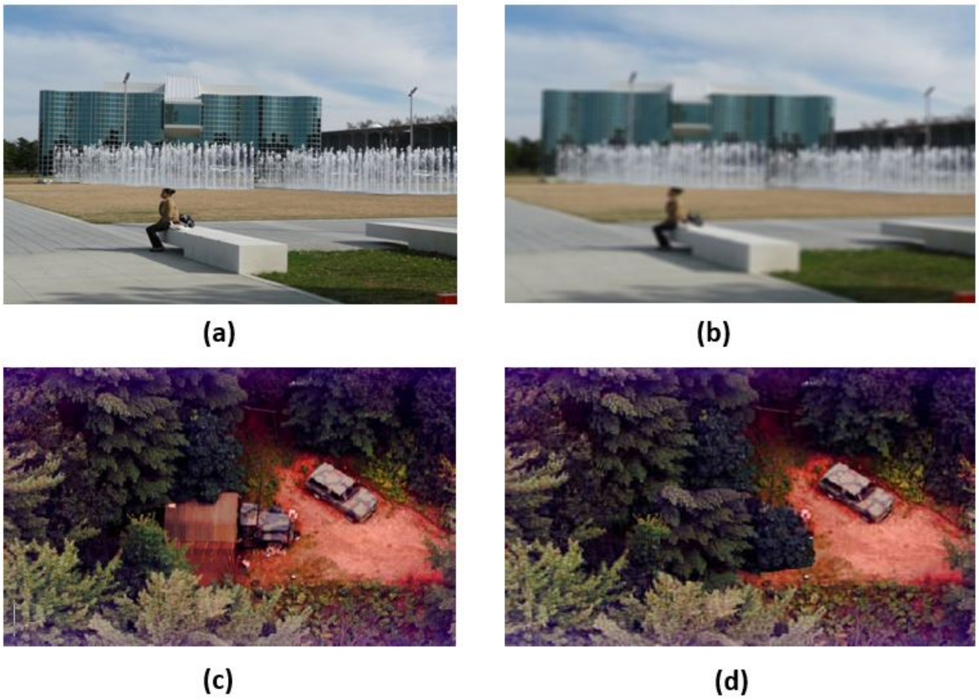


Figure 4.1: Some images of our dataset (a) original image (b) forged image after blurring (c) original image (d) forged image after compression

4.2 Performance Measurement

We have measured the performance by calculating Precision and Recall. Precision (also called Positive Predictive Value) is the fraction of retrieved instances that are relevant; While Recall (also called sensitivity) is the fraction of relevant instances that are retrieved. High recall means that an algorithm returned most of the relevant results, while high precision means that an algorithm returned substantially more relevant results than irrelevant. For classification tasks, the terms true positives, true negatives, false positives, and false negatives compare the results of the classifier under test with trusted external judgments. The terms positive and negative refer to the classifier's prediction (sometimes known as the expectation), and the terms true and false refer to whether that prediction corresponds to the external judgment (sometimes known as the observation). This is illustrated by the table: 4.1 below:

Table 4.1: Predicted class and Actual class

	actual class(observation)	
predicted class(expectations)	T_p correct result	F_p unexpected result
	F_n missing result	T_n correct absence of result

For several benchmark data images we have calculated the average Precision and Recall values according to

$$Precision = \frac{T_p}{T_p + F_p} Recall = \frac{T_p}{T_p + F_n}$$

Table: 4.2 gives us the result of precision and recall for different Signal to Noise Ratio (SNR) after simulating with our detection method in Matlab.

Table 4.2: Precesion & Recall values for different Gaussian Noise of our proposed method

Noise (SNR db)	**Precision**	**Recall**
40	0.98	0.94
35	0.98	0.93
30	0.95	0.91
25	0.90	0.88
20	0.86	0.82

Table: 4.3 provides us the performance with blurred images with respect to precision and recall after adding Gaussian blurring with different standard deviation (window size 5*5).

Table 4.3: Precesion & Recall values for different Gaussian Blurring of our proposed method

Gaussian Blurring (Standard deviation)	**Precision**	**Recall**
1	0.98	0.96
2	0.98	0.95
3	0.97	0.91
4	0.97	0.92
5	0.97	0.90
6	0.97	0.89

Table: 4.4 gives us the result of precision and recall for different JPEG quality factor.

Table 4.4: Precesion & Recall values for different Gaussian Blurring of our proposed method

JPEG Compression (Quality factor)	**Precision**	**Recall**
90	0.96	0.93
80	0.95	0.88
70	0.92	0.88
60	0.90	0.84
50	0.88	0.82
40	0.87	0.74
30	0.84	0.72

4.3 Comparative Analysis

In this section, we will show the output images for out detection method for each and every dataset. After that, we will also compare the result of our detection method with others method graphically. As we have five parts of dataset, we will gradually describe each part. Most of the cases our method performs well. For example, in case of JPEG compression other methods are not able to detect if the quality factor is less than 50. But our method is able to detect even if the quality factor is 30 with an acceptable accurate result. In case of Gaussian blurring, for a window size of 5*5 and standard deviation from 1-10, our method is able to detect Copy-Move forgery with high accuracy where the other methods performance is quite poor. In case of noise, our method is also able to detect with Gaussian noise up to SNR db 20 with a better performance. As we have taken circular block for feature extraction, our method is also able to detect if the copied region is rotated or flipped without calculating any complex calculation.

4.3.1 Without Any Modification

At first we have tested the dataset images which contain no modification and the result of precision and recall are shown in the following diagram figure: 4.2 for implemented five methods as well as our proposed method. Although most of the methods accuracy is good enough, our proposed method provides a little bit more accuracy than others method. After running all the methods to our first dataset images, we observed that output results are quite good compared with the ground truth. Almost all the output images are similar with ground truth. Because of the use of distance threshold and frequency threshold false positives are canceled

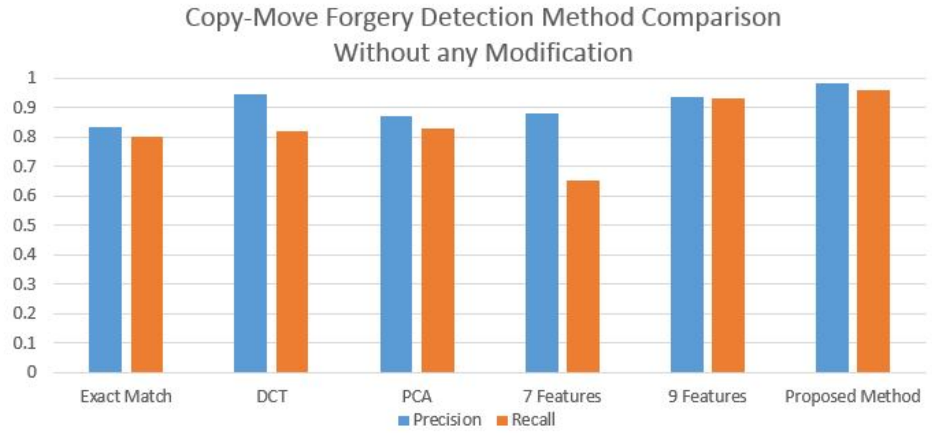


FIGURE 4.2: Comparison with other methods

mostly and provide better performance in spite of uniform regions also. Figure: 4.3 shows one of the output result where the fist image is the original image, next one is the forged image, then the ground truth and finally the output result image of our own detection method.

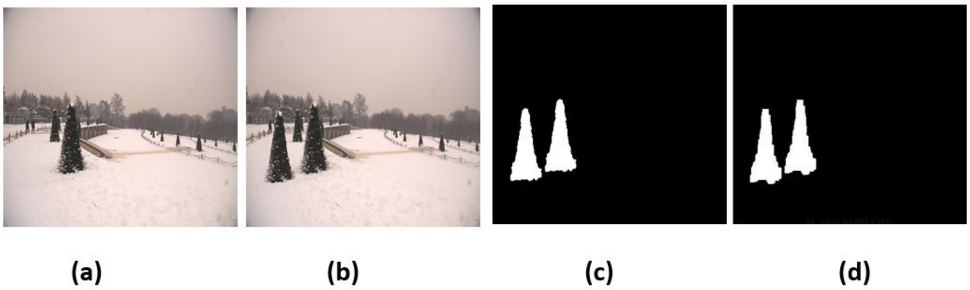


FIGURE 4.3: Performance without any modification (a) original image (b) forged image (c) ground truth (d) our detection result

4.3.2 Gaussian Noise

Our second dataset contains noisy images with different Signal to Noise Ratio (SNR) in db. After running the simulation we found that our method can almost correctly detect any Copy-Move forgery if the SNR is above 20, otherwise it provides poor result. Figure: 4.4 shows output result of our detection method with different SNR db and we can visualize that the performance of our method is quite robust. Precision and Recall value is also calculated for each images for our and others implemented methods. And finally average value of precision and recall is taken for comparing our method with others method. The following graph figure:

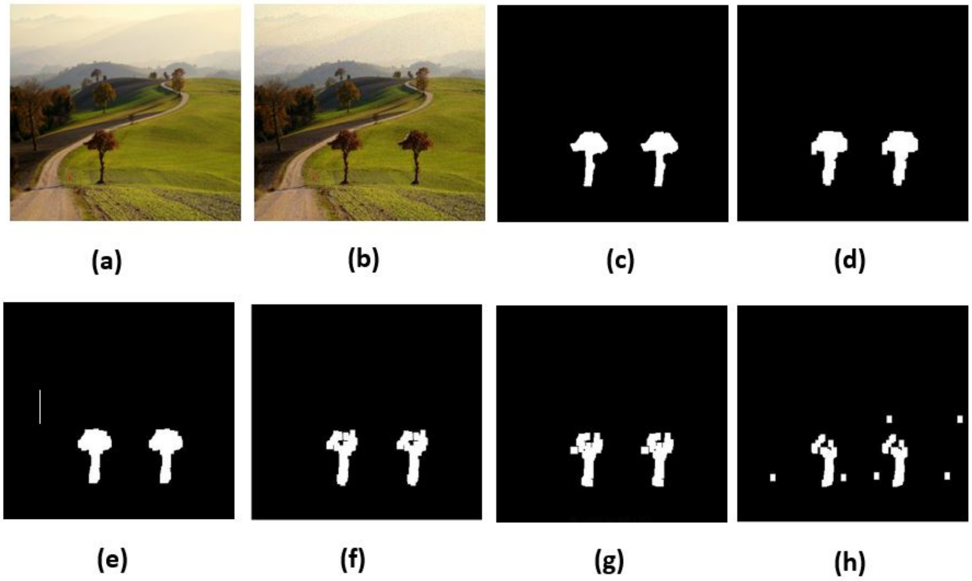


FIGURE 4.4: Performance with Gaussian noise (a) original image (b) forged image after adding noise (c) ground truth (d) Output result image without noise (e), (f), (g), (h) Output result image with SNR (db) 40,35,30,25 respectively

4.5 stands for the comparison of our detection method and previous methods for Noisy images with different SNR db.

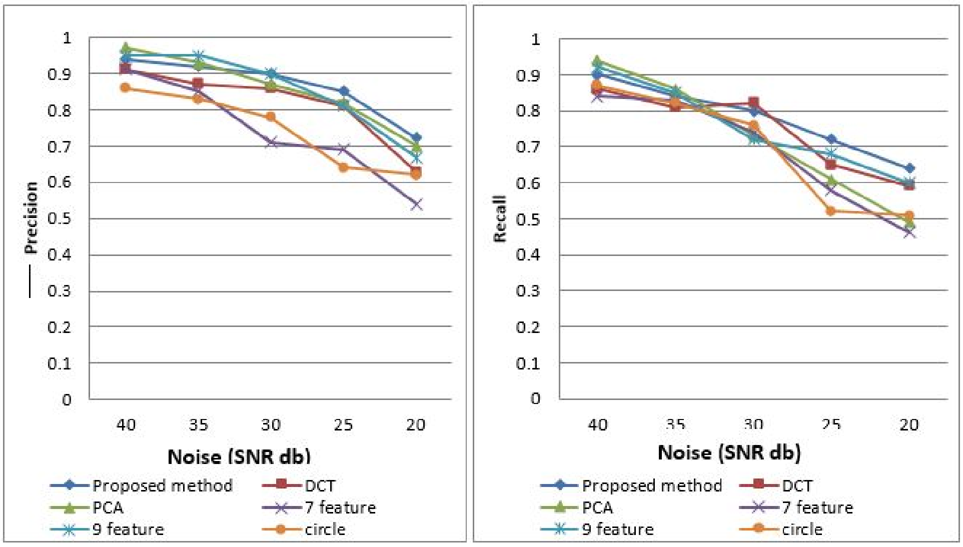


FIGURE 4.5: Comparison of our detection method and previous methods for Noisy images

4.3.3 Gaussian Blurring

Our third dataset contains blurred images with different Standard deviation (window size 5*5). After running the simulation we found that our method can almost correctly detect any Copy-Move forgery if the standard deviation is less than 10 for 5*5 window. Figure: 4.6 shows output result of our detection method with different Standard deviation and we can visualize that the performance of our method is highly robust. Precision and Recall value was also calculated for each images for our and others implemented methods. And finally average value of precision and recall was taken for comparing our method with others method. The following graph figure: 4.7 stands for the comparison of our detection method and previous methods for Blurred images with different Standard deviation.

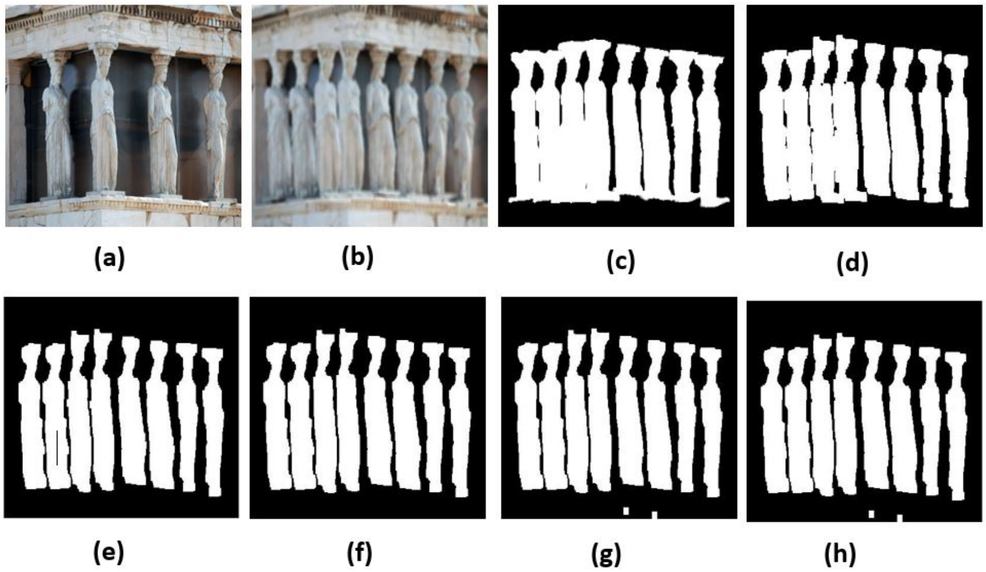


Figure 4.6: Performance with Gaussian blurring (a) original image (b) forged image after blurring (c) ground truth (d) Output result image without blurring (e), (f), (g), (h) Output result image with standard deviation 2,4,6,8 respectively,(window size 5*5)

4.3.4 Rotation and Flipping

Our forth dataset contains some rotated and flipped images. Rotation was done with different random angles including 90, 180, 270 degree. As our features are calculated from circular block, for any random rotation our method will work correctly. As frequency threshold is not valid in case of rotation, some false positive results will arise especially if the image contains uniform regions. But still our features are robust enough to detect any Copy-Move forgery with rotation and flipping with an acceptable level of accuracy. Figure:4.8 shows that, the first image

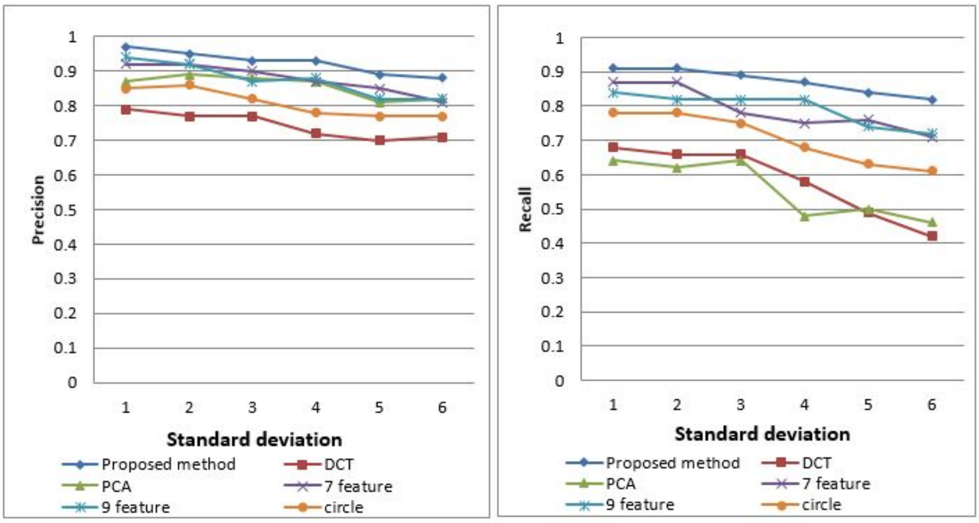


Figure 4.7: Comparison of our detection method and previous methods for Blurred images

is the original image. Then horizontal flipped was performed using Photoshop. Next image is the ground truth and finally output with our detection method.

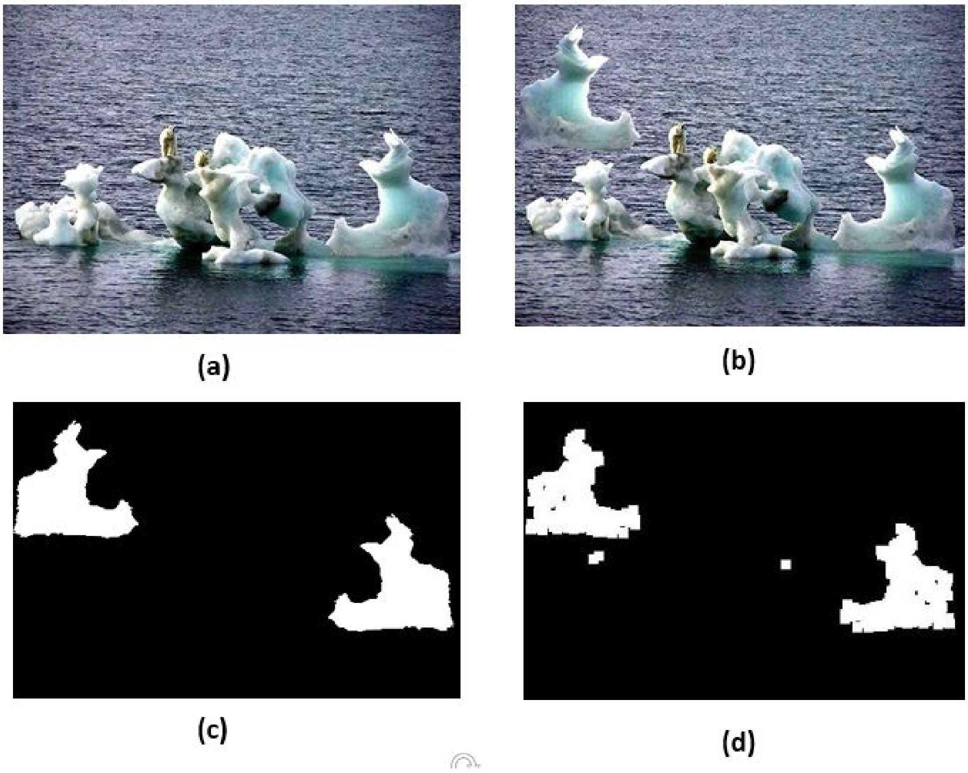


Figure 4.8: Performance with horizontal flipping (a) original image (b) forged image after flipping (c) Ground truth (d) Output result image with our detection method

4.3.5 JPEG Compression

Next we evaluated the result for different JEPG compression quality factors (30, 40, 50, 60, 70, 80, and 90) with our fifth part of dataset. Figure: 4.9 shows that, the image is the original image; the next one is the forged compressed image. After that ground truth is given and the forth picture is the output result of our detection method without compression. Next four images provide the output result of our detection method with JPEG compression quality factor 90, 70, 50 and 40 respectively. The following graph figure: 4.10 gives us the view of performance result with respect to precision and recall. We can see that up to quality factor 60 most of the methods are able to detect the forgery but the accuracy is very less when quality factor goes under 60 and the performance of recall is highly less than the performance of precision. But our method is able to detect even if the quality factor is 40. Combination of our robust features and checking nth consecutive blocks for matching help to achieve this robust result.

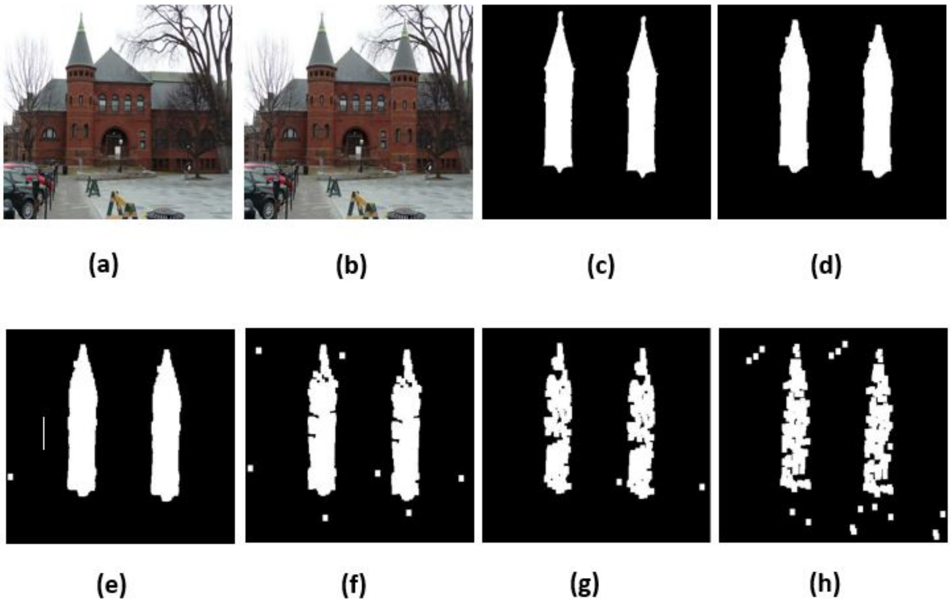


Figure 4.9: Performance with JPEG Compression (a) original image (b) forged image after compression (c) ground truth (d) Output result image without compression (e), (f), (g), (h) output result image with quality factor 90, 70, 50, 40 respectively

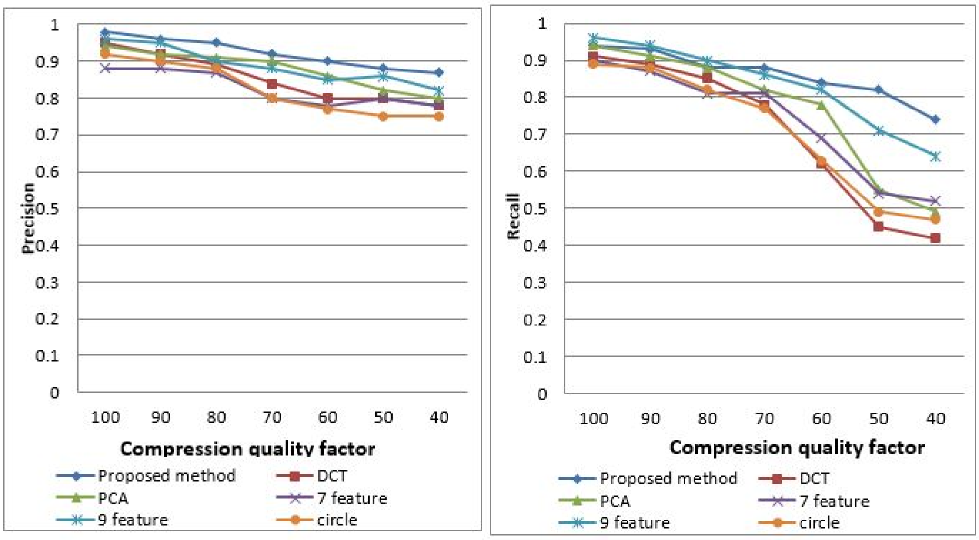


FIGURE 4.10: Comparison of our detection method and previous methods for JPEG Compression

Chapter 5

Conclusion

5.1 Summary of Contributions

Copy-move forgery detection has been the most popular in the ?eld of digital image forensics for the increasing nature of copy-moved forged image. Already many methods have been proposed for the detection of copy-move forgery. The main challenge of this detection is to be feasible in any kinds of challenging situation like compression, scaling, noise addition, rotation, flipping etc. Another important issue is computational time means time complexity which is important considering the large images. We explored the common steps of different existing methods and figured out their contributions as well as limitations. For getting clear analysis we have implemented several existing methods and comparison among themselves. So that our findings are proper and precise. This thesis paper describes a detection method for copy-move forgery which is simple but efficient in the challenge of JPEG compression, flipping and rotation for any angles as well as good for noise, blurring also. The reason for any angle rotation detection is taking the circular blocks. Moreover, we have considered the comparison after sorting up to nth consecutive blocks instead of comparing two consecutive blocks. As a result our proposed method is showing better performance in case of JPEG Compression for low quality factor and Blurring. Besides, time complexity is also good enough for detecting the forged image. We figured out that our proposed method is giving better performance but having some false positive results for rotation.

5.2 Limitations and Future Work

Although our proposed method gives better performance comparing to others but it has got some false positive results for rotation if the image area is bested with flat or uniform region. The reason behind this is we cannot use one of our proposed threshold value in case of rotation. That's why we may get some false result for some rotated images. More over our proposed method is not scale invariant. So our future work is to decrease the false positive response for rotation and try to make it scale invariant detection method.

Appendix A

Matlab Simulation Code of Proposed Method

The matlab simulation code of our proposed method is given below:

```
I2=imread(’Sample.jpg’);
I=rgb2gray(I2);
figure, imshow(I);
I=double(I);
[row, col] = size(I);
N = row * col;
C = zeros(row,col);  %for binary output image
freq = zeros(row,col); %for counting the shift frequency
bSide = 9; % Size of side of one block
b = bSide^2; % Size of the total block
rowblock=(row-bSide+1);  colblock=(col-bSide+1);
Nb = rowblock*colblock;  %total number of overlapping blocks
nth_consecutive=30; total_feature=12;
mean1 =2;   contrast1 =.15; ratioMean=.01; ratioCon=.01;  sft_dist=35; frequency=5; %threshold values
A = zeros(Nb,total_feature+3); %Block representation of input image

mask=ones(9,9);  %first mask
mask(1,1:2)=0;mask(1,8:9)=0;mask(9,1:2)=0;mask(9,8:9)=0;
mask(2,1)=0;mask(2,9)=0;mask(8,1)=0;mask(8,9)=0;

mask2=ones(9,9);  %second mask
mask2(1,:)=0;mask2(9,:)=0;mask2(:,1)=0;mask2(:,9)=0;
mask2(2,1:3)=0;mask2(2,7:9)=0;mask2(8,1:3)=0;mask2(8,7:9)=0;
mask2(3,2)=0;mask2(3,8)=0;mask2(7,2)=0;mask2(7,8)=0;

mask3=ones(9,9);  %third mask
mask3(1,:)=0;mask3(9,:)=0;mask3(:,1)=0;mask3(:,9)=0;
mask3(2,:)=0;mask3(8,:)=0;mask3(:,2)=0;mask3(:,8)=0;
mask3(3,3)=0;mask3(3,7)=0;mask3(7,3)=0;mask3(7,7)=0;

tot=1;
```

```
for i=1:rowblock,
    for j=1:colblock,
        part1 = 0.0; part1n =0; var1=0.0;
        part2 = 0.0; part2n =0; var2=0.0;
        part3 = 0.0; part3n =0; var3=0.0;
        for x=1:bSide
            for y=1:bSide
                if mask(x,y)==1
                    part1= part1 + I(i+x-1,j+y-1);
                    part1n=part1n+1;
                end
                if mask2(x,y)==1
                    part2= part2 + I(i+x-1,j+y-1);
                    part2n=part2n+1;
                end
                if mask3(x,y)==1
                    part3= part3 + I(i+x-1,j+y-1);
                    part3n=part3n+1;
                end
            end
        end
        f1= part1/part1n;
        f2= part2/part2n;
        f3= part3/part3n;

        f4= f1/(f1+f2+f3);
        f5= f2/(f1+f2+f3);
        f6= f3/(f1+f2+f3);
        for x=1:bSide
            for y=1:bSide
                if mask(x,y)==1
                    var1= var1 + (I(i+x-1,j+y-1)-f1)^2;
                end
                if mask2(x,y)==1
                    var2= var2 + (I(i+x-1,j+y-1)-f2)^2;
                end
                if mask3(x,y)==1
                    var3= var3 + (I(i+x-1,j+y-1)-f3)^2;
                end
            end
        end
        f7= var1/(var1+var2+var3);
        f8= var2/(var1+var2+var3);
        f9= var3/(var1+var2+var3);

        A(tot,1)=floor(f1);
        A(tot,2)=floor(f2);
        A(tot,3)=floor(f3);
        A(tot,4)=var1/(f1*part1n);
        A(tot,5)=var2/(f2*part2n);
        A(tot,6)=var3/(f3*part3n);
        A(tot,7)=f4;
        A(tot,8)=f5;
        A(tot,9)=f6;
        A(tot,10)=f7;
        A(tot,11)=f8;
        A(tot,12)=f9;
        A(tot,total_feature+1)=i;
        A(tot,total_feature+2)=j;
        A(tot,total_feature+3)=tot;
        tot=tot+1;
    end
end

Z1=sortrows(A);  % sort rows lexicographycally

j=1; maxfreq=0;
for i=1:Nb-nth_consecutive-1
    for h=i+1:i+nth_consecutive
    flag=0;
      for k=1:3
        if abs(Z1(i,k)-Z1(h,k)) >= mean1
            flag=1;
            break;
        end
      end
      for k=4:6
        if abs(Z1(i,k)-Z1(h,k)) >= contrast1
            flag=1;
            break;
        end
      end
      for k=7:9
        if abs(Z1(i,k)-Z1(h,k)) >= ratioMean
            flag=1;
            break;
        end
      end
      for k=10:12
        if abs(Z1(i,k)-Z1(h,k)) >= ratioCon
            flag=1;
            break;
        end
      end

      shift=sqrt((Z1(i,total_feature+1)-Z1(h,total_feature+1))^2+(Z1(i,total_feature+2)-Z1(h,total_feature+2))^2);

      if flag==0 && shift >sft_dist
        Z5(j,1) = Z1(i,total_feature+3);
        Z5(j,2) = Z1(h, total_feature+3);
        x=abs((Z1(i,total_feature+1)-Z1(h,total_feature+1)))+1;
        y=abs((Z1(i,total_feature+2)-Z1(h,total_feature+2)))+1;
        Z5(j,3) = Z1(i,total_feature+1);
        Z5(j,4) = Z1(i,total_feature+2);
        Z5(j,5) = Z1(h,total_feature+1);
        Z5(j,6) = Z1(h,total_feature+2);
        freq(x,y)= freq(x,y) +1;
        if(freq(x,y)>maxfreq)
            maxfreq=freq(x,y);
        end
        j=j+1;
      end
    end
```

```
end
maxfreq

%mapping binary image
for i=1:j-1
    x11=Z5(i,3);
    y11=Z5(i,4);
    x22=Z5(i,5);
    y22=Z5(i,6);
    f1=abs(x11-x22)+1;
    f2=abs(y11-y22)+1;
    if freq(f1,f2) > max(maxfreq/1.2,0)

      for a1=x11:x11+bSide-1
        for b1=y11:y11+bSide-1
            C(a1,b1)=1;
        end
      end
      for a1=x22:x22+bSide-1
        for b1=y22:y22+bSide-1
            C(a1,b1)=1;
        end
      end
    end
end
figure, imshow(C);

%for precision & recall
BI2=imread('Sample_GroundTruth.jpg');
BI=rgb2gray(BI2);
BI=im2bw(BI);
tp=0; tn=0; fp=0; fn=0;
for i=1:row
    for j=1:col
        if BI(i,j)==1 && C(i,j)==1
            tp=tp+1;
        elseif BI(i,j)==1 && C(i,j)==0
            fn=fn+1;
        elseif BI(i,j)==0 && C(i,j)==1
            fp=fp+1;
        else
            tn=tn+1;
        end
    end
end
        precision=tp/(tp+fp);
        recall=tp/(tp+fn);
        accuracy=(tp+tn)/(tp+tn+fp+fn);
```

Bibliography

[1] J. Fridrich, David Soukal, Jan Lukas, "Detection of Copy-Move Forgery in Digital Images", *in Proceedings of Digital Forensic Research Workshop*, August 2003.

[2] A. Popescu, H. Farid, "Exposing digital forgeries by detecting duplicated image regions", *Technical Report TR2004-515, Department of Computer Science, Dartmouth College*, 2004.

[3] Guohui Li, Qiong Wu, Dan Tu, Shaojie Sun, "A sorted neighborhood approach for detecting duplicated region in image forgeries based on DWT and SVD", *IEEE, ICME*, 2007.

[4] W. Luo, J. Huang and G. Qiu, "Robust detection of region-duplication forgery in digital image", *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition, IEEE Computer Society, Washington, DC, USA*, 2006.

[5] H. J. Lin, C.-W. Wang, Y.-T. Kao, "Fast copy-move forgery detection", *WSEAS Transactions on Signal Processing 5 (5)*, 2009.

[6] Vivek Kumar Singh and R.C. Tripathi, "Fast and Efficient Region Duplication Detection in Digital Images Using Sub-Blocking Method", *International Journal of Advanced Science and Technology Vol. 35*, October, 2011.

[7] Hailing Huang, Weiqiang Guo, Yu Zhang, "Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm", *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008.

[8] Leida Li, Shushang Li,Hancheng Zhu, "An Efficient Scheme for Detecting Copy-move Forged Images by Lo cal Binary Patterns", *Journal of Information Hiding and Multimedia Signal Processing, Volume 4*, January, 2013.

[9] Junwen Wang, Guangjie Liu, Hongyuan Li, Yuewei Dai, Zhiquan Wang, "Detection of Image Region Duplication Forgery Using Model with Circle Block",

Vol. 1, 2009 International Conference on Multimedia Information Networking and Security, 2009.

[10] Weihai Li, Yuan Yuan, and Nenghai Yu, "Detecting Copy-Paste Forgery of JPEG image vai block artifact grid extraction", 2006.

[11] Shuiming Ye, Qibin Sun and Ee-Chien Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact", *IEEE*, 2007.

[12] M.Sridevi, C.Mala and S.Sandeep, "Copy-move image forgery detection in a parallel environment", 2012.

[13] Vincent Christlein, Christian Riess, Elli Angelopoulou, "A Study on Features for the Detection of Copy-Move Forgeries ", 2010.

[14] Sunil Kumar et al. "Copy-Move Forgery Detection in Digital Images: Progress and Challenges", *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 3 Feb, 2011.

[15] Christlein et al. "An evaluation of popular copy-move forgery detection approaches", *IEEE Transactions on information forensics and security*, 2012.

[16] Sevinc Bayram et al. "A Survey of copy-move forgery detection techniques", 2009.

[17] B.L.Shivakuma et al. "Detecting Copy-Move Forgery in Digital Images: A Survey and Analysis of Current Methods", *Global Journal of Computer Science and Technology Vol. 10*, 2010.

[18] Mohamadian Zahra et al. "Image Duplication Forgery Detection using Two Robust Features", *Research Journal of Recent Sciences Vol. 1(12)*, 2012.

[19] Babak Mahdian and Stanislav Saic "Detection of copy-move forgery using a method based on blur moment invariants", *Forensic Science International 171 (2007), Elsevier*, 2006.

[20] Sergio Bravo-Solorio and Asoke K. Nandi "Passive Forensic Method for Detecting Duplicated Regions Affected by Reflection, Rotation and Scaling", *17th European Signal Processing Conference (EUSIPCO)*, 2009.

[21] Nattapol Chaitawittanun "Detection of Copy-Move Forgery by Clustering Technique", *2012 International Conference on Image, Vision and Computing (ICIVC)*, 2012.

[22] Sevinc Bayram et al. "An Efficient and Robust Method for Detecting Copy-move Forgery", *2012 International Conference on Image, Vision and Computing (ICIVC)*, 2009.