

# SWT and SIFT Based Copy-Move Image Forgery Detection



Thesis submitted in partial fulfilment of the requirement for the degree of  
Bachelor of Computer Science and Engineering

Under the Supervision of

Dr. Jia Uddin

By

Taposh Das (13101093)

Md. Rasel Azam (13101295)

Rizbanul Hasan (13301065)

**School of Engineering and Computer Science**

December 2017

BRAC University, Dhaka, Bangladesh

## **Declaration**

We hereby declare that this thesis is based on results obtained from our own work. Due acknowledgement has been made in the text to all other material used. This thesis, neither in whole nor in part, has been previously submitted to any other University or Institute for the award of any degree or diploma.

### **Signature of Supervisor**

---

Dr. Jia Uddin

### **Signature of the Authors**

---

Taposh Das(13101093)

---

Md. Rasel Azam(13101295)

---

Rizbanul Hasan(13301065)

## **Acknowledgement**

First and foremost, we would like to thank Almighty Allah for enabling us to initiate the research, to put our best efforts and successfully conclude it.

Secondly, we are grateful to our respected Supervisor Dr. Jia Uddin for his contribution, guidance and support in conducting the research and preparation of the report. From the very beginning to the end, he had inspired us with his kindest words to effectively complete the paper.

We would like to express our gratitude to our parents as well as our friends, who helped us with their direct or indirect suggestions. We would also like to acknowledge the assistance we received from numerous resources over the Internet especially from fellow researchers' work.

Last but not the least, we thank BRAC University for providing us the opportunity of conducting this research and for giving us the chance to complete our Bachelor degree.

# Table of Contents

<b>Acknowledgement</b> .....	ii
<b>List of Figures</b> .....	v
<b>List of Tables</b> .....	vi
<b>List of Abbreviations</b> .....	vii
<b>Abstract</b> .....	1
<b>Chapter 1: Introduction</b> .....	2
1.1 Motivation.....	3
1.2 Contribution Summary.....	3
1.3 Thesis Outline.....	4
<b>Chapter 2: Background Analysis and Related Works</b> .....	5
2.1 RGB to Greyscale Conversion.....	5
2.1.1 Algorithm.....	5
2.2 Stationary Wavelet Transform (SWT).....	6
2.3 Scale Invariant Feature Transform.....	7
2.3.1 Scale Space Extrema Detection.....	8
2.3.2 Key-point Localization.....	10
2.3.3 Orientation Assignment.....	11
2.3.4 Generation of Key-point Descriptor.....	11
2.3.5 SIFT Algorithm.....	11
2.4 Key-point Matching Using g2NN.....	12
2.5 Agglomerative Hierarchical Clustering.....	14
2.6 False Positive Removal.....	16
2.6.1 RANSAC Algorithm.....	17

2.7 Related Works.....	17
<b>Chapter 3: Proposed Model</b> .....	<b>20</b>
3.1 Workflow of Proposed Model .....	20
3.1.1 Preprocessing.....	21
3.1.2 SIFT Feature Extraction.....	22
3.1.2.1 Scale Space Extrema Detection .....	23
3.1.2.2 Key-point Localization .....	24
3.1.2.3 Assignment of Orientation.....	25
3.1.2.4 Generation of Key-point Descriptors.....	25
3.1.3 Clustering.....	26
3.1.4 Key-point Matching.....	26
3.1.5 False Matches Removal .....	26
3.2 Data Collection .....	26
3.3 Tools Used .....	27
<b>Chapter 4: Experimental Setup and Result Analysis</b> .....	<b>28</b>
<b>Chapter 5: Conclusion and Future Work</b> .....	<b>36</b>
5.1 Conclusion .....	36
5.2 Future Work.....	36
<b>References</b> .....	<b>37</b>

## List of Figures

Figure 1: (a) Original Image, (b) Forged Image .....	2
Figure 2: RGB to Grey Conversion Algorithm.....	5
Figure 3: 2-D SWT Decomposition.....	6
Figure 4: Final Computation.....	7
Figure 5: DoG Pyramid Formation.....	9
Figure 6: Computation of Maxima and Minima of DoG.....	10
Figure 7: Pseudo Code of SIFT Algorithm.....	12
Figure 8: Dendrogram of Agglomerative Hierarchical Clustering.....	15
Figure 9: Hierarchical Clustering Using Linkage Method Ward (Represents a Clustering Overview of Four Colors) .....	16
Figure 10: Workflow of Proposed Model.....	20
Figure 11: Conversion of RGB to Gray Scale .....	21
Figure 12: 2-D SWT Decomposition of Input Image .....	22
Figure 13: DoG Pyramid Formation of Approximate Image.....	23
Figure 14: Initial Location of Key-points of Different Views Component .....	24
Figure 15: Accurately Selected Key-points .....	25
Figure 16: (a) Original image, (b) Forged image, (c) Forgery Detection.....	29
Figure 17: (a) Original, (b) (d) (f) and (h) are Tampered Images and (c), (e), (g) and (i) are Their Corresponding Detection of Forgery .....	33

## List of Tables

Table 1: Recorded Data from A Set of 20 Non-Tampered Images .....	30
Table 2: Recorded Data from A Set of 20 Tampered Images.....	31
Table 3: Outcome of Proposed Method .....	32
Table 4: Performance of Proposed Method .....	32
Table 5: Performance Analysis Based on Different Attacks on the Image .....	34
Table 6: Comparative Result with Existing Models.....	35
Table 7: Comparison of Robustness with Existing Methods.....	35

## List of Abbreviation

DCT	Discrete Cosine Transform
DoG	Difference of Gaussian
DWT	Discrete Wavelet Transform
FPR	False Positive Rate
FWT	Fast Wavelet Transform
g2NN	Generalized 2 Nearest Neighbor
HOG	Histogram of Oriented Gradients
MRA	Multiresolution analysis
PCA	Principal Component Analysis
RANSAC	Random Sample Consensus
SIFT	Scale Invariant Feature Transform
SURF	Speeded-Up Robust Features
SVD	Singular Value Decomposition
SWT	Stationary Wavelet Transform
swa	Decomposed approximate sub image by Stationary wavelet transform
swh	Decomposed horizontal sub image by Stationary wavelet transform
swv	Decomposed vertical sub image by Stationary wavelet transform
swd	Decomposed diagonal sub image by Stationary wavelet transform
TPR	True Positive Rate



## **Abstract**

In our proposed model we have implemented copy-move image forgery detection technique. Copy-move image forgery is one of the types of image forgery where a part of image is copied and then it is pasted in the same image having an intention to make a false image or to hide some important object within the image. Our purpose is to make an efficient and robust solution to this kind of image forgery. Our proposed system consists of few steps: (1) Stationary Wavelet Transform (SWT) is used to decompose the input image into four parts from which approximate image is taken as input for the next step. (2) Scale Invariant Feature Transform (SIFT) algorithm is then run on the approximate image extracted by SWT to extract the key point descriptor features. (3) The descriptor features are clustered using linkage method ward. (4) Clustered key points are compared to take decision whether image is tampered or not. (5) In post processing step false positive removal is done using Random Sample Consensus (RANSAC). Our proposed model after implementations performs 93% accurately over a certain dataset.

# Chapter 01

## Introduction

Digital image forgery detection is a technology that is used to detect whether an image is manipulated or not. There are various ways to manipulate an image e.g. copy-move forgery, image splicing, image retouching etc. Therefore, the task of detecting a forged image is very complex. Hence, the approach to handle and detect different types of forgery is different [21]. Among the various types of image tampering approach, copy-move is widely and commonly used. In copy-move image forgery a part of image is copied and then it is pasted in the same image having an intention to make a false image or hide some important object within the image. There are a number of copy-move image forgery detection algorithms but most of them are not robust and efficient in terms of computational expense and affine or geometric transformation. The goal of this proposed method is to detect forgery irrespective of all the ways of copy-move tampering including the tampering with geometric transformation with giving importance on the reduction issue of time complexity. Figure 1 shows an example of copy-move forgery performed on a JPEG image from the data set MICC-F220 [12].



**Figure 1: (a) Original Image, (b) Forged Image**

## 1.1 Motivation

Digital image contains huge amount of information. Human visual sensory system can receive pictorial information in less than a millisecond compared to textual information. Hence digital images are widely used as a means to transferring and conveying information. For instance, Instagram has become tremendously popular because it's primarily a social media platform based on image sharing. Other than that image is used in newspaper, magazines, and research based journals. Imagery information is used as a vital proof against various types of crime and acts as evidence for multifarious purpose. However, at the same time heavy image editing tools and software are widely and cheaply available using which content of an image can be easily tampered [22]. This process of manipulation of original image by applying various type of geometric transformation (rotation, scaling, resizing), adding or removing an object in the real image with the use of this editing tools is called digital image forgery [1,3]. These modifications are almost impossible to trace and detect by naked eyes. The purpose for such modification widely varies depending of circumstances and often can be as serious as hiding evidences or manipulating the awareness of the recipient. Therefore, to ensure the authenticity of image, there are many algorithms and models that are being developed to solve this issue. However most of these models have limitation either in time complexity or in detection accuracy. So, the detection system should overcome these limitations and difficulties.

## 1.2 Contribution Summary

The summarized overview of the main contributions is given as follows

- Decomposing the input image using 2D SWT
- Extracting the descriptor feature from the decomposed approximate image
- Clustering of features using linkage method –ward
- Comparison among the clustered features for forgery detection

### **1.3 Thesis Outline**

The rest of the thesis is organized as follows:

- Chapter 02 includes background analysis and related works.
- Chapter 03 presents our proposed model and its implementation
- Chapter 04 demonstrates the experimental results and comparison.
- Chapter 05 concludes the thesis and states the future research directions.

## Chapter 02

### Background Analysis and Related Works

#### 2.1 RGB to Greyscale Conversion

It is the process of converting an RGB image i.e. colorful image to grey image by the formula shown in Figure 2.

$$\text{IMG} = 0.2989R + 0.5870G + 0.1140B \dots \dots \dots (1)$$

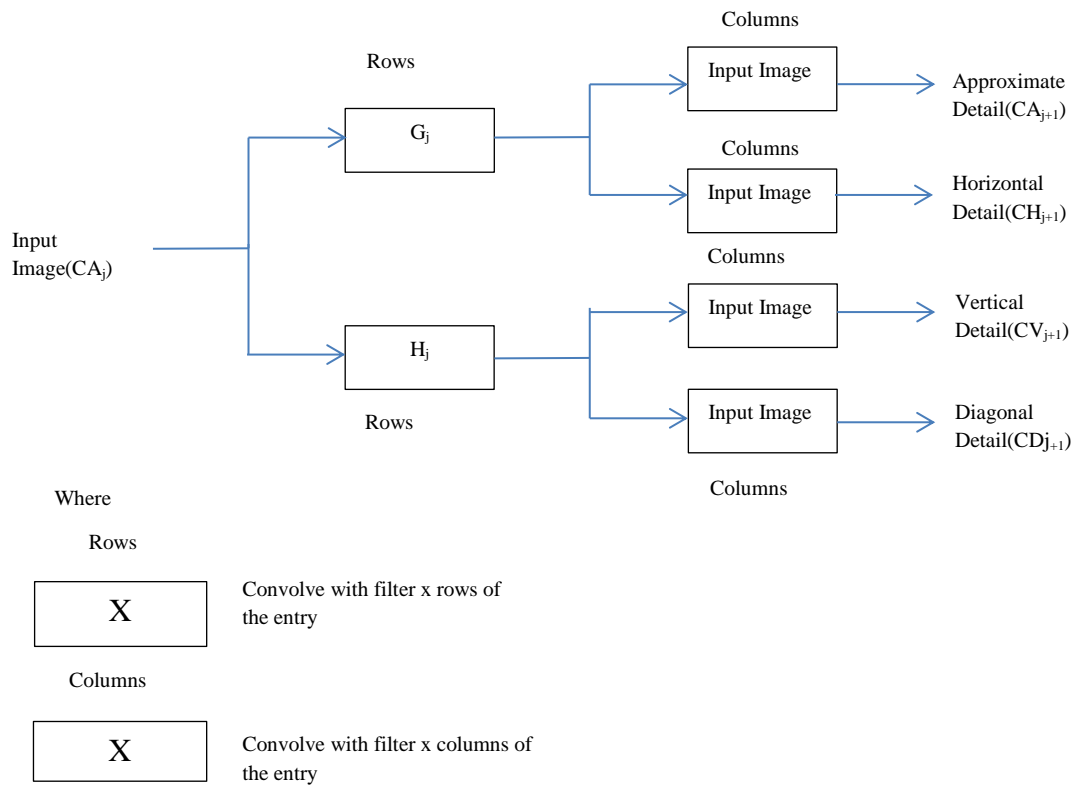
##### 2.1.1 Algorithm

```
For Each Pixel in Image  
  
    Set Red = Pixel.Red  
  
    Set Green = Pixel.Green  
  
    Set Blue = Pixel.Blue  
  
        Set Grey = (Red * 0.299 + Green * 0.587 +  
Blue * 0.114)  
  
    Set Pixel.Red = Grey  
  
    Set Pixel.Green = Grey  
  
    Set Pixel.Blue = Grey  
  
End For
```

**Figure 2: RGB to Grey Conversion Algorithm**

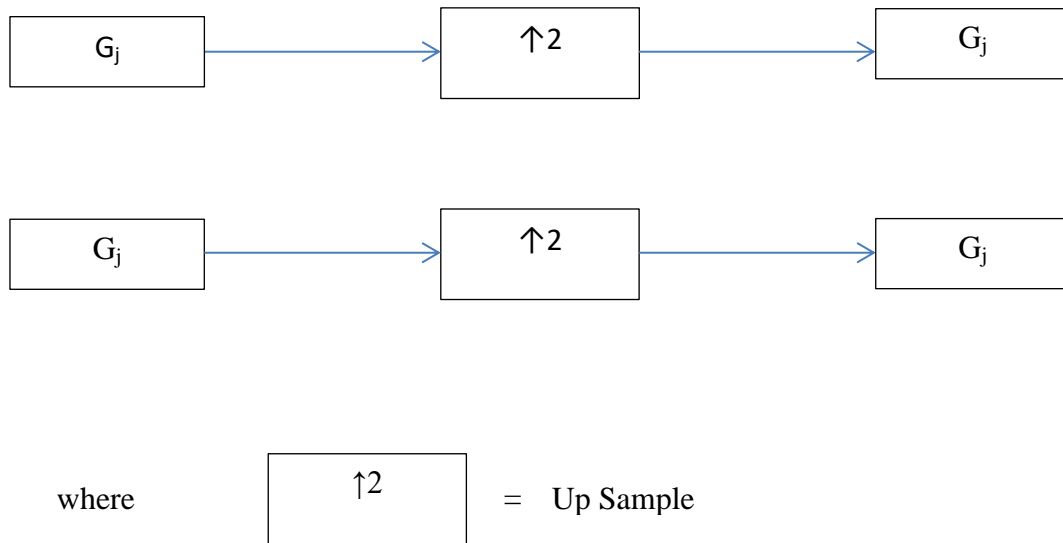
## 2.2 Stationary Wavelet Transform (SWT)

SWT is one of the several types of wavelet transformation e.g. discrete wavelet transform (DWT), Multiresolution analysis (MRA), Fast wavelet transform (FWT) etc. Wavelet transform has a wide field of application such as de-noising, image categorization, and recognition of patterns and also in medical field e.g. Magnetic resonance imaging. SWT is actually a modified version of DWT which is shift variant. SWT is not only shift invariant but also noise and blur invariant i.e. unlike DWT even if there is a shift in the signal, the transformed coefficients do not reorient and it shows better performance in terms of de-noising and edge identification [15]. SWT implement high and low pass filters to the input image at each level and at next stage produces two sequences each have the same length as the original sequence. But unlike DWT, the filters are changed at every level and it is done by padding them with zeros. However, the computational expense of SWT is more. The process of two dimensional decomposition of input image by SWT is shown in Figure 3.



**Figure 3: 2-D SWT Decomposition**

The shift invariance feature is achieved by removal of up and down sampling in DWT and by up sampling of the coefficient by the factor of  $2^{j-1}$  in that particular level. SWT actually enhances the resolution of edges with odd groups of coefficients. The final computation process is shown in Figure 4.



**Figure 4: Final Computation**

### 2.3 Scale Invariant Feature Transform(SIFT)

SIFT [11] is an algorithm developed by David Lowe which is used to extract features from digital images and these features are scale invariant and rotation invariant. More over SIFT is also to some extent invariant to different point of view of 3D camera and illumination [25]. With the implementation of SIFT algorithm a humongous number of features can be withdrawn and these features are invariant to different factors and thus these are considered to be eminently distinctive. So, the chance of finding a match between one feature to a huge database of features is highly probable and this would raise the issue of computational complexity. In order to reduce the computational expense, cascade filtering approach is adopted. Thus, whole process of initial detection of key-points to generating key-point descriptors is divided into four main stages.

### 2.3.1 Scale Space Extrema Detection

In this first step, the location and scale space of extrema are found out. The focus of this step is to actually detect those locations of key-points which are invariant to scale and rotation in image and it is done by seeking for distinctive features over various possible scales. The function that defines the scale space of an image is denoted by  $L(x, y, \sigma)$  which is produced by convolving Gaussian Function  $G(x, y, \sigma)$  with the input image  $I(x, y)$  [11].

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \dots\dots\dots(2)$$

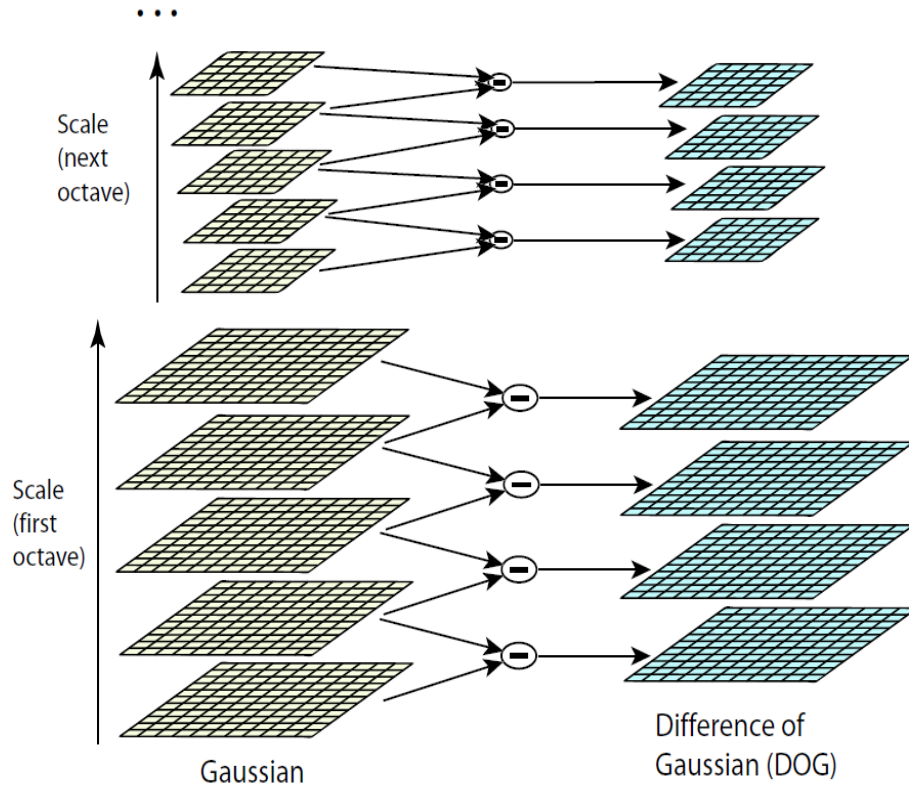
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \dots\dots\dots (3)$$

Where  $*$  denotes convolution operation, sigma is the scale space factor of Gaussian normal distribution,  $(x, y)$  denotes the pixel coordinates of image. To make the detection of key-points more reliable, efficient and stable DoG Function  $D(x, y, \sigma)$  is required. It is computed by convolving the difference of two nearby scales separated by a constant scaling factor  $k$  with the input image.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \dots\dots\dots(4)$$

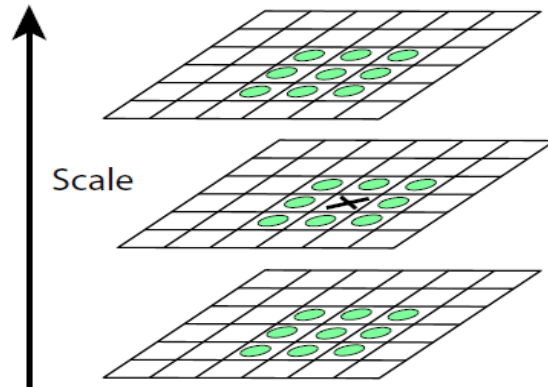
The purpose behind using  $D(x, y, \sigma)$  are it is comparatively cost-effective function since the smoothed image requires to be calculated for all types of view point for the description of scale space features. Moreover, by applying plain sailing image subtraction  $D(x, y, \sigma)$  can be very easily computed. For each octave the reiterated convolving of input image with  $G$  is shown in the Figure 5[11].





**Figure 5: DoG Pyramid Formation**

After computation of  $D(x, y, \sigma)$ , maxima and minima of DoG images are identified by making a pixel(X) comparison with the 26 neighboring pixels in current and two other adjacent scales. It is illustrated in Figure 6[11].



**Figure 6: Computation of Maxima and Minima of DoG**

### 2.3.2 Key-point Localization

In this step more accurate key-points are selected. For achieving this purpose Taylor series expansion of scale space is applied and those extrema with intensity value less than a preset threshold value are rejected. This series shifts the function  $D(x, y, \sigma)$  in a way so that sample point is positioned at the origin [11].

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \dots \dots \dots (5)$$

Now in order to determine the location of extrema we differentiate  $D(x)$  with respect to  $x$ ,

$$\hat{x} = - \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \dots \dots \dots (6)$$

Then we substitute the value of  $\hat{x}$  from (4) in (3) to discard the unstable extrema.

$$D(\hat{x}) = D + \frac{\partial D^T}{\partial x} \hat{x} \dots \dots \dots (7)$$

### 2.3.3 Orientation Assignment

According to the local image properties, each key-point is given an orientation. Firstly, the Gaussian smoothed image  $L$  is calculated by the function  $L(x, y, \sigma)$ . Secondly calculation of the magnitude of gradient denoted by  $m$  and orientation denoted by  $\theta$  is done by using the following two equations.

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \dots\dots\dots (8)$$

$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right) \dots\dots\dots (9)$$

Histogram of oriented gradient is used to calculate gradient direction of feature points. Dominant direction of the local gradients is represented by orientation histogram peaks.

### 2.3.4 Generation of Key-point Descriptor

The measurement of the local image gradient is taken at the selected scale in the area around every key-point. These image gradients are transformed to represent significant levels of local shape distortion. Key-point descriptors use a set of 16 histograms each having 8 elements which results in the feature vectors having 128 elements.

### 2.3.5 SIFT Algorithm

The SIFT algorithm from key-point localization to generation of key-point descriptor is show in Figure 7 in a simple pseudo code.

```

Parse options
Load image
Resample image to double size
For each octave
    Create Gaussian blur intervals
    Create difference-of-Gaussian intervals
    Compute edges for each interval
End for
Search each octave for stable extrema
Create key-points at dominant orientation of extrema
For each key-point
    Rotate sample grid to key-point orientation
    Sample region and create descriptor
End for
Optionally save pyramid images
Save out images
Save descriptors

```

**Figure 7: Pseudo Code of SIFT Algorithm**

#### **2.4 Key-point matching using g2NN (generalized 2 nearest neighbor)**

Each key-point extracted from SIFT belongs to a specific key-point descriptor vector. To get the set of matching key-points, each key-point is compared with the other key-points which belongs a descriptor vector. A generalized 2 nearest neighbor matching method is applied to get the matching key-points. In this method, Euclidean distance is computed for the finding the ratio of the two closest neighboring key-point. Then a threshold value (T) is approximately 0.6 is used to

compare the computed ratio of neighboring key-points [17]. In mathematical, if the similarity vector is denoted by D for a random key-point extracted from the SIFT algorithm;

We get,

$$D = \{d_1, d_2, d_3, \dots, d_{n-1}\} \dots \dots \dots (10)$$

Which is used as the sorted Euclidean distance for the descriptor vectors. To satisfy the matching criteria the following condition must be met,

$$\frac{d_1}{d_2} < T$$

Where,  $d_1$  and  $d_2$  denotes respectively the closest neighbor and second closest neighbor for a certain key-point. T denotes the threshold value that is in between 0 to 1. If the ratio of  $d_1/d_2$  is less than the predefined threshold value, then we can say that the key-points ( $d_1$  and  $d_2$ ) can be candidate for matching similar key-points. In case of random selection, the ratio may be greater than the threshold value. Since generalized version of 2NN is an iterative approach, it continuously performs the 2NN test until the ratio of  $d_1/d_2$  is 0.5. This g2NN helps to find multiple copy-move forgery in the image because of its iterative nature. To minimize the time complexity, a value of k is defined for stopping condition of 2NN test, where k is in between 1 to n when

The similarity vector i.e. sorted Euclidean distance vector  $D = \{d_1, d_2, \dots, d_{n-1}\}$  is then becomes as

$D = \{d_1, d_2, \dots, d_k\}$  that is used as minimized sorted Euclidean distance vector for finding the matching key-points. Then for later processing, the matched-key-points are considered and the other key-points is discarded [13]. But, there may have some identical points that contains intrinsically in the image. As a result, we can get some false positive matches that effects the performance and

result of the detection algorithm. Hence, a method is applied to remove the false positive rate for better detection accuracy of proposed model that is described in later part.

## 2.5 Agglomerative Hierarchical Clustering

Clustering is a process of grouping similar data in a group or cluster in such a way that the elements of the same cluster are more similar than compare to others clusters' elements. Hierarchical Clustering is one of most used clustering technique that is used in different fields of data analysis, computation and simulation. It forms a hierarchy to the clusters. Agglomerative hierarchical clustering follows a bottom-up clustering technique where each cluster belongs to another cluster and it has a sub cluster. In case of bottom level clusters, it has no sub clusters. In agglomerative hierarchical clustering every single data is considered as a single cluster. In next successive iterative processes, it merges (agglomerates) the pair of clusters based on the distance value between them.

For copy - move forgery detection each matched key-point is considered as a single cluster [13]. The spatial locations, in other words, the coordinate of matched key-points is used to find spatial distance among the clusters. Based on this distance the two closest clusters are then identified to merge them as a new single cluster. This iterative approach of computation continues until all the clusters are merged to form a final single cluster. This process is performed based on a specific linkage method and a threshold value which determiners when to stop for forming the final merged cluster. This type of agglomerative hierarchical clustering uses a dendrogram or cluster tree to finish the clustering process of the matched key-points. The linkage method ward is used to agglomerates the matched key-points by finding the coordinate distance among the points [24]. It uses the following formula to generate the distance vector;

$$d(r, s) = \sqrt{\frac{2n_r n_s}{(n_r + n_s)}} \|\bar{x}_r - \bar{x}_s\|_2 \dots \dots \dots (11)$$

Where,  $\|\cdot\|_2$  denotes Euclidean distance

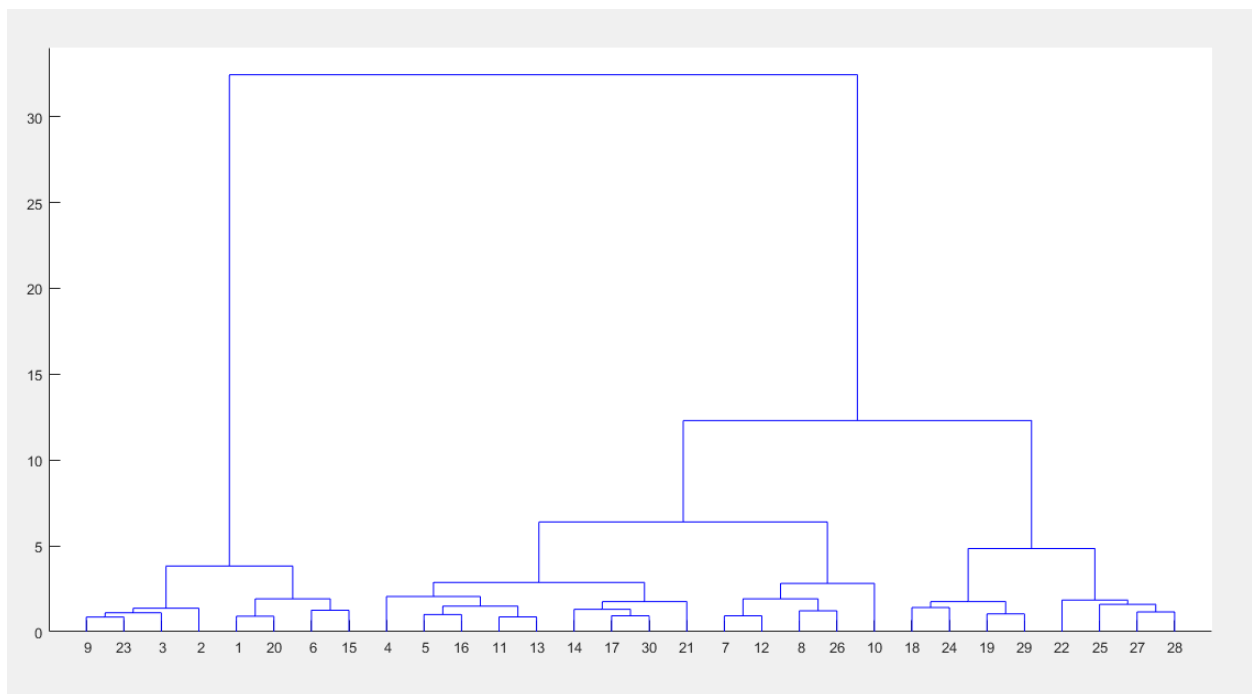
$\bar{x}_r$  and  $\bar{x}_s$  are the centroids of clusters  $r$  and  $s$

$n_r$  and  $n_s$  are the number of elements in clusters  $r$  and  $s$

The following steps forms Agglomerative Hierarchical Clustering i.e. algorithm.

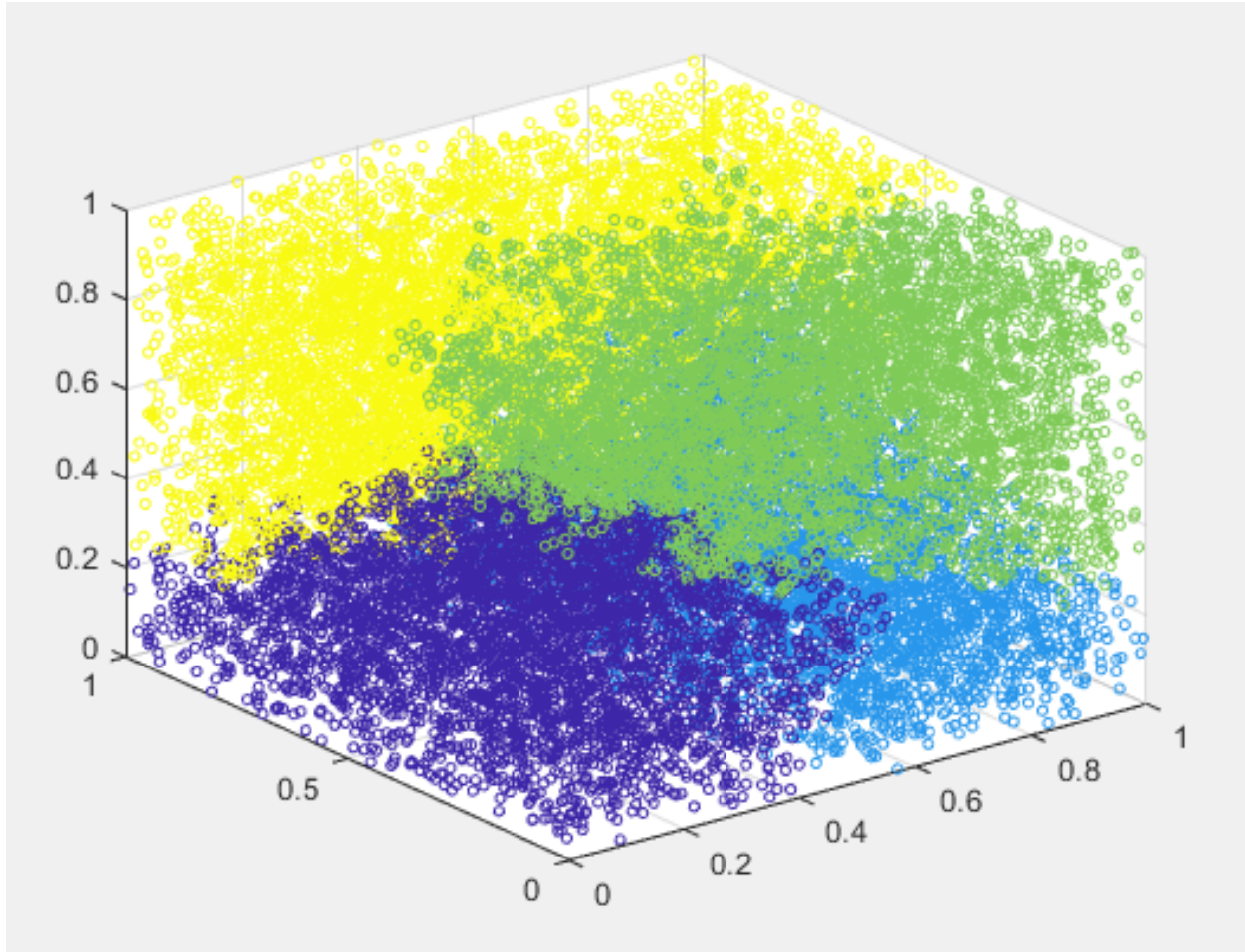
1. Each key-points is considered as a single cluster
2. Compute the distance among clusters pair wisely.
3. Make the distance matrix based on the distance value from the previous step.
4. Identify the pair having shortest distance.
5. Discard the pair having shortest distance from the distance matrix to form a new cluster by merging them.
6. Compute the all pair wise distances from the newly created cluster to the all other clusters for updating the distance matrix.
7. Continues until having the distance matrix with a single cluster or key-point.

Figure 8 shows the cluster tree of agglomerative hierarchical clustering generated by MATLAB.



**Figure 8: Dendrogram of Agglomerative Hierarchical Clustering**

Figure 9 shows the hierarchical clustering using linkage method ward generated by MATLAB.



**Figure 9: Hierarchical Clustering Using Linkage Method Ward (Represents a Clustering Overview of Four Colors).**

## 2.6 False Positive Removal(RANSAC)

In this step, we use Random Sample Consensus algorithm (RANSAC) [16] to remove false positive matches. RANSAC is used as the mismatched points or outliers can (outliers) hamper the estimated homography. In RANSAC algorithm, a set of matched points are randomly selected and then the homography is estimated. After that other remaining matched points are transformed and then compared in terms of distance with respect to their respective matches. A threshold value is set. If this distance is under the threshold value it is marked as inliers and if it is above the threshold is catalogued as outliers. After a predefined number of iterations, the



estimated transformation which is associated with the higher number of inliers is chosen [13]. In our experimental tests, has been set to 1000 and the threshold to 0.05; this is due to the fact that we used a standard method of normalization of the data for homography estimation. The points are translated so that their centroid is at the origin and then they are scaled so that the average distance from the origin is equal to. This transformation is applied to both of the two areas and independently.

### **2.6.1 RANSAC Algorithm**

The RANSAC algorithm can be described as follows:

1. Randomly selecting a subset of the data set
2. Fitting a model to the selected subset
3. Determining the number of outliers
4. Repeating steps 1-3 for a prescribed number of iterations

### **2.7 Related Works**

Since copy-move forgery detection is comparatively more difficult, most of researchers in this field have tried to find out a stable and robust solution for detecting manipulation in image mostly either in key point or block based approach. In block based approach image is divided into blocks and feature matching among blocks provides tampering confirmation of image where as in key point based approach key-points are extracted from the image using different key-points extracting algorithms, then matching among the key-points feature vector provides tampering information about the image.

In [6,13,1] authors propose a key-point based copy-move forgery detection method. In [6] propose an efficient and robust method for detection copy-move image forgery detection. Three ways of detection approach for copy-move forgery is described here. In first approach, SIFT based detection approach is mentioned that is comprised of extraction of sift features, computing Euclidean distance between the pair of sift key points, determining the best matches key points on the basis of threshold prioritizing minimum distance pair of key points, forming cluster among the best matches key points on the basis of threshold value of Euclidian distance, cluster size .Finally the matching between the cluster is performed and forged region is shown in the image if at least two key points are matched among the clusters.

In the second approach of detecting copy-move forgery, SURF based detection algorithm is proposed that is faster than the SIFT algorithm. Here, a set of steps is followed to get the desired forgery detection of copy-move forgery in digital image. Same for the SIFT approach, here in the SURF based detection approach, first the key points of image are extracted that is later used for finding Euclidian distance between the pairs of SURF key points. Then, a threshold value based minimum distance pair of key points are chosen for best match candidates which is used in later steps to form cluster providing a threshold value of Euclidian distance and cluster size. At the final step, matching among the cluster is done to find the forged region if any key points are found between two clusters.

In the third approach mentioned in the paper is HOG based approach to find copy-move detection forgery. In this approach 1-D DWT is used to the input image to get the most information containing approximate image which is used in the next step to divide in overlapping blocks. For Each block then HOG features are extracted to get the lexicographically sorted feature vector. Then block matching is done to get the matched pair to form clusters. Then matching between cluster is performed to get the forged region if any matches found between the two clusters.

In [7], the authors proposed a method to detect copy-move forgery using Discrete Cosine Transform (DCT) of the image blocks. At first the blocks are sorted lexicographically and then the adjacent identical pair of blocks are considered copy-move region. The problem with this method is that it cannot detect small duplicate regions.

In [18], the authors propose an improved copy- move forgery detection based on DCT. In this approach, image is divided into overlapping blocks and then features vectors found by each block is sorted lexicographically which finally identifies the duplication region. Compare to others method based on DCT, its good side is that it can detect forged region if the image is even distorted by blurring, Gaussian noise and JPEG compression.

In [14], authors used region duplication detection algorithm with an improved DCT which reduced the computational complexity. The difference of this method with others is that the quantized block is characterized by a circle block. The feature vectors were calculated after the circle block is divided into a fixed number of parts. Then the feature vectors were lexicographically sorted and Euclidean distance between adjacent pairs is calculated. The benefit

of using this method was it can identify multiple region duplications. However, it is not robust to geometrical operation and also fails to work with poor image quality.

A method using Principal Component Analysis (PCA) is described in [8]. Here, at first the image is divided into several blocks. Then their feature vectors are calculated and sorted lexicographically. PCA is used to represent the dissimilar blocks. The benefit of this model is that it can reduce time complexity, works good for large images. But its accuracy decreases for small block sizes and low JPEG qualities.

In [9] author proposed a block based copy-move image forgery detection model based on DWT to compress the input image. Afterwards the compressed image is divided into some overlapping blocks. The blocks are then lexicographically sorted to identify the duplicated blocks.

In [10] author proposed a copy-move image forgery detection technique based on SWT-SVD. SWT is shift invariant and noise invariant which is used to decompose the input image and helps in finding similarities between blocks of an image. The blocks are represented using features extracted by SVD. This model detects image forgery for blurred image successfully.

In [19], the authors propose segmentation based copy-move forgery detection. In this method, first the input image is semantically segmented into different independent patches. Later for each segments key-points are extracted. Affine transform matrix and an Expectation-Maximization-based algorithm are used to confirm detection of copy-move regions.

In [20], the authors propose a robust and efficient method for detection of copy-move image forgery. Here, authors propose Fourier Melon Transform for extracting the features from the image blocks. This method is robust to scaling, rotation, lossy JPEG compression, noise and blurring. For better performance and time complexity reduction, authors use lexicographic sorting and counting bloom filter to find copy-move region detection.

## Chapter 03

### Proposed Model

#### 3.1 Workflow of Proposed Model

Our proposed model is mainly based on SWT and SIFT algorithm which can detect tampered region in a copy-move forged image. Firstly, we convert the input image from RGB to Gray Scale. 2-D SWT is applied on the dataset and approximate component of the decomposed image is passed as input parameter in SIFT algorithm to extract the descriptor vectors. Then matching operation is performed on the descriptor vectors to detect copy-move tampering. Finally, RANSAC algorithm is applied to remove outliers that help to reduce the false positive rate. The block diagram of our proposed model is shown in Figure 10.

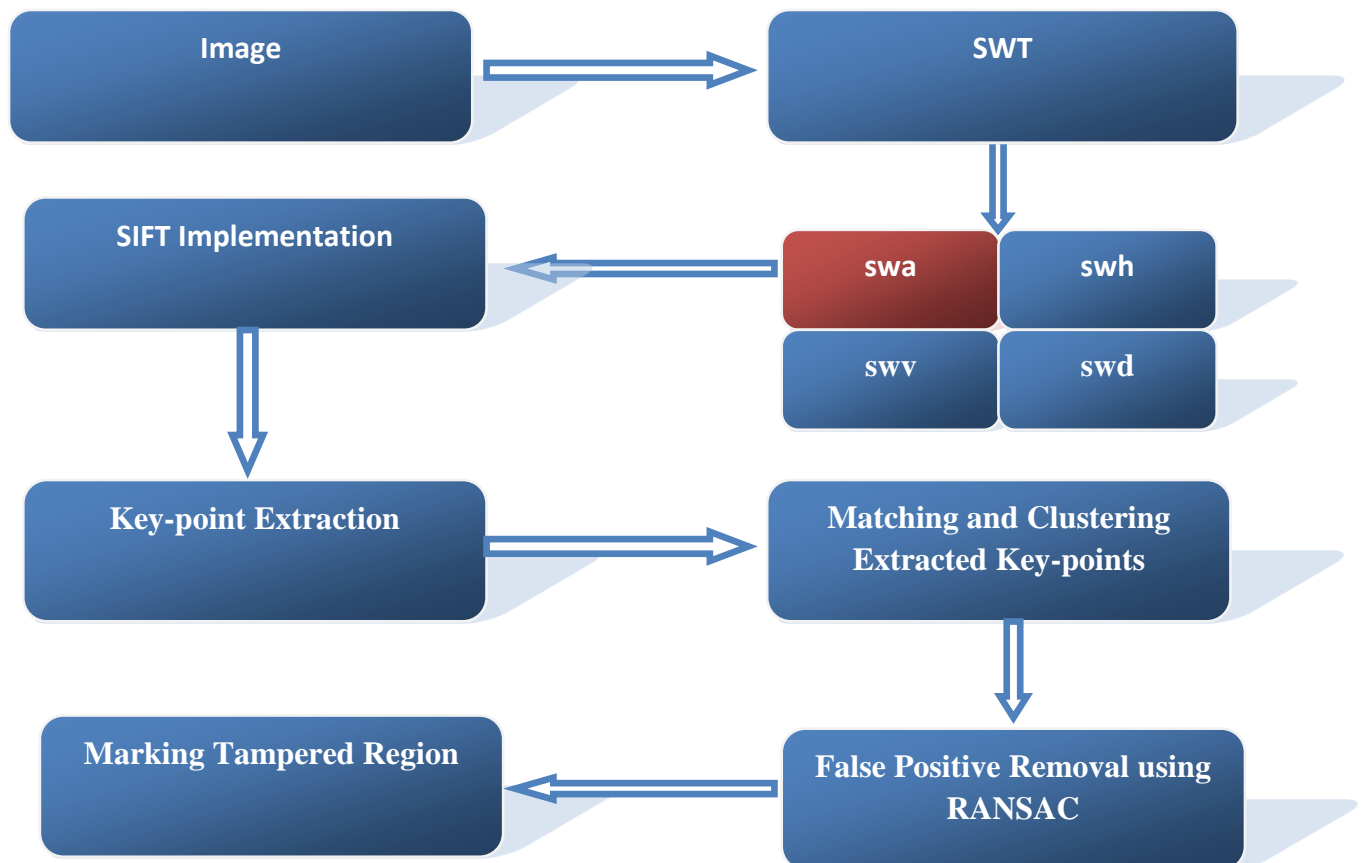


Figure 10: Workflow of Proposed Model

### 3.1.1 Pre-processing

The pre-processing step of the model comprises of two sub-steps. Firstly, the input image is converted to grayscale if it is a RGB image. The reason behind converting it into grayscale is to reduce complexity by converting a 3D pixel value (R, G, B) to a 1D value. Besides the color information does not contribute in identifying key-point features. The following formula is used to convert the RGB values to grayscale value.

$$\text{IMG}=0.2989R + 0.5870G + 0.1140B.....(12)$$

Figure 11 shows the conversion of RGB image to gray scale.



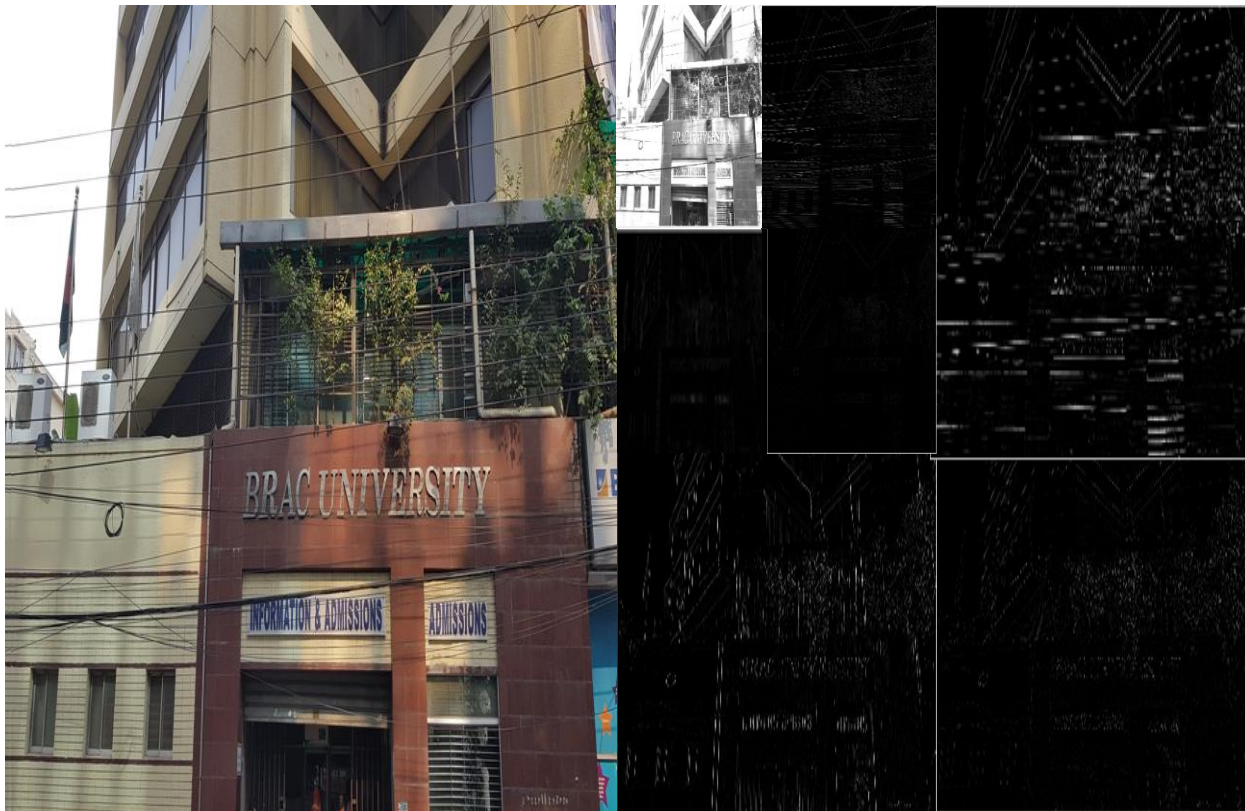
(a)



(b)

**Figure 11: Conversion of RGB to Gray Scale**

Secondly, SWT is used to obtain four sub bands such as swa, swh, swv and swd which represents respectively approximate image, horizontal image, vertical image, diagonal image. The approximate sub band is later passed as input parameter in SIFT algorithm to extract features. The primary reason for choosing SWT over DCT or DWT is that it is shift invariant, translation invariant and efficient at finding similarities and dissimilarities despite of having noise or blurring in the image. Figure 12 shows the SWT decomposition output.



**Figure 12: 2-D SWT Decomposition of Input Image**

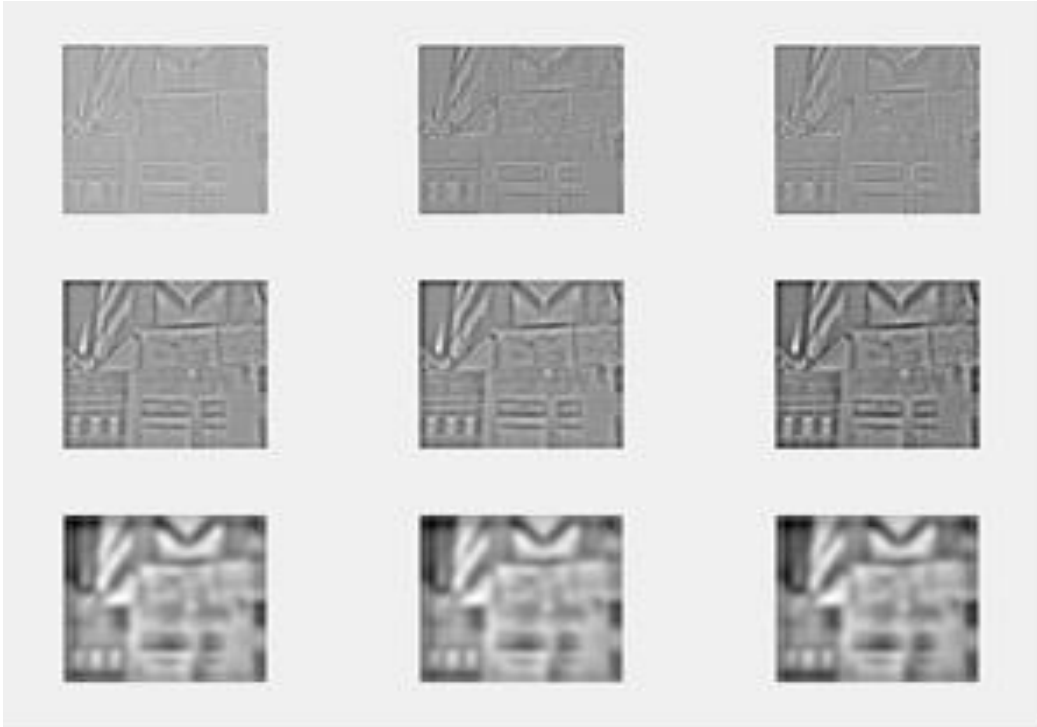
### **3.1.2 SIFT Feature Extraction**

SIFT is one of the best feature extracting algorithm proposed by David Lowe [11]. It is invariant to image rotation, geometrical transformation, intensity and change of viewpoint in matching features. The algorithm is divided into 4 main steps. They are as follows:

### 3.1.2.1 Scale Space Extrema Detection

In this step Gaussian of Difference(DoG) is used to find possible points of interest which are invariant to orientation and scaling. To make the detection of key-points more reliable, efficient and stable DoG Function  $D(x, y, \sigma)$  is required. It is computed by convolving the difference of two nearby scales separated by a constant scaling factor  $k$  with the input image as shown in the Figure 13.

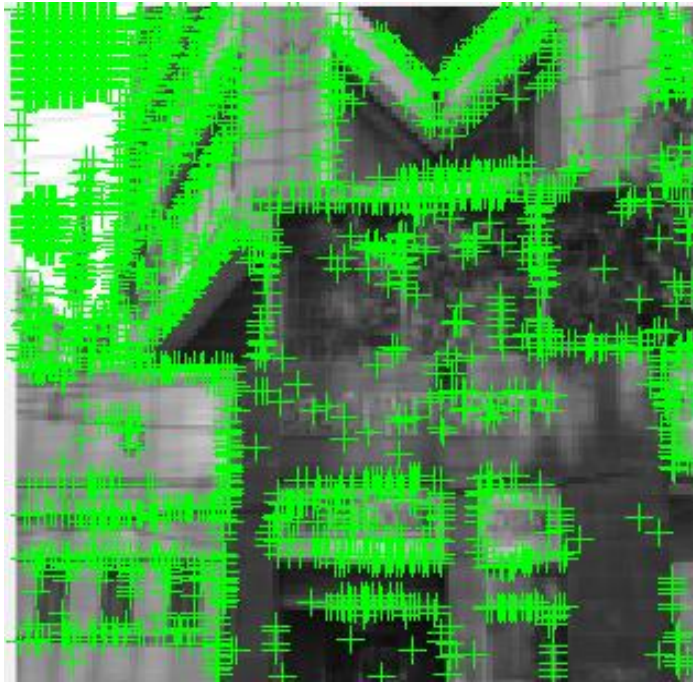
$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \dots \dots \dots (13) \end{aligned}$$



**Figure 13: DoG Pyramid Formation of Approximate Image**

The key-points that are initially identified on the approximate component of decomposed image are shown in the Figure 14.



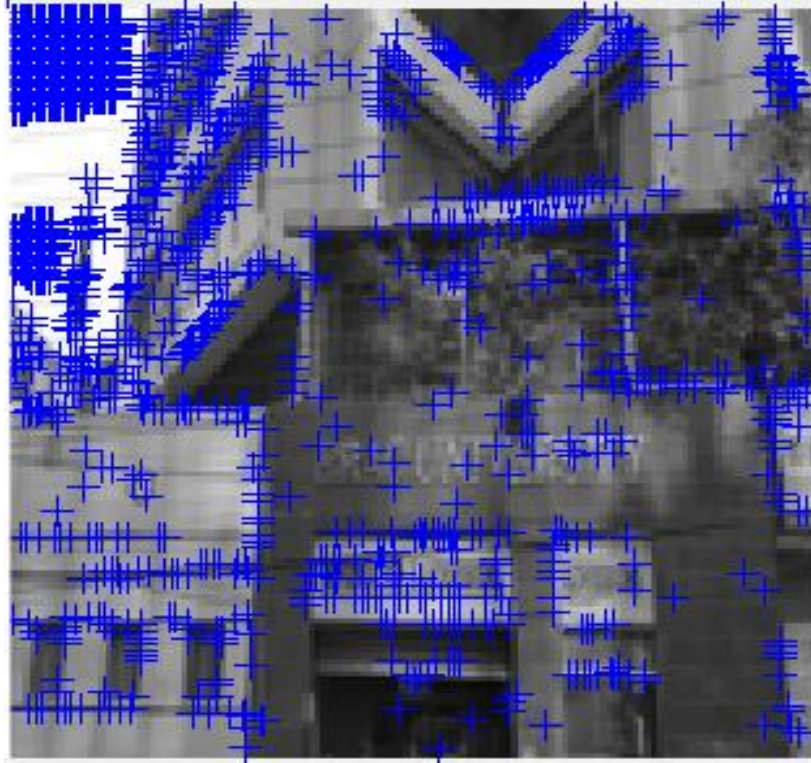


**Figure 14: Initial Location of Key-points of Different Views Component**

### **3.1.2.2 Key-point Localization**

In this step more accurate key-points are selected. For achieving this purpose Taylor series expansion of scale space is applied and those extrema with intensity value less than a pre-defined threshold value are rejected. The accurately selected key-points on the approximate image after discarding the ones having poor contrast are shown in Figure 15.





**Figure 15: Accurately Selected Key-points**

### **3.1.2.3 Assignment of Orientation**

According to the local image properties, each key-point is given an orientation. Histogram of oriented gradient is used to calculate gradient direction of feature points. Dominant direction of the local gradients is represented by orientation histogram peaks.

### **3.1.2.4 Generation of Key-point Descriptors**

The measurement of the local image gradient is taken at the selected scale in the area around every key-point. Key-point descriptors use a set of 16 histograms each having 8 elements which results in the feature vectors having 128 elements.

### **3.1.3 Clustering**

Agglomerative hierarchical clustering is used to group the extracted SIFT key-point descriptors. Linkage method ward is used to complete the clustering process.

### **3.1.4 Key-point Matching**

In order to perform matching among any two clusters, comparison is done for each point in one cluster's descriptor vector with the descriptor vector of the other cluster. Efficiency is obtained by calculating the angle between the descriptor vectors of the two clusters. Only those key-points are accepted as matched key-points when the ratio of minimum angles between the descriptors vector are less the threshold value 0.5[13].

### **3.1.5 False Matches Removal**

In this step we use a sorting algorithm named RANSAC to remove false positive matches [16]. In this algorithm four arbitrary points from the matched points are chosen to estimates the homography  $H$ . Other remaining matched points are transformed and then compared in terms of distance with respect to their respective matches [23]. A threshold value is set and points with distance less than the threshold value are rejected.

## **3.2 Data Collection**

Since our proposed model is designed to detect multifarious types of copy-move forgery such as copy-move without geometric translation, with geometric transformation including rotation and scaling, we need a dataset of images which have such type of forgeries. In order to meet the need, we have chosen the MICC-F220 dataset [12]. Moreover, since our proposed model can detect forgeries in images having blur or noise so we have created such type of dataset to test the performance of our proposed model.

### **3.3 Tools Used**

We have used MATLAB 2017a software with 8GB Ram and core i5 processor to develop and record the performance of your implemented model.

## Chapter 04

### Experimental Setup and Result Analysis

We applied our model over the standard dataset MICC-F220[12] as well as some of our own images. We have used MATLAB 2017a software with 8GB Ram and core i5 processor. Firstly, 2-D SWT is applied on the dataset and approximate component of the decomposed image is passed as input parameter in SIFT algorithm to extract the descriptor vectors. Finally, matching operation is performed on the descriptor vectors to detect copy-move tampering.

We have calculated result of proposed model by finding True Positive(TP), False Positive(FP), True Negative(TN), False Negative(FN) by running our model on images from dataset MICC-F220[12].

TP: Number of forged images detected as forged

FP: Number of authentic image identified as forged

TN: Number of authentic image identified as authentic

FN: Number of forged image identified as authentic

$$\text{Accuracy} = (TP+TN)/(TN+FP+TP+FN)$$

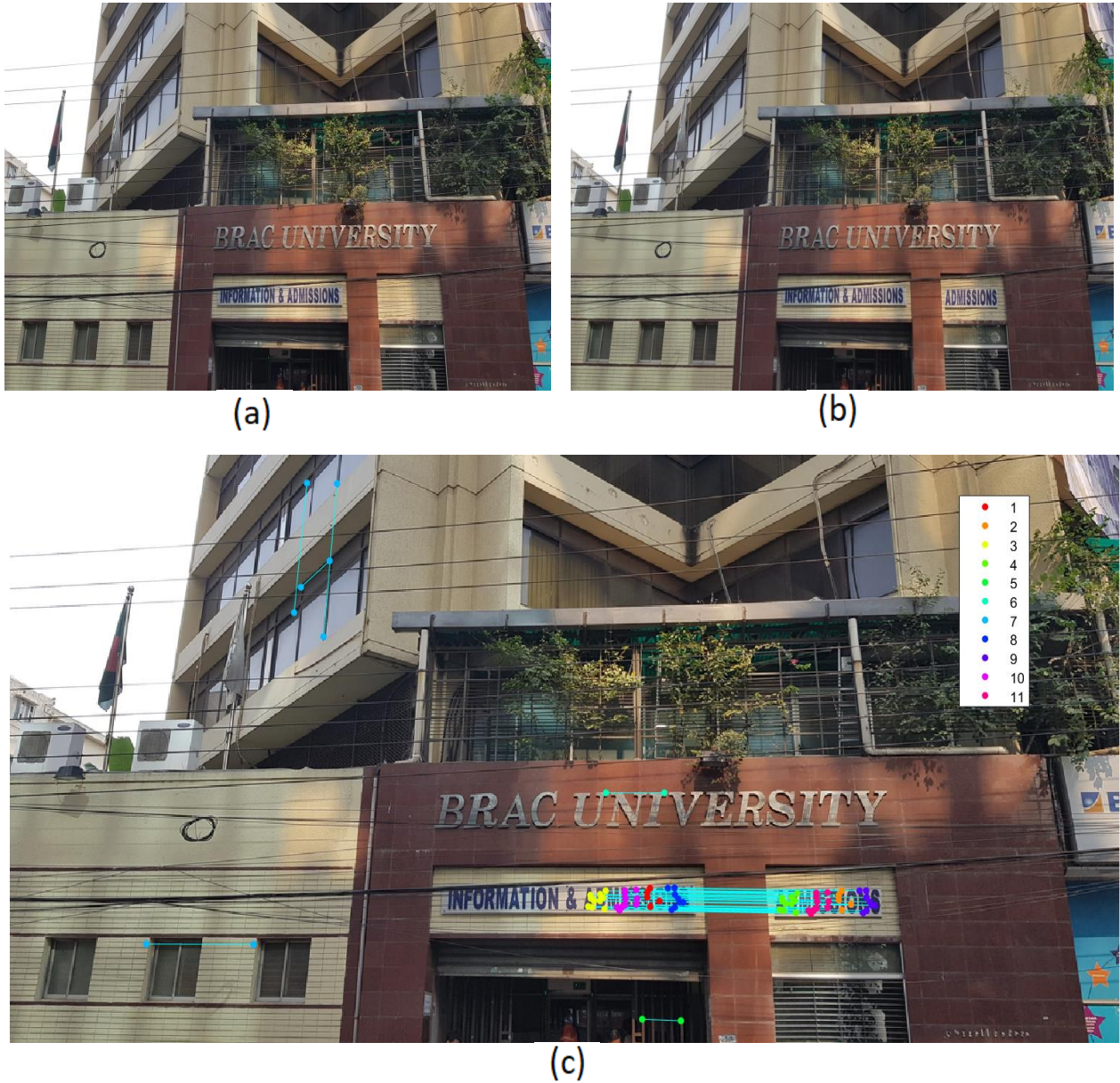
$$\text{Sensitivity} = TP/ (TP + FN)$$

$$\text{Specificity} = TN/ (TN + FP)$$

$$TPR = \frac{\# \text{ images detected as forged being forged}}{\# \text{ forged image}}$$

$$FPR = \frac{\# \text{ images detected as forged being original}}{\# \text{ original image}}$$

Figure 16 shows the original and forged image side by side and the bottom image shows the output after detection of forged region.



**Figure 16: (a) Original Image, (b) Forged Image, (c) Forgery Detection**

We have applied our proposed model on a set of 50 original images and a set of 50 forged images and we have recorded the number of key-points found, time required for finding those key-points, number of matched feature, time required to find the matches and decision on whether the image is forged or not. The data for original set of 20 images is shown in Table 1 and data for forged set of 20 images is shown in Table 2.

**Table 1: Recorded Data from a Set of 20 Non-Tampered Images**

Serial	No of key-points	Required time to find key-points in sec(t1)	No of matched key-points	Required time to match key-points in sec(t2)	Total time in sec (t1+t2)	Decision
1	3820	2.26	57	3.50	5.76	FP
2	4325	2.46	16	3.89	6.49	TN
3	2879	1.98	6	2.12	4.1	TN
4	3687	2.23	15	3.09	5.32	TN
5	3256	2.04	16	2.61	4.65	TN
6	3741	2.43	7	3.18	5.61	TN
7	2936	1.94	12	2.10	4.04	TN
8	2377	1.83	2	1.62	3.45	TN
9	5351	2.77	12	3.15	5.92	TN
10	3695	2.13	1	3.13	5.26	TN
11	3255	2.56	13	3.15	5.71	TN
12	3566	2.25	8	3.20	5.45	TN
13	3645	1.89	15	2.15	4.04	TN
14	2985	2.49	12	3.16	5.65	TN
15	2859	2.98	18	2.59	5.57	TN
16	3655	1.97	8	2.23	4.20	TN
17	4025	2.15	16	2.71	4.86	TN
18	3009	1.79	10	3.15	4.94	TN
19	4103	2.49	15	3.50	5.99	TN
20	3579	2.19	14	3.01	5.20	TN

The total required time from finding key-points to detection of forgery from Table 1 is 102.21 seconds

**Table 2: Recorded Data from A Set 20 Tampered Images**

Serial	No of key-points	Required time to find key-points in sec(t1)	No of matched key-points	Required time to match key-points in sec(t2)	Total time in sec (t1+t2)	Decision
1	5481	2.77	54	4.18	6.95	TP
2	5538	2.79	55	4.33	7.12	TP
3	660	1.06	22	0.50	1.56	TP
4	696	1.06	25	0.48	1.54	TP
5	6338	3.07	27	4.59	7.66	TP
6	6360	3.19	23	4.79	7.98	TP
7	707	1.34	45	0.63	1.97	TP
8	3136	2.09	62	2.37	4.46	TP
9	1026	1.37	26	0.92	2.29	TP
10	4452	2.54	25	4.25	6.79	TP
11	5256	2.68	49	3.75	6.01	TP
12	589	0.98	18	0.45	1.43	FN
13	6189	2.45	23	4.05	6.50	TP
14	690	1.28	39	0.59	1.87	TP
15	1009	1.29	22	0.89	2.18	TP
16	5248	2.60	50	3.89	6.49	TP
17	628	1.02	13	0.45	1.47	TP
18	6225	2.80	21	3.87	6.67	TP
19	3089	2.03	59	2.49	4.52	TP
20	4129	2.49	23	3.10	5.59	TP

The total required time from finding key-points to detection of forgery from Table 2 is 91.05 seconds

Based on the data achieved from applying our model on original set of 50 images and forged set of 50 images we have recorded the number of true positive, true negative, false positive and false negative. Table 3 shows the data.

**Table 3: Outcome of Proposed Method**

No of Original Images	No of Forged Images	TP	TN	FP	FN
50	50	45	48	2	5

Based on the result from Table 3 i.e. the number of TP, TN, FP and FN we calculate sensitivity, specificity, accuracy, false positive rate(FPR) and true positive rate(TPR). We show the performance of our model in Table 4.

**Table 4: Performance of Proposed Method**

Sensitivity	Specificity	Accuracy	FPR (%)	TPR (%)
90%	96%	93%	4%	90%

Since our proposed model is geometric transformation invariant so we selected an image from MICC-F220 [12] which has various combination of transformation. We apply our proposed model on these tampered images and the outputs are shown in Figure 17.

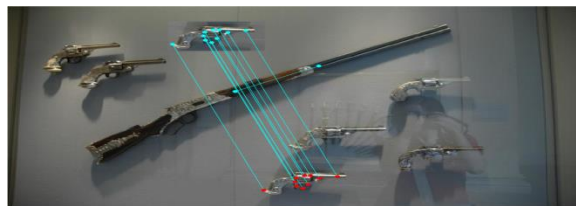




(a)



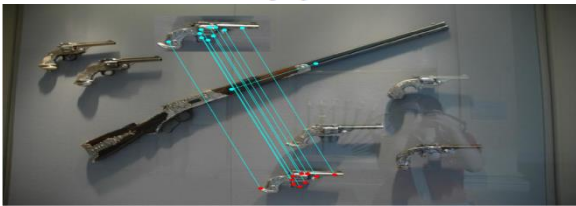
(b)



(c)



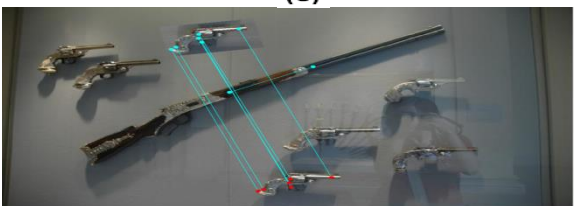
(d)



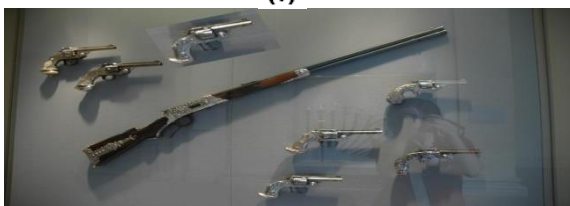
(e)



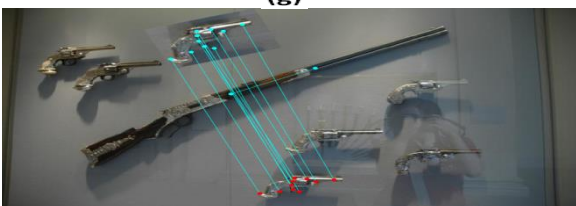
(f)



(g)



(h)







(i)

**Figure 17: (a) Original, (b) (d) (f) and (h) are Tampering of Images and (c), (e), (g) and (i) are Their Corresponding Detection of Forgery.**

In Figure 17, we have detected forgery for different transformations including rotation and scaling. We have showed several transformed images with their corresponding key-points, no of matched key-points and computational time in Table 5.

**Table 5: Performance Analysis Based on Different Attacks on the Image**

<b>Types of attack</b>	<b>Numbers of key-points found</b>	<b>Number of matches detected</b>	<b>Computational Time(s)</b>	<b>Forged Image of Different Geometric Transformation</b>
<b>Without Rotation or Scaling</b>	397	22	0.871	
<b>Scaling</b>	376	20	0.716	
<b>Rotation</b>	259	13	0.549	
<b>Rotation and Scaling</b>	287	21	0.763	

The total time of forgery detection from Table 1 and Table 2 are used to calculate the average time.

$$\text{Average Time, } T_A = \frac{102.21+91.05}{40} \text{ seconds}$$

$$=4.83 \text{ seconds.....(14)}$$

Comparison among other existing model and our model based on true positive rate, false positive rate and average of total execution time is shown in Table 6.

**Table 6: Comparative Result with Existing Models**

<b>Method</b>	<b>FPR (%)</b>	<b>TRP (%)</b>	<b>Time(s)</b>
Popescu and Farid [8]	86	87	70.97
Fridrich et al [7]	84	89	294.69
Our Method	4	90	4.83

We compared our model with several existing models and found better result. Our model works perfectly in terms of rotation, scaling and noisy image. Robustness of our model is compared with other existing methods which is shown in Table 7.

**Table 7: Comparison of Robustness with Existing Methods**

<b>Method</b>	<b>Types of Attack</b>				
	<b>Without Rotation and Scaling</b>	<b>Scaling</b>	<b>Rotation</b>	<b>With Rotation and Scaling</b>	<b>Noisy Image</b>
<b>DCT</b>	yes	no	no	no	no
<b>DWT</b>	yes	no	no	no	no
<b>SIFT</b>	yes	yes	yes	yes	no
<b>Our Proposed Model</b>	yes	yes	yes	yes	yes

## Chapter 05

### Conclusion and Future Work

#### 5.1 Conclusion

In this paper, Stationary Wavelet Transform(SWT) has been used with SIFT features to detect copy-move forgery in a digital image in a faster way. SWT is shift invariant, blur and noise invariant. With the simulation performed on original and copied images, it shows that SWT and SIFT perform better in terms of time complexity and accuracy. The combination of SWT and SIFT also shows better performance than of DWT and SIFT. We found overall accuracy of 94% with a database of 100 images (50 authentic and 50 forged images). Robustness of the algorithm was checked with the MICC-F220 database. To check the robustness, copied part has been rotated, scaled and then pasted in the image. The accuracy rate has been found higher than most of the existing algorithms. Computational time was less hence it can be said that the computational complexity was reduced. Our proposed model doesn't require any training and can efficiently detect the forgery. The main objective behind this approach is to obtain a unique and more robust technique to detect copy-move image forgery. From the results obtained we can conclude that our proposed algorithm has better precision rate than other existing copy-move forgery detection algorithms.

#### 5.2 Future Work

In our next work, we will make our model more efficient. We will try to improve our algorithm so that time complexity can be reduced more.

Currently, our algorithm detects forgery if some part is copied and pasted within the same image. We will extend our model to detect image forgery if some part is copied from other images also.

We will try to improve the detection technique reducing the false positive combining with some other techniques. Furthermore, we want to integrate other methods of copy-move detection in our model so that we can achieve invariance to affine transformation, illumination and 3D view point.

Also, we have future plans to build desktop and mobile version of our software for mass use.

## Reference

- [1] Huang, Hailing, Weiqiang Guo, and Yu Zhang, "Detection of copy-move forgery in digital images using SIFT algorithm", In Proceedings of Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA-2008), vol. 2, pp. 272-276.
- [2] Birajdar, Gajanan K., and Vijay H. Mankar. "Digital image forgery detection using passive techniques: A survey." *Digital Investigation* 10.3 (2013): 226-245.
- [3] Saiqa Khan, Arun Kulkarni, "Reduced Time Complexity for Detection of Copy-move Forgery Using Discrete Wavelet Transform" *International Journal of Computer Applications* (0975 – 8887) Volume 6– No.7, September 2010, pp31-36.
- [4] Muhammad, Najah, Muhammad Hussain, Ghulam Muhammad, and George Bebis . "Copy-move forgery detection using dyadic wavelet transform.  
" In Proceedings of IEEE Eighth International Conference on Computer Graphics, Imaging and Visualization (CGIV-2011), pp. 103-108, 2011.
- [5] Jing, Li, and Chao Shao. "Image Copy-move Forgery Detecting Based on Local Invariant Feature." *Journal of Multimedia* vol. 7, no. I, pp.90-97,2012.
- [6] S. Prasad and B. Ramkumar, "Passive copy-move forgery detection using SIFT, HOG and SURF features," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2016, pp. 706-710.
- [7] J. Fridrich, D. Soukal, and J. Lukas, "Detection of copy-move forgery in digital images," Proceedings of the Digital Forensic Research Workshop. Cleveland OH, USA, 2003
- [8] Popescu, A. and Farid, H. (2004), Exposing Digital Forgeries By Detecting Duplicated Image Regions, Tech. Rep. TR2004-515, Dartmouth College, Computer Science, Hanover, Conn, USA
- [9] YADAV, P., RATHORE, Y., YADU, A.. DWT Based Copy-move Image Forgery Detection. *International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE)*, North America, 1, Jul. 2012
- [10] R. Dixit, R. Naskar and S. Mishra, "Blur-invariant copy-move forgery detection technique with improved detection accuracy utilising SWT-SVD," in *IET Image Processing*, vol. 11, no. 5, pp. 301-309, 5 2017.

- [11] Lowe, David G., "Distinctive image features form scale-invariant key-points", *International journal of computer vision*, vol. 60, no. 2, pp. 91-110, 2004.
- [12] Amerini, Irene & Ballan, Lamberto & Caldelli, Roberto & Del Bimbo, A & Serra, Giuseppe. (2013). MICC-F220.
- [13] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo and G. Serra, "A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery," in *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099-1110, Sept. 2011.
- [14] Cao, Y. Gao, T. Fan, L. Yang, Q. (2012), A Robust Detection Algorithm For Copy-move Forgery in Digital Images, *Forensic Science International*, vol. 214, No. 1–3, pp. 33–43.
- [15] R.Gupta, D. Awasthi, Wave-packet image fusion technique based on Genetic Algorithm, *IEEE 5th International Conference on the Next Generation Information Technology Summit (Confluence)*, 2014, 280-285.
- [16] M.x. and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [17] P. P. Panzade, C. S. Prakash and S. Maheshkar, "Copy-move forgery detection by using HSV preprocessing and key-point extraction," *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Waknaghat, 2016, pp. 264-269.
- [18] Yanping Huang, Wei Lu, Wei Sun, Dongyang Long, Improved DCT-based detection of copy-move forgery in images, In *Forensic Science International*, Volume 206, Issues 1–3, 2011, Pages 178-184.
- [19] J. Li, X. Li, B. Yang and X. Sun, "Segmentation-Based Image Copy-move Forgery Detection Scheme," in *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507-518, March 2015.
- [20] S. Bayram, H. Taha Sencar and N. Memon, "An efficient and robust method for detecting copy-move forgery," *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, 2009, pp. 1053-1056.

- [21] B, Shwetha & Sathyanarayana, S. (2016). Digital image forgery detection techniques: a survey. ACCENTS Transactions on Information Security. 2. 22-31.
- [22] Dixit, Anuja & K. Gupta, R. (2016). Copy-move Image Forgery Detection using Frequency-based Techniques: A Review. International Journal of Signal Processing, Image Processing and Pattern Recognition. 9. 71-88.
- [23] A. Shahroudnejad and M. Rahmati, "Copy-move forgery detection in digital images using affine-SIFT," 2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS), Tehran, 2016, pp. 1-5.
- [24] G., S. and V., S. (2016). Copy-move Detection of Image Forgery by using DWT and SIFT Methodologies. International Journal of Computer Applications, 148(7), pp.37-41.
- [25] Lowe, D. G. (1999, September). Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision (Vol. 2, pp. 1150-1157).