Forgery Detection

In

Spliced Images

Submitted by

Bhashwitha Kolluri – 11526264

CONTENTS

ABSTRACT

Digital Image Forensics is a of study that evaluates the originality of images by retrieving information or data from their early history. Two major issues must be addressed: identifying the imaging device that acquired the image and detecting forgery traces. The identification of the source of the image requires the data base of the information regarding the camera such as its model, lens type etc., which might not be available. In this research I studied the latter problem i.e., Detecting the Forgery in Spliced Images. In this, an image is taken and is put under analysis for traces of any forgery. Various earlier methods that have been introduced uses different methods for detecting the forgery. In this research I learned about using Noise Discrepancies in for representing Forgery Detection in spliced image at different Multiple Scales. The test image at multiple scales is segmented into super-pixels. At each seperate scale, the Noise Level Function calculates the relation between noise level and brightness of each segment. Here I explained about an Outlier Map, which consists of regions regarded as spliced, which when becomes stable we output the regions in it as spliced.

1.INTRODUCTION

1.1 Importance

Images and videos have emerged as the main information carriers in the digital age. Visual media are increasingly being utilized to carry information, even intelligent information, because to their expressive power and ease with which they may be acquired, transferred, and stored. As a result, pictures and videos are increasingly being used as proof in court cases as well as daily discussions. In TV news, even the simplest of images is frequently used as evidence of the news' veracity. In a court of law, video surveillance footage might be utilized as crucial evidence regarding probation.

The availability of digital visual media has unquestionable benefits but also a serious drawback. Specialists in image processing may easily access and change an image's content and, consequently, its meaning, without leaving any obvious evidence. Furthermore, the practice of manipulating and fabricating visual representations is no longer exclusive to experts due to the spread of inexpensive, user-friendly editing tools. As a result, malicious image alteration is now more common than ever. Along with digital watermarking, Digital Picture Forensics, a part of multimedia security, aims to identify and reveal fraudulent image manipulation.

1.2 Forgery Incident

For his efforts to the international aid organization Médecins Sans Frontières, Malaysian lawmaker Jeffrey Wong Su En declared in July 2010 that Queen Elizabeth II had knighted him. Along with his comment, which was extensively reported in local media, came a picture of him getting the award from the Queen of England (Fig.1.1a). When asked about the honor, the British High Commission in Kuala Lumpur claimed that Mr. Wong's name wasn't on the official list of recipients and that the photo had been taken improperly, which was against the protocol for knighthood ceremonies. Eventually, it was revealed that the picture was a mash-up of Mr. Wong's face and a legitimate ceremonial shot (Fig.1.1b), made to increase his notoriety.

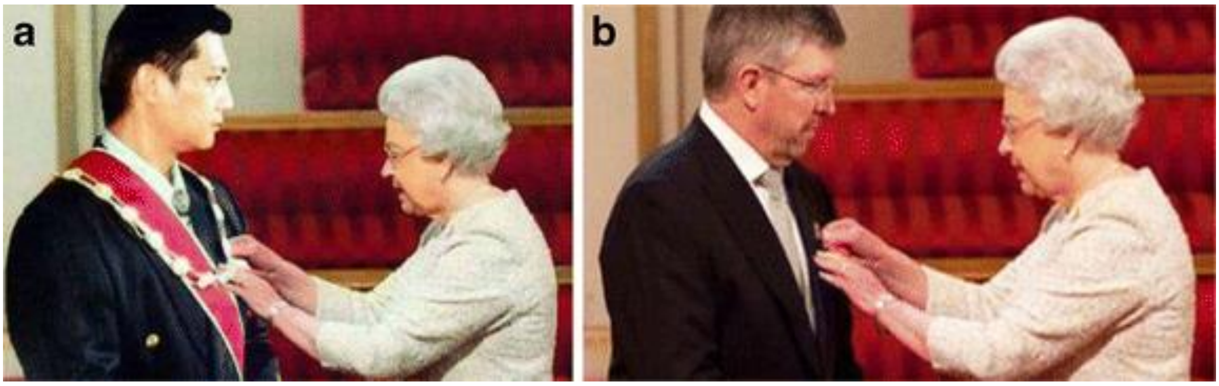


Figure 1.1: Spliced Image

1.3 Digital Image Forensics

Providing tools to aid in blind investigations is the aim of digital image forensics (DIF). This new area of research draws inspiration from existing multimedia security-related study subjects (such watermarking and steganography) and uses image processing and analysis techniques to obtain information about an image's past. Two major study areas come into focus while looking at Digital Image Forensics. The first category includes methods that use ballistic analysis to pinpoint the camera that took the picture or, at the very least, the cameras that missed it. These methods will all be included under the heading "image source device identification techniques." The second group of methods, on the other hand, looks for signs of semantic forgeries by analyzing irregularities in natural picture statistics.

1.4 Methodology

In this paper we will discuss about Detecting the Forgery in Spliced Images. An image is taken and is put under analysis for traces of any forgery. Various earlier methods that have been introduced uses different methods for detecting the forgery. Let's talk about detecting image splicing forgeries utilizing noise discrepancies across many scales. Super-pixels of various scales are used to segment the test image. In each individual scale, a Discrete Cosine Transformation (DCT) value is calculated for every pixel. By comparing these DCT values with the surrounding pixel DCTs we can cluster the pixels and categorize them as spliced and non-spliced regions. We consider an Outlier Map, which consists of regions regarded as spliced, which when becomes stable we output the regions in it as spliced. In this way we can identify the spliced region in the given image.

Two different methods are proposed and investigated for copy-move image forgery detection. The first method uses statistical features and dendrogram clustering technique, while the second method extracts textual descriptors in the form of local binary pattern (LBP) and uses neighborhood clustering technique. Both methods are based on the clustering techniques and follow the same steps at the beginning; The features should be extracted from each block once the image has been segmented into overlapping or non-overlapping blocks.

2.LITERATURES REVIEW

Judith A. Red, Wiem Taktak and Jean-Luc Dugelay introduces to the branch of Image Forensics and its importance in the present world. They specify how the images or videos can be manipulated and are show casted as original ones. Due to the easiness in which images' origin and content may be forged, digital image forensics aims to authenticate their authenticity by retrieving information about their origins. They focused on two major issues: identifying the imaging instrument that acquired the image and detecting forgery traces. They have mostly concentrated on the function of digital image forensics in multimedia security, in which a mark or message is concealed in a picture to protect the copyrights (Watermarking) or is used to communicate secret communications (Steganography). They also elaborate how the source of the image can be identified and the importance of its identification in the process of forgery detection. The focus is the Tampering Detection i.e., detecting that the image is spliced. There are various methods proposed that they have proposed for the forgery detection and provides us with the insights on how the fingerprints of the forgery can be erased i.e., covering the traces of image manipulation.

The key idea of Heng Yao is that locations with distinct origins have varied noise characteristics, which can be used as proof of forgeries. For digital photographs, he created a noise level evaluation approach that is used to spot image splicing. The noise distribution that was used in this study depends on intensity. A noise level function (NLF) that more accurately captures the properties of the actual noise can be used to characterize this model. The standard deviation of noise varies in relation to image intensity, according to NLF. The assessment of noise characteristics must be done in tiny areas because, in contrast to denoising issues, noise in forensic applications is typically minimal and content related. The NLF curve is fitted under the CRF restrictions by examining the relationship between NLF and the camera response function (CRF). Then, a Bayesian maximum a posteriori (MAP) framework is developed to improve the NLF estimate and create a technique for image splicing detection based on noise level discrepancy in image blocks acquired from various sources. The effectiveness of the suggested approach is demonstrated by the experimental findings.

Across this study, Chi-Man Pun describes how noise disparities in several scales are used as markers for the detection of image splicing fraud. This segmentation of the test image uses super pixels with various scales. A noise level function is produced for each scale, which depicts the relationship between the level of noise and the brightness of each segment. Suspicious regions are those regions that do not fall within the noise level function. The essential morphological processing is followed by the designation of the pixels that appear in suspicious areas of each scale as "spliced regions" (s). It is suggested to use the Optimal Parameter Combination Searching (OPCS) Algorithm to find the best parameters throughout the procedure. In order to evaluate the suggested scheme and train the best parameters, two datasets have been produced. The experimental findings indicate that the suggested technique is successful, particularly for the splicing of many items. The proposed plan has also been shown to outperform the most recent cutting-edge technique.

According to William T. Freeman, it is crucial to assess the picture noise level because many image processing algorithms need their parameters to be changed in accordance with the image noise level to function properly. He describes how to calculate an upper bound on the noise level from a single image using a piecewise smooth image prior model and observed CCD camera response functions. Additionally, he discusses the space of noise level functions, how noise level fluctuates with brightness, and how to use Bayesian MAP inference to estimate the noise level function from a single image. He demonstrates how two algorithms—edge detection and feature-preserving smoothing with bilateral filtering—can exploit this noise estimation.

3.FORGERY DETECTION

3.1 Introduction

As we are increasingly bombarded with visual imagery, it can be hard to know what to trust. With today's digital technology, it is easy for anyone to doctor an image. Forgery detection is a way to identify whether an image has been modified. Forgery detection can be done using a wide range of techniques. The two main categories of image forgery detection methods are Active Approach and Passive Approach.

3.1.1 Active Approach

Watermarking and digital signatures are two active picture protection techniques. These are also referred to as non-blind techniques. The main disadvantage of the watermark method is the requirement for watermarks to be inserted in the image prior to distribution. Most cameras on the market today lack the ability to insert watermarks. Utilizing these techniques also reduces the quality of the images.

3.1.1.1 Watermarking

In order to protect an image's copyright, a mark or message must be hidden in the image through the technique of watermarking. Watermarking aims to conceal a message pertaining to the real digital signal content.

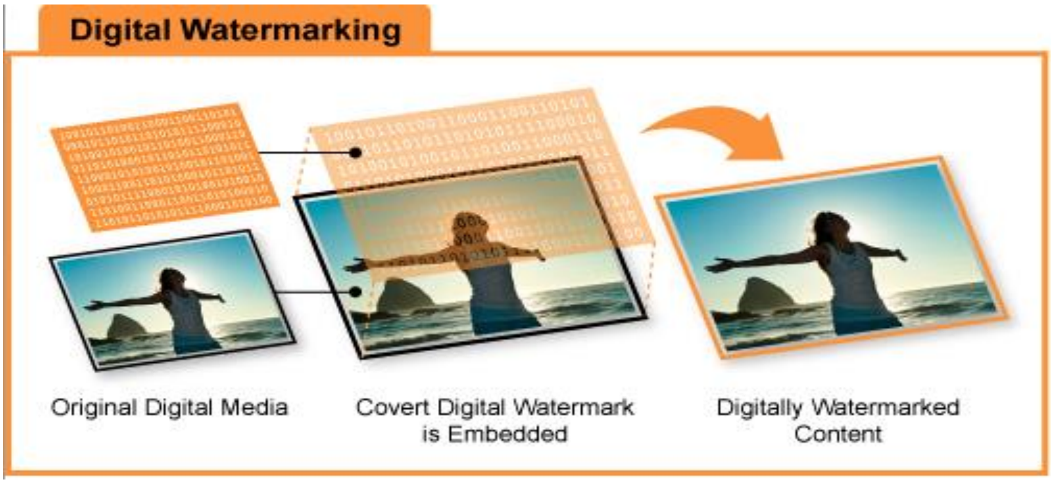

Digital Watermarking
Original Digital Media
Covert Digital Watermark is Embedded
Digitally Watermarked Content

Figure 3.1: Digital Watermarking

3.1.1.2 Steganography

Steganography is primarily used for covertly sending messages through pictures or movies. The digital signal in steganography serves just as a disguise for the message and has no connection to it.

Figure 3.2: Steganography

3.1.2 Passive Approach

No data needs to be inserted in photos for distribution when using the passive method of digital image authentication. Because the main image is not necessary for verification, these techniques are also referred to as "blind" techniques. Therefore, the field of visual forensics can likewise benefit from these techniques.

3.1.2.1 Image Splicing

The process of merging image segments from the same or distinct photos without additional processing or editing, such as smoothing the boundaries between separate pieces, is known as image splicing, and the result is a composite image known as a spliced image.

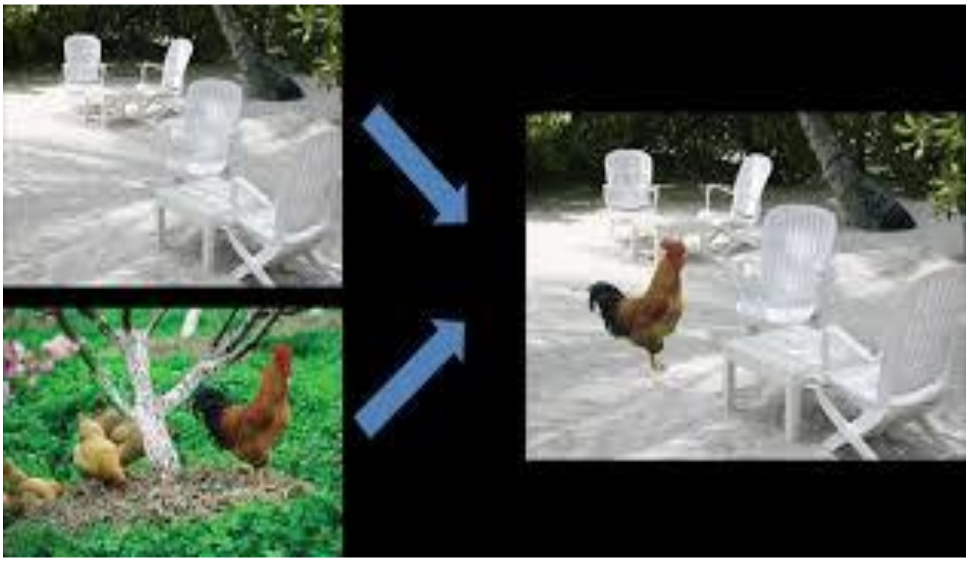


Figure 3.3: Image Splicing

3.1.2.2 Re-Sampling

A segment of an image can be copied, then pasted into another image, changing some orientational characteristics of the pasted section. The resampled image is shown in the next figure. It was made by taking a piece of the foreground picture, flipping it around, and pasting it in another place.

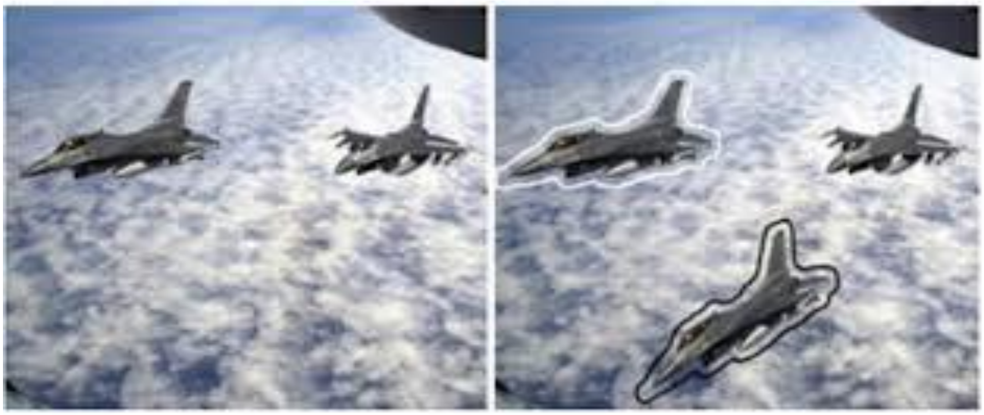


Figure 3.4: Re-Sampling

3.1.2.3 Copy-Move

A portion of the image is duplicated in Copy-Move and then placed into another portion. The following figure is as an illustration of a copy-move forgery in which some foreground images have been eliminated and others have been duplicated and pasted exactly in another location.

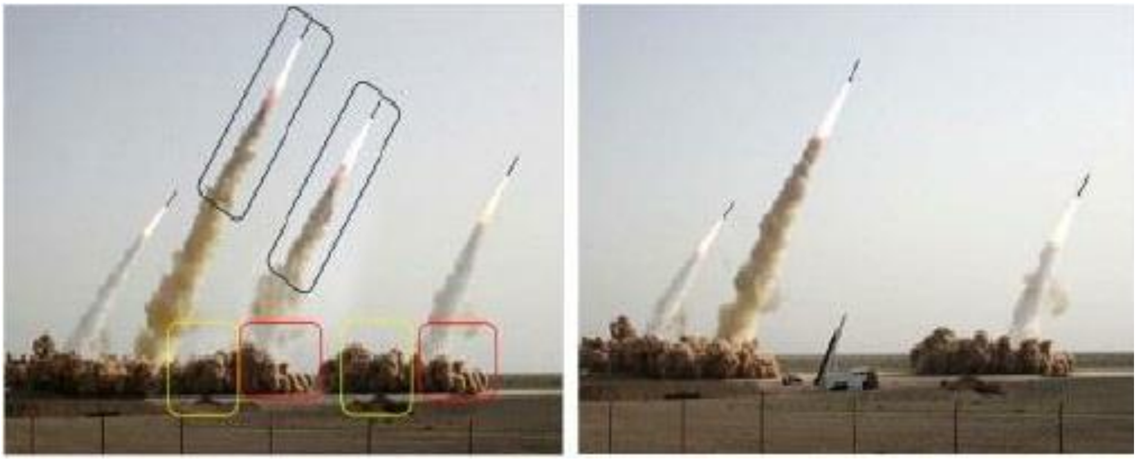


Figure 3.5: Copy-Move

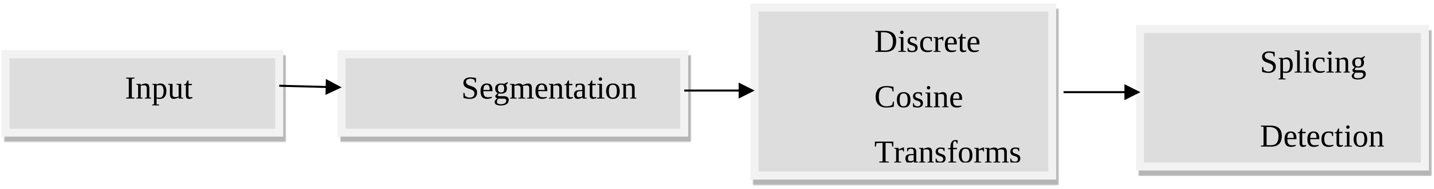
3.2 Forgery Detection Methods

Pixel-based, cloning, resampling, splicing, statistical, format-based, JPEG quantization, double JPEG, JPEG blocking, camera-based, chromatic aberration, color filter array, camera response, sensor noise, physics-based, light direction are some of the parameters that can be used to detect forgery (2-D and 3-D). In my research, I consider that the variables Noise and Intensity are crucial for finding forgery-related traces.

3.3 Forgery Detection

Taking the image to be examined for the detection of the spliced portions is the first step in the forgery detection process. Using segmentation techniques, the image is then sampled, and each segment is then taken independently to estimate the noise for detecting discrepancies. The forged sections must then be located using the suggested splicing detection algorithm.

Figure 3.6: Method Overview



The method that is chosen to implement is that, when an image is given, it is divided into different segments or clusters. These segments are then tested for forgery with the help of Discrete Cosine Transforms (DCT).

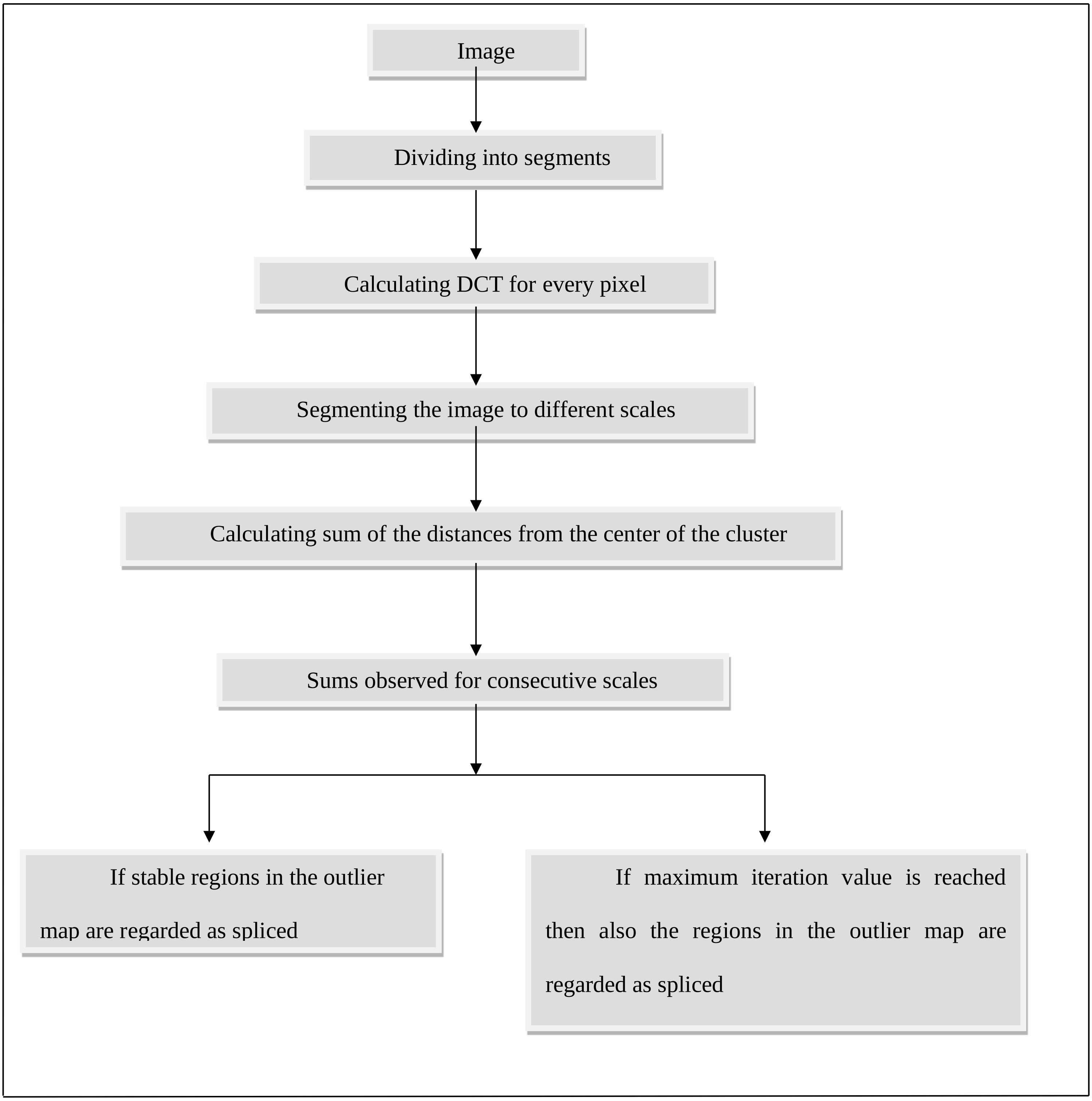


Image
Dividing into segments
Calculating DCT for every pixel
Segmenting the image to different scales
Calculating sum of the distances from the center of the cluster
Sums observed for consecutive scales
If stable regions in the outlier map are regarded as spliced
If maximum iteration value is reached then also the regions in the outlier map are regarded as spliced

Figure 3.7: Detection Flow Steps

The complete process of forgery detection, for instance, is summarized in the diagram below. Before being separated into blocks, the fake image is preprocessed. After checking if they match the pixels of the same block, which is verified, these blocks are used to extract the features. Next, the forged areas are detected.

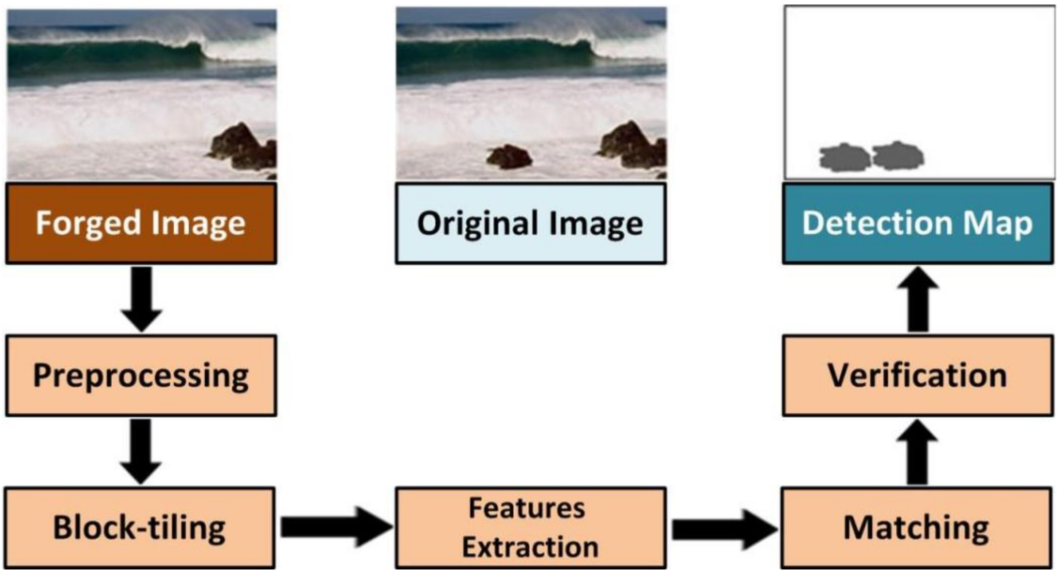


Figure 3.8: Detection Flow Steps

3.3.1 Statistical Features and Dendrogram Clustering

A flowchart for the first suggested strategy, which employs dendrogram clustering and statistical characteristics, is shown in Figure 3.10. The method involves four main steps: preparing the image, extracting basic statistical data, choosing the primary candidate by computing the block-pair distance, and then filtering the primary candidates by dendrogram clustering. The steps are thoroughly explained in the ensuing subsections.

3.3.1.1 Image Preparation

In this stage, a color image that was input is first turned into a grayscale image. Then, m x m pixel square pieces of the image are separated. It is possible for the blocks to overlap or not. We utilize an overlapping size of p p pixels when there is overlap. Because of this, there is no overlapping if p = m.

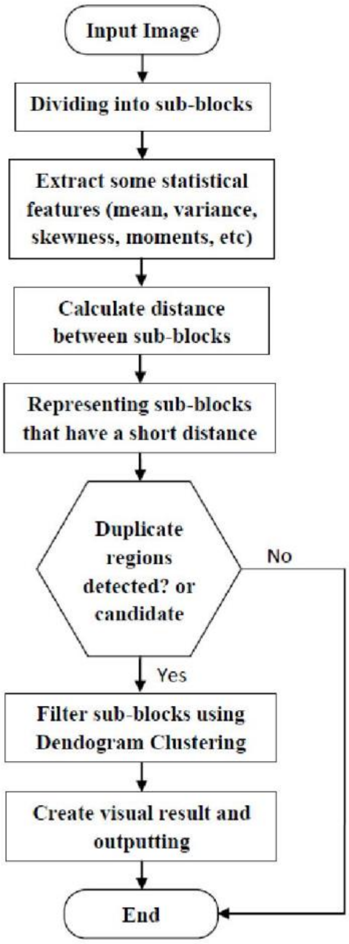

Input Image
Dividing into sub-blocks
Extract some statistical features (mean, variance, skewness, moments, etc)
Calculate distance between sub-blocks
Representing sub-blocks that have a short distance
Duplicate regions detected? or candidate
No
Yes
Filter sub-blocks using Dendogram Clustering
Create visual result and outputting
End

Figure 3.10: Flow chart of the first proposed method using statistical features and dendrogram clustering.

3.3.1.2 Statistical Features

In this step, we have extracted six statistical features from each block separately. The statistical features are as follows.

Mean**:** calculate the average value for each block. It is equal to the sum of all elements divided by the number of elements in each block. If *m* × *m* = *N*, the mean is calculated by the following equation:

$$\mathbf{M\ (mean)} = \frac{(\mathbf{i1+i2+\ldots iN})}{\boldsymbol{N}}$$, where N = number of elements in each block.

Variance**:** It is a measure of how far each value in the data set (block) is from the mean. Here is how it is calculated:

1. Subtract the mean from each value in the block. This gives a measure of the distance of each value from the mean.
2. Square each of these distances (so that they are all positive values), and add all of the squares together.
3. Divide the sum of the squares by the number of elements in each block.

***Standard Deviation*:** It is a quantitative measure of how much a group's members differ from one another. The standard deviation is large if individual observations differ significantly from the group mean, and vice versa. (Figure 3.11 illustrates this).

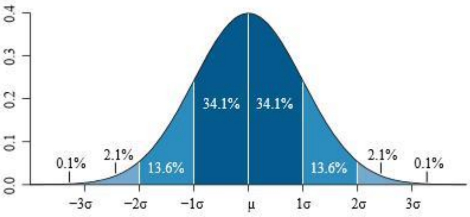


Figure 3.11: Standard deviation diagram (with plot of a normal distribution where each band has a width of 1 standard deviation).

The standard deviation of a population is defined by the following formula:

$$\boldsymbol{\sigma}\ \textbf{(population standard deviation)} = \sqrt{\frac{\sum(\mathbf{i}-\mathbf{M})\mathbf{2}}{\boldsymbol{N}}}\ \text{, where} \begin{cases} M = mean \\ N = number\ of\ elements \end{cases}$$

The standard deviation of a sample is defined by slightly different formula as follows:

$$\boldsymbol{\sigma}\ \textbf{(population standard deviation)} = \sqrt{\frac{\sum(\mathbf{i}-\mathbf{M})\mathbf{2}}{(\mathbf{N}-\mathbf{1})}}\ \textbf{, where} \begin{cases} \mathbf{M = mean} \\ \mathbf{N = number\ of\ elements} \end{cases}$$

***Skewness*:** It serves as a metric for the asymmetry of data relative to the sample mean. The data are spread out more to the left of the mean than to the right if skewness is negative (this is illustrated in Figure 3.12.a). Positive skewness causes the data to be more evenly distributed to the right (this is illustrated in Figure 3.12.b). The normal distribution has zero skewness, as does any completely symmetric distribution. The skewness of a distribution is defined as:

$$\textbf{s (skewness)} = \frac{\mathbf{E(i-M)3}}{\boldsymbol{\sigma}\mathbf{3}}\text{, where}\left\{\begin{array}{c}\mathbf{E(i) = expected\ value\ of\ the\ quantity\ 'i'} \\ \mathbf{M = mean} \\ \boldsymbol{\sigma}\mathbf{ = standard\ deviation}\end{array}\right.$$

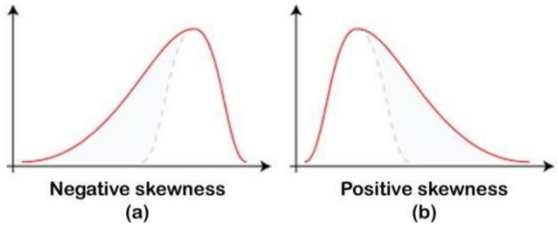


Figure 3.12: Shows the distribution of data about mean when: (a) the value of skewness is negative, (b) the value of skewness is positive.

***Kurtosis*:** It is a measure of whether the data are peaked or flat relative to a normal distribution.

$$\textbf{\textit{k (kurtosis)=}}\ \frac{\mathbf{E(i-M)4}}{\boldsymbol{\sigma}\mathbf{4}}\text{, where}\left\{\begin{array}{c}E(i) = expected\ value\ of\ the\ quantity\ 'i' \\ M = mean \\ \sigma = standard\ deviation\end{array}\right.$$

***Moment*:** For the time being, it can be physically defined as the separation between a point and a line or surface that is perpendicular to them. The core sample moment of the data (block), as determined by the positive integer order, can also be described mathematically.

The central moment of order *k* of a distributions defined as:

$$\mathbf{m_k\ (moment\ of\ order\ k) = E(i\text{-}M)^k}\text{, where}\left\{\begin{array}{c}E(i) = expected\ value\ of\ the\ quantity\ 'i' \\ M = mean \\ \sigma = standard\ deviation\end{array}\right.$$

3.3.1.3 Calculate the Distance:

The distance (dissimilarity) between the blocks of the image can be calculated in a number of methods

***Euclidean Distance*:** Figure 3.13 appears to depict the normal, simple distance between two points that may be determined with a ruler. If, for example, the length of the line segment connecting points I and j is the Euclidean distance between them, then the distance from I to j or from j to I is given by the following equation:

$$\mathbf{D(i, j) = D(j, i)} = \sqrt{\sum_{k=1}^{N}(\mathbf{ik}-\mathbf{jk})\mathbf{2}}\text{ , where N = number of element.}$$

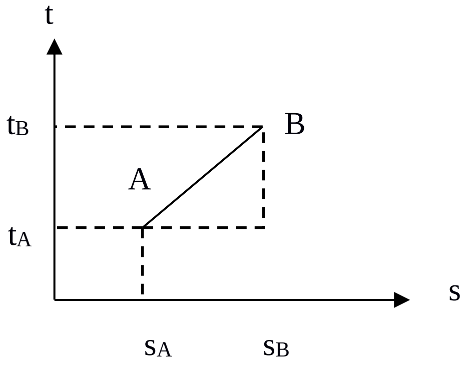
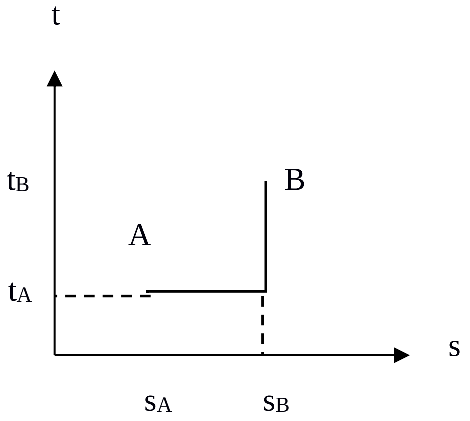


Figure 3.13: Calculation of the Euclidean distance between two vectors B= (tB, sB) and A= (tA, sA).

***City-Block*:** It denotes the shortest path taken by piecewise linear pathways whose line segments are all parallel to the same coordinate axis. Figure 3.14 seems to illustrate the mental picture that you are supposed to have of trying to get from one place to another in a city where the streets are built out in a grid-like layout, preventing you from moving in any direction other than horizontally and vertically.



$$\mathbf{D_{CB}(i, j)} = \sum_{k=1}^{N}|\mathbf{ik}-\mathbf{jk}|\text{ , where N = number of elements}$$

Figure 3.14: Calculation of the City-Block distance between two objects B=(tB, sB) and A=(tA, sA).

The city-block between the objects *i* and *j* is given in the above equation.

***Correlation distance*: **It is a measurement of the statistical relationship between two random variables or random vectors, which need not have the same dimensions. The following equation, for instance, gives the correlation distance between vectors I and j:

$$\mathbf{D_{cr}(i,j)} = \frac{\frac{1}{N}\sum_k(\mathbf{ik}-\mathbf{jk}) -(\mathbf{Mi\ Mj})}{\boldsymbol{\sigma}\mathbf{i}\ \boldsymbol{\sigma}\mathbf{j}}\text{, where}\begin{cases} Mi\ and\ Mj = mean \\ \sigma i\ and\ \sigma j = standard\ deviation \end{cases}$$

The difference between the mean of the product of I and j subtracted from the product of the means is what the equation's numerator, covariance of I and j, is called.

***Hamming distance*:** It uses the number of mismatches between each pair of variables to measure the distance between two objects. Although it can be helpful for numerical variables, it is primarily utilized for string and bitwise analyses. Despite being a metric, the fundamental Hamming distance can be used to specify a threshold. The following equation provides the Hamming distance between the objects I and j:

$$\mathbf{D_{HAM}(i,j)} = \sum_{k=1}^{N}[\mathbf{yi}, \mathbf{k}\ \neq\ \mathbf{yj}, \mathbf{k}]$$

where *k* is the index of the respective variable reading *y* out of the total number of variables *n*. The Hamming distance itself gives the number of mismatches between the variables paired by *k*.

3.3.1.4 Dendrogram Clustering:

A measure of dissimilarity between sets of data is necessary to determine where clusters should be joined (for agglomerative) or where a cluster should be divided (for divisive).

This is done by utilizing an appropriate metric and a linkage criterion, which defines the dissimilarity of sets as a function of the pair-wise distances of observations in the sets (a measure of distance between pairs of observations; it is detailed in section [3.1.3]).

3.3.1.4.1 Linkage Criteria: Ce Liu William T. Freeman

The pairwise distances between observations are a function of the linkage criterion, which establishes the distance between sets of observations. Here are four different methods for linkage criteria:

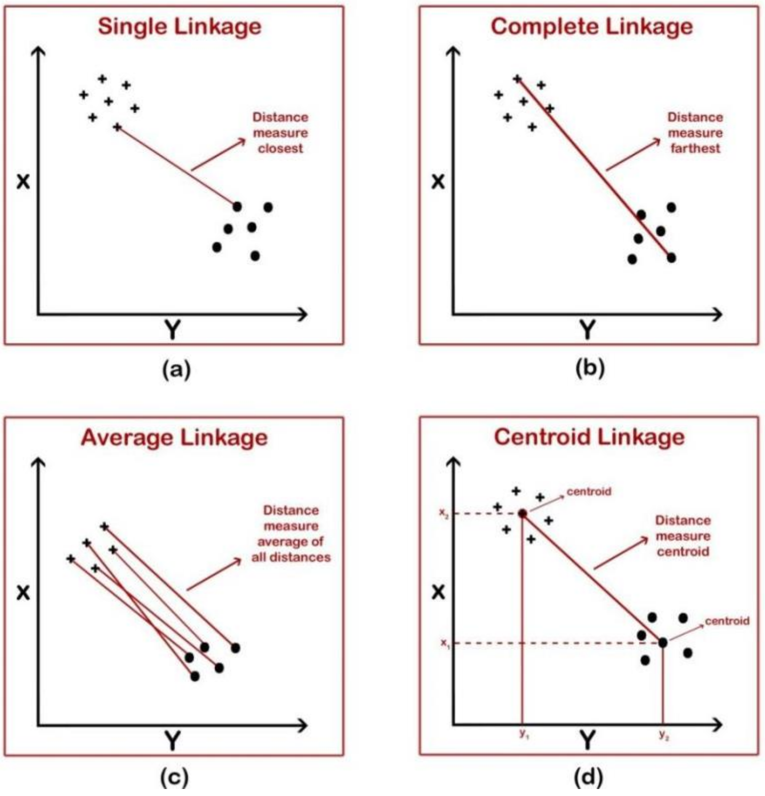


Figure 3.16: Illustration of the four different methods for linkage criteria, (a) single linkage, (b) complete linkage, (c) average linkage, and (d) centroid linkage.

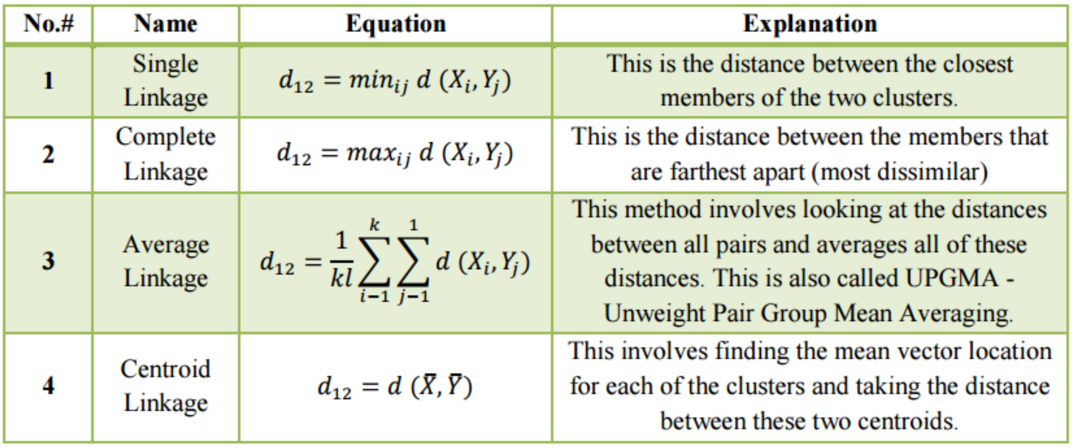
Single Linkage: In single linkage, the distance between the two clusters is the shortest distance between any two individual data points in the first cluster and any individual data point in the second cluster. We combine the two clusters at each stage of the process that have the smallest single linkage distance, as illustrated in part (a) of Figure 3.16, in accordance with this definition of the distance between clusters.

Complete Linkage: According to complete linkage, the maximum distance between any two data points—one from the first cluster and one from the second—is what we refer to as the distance between two clusters. According to this definition of the distance between clusters, we join the two clusters that have the least complete linkage distance at each stage of the procedure, as shown in portion (b) of Figure 3.16.

Average Linkage: In average linkage, we define the distance between two clusters as the average distance between the data points in the first cluster and the data points in the second cluster. We unite the two clusters that have the smallest average linkage distance at each stage of the process, as illustrated in part (c) of Figure 3.16, using this definition of the distance between clusters.

Centroid Linkage: The distance between two clusters in a centroid linkage is determined by the separation of their respective mean vectors. As seen in part (d) of Figure 3.16, we merge the two clusters that have the shortest centroid distance at each stage of the procedure.

Table 3.1: Four different methods for linkage criteria.



No.#	Name	Equation	Explanation
1	Single Linkage	$d_{12} = min_{ij}\, d\,(X_i, Y_j)$	This is the distance between the closest members of the two clusters.
2	Complete Linkage	$d_{12} = max_{ij}\, d\,(X_i, Y_j)$	This is the distance between the members that are farthest apart (most dissimilar)
3	Average Linkage	$d_{12} = \frac{1}{kl}\sum_{i-1}^{k}\sum_{j-1}^{1} d\,(X_i, Y_j)$	This method involves looking at the distances between all pairs and averages all of these distances. This is also called UPGMA - Unweight Pair Group Mean Averaging.
4	Centroid Linkage	$d_{12} = d\,(\bar{X}, \bar{Y})$	This involves finding the mean vector location for each of the clusters and taking the distance between these two centroids.

3.3.1.4.2 Dendrogram

The organization of the clusters created by hierarchical clustering is typically illustrated using a tree diagram. The spot correlation data can be seen visually in a dendrogram. The distinct locations, known as leaf nodes, are organized along the dendrogram's base. Spot clusters are created by connecting individual spots or already-existing spot clusters at a node.

Numerous U-shaped lines connecting data points in a hierarchical tree make up a dendrogram. According to Figure 3.17, the height of each U denotes the separation between the two connected data points.

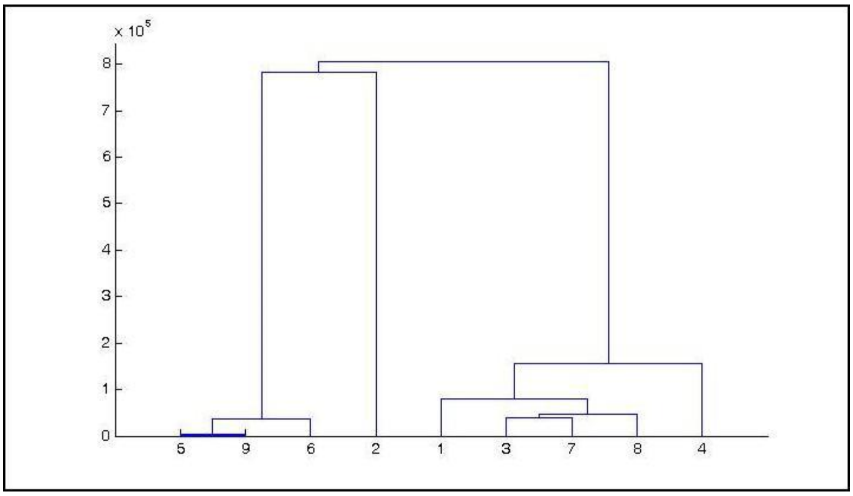


Figure 3.17: Illustration the output of dendrogram clustering algorithm. The x-axis numbers represent candidate sub-blocks, and y-axis numbers represent linkage (distance) values.

We are forced to look for a new and more effective method, called (LBP Feature and Neighborhood Clustering), which will be detailed in the following part, even though dendrogram clustering works well and produces outstanding and favorable outcomes. There are three basic causes for this:

1. The dendrogram produces accurate findings when the objects to be recognized are large in the images, but the results are inaccurate when the targets to be detected are small.
2. The dendrogram (by default) uses the first 30 blocks of any image that has been fragmented into sub-blocks (more than 30 blocks) and ignores the remaining blocks. The dendrogram can be longer than 30 blocks, but the findings won't be precise and obvious.
3. In contrast to this method, which used basic features, the second method used the extracted feature (local binary pattern) for all blocks. Additionally, this weakens the outcomes.

3.3.2 Second Method: LBP Feature and Neighborhood Clustering

The second proposed method in this research is to detect the copy-move image forgery using LBP feature, and neighborhood clustering technique.

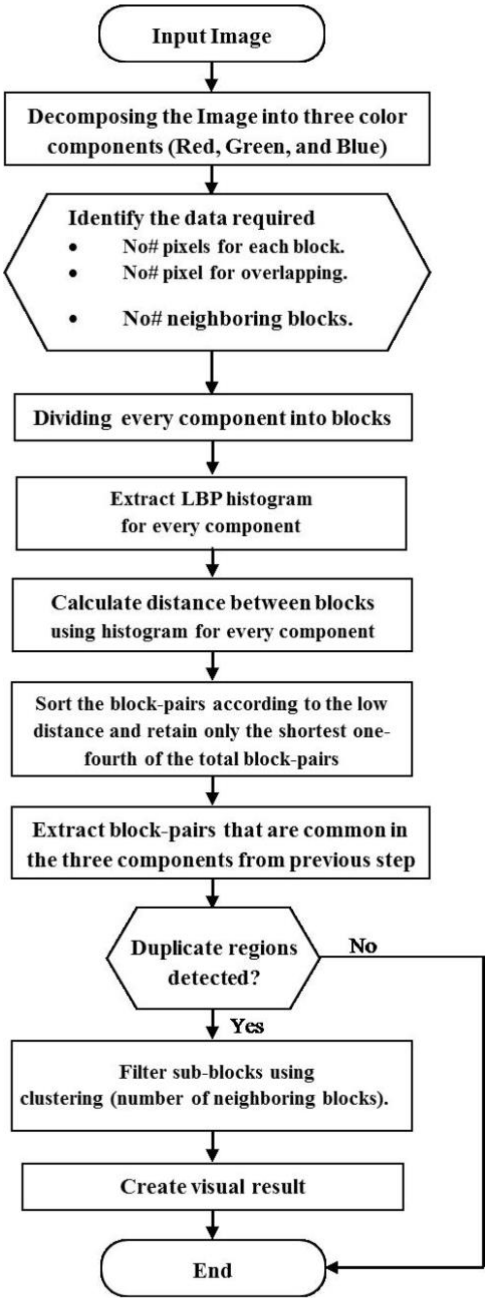

Input Image
Decomposing the Image into three color components (Red, Green, and Blue)
Identify the data required
• No# pixels for each block.
• No# pixel for overlapping.
• No# neighboring blocks.
Dividing every component into blocks
Extract LBP histogram for every component
Calculate distance between blocks using histogram for every component
Sort the block-pairs according to the low distance and retain only the shortest one-fourth of the total block-pairs
Extract block-pairs that are common in the three components from previous step
Duplicate regions detected?
No
Yes
Filter sub-blocks using clustering (number of neighboring blocks).
Create visual result
End

Figure 3.18: Flow chart of the second proposed method using LBP features and Neighborhood clustering.

The second suggested approach for (copy-move picture forgery detection) is shown as a flowchart in Figure 3.18. It consists of "five stages," including image preparation and data identification, LBP histogram extraction, distance calculation, primary candidate selection, and neighborhood clustering. More in-depth descriptions of each stage are provided in the following sections.

3.3.2.1 Image Preparation

An input image is divided into its three color (red, green, and blue) components in this step. This breakdown was done in order to make use of the various components of information that make up different color components. Most ways (like the first method) start by turning the color image into a grayscale image. To create the grayscale during this conversion, various approximations in weight are applied to the three-color components. Some mild but significant signs of fraud could be lost during this approach.

3.3.2.2 LBP Histogram

Local binary patterns are a type of feature used in computer vision to categorize objects (LBP). Also described is the LBP texture analysis operator, which produces a gray-scale invariant texture measure given a broad description of texture in a nearby area. The LBP operator's present form differs greatly from its initial form since various extensions have been created, and the original concept has been expanded to include arbitrary circular neighborhoods.

The LBP can be expressed in the decimal form as:

$$\mathbf{LBP_{P,R}(x_c,y_c) = \sum_{p=0}^{P-1} s(i_p - i_c)\, 2^p}\text{, where } s(x) = \begin{cases} 1, if\ x \geq 0 \\ 0, otherwise \end{cases}$$

where i_c and i_p are respectively, gray-level values of the central pixel and P surrounding pixels in the circle neighborhood with a radius R, in the original case $P=8$ and $R=1$, (as illustrated in Figure 3.18).

With the help of the normalized LBP histogram, a feature vector is produced for the matched block. The histogram's 256 bins stand in for the 256 various colors of gray. We utilize LBP because we are interested in texture, which is still there in the copied and pasted area even after forging and some post-processing. The texture pattern may be a reliable indicator of fakes.

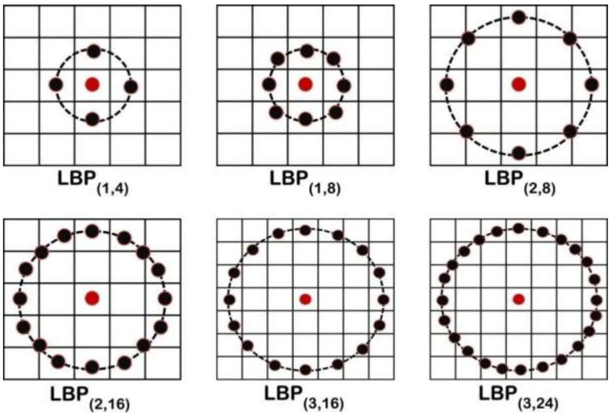


Figure 3.20: Examples of the extended LBP operator.

Figure 3.20 illustrates an LBP histogram as an example. It has three color components and four blocks, three of which display copy-move and one of which does not. For the sake of simplicity, blocks are shown as not overlapping. In their respective color components, the cloned blocks (blocks 23, 28, and 63 in the figure) show comparable LBP histograms, but block number 79's histogram is entirely different.

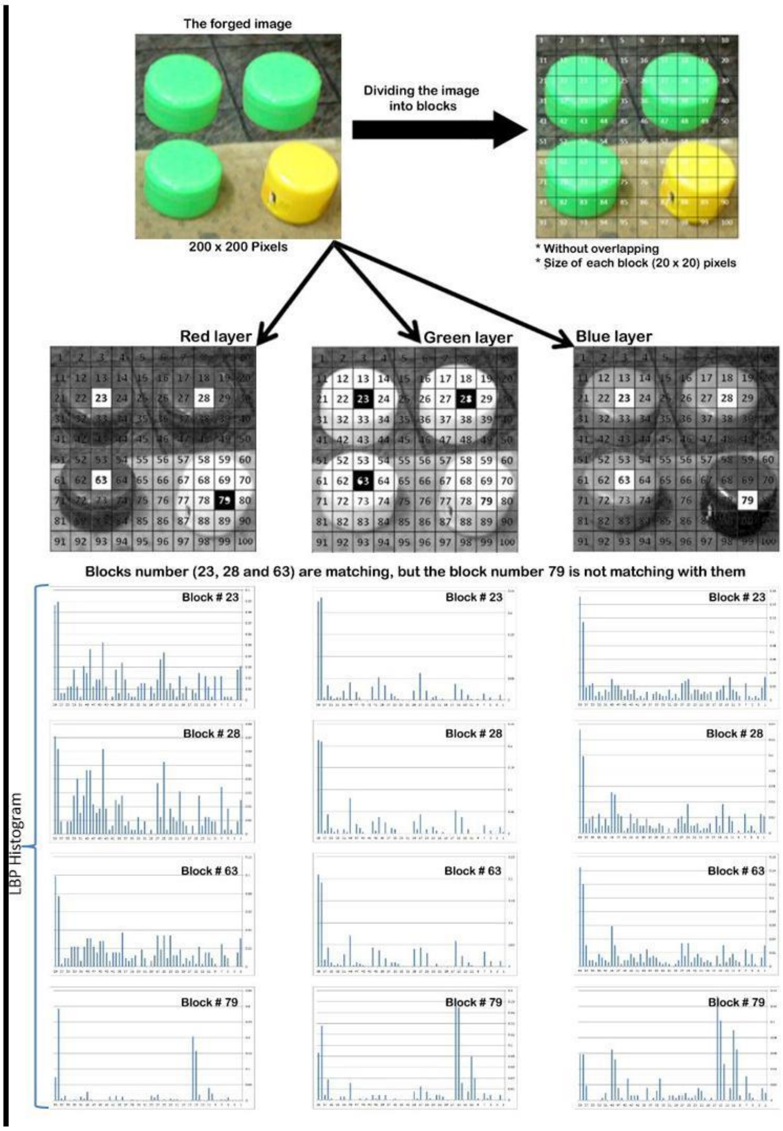

The forged image
200 x 200 Pixels
Dividing the image into blocks
* Without overlapping
* Size of each block (20 x 20) pixels
Red layer
Green layer
Blue layer
Blocks number (23, 28 and 63) are matching, but the block number 79 is not matching with them
LBP Histogram
Block # 23
Block # 28
Block # 63
Block # 79

Figure 3.21: Illustration of LBP histogram similarity between the copied and pasted blocks, and dissimilarity between non-copy-move parts in three color components.

3.3.2.3 Calculate the Distance

The dissimilarity (distance) between the blocks of the image can be calculated in a number of ways, as detailed in section [3.1.3]). Additionally, we concentrate on Euclidean distance and city-block in this method because the results are straightforward and can be applied to the clustering process.

3.3.2.4 Primary Candidate Selection

The recovered LBP histograms of the blocks are used to compute some distance metric between each pair of blocks. Since it is simple and works well in histograms, city block distance is used in the suggested technique. The block pairs are organized according to increasing distance to generate a list. The list will then be sorted so that the similar block pairings are displayed first. In an image with M total blocks, there will be the following number of block pairs:

$$^{M}C_{2} = \frac{M\ X\ (M-1)}{2}$$

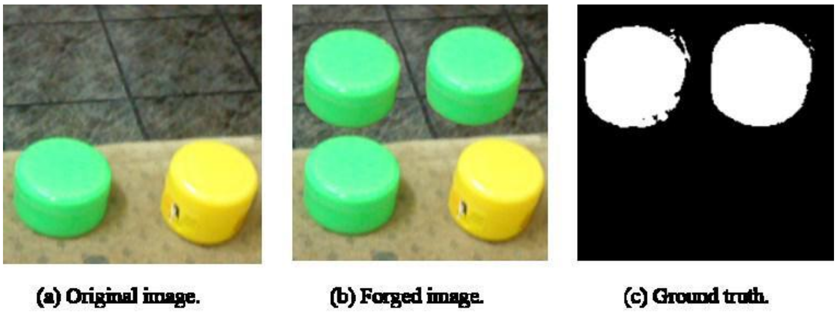

(a) Original image. (b) Forged image. (c) Ground truth.

Figure 3.25: An example of (a) original image, (b) forged image, and (c) the ground truth image.

4.ALGORITHMS

In order to comply with the methodology's requirements, we must implement several algorithms. The Forgery Detection by Noise Level Function algorithm, the first one listed below, provides a general overview of how the selected methodology is used. The second algorithm mentioned is the Optimal Parameter Searching Algorithm, which is necessary to provide the main algorithm's terminal conditions. The image can be divided into different scales using the third approach listed, the Simple Linear Iterative Clustering Algorithm.

4.1 Optimal Parameter Combination Searching (OPCS) Algorithm

Input: ROC curves A_j

Output: Optimal Parameters Cmb = $\{K, C, \tau\}$

```
for each ROC $A_j$
    for each point on ROC $a^i_j = \{p,r|K, C, \tau\} \epsilon A_j$
        Calculate the distance from $a^i_j(p^i_j, r^i_j)$ to $Idl(p_{idl},r_{idl})$. as $Dist^i_j$
    end for
end for
for each $A_i$
    Sort $A_i$ in ascending order according to $Dist^i_j$, and store as $A\_sort_i = [a_{1,j}, a_{2,j}, ... , a_{N\_Cmb,j}]$
end for
for j = 1 : N_Cmb
    if $\bigcap^{N\_S}_{j=1}\bigcap^{N\_Cmb}_{i=1} a_{i,j}(K, C, \tau) \neq \phi$
        Optimal Parameter Combination
        $\bigcap^{N\_S}_{j=1}\bigcap^{N\_Cmb}_{i=1} a_{i,j}(K, C, \tau) \Rightarrow \{K, C, \tau\}$
        return Cmb = $\{K, C, \tau\}$
    end if
end for
```

Three parameters are defined to judge the stability of the outliers map: the number of clusters, K; the threshold of the change ratio, s; and the number of successive scales, C. First, the outliers map Mi is clustered into K clusters, and the distance from each pixel in the cluster to its corresponding cluster center are summed, then, K inner sums are accumulated as $SUMD_i$, with which, the change rate of outliers map between two successive scales can be determined by

$$\frac{\tau i = |SUMDi - SUMDi - 1|}{SUMDi - 1}$$

where $SUMD_i$ and $SUMD_{i-1}$ indicate the accumulated value of the successive ith and (i-1)th scale, respectively; and τ_i is the change rate of the successive ith and (i-1)th scale.

4.2 DCT Algorithm

Input: Spliced image

Output: Spliced regions

1. Take tampered image as a input.
2. Divide input image into 8x8 block of pixels.
3. Apply DCT to each block of pixels.
4. Compare DCT over all blocks.
5. Analyse forgery.
6. Output forged regions.

In this DCT Algorithm, initially the input image is divided into an 8x8 block of pixels. Each pixel in every block is applied a DCT. Now these DCT values of the pixels are compared, and its results are given as input to the forgery detection algorithms.

4.3 Hierarchical Clustering

Giving a set of *N* candidate blocks to be clustered, *N**N* matrix will be produced:

1. Start by assigning each block to a cluster.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster.

Note: Step 3 can be done in different methods of linkage criteria, but in our method we used the single linkage. Now we explain how the single linkage clustering algorithm works.

4.4 Single Linkage Clustering Algorithm

The algorithm is composed of the following steps:

1. Begin with the disjoint clustering having level *L (0)=0* and sequence number *m = 0*.
2. Find the least dissimilar pair of clusters in the current clustering, say pair *(r), (s),* according to *d[(r), (s)] = min d[(i), (j)]*.
3. Increment the sequence number: *m = m +1*. Merge clusters *(r)* and *(s)* into a single cluster to form the next clustering *m*. Set the level of this clustering to *L(m) = d[(r), (s)]*.
4. Update the proximity matrix, *D*, by deleting the rows and columns corresponding to clusters *(r)* and *(s)* and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted *(r,s)* and old cluster *(k)* is defined in this way *d[(k), (r,s)] = min d[(k),(r)], d[(k),(s)]*.
5. If all objects are in one cluster, stop. Else, go to step 2.

Recently, there was a clear spread for using "Clustering Technique" in the process of image forgery detection. This technique is often used in two ways:

1. After the process of extracting properties, by using "Clustering Technique" the blocks are classified in groups, and then search for the groups which have the same properties.
2. After the process of extracting properties, "Clustering Technique" filters the remaining blocks, to keep the matched blocks.

4.5 Results

The following are the test images upon which Forgery Detection Algorithms have been applied so as to find the Spliced regions. There is a threshold that is fixed i.e., 0.9. All the regions above the threshold are considered as Spliced regions and the remaining as non-spliced.

4.5.1 Image 1

Following are the input and output images of a forgery detection algorithm. We input a forged image and we get the output image with the indication of the forged areas with a red mark.

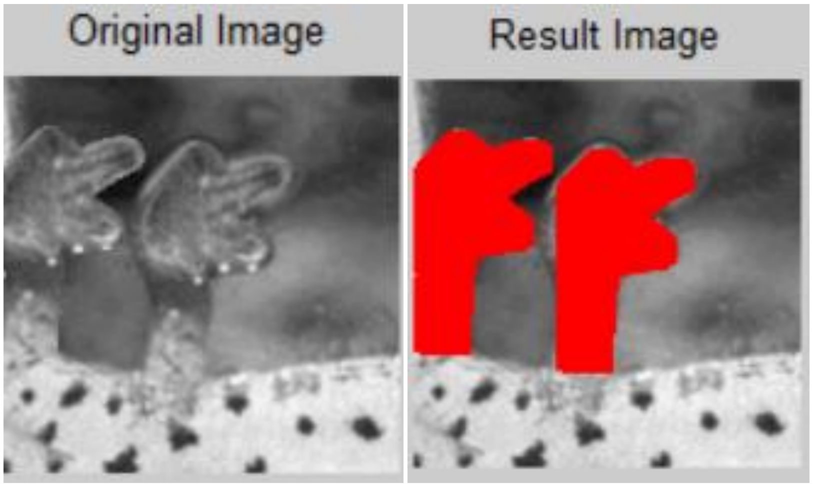


Figure 4.1: Original and Result Image after Splicing Detection

CONCLUSION

Forgery Detection can be done in many ways more than what is mentioned in the project. The method used in the project has tried to efficiently extract only the spliced regions with very precise accuracy. Since the forgery detection is done at many scales the possibility that a non-spliced region is marked splice is almost 'zero'. But the limitation of this project is that we need to find the images having a certain threshold difference at the edges of the spliced regions. When such kind of images are found this algorithm proposed works at its best precision. There are three images which have been found with the requirements and they are used for testing the project. The results for those images came out perfect and as enclosed in the results section of the documentation. The scope of it is not just restricted to images whereas it can be extended to videos as well, since video is nothing, but a series of images projected at a certain rate.

BIBLIOGRAPHY

[1] S. Kumar, and S. Mukherjee, *"Copy-Move Forgery Detection in Digital Images: Progress and Challenges"*, International Journal on Computer Science and software Engineering (IJCSE), Vol. 3, no. 2, Feb 2011, pp.652-663.

[2] L. Wu, X. Kong, B. Wang, and S. Shang, *"Image Tampering Localization via Estimating the Non-Aligned Double JPEG compression"*, Media Watermarking, Security, and Forensics, Vol. 8665, 2013, pp.1-7.

[3] Muneer H. Al-Hammadi, *Copy moves forgery detection in digital images based on multiresolution techniques*, Master Thesis, Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, May 2013.

[4] Available at: http://zoi.utia.cas.cz/image-forensics, accessed on December 3, 2013.

[5] S.J. Sun, Q. Wu and G.H. Li, *"Detection of image compositing based on a statistical model for natural images"*, Acta Automatica Sinica, Vol. 35, No. 12, December 2009, pp.1564-1568.

[6] F. Peng, Y. Y. Nie and M. Long, *"A complete passive blind image copy-move forensics scheme based on compound statistics features"*, Forensic Science International 212 (2011), pp.e21-e25.

[7] Shrishail Math, R.C.Tripathi, *"Digital Forgeries: Problems and Challenges"*, International Journal of Computer Applications, , Vol. 5, No.12, August 2010, pp.9-12**.**

[8] O. Al-Qershi and B. Khoo, *"Passive detection of copy-move forgery in digital images:State-of-the-art"*, Forensic Science International 231, 2013, pp.284–295.

[9] Syed Z. M. Shaid, "Image Forgery Detections", available at: http://csc.fsksm.utm.my/syed/research/image-forensics/12-image-forgery-detection.html, accessed on April 28, 2012.

[10] S. Theodoridis and K. Koutroumbas, Pattern Recognition, Fourth Edition, Chapter 13, Academic Publisher, 2009.

[11] G. Li, Q. Wu, D. Tu and S .Sun, *"A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD"*, International Conference on Multimedia and Expo (ICME) 2007, pp.1750-1753.

[12] S. Bravo-Solorio and A. K. Nandi, *"Passive forensic method for detecting duplicated regions affected by reflection, rotation and scaling"*, 17th European Signal Processing Conference (EUSIPCO 2009), pp.824-828.

[13] B. Mahdian and S. Saic, *"Using noise inconsistencies for blind image forensics"*, Image and Vision Computing, vol. 27(10), 2009, pp. 1497–1503.

[14] V. Christlein, C. Riess and E. Angelopoulou, *"A Study on Features for the Detection of Copy-Move Forgeries"*, Proc. Information Security Solution Europe (ISSE 2010), pp.105-116.

[15] Y. Huang, W. Lu, W. Sun and D. Long, *"Improved DCT-based detection of copy-move forgery in images"*, Forensic Science International 206 (2011), pp. 178-184.

[16] X. Quan and H. Zhang, *"Copy-Move Forgery Detection in Digital Images Based on Local Dimension Estimation"*, International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), June 2012, pp. 180-185.

[17] L. Jing and C. Shao, *"Image Copy-Move Forgery Detecting Based on Local Invariant Feature"*, Journal Of Multimedia, Vol. 7, No. 1, February 2012, pp.90-97.

[18] I. Amerini, et al., *"Copy-move forgery detection and localization by means of robust clustering with J-Linkage"*, Signal Processing-Image Communication (2013), http://dx.doi.org/10.1016/j.image.2013.03.006.

[19] R. Davarzani, K. Yaghmaie, S. Mozaffari and M. Tapak, *"Copy-move forgery detection using multiresolution local binary patterns"*, Forensic Science International 231 (2013), pp.61–72.

[20] H. Hsu and M. Wang, *"Detection of Copy-Move Forgery Image Using Gabor Descriptor"*, International Conference on Anti-Counterfeiting, Security and Identification (ASID), Aug 2012, pp.1-4.

[21] Di Huang, Caifeng Shan, Ardabilian, M., Yunhong Wang, Liming Chen, "*Local Binary Patterns and Its Application to Facial Image Analysis: A Survey*", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol.41, Nov.2011, pp.765-781.

[22] Ojala T., Pietikainen, M., Maenpaa, T., "*Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, Jul 2002, pp.971-987.

[23] Ahonen, T., Hadid, A., Pietikainen, M., "*Face Description with Local Binary Patterns: Application to Face Recognition*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.28, Dec. 2006, pp.2037-2041.

[24] Babak Mahdian and Stanislav Saic,*"Detecting Double Compressed JPEG Images"*, Proc. 3rd Int. Conf. Imaging for Crime Detection and Prevention, 2009.

[25] A. F. Martin, G. R. Doddington, T. Kamm, M. Ordowsky, M. A. Przybocki, "*DET curve in assessment of detection task performance"*, Proc. Eurospeech'97, vol. IV,

1997, pp.1895-1898.

[26] M Babak and S Stanislav, *"A bibliography on blind methods for identifying image forgery"*, Signal Processing: Image Communication, Vol. 25, 2010, pp.389–399.

[27] P. Kakar,and N. Sudha, *"Exposing Post processed Copy-Paste Forgeries through Transform-Invariant Features"*, Information Forensics and Security, IEEE Transactions on,vol.7, 2012, pp.1018 - 1028.

[28] P. Yadav,and Y.Rathore, *"Detection of Copy-Move Forgery of Image Using Discrete Wavelet Transform"*, International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 4, April 2012, pp.2-5.

[29] B.L. Shivakumar, and S. Santhosh, *"Automated Forensic Method for Copy-Move Forgery Detection based on Harris Interest Points and SIFT Descriptors"*, International Journal of Computer Applications, vol. 27, no.3, August 2011, pp.9-17.

[30] Redi, J.A., Taktak, W. & Dugelay, JL. Multimed Tools Appl (2011) 51: 133. doi:10.1007/s11042-010-0620-1 http://link.springer.com/article/10.1007/s11042-010-0620-1

[31] Chi-Man Pun *, Bo Liu Xiao-Chen Yuan http://dx.doi.org/10.1016/j.jvcir.2016.03.005

[32] Yao, H., Wang, S., Zhang, X. et al. Multimed Tools Appl (2016). doi:10.1007/s11042-016-3660-3. Copyright © 2016 Elsevier B.V. or its licensors or contributors. ScienceDirect ® is a registered trademark of Elsevier B.V. http://link.springer.com/article/10.1007/s11042-016-3660-3

[33] CVPR '06 Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1. IEEE Computer Society Washington, DC, USA ©2006. ISBN: 0-7695-2597-0 doi>10.1109/CVPR.2006.207. http://dl.acm.org/citation.cfm?id=1153435

[34] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua,
and Sabine S¨usstrunk. http://www.kev-smith.com/papers/SLIC_Superpixels.pdf