Detection of Copy More Forgery in Digital Images Using an optimized Block Based Approach

A Dissertation submitted in partial fulfilment of the requirement for the award of degree of

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted by

PRASHIT YADAV

(2K12/ISY/20)

Under the esteemed guidance of

RITU AGARWAL

(Asst. Professor, IT Department)

Department of Information Technology

Delhi Technological University

Bawana Road, Delhi – 110042

(2012-2014)

CERTIFICATE

This is to certify that **Prashit Yadav (2K12/ISY/20)** has carried out the major project titled **"Detection of Copy More Forgery in Digital Images Using an optimized Block Based Method"** for the award of Master of Technology degree in Information Systems by Delhi Technological University.

The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session **2012-2014**. The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)

Ms Ritu Agarwal

Assistant Professor

Department of Information Technology

Delhi Technological University

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I express my sincere thanks and deep sense of gratitude to my project guide**,** **Ms Ritu Agarwal**, Assistant Professor, Department of Information Technology, Delhi Technological University, for her valuable motivation and guidance, without which this study would not have been possible. I consider myself fortunate for having the opportunity to learn and work under her supervision and guidance over the entire period of association.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

Prashit Yadav

Roll No. 2K12/ISY/20

M.Tech. (Information Systems)

E-mail: prashit_yadav@yahoo.com

ABSTRACT

Technology is growing exponentially and so does the sophistication of various imaging tools that makes it easier for a person to forge digital images. There are several types of forgeries which are done on images out of which Copy-Move is one of the most widely used. In this type of forgery a portion of an image is copied and pasted over to another portion to conceal some information. Several approaches have been developed so far to detect such kind of forgery. In the block based approach the image is divided into overlapping blocks and each block is matched against every other block for similarity. However such a matching would be computationally expensive, thus each block's dimensionality is reduced by extracting its vital features and using this information for representation of the block. To extract the features we first applied the Laplacian of Gaussian filter on the image to highlight the areas of interest such as edges, blobs, etc. Subsequently we used Discrete Cosine Transform algorithm on each block to generate DCT coefficient matrix for each block. The DCT coefficients matrix is quantized and averaging is used to extract a feature set with reduce dimensionality. Now these feature vectors with respect to each block are stored in a matrix called feature vector matrix. The matrix is then lexicographically sorted so that vectors corresponding to similar blocks come adjacent to each other. The decision of forgery can then be taken if a number of connected/similar blocks are equidistant to each other. The feature extraction is vital as not only does it reduce the computation complexity but choosing the correct set of features makes an approach invariant to various image manipulations such as rotation, translation, scaling, etc.

Table of Contents

Chapter 1
Introduction

1.1 Introduction

We live in an age where technology has become a necessity. We can see digitalization and computerization in every aspect of life. With advent of technology and growing number of digital devices, digital images and videos have become a main source of information carrier. In the present times, digital images play a major role in myriads of areas like Weather forecasting, Astronomy, Forensic Investigation, Surveillance Systems, Media Industry, Medical imaging, Intelligence Services, normal day to day photography and so on so forth.

With this increased dependence on digital images, a natural need that arises is the images that we see should be authenticated and not doctored. The same technology that gave us this benefit also gave the potential forger power to forge the digital images. There are several tools and software available with which even a non professional can forge the images very easily. As a result we see some form of forgery in our everyday life from the fashion industry to the tabloid magazines and in newspapers, scientific journals, courtrooms, campaigns – political and apolitical, and numerous photo frauds that come in our e-mails, forged photographs are appearing with an ever-increasing frequency and erudition. This development show stern vulnerabilities and lessens the credibility of the digital images. Considering the fact the images are also presented in court of law as evidence, as medical images for examination purposes, as financial documents and in many other areas, it became necessary to develop techniques and methodologies through which the authenticity and integrity of digital images can be verified. However before moving to detection of image forgery we must acquaint ourselves with the various forms of digital image forgeries.

Digital Image: Any such image which is derived from or processed with a digital device. The digital device or gadget may be a digital camera, computer, etc. In simpler words any image that is either derived via a digital device or an image which can be viewed in digital device are called digital images.

“Digitization” is a process through which a real world scene or an analog colour picture/image is converted to numerical data in the form of matrices where each data element represents the intensity levels of the original scene or picture. A digital image is stored as 2D or 3D matrices where size of the matrix gives us the size/resolution of image in pixels and each element of the matrix gives an intensity level to the corresponding pixel.

1.2 Types of Image Forgery

Digital Image forgery does not differ very much in nature as compared to conventional image forgery. Instead of using photograph, digital image forgery deals with digital representation of the image. Digital Image forgery is a method in which the digital representation of the image is modified to hide some information, add some information or to create some kind of visual artefact. The process of image forgery have become enormously easy with the upcoming of potent graphics amending software like PhotoEdit, Corel Draw, lunapic, etc., most of which are available for free.

Among all the forgeries, some of the most common types of forgeries being done in prevalent times are four which are explained below.

1.2.1 RETOUCHING

This is a type of forgery which came into existence long ago, just about the same time when photography was discovered. Compared to other forgeries this is still less dangerous as the implications are less harmful. This does not completely change an image but alters certain features such as enhancing some features while reducing others. A lot of photographers and editors of various magazines and newspapers very often make use of this method to make their pictures more appealing and eye-catching. An example retouching is shown in Fig 1.

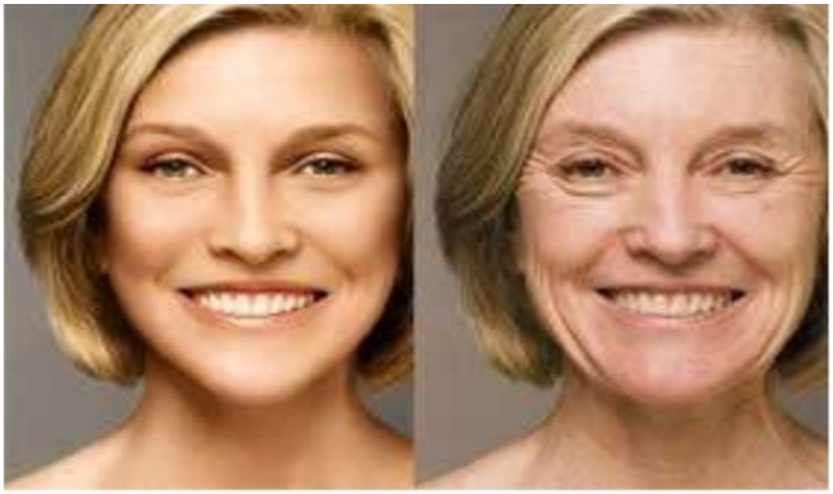


Figure 1 Retouching

1.2.2 SPLICING

Another common type of forgery that is seen around is Image Splicing. It is defined as the process where two or more than two images are combined and a single image is formed. In this, portion of one image is taken and combined onto another image. For example, we see in tabloid magazines face of some other celebrity on the body of some other person. When carried out cautiously, the boundary linking the spliced portions can be visually undetectable. This method is more aggressive than the previous one, that is, image retouching.

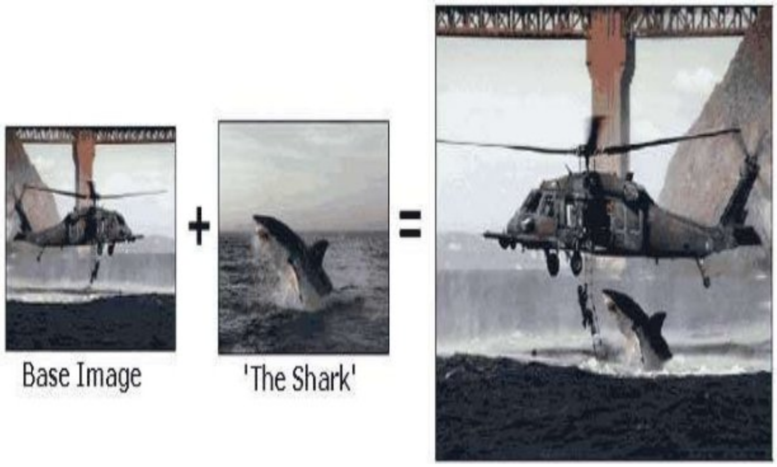


Figure-2 Splicing

1.2.3 RESAMPLING

When we combine two or images into one; or when we take a portion of the image and paste it over other region in the same image then sometimes it requires to make some modifications in the image such as rotation, resizing or stretching the portions of the image. For example if we were to combine two people from two different images, then resizing may be needed to match their relative heights. This is actually not a forgery in itself, but it is more of a technique that aid in making other types of forgeries more sophisticated and difficult to detect. For a detection method to be effective, the method has to consider these artefacts as well.

1.2.4 Copy-Move Forgery

This type of forgery is the most common in the prevalent times. Here the forger do not combine two images into one, instead a portion of an image is copied and pasted on another portion of the image to hide or conceal some object or person from the image; or to emphasize on certain aspects like a crowd, etc. It is analogous to Image Splicing since it also combines one region of an image with another region of the image, but the difference here is that both the source and target regions are from same image. To make the forgery appear invisible, forger usually applies some post processing modifications such as re-sampling or blurring. An example of such forgery is shown in Fig 3.

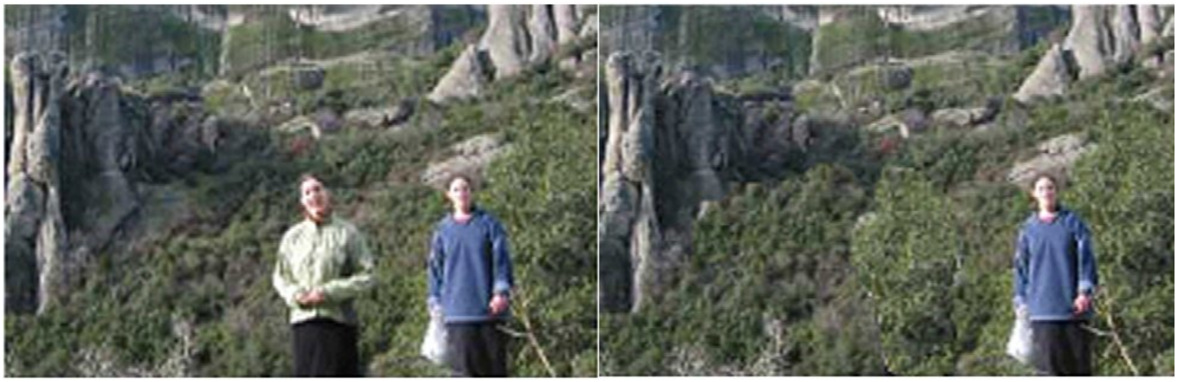


Figure 3

As we already know that making forged image is not a difficult task these days. There are numerous tools and software applications through which even a naïve person can do image forgery. This is the reason why there are a lot of forged images we see in our daily life. Among all the types of forgery [6],[19] that is done in present times, the Copy Move forgery is most commonly done. The reason for this is it is very simple to do and very difficult to detect. It’s simple because a forger requires nothing in order to do copy-move forgery. It requires no special knowledge or training, no special apparatus, and no other image to aid the forger in the forgery. All one needs is an image which is to be forged. Another reason for its prevalence is availability of many dedicated applications and tool that helps in doing such kind of forgery. It is difficult to detect since the copied region is a part of the original image it does not considerably change the statistical or texture related properties of the image. Another reason why it is difficult to detect is there is no way to know the size and location of forgery. Without knowing these it is difficult to compare each possible set of pixels with every other possible set of pictures. Copy- move forgery is known by many names such as Cloning, Region Duplication, etc.

1.3 Detection of Digital Image Forgery

Some forgeries are so sophisticated that it is impossible to detect it from naked eye. The question here arises is how does computer identify the forgery. The answer to this question lies in the fact that digital images are stored as 2D(for grayscale) or 3D(for color) matrices, where size of the matrix is the size of the image and each element of the matrix suggest the intensity values of the corresponding pixels. When a forgery happens, no matter how sophisticated it is, it is bound to bring some changes in the statistical characteristics of its digital representation. Thus, in detection methodologies we look for these statistical or other forms of anomalies so as to ascertain if the image under consideration is a forged one.

The image forgery detection approaches can be broadly categorized into two i.e., Active and Passive.

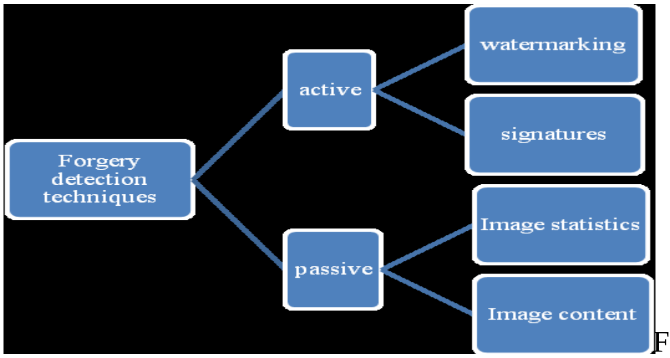


Figure 4

1.3.1 Active Approach

Here, some sort of pre-processing is done on image at the time of image acquisition. The pre-processing generally done is watermark [13] or signature embedding in the image. There are certain cameras that leave a signature on each image. Besides there are certain authorities that provide these watermark and embed it into the images. Individuals can themselves watermark their image but this would require proper knowledge and training. However, such a pre-processing would bind the scope of their application in practice. Thus it can be said that Active approach for forgery detection is not a feasible solution since it requires hard effort and knowledge. Additionally, there exist billion of images on cyberspace without any security scheme embedded in them. Considering all this, it can be said that Active approach of forgery detection is not a viable solution.

1.3.2 Passive Approach

As the name suggests, these require no pre-processing like watermarking or signature embedding unlike Active approaches. Passive techniques of forgery detection takes an image as input and attempts to find out if there are any anomaly at statistical level of the image. The logic is that while there may not be any visible sign of forgery but the forgery must have brought some changes in the image statistics. The Passive approach tries to find out these changes to take a decision about forgery. Depending on the type of changes that forgery can make in the characteristics properties of the image, passive method of detection can also be divided into sub categories. Some of the categories are named below:

1) Pixel-based techniques: The techniques that work on the pixels of image and detect statistical anomalies (if any) brought in the image's pixels.

2) Format-based : Those techniques that take advantage of the statistical correlations due to a particular compression methodology.

3) Camera-based: These are the ones that can exploit and detect the artifacts introduced at the time of capturing the image. These artifcats are caused by a camera lens, sensor, etc.

4) Physically based methods: They find out abnormality introduced in 3-D space, that is, among light, camera, physical objects, etc in the image.

5) Geometric-based: These techniques make note of the shape and sizes of the objects in the world and find out if there is any abnormality in their relative postion or orientation in the image.

Since active approach of forgery detection is seldom used for practical purposes, our focus remains on Passive approaches only. In passive approaches, some authors have divided the detection of copy move forgery into two broad segments that is Block based approach and Key point based approach.

1.3.3 Block Based Approach

In this the image is divided into blocks of fixed size so that instead of randomly matching the pixels with each other these blocks are matched against each other. The image is divided into overlapping blocks of fixed size and it is presumed that size of duplicated region would be larger than the block's size. With this, we are sure of the fact that there are many blocks within the duplicated regions. The idea of detection is to match each block against each other and find a number of connected blocks that matches with another set of blocks that have same number of blocks and are connected. When we match two blocks, it is not the exact pixel by pixel values that we are matching since it is not computationally viable. So we extract some important features [25] from each block, represent each block by these few important features thereby reducing the dimension of the block and hence make matching of the blocks easier and computationally viable.

1.3.4 Key-point based Approach

Key-points [14] are those locations within an image that carry the distinct information of the image. It may be local extrema or minima or other such factor by which a region can be clearly distinguished from its neighbouring regions.

In this detection scheme, we do not divide the complete image into fixed size, overlapping blocks, instead some distinct features or region of interests are extracted from the image itself. When these locations are found, then we extract useful features from them. In this way the comparisons reduce drastically. There are several approaches such as SIFT [20], [26] and MIFT [23] to extract the features from an image. A number of features so extracted are known as key-points of the image. The region around these key-points is matched against each other for similarity. This is different from the block based approach in the way that only the key-points (area around key-points) are matched against each other instead of matching each block of image with each other. From the neighbourhood of each key-point say 128 pixel neighbourhood some characteristics features are extracted to represent each key-point. Now instead of matching the whole neighbourhood of key-points, the characteristics feature vector with respect to each key-point is matched against each other. Naturally there would be less key-point than number of overlapping blocks in an image. Hence it computational complexity is better. However there are some inherent problems with this approach. Many a times it fails to cover the whole image and there may be significant forgeries left undetected.

1.4 Feature Extraction and Dimension Reduction

Feature [25] – A feature in general sense is a piece of information that is required to solve a computational work with reference to certain applications. In context of the images and image processing applications features can be some specific structures like points, lines, edges, blobs or ridges. There are some other statistical features such as color, texture, entropy, etc which are very important in context of the images and its applications. Other than these there are a lot of complex features which can be derived by using these basic features.

The term feature extraction can be defined as obtaining important information from a large set of data in such a manner that this important information can be used to represent the whole data without much loss of detail.

With the help of feature extraction we obtain dimension reduction. Many a times the data is hugely enormous to be processed or it is just computationally expensive to process that much amount of data. In such scenario, we look for a small subset of data which can be used for the actual data. This small subset is expected to contain all the vital information of the actual data minus the redundancy.

Dimension Reduction is the mapping of big data onto lower dimensional space such that the ambiguous and uninformative variance in the data is discarded. In other words, a subspace is extracted where informative data resides. The desired low dimensional features depend on the problem that we wish to solve. Dimension Reduction techniques can be placed in two broad categories. In one category are the methods that rely on projections and in the second category are the methods that attempt to model the manifold on which the data lives.

There exist a number of methods and techniques for feature extraction [25] and dimensionality reduction purpose. Some of the most common techniques are PCA, DCT, FMT, DWT, SIFT, Zernike Moments, Fourier Transforms, etc. Since dimensionality reduction is needed in myriads of fields and areas, the researchers are constantly working on developing new and better mechanism to extract features and reduce dimensionality. Here we briefly describe two prevalent techniques used in dimensionality reduction of images.

1.4.1 Principal Component Analysis (PCA)

PCA is one of the most widely used techniques for feature extraction and dimensionality reduction purposes. It is a way by which one can identify patterns in the data in such manner that their similarities and difference get highlighted. Finding such patterns is a difficult task where data is high dimensional. Thus it is potent tool for analyzing and evaluating data.

The other main benefit of PCA is its ability to compress the data, that is, once patterns are identified in the data, it uses these patterns to compress the data, that is, it reduces the number of dimensions without any significant loss of information.

The very first step is to calculate zero-mean data, that is, an equivalent of actual data but this equivalent will be having its mean as zero. To attain this objective we calculate the mean of actual dataset and subtract it from every data element to generate zero-mean data. Subsequently, we calculate the co-variance matrix of the data. Further we calculate the eigen vectors and eigen values from the co-variance matrix. Now these eigen vectors and eigen values are used for finding patterns in the data. The eigen vectors that corresponds to high eigen values are taken while those vectors that corresponds to low eigen values can be discarded. In this way we can reduce the dimensions with the help of PCA

1.4.2 Discrete Cosine Transform (DCT)

In many ways DCT is analogous to DFT, that is, Discrete Fourier Transform, however DCT has an advantage over DFT, that it provides better energy compaction. Energy compaction is defined as the skill to bundle the energy of spatial sequence into as few frequency coefficients as feasible. This is an important factor as it plays a major role in the image compression and other applications as well.

The Discrete Cosine Transform separates the image/picture into different parts with varying frequencies. The compression actually begins with the beginning of Quantization. During Quantization, the less significant frequencies are removed; hence it is given the term “lossy”. After this we are left with the coefficients corresponding to highly significant frequencies and these coefficients are used to compress the image. As a consequence of the fact that some frequencies are discarded, the reconstructed image contains a little distortion, but the levels of distortion’s level can be attuned at while compressing the image.

Chapter 2
Literature Review

As discussed earlier, copy move forgery is the most common forgery that is done in digital images because it is very simple to carry out and very difficult to detect. Since this forgery is so common and prevalent, there exists a lot of literature on detection of copy move forgery. Several researchers have shown interest on this topic and thus there exist a number of different purposed algorithms for the detection of such forgery.

Copy move forgery detection comes under blind image forensics or passive image forensics in which no security mechanism like watermarking [12] or signature [17] is embedded in the image. All one is provided with is an image and the goal is to find out if such forgery has been done in the image. In Copy move forgery, a portion of the image is copied and pasted onto another portion to hide or conceal some information like person or object; or to highlight or exaggerate some areas like crowd, etc. Since both the pasted region belongs to the same image, the statistical properties of the image like noise, color intensity levels, contrast, texture, etc do not change arbitrarily and thus the duplicated region mixes in the very easily. The same fact, that both the regions, that is, the copied and the pasted portion lie somewhere in the same image, is utilized by several algorithms.

2.1 Watermarking and Signature based Methods

The technique of watermarking [12],[13],[16] and signature embedding [17],[18] is probably one of the oldest techniques to detect the authenticity and integrity of the data. Both of these techniques are general techniques and can be used for any kind of data such as numerical, image, audio, etc.

In case of image authentication, the watermark [13] is embedded in the least significant bits of the image. The disadvantage here is that, it will distort the image a little bit. Contrastingly, in signature based approach the hash code of the whole image is calculated and inserted in the image itself. The embedding of watermark or signature is carried out at the time of generation of the image or typically before transmitting it over a network. At the time of detection of an image’s authenticity and integrity, the hash code is calculated again and verified with the original hash code. The plain and simple logic behind the method is that if the image is forged, it will affect and alter its watermark or signature. Though some methods [12] are quite robust to several manipulations but overall this approach still has some major disadvantages. The disadvantage is that it requires pre-processing of the image and only those images can be verified in which such a security mechanism is pre-embedded and there are a lot of digital images without any such embedding. Additionally, it requires knowledge and training to carry out this method.

One of the direct and simplest approaches to detect such forgery is proposed in [1] where the authors performed an exhaustive comparison that is they compared every cyclic shifted version of the image with the image itself. However the computational complexity of the process is extremely high that is $(MN)^2$ and thus is not feasible. Same authors proposed another method called Autocorrelation [1], where they used Fourier transform. The logic they used that the there will be peaks in the Autocorrelation corresponding to the source and target areas where forgery is done but this method needed the duplicated region to be sufficiently large to be detectable. In [4], the authors proposed a method Block Artifact Grid Extraction for JPEG compressed images but it can detect forgeries up till a level. When compression increases significantly, the method failed.

Most of the copy move forgery detection algorithms makes use of one of the two methods that is either key-point [14] based method (e.g.,[20], [23]) or block based method (e.g.,[1], [2], [3], [21]). In both the cases the pre-processing of the input image is possible for example; many methods first convert the image into its greyscale equivalent and work on it. The aim of this conversion is reduction of computational complexity.

2.2 Block Based Methods

Here the image is divided into overlapping blocks and these blocks are matched against each other. The size of the block must be smaller than size of duplicated region. This means that there will be a number of blocks in the duplicated region. Now each block is matched with every other block. Naturally the blocks that are from the region which is copied and from the region where the portion is pasted will be same or extremely similar. If there are a number of blocks which are same and connected as well, then a decision of forgery can be taken. When we compare the blocks, it would be computationally expensive if we compare them as it is. For example let’s say block size, b is 16 x 16 which means each block will have 16 x 16 = 256 elements. Clearly comparing these 256 elements each time a block is compared with other block is computationally expensive. Hence, researchers tried to extract a few essential features from each, thereby reducing the dimensionality. These blocks with much less number of elements, usually in form of row vector are used for comparison. The logic is very simple, that is, if two blocks are similar, their extracted feature are ought to be similar. However, it was more difficult than it seems, the problem is that some forger used techniques like noise addition, compression, rotation, scaling, blurring, etc to make the duplicated region undetectable. Thus researchers focused on this aspect to find an appropriate and robust scheme for feature extraction and dimensionality reduction.

Authors in [1] also proposed to use DCT coefficients, that is, with respect to each block; they computed the DCT coefficients and used these coefficients to represent a block. The extracted coefficients are quantized by a factor Q which also plays a role in detection rate and degree of robustness. Next, instead of matching actual pixel values, they matched the extracted DCT values. The extracted features are robust to some modification including low pass filtering and compression. The inherent benefit in using DCT coefficients [22] is that the energy gets concentrated on first few components while the remaining coefficients are relatively very small. Hence, the operations like noise addition, compression and retouching which bring changes in high frequencies should not be influencing or altering these first few components. The only robustness that they proved is that their method was robust to retouching operations. However when it comes to detection of duplicated regions with additive noise, their method was sensitive. The method [1] proposed was not very robust to several kinds of modifications but it definitely set a benchmark in CMFD. Later, in [7], Huang et al. reduced the feature vector in dimensionality by truncating some of the DCT coefficients and improved the performance. However their method did not take multiple copy move forgery into account.

Popescu et al [2] proposed to use PCA [5] to extract features and reduce dimensionality. They chose PCA because of the reason that it is more invariant to compression and additive noise. They converted each block into row vector and calculated the covariance matrix of the vector. Next they calculated eigen values and eigen vectors of the covariance matrix. Now to reduce dimensionality they keep the high eigen values and truncated the lower ones. They projected the image data onto these selected eigen vector to obtain new representations. Their method was robust to the additive noise and compression [8] but the accuracy of detection was low and if there is any re-sampling like rotation or scaling, the derived eigen values get affected. They showed that their technique was invariant to compression till jpeg quality level 50 and can work under additive noise with SNR 36db and 29db.

Li et al [9] used the similar approach differently. They did not take the approach where they would divide the image into blocks and then extract features from them; instead first they applied discrete wavelet transform (DWT) onto the image and divided it into four sub bands. Now, knowing the truth that low frequency sub band would be having most of the energy concentrated it, they chose only low frequency sub and divided it into overlapping blocks. This way, they also reduced the number of blocks and made the process fast. Next, to extract features from these blocks they chose singular value decomposition (SVD) which is directly related to PCA. Thus it is expected that this method would give similar result as in [2] though in the paper they proved the robustness of their method only up to quality level 70.

Not similar to the conventional methods that were being used at that time, Luo et al [10] proposed to use an alternative method of feature extraction. They extracted the features based on the color information of the blocks. One set of extracted features contained the R, G, B components while to derive the other set, they divided the blocks in two parts, in four directions. They extracted the second feature set in the form of ratio that is, they divided the overall intensity values of one part with the overall intensity value of the whole block to get the ratio. In the similar fashion, they calculated the ratio with respect to each part and formed a feature vector. It was claimed them that the method is highly robust to Compression to the tune of Q_{30}; to noise up to SNR 24 db and also to Gaussian smoothening with sigma = 1 and 5x5 window.

More recently Bayram et al [11] used Fourier Mellin Transform to extract the features. With respect to each block, they calculated the Fourier transform and used the log polar coordinates to re-sample the transformed block. Next, the log polar values so generated are projected onto 1-D and the representations so obtained are used as features. Wu et al [12] had previously used this method in context of watermarking and shown its robustness to various modifications like rotation, translation and scaling. In [12], it was shown that method is invariant to compression till Q_{20} and to rotation till 10^{o} and 10 % scaling.

2.2 Matching the Blocks

When we get a reduced and robust representation of each block, the further step is to match each block with every other block. For this purpose, most of the methods described above e.g., [1], [2], [9], [10], [11] uses lexicographic sorting which is similar to the way dictionary is sorted. First the new representation of each block is vectorized and stored in a matrix called feature vector matrix (say A) of size NB x F (i.e., no of blocks x no of features). In accordance with this, the feature vector matrix of [1], [2], [10], [11] would contain (M – b+1) (N - b + 1) rows and F columns whereas in [9], the feature vector matrix would be having (M/2 – b+1) (N/2 - b + 1) rows since only a sub band of the DWT frequency band is used for generating blocks. Each row in this matrix corresponds to the feature vector of a block and the column contains their respective feature vectors. In lexicographic sorting, the rows are sorted in ascending order in such a way that first column of each row is matched against each other and sorted in ascending order. If two rows have the same element in their first columns, they are sorted as per the elements in their second column and so on. In this way, the similar or same rows come adjacent to each other. The running time of lexicographic sorting depends on number of rows and the number of features in each row but since number of features is usually a small number, it is not considered that important. Thus lexicographic sorting requires Rlog(R) steps where R is the number of rows in feature vector matrix. Clearly, the method in [9] has an advantage as it has reduced its number of blocks to 1/4, hence would take 4 times lesser steps. The authors of [11] also proposed another method for matching process where they did not compare the features. Instead they calculated the hashes of the feature vectors and compared the hashes with each other. The method is known as “Counting Bloom Filter” in which they created an array K of size k. Initially, all values of K are initialized to zero. After calculating the hash value of a block’s feature vector, array K is incremented by 1 (at the index = hash value), that is, K (hash value) = K (hash value) +1. The logic given is this, since two same blocks will generate same hash value, the K array will be incremented twice (at the index = hash value of same blocks). Hence, the same blocks can be found out by checking which blocks set the hash value two or more than two. Since hash values can be calculated at the feature extraction step only, the only running time difference that it makes is to find out the corresponding blocks due to which hash array has elements with value 2 or more. Clearly this additional time depends on k which ought to be M x N.

2.3 Performance Evaluation

To see a comparison between existing methods, the findings of survey [3] is used where they compared three feature extraction techniques that is, PCA, DCT and FMT [24]. The block size is kept 16 and it is assumed that size of duplicated region would be 32 x 32 at least. Thus there ought to be (32-16+1) (32-16+1) = 289 duplicated blocks. So they kept their threshold as 150. Next they performed two sets of experiments, one without any modification and one where duplicated region is modified. It was found that all three techniques gave accurate result when there was no modification. In second scenario when modifications like scaling and rotation is applied on the duplicated region, the performance of the three methods is illustrated in the Table 1.

Table 1 : Performance Results [3]

Manipulation Type	FMT[24]	DCT[1]]	PCA[2]
JPEG	20	40	50
Rotation	10^o	5^o	0^o
Scaling	10%	10%	0%

Seeing the performance result [3] in Table 1, it can be understood that out of the existing methods, DCT feature extraction method gives a better trade-off between various types of manipulations.

Chapter 3

Discrete Cosine Transform

3.1 Introduction

The primary application of DCT has always been data compression and it is widely used in jpeg compressions, compression of MP3 and for solving partial differential equations.

In many ways DCT is analogous to DFT, that is, Discrete Fourier Transform, however DCT has an advantage over DFT, that it provides better energy compaction. Energy compaction is defined as the skill to bundle the energy of spatial sequence into as few frequency coefficients as feasible. This is an important factor as it plays a major role in the image compression and other applications as well.

The Discrete Cosine Transform separates the image/picture into different parts with varying frequencies. The compression actually begins with the beginning of Quantization. During Quantization, the less significant frequencies are removed; hence it is given the term "lossy". After this we are left with the coefficients corresponding to highly significant frequencies and these coefficients are used to compress the image. As a consequence of the fact that some frequencies are discarded, the reconstructed image contains a little distortion, but the levels of distortion can be attuned while compressing the image.

3.2 The Method

a) The input picture is divided in overlapping blocks of size, say 8 x 8 pixels.
b) DCT is applied on each and every block to extract DCT coefficients.
c) Quantization is done on each block's coefficients in order to obtain compression. After quantization the coefficients corresponding to higher frequency tends to become negligible.

d) The matrix containing compressed or reduced representation of blocks can be stored in a significantly reduced space.

e) The original picture can be recreated using decompression, which is a process that makes use of the inverse of DCT.

3.3 The Equation

The following Equation (eq 1) calculates the i^{th}, j^{th} entry of the DCT of an image p(x,y).

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N} \right] \cos\left[\frac{(2y+1)j\pi}{2N} \right] \tag{1}$$

$$C(u) = \left\{ \begin{array}{ll} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{array} \right\} \tag{2}$$

where p(x,y) → x,y^{th} element of the image which corresponds to matrix p.

N→ the block's size on which DCT is applied on

For an 8 x 8 standard block, the algorithm makes use of, N = 8 and x and y range from 0 to 7.

Hence D(i,j) corresponds to equation(3).

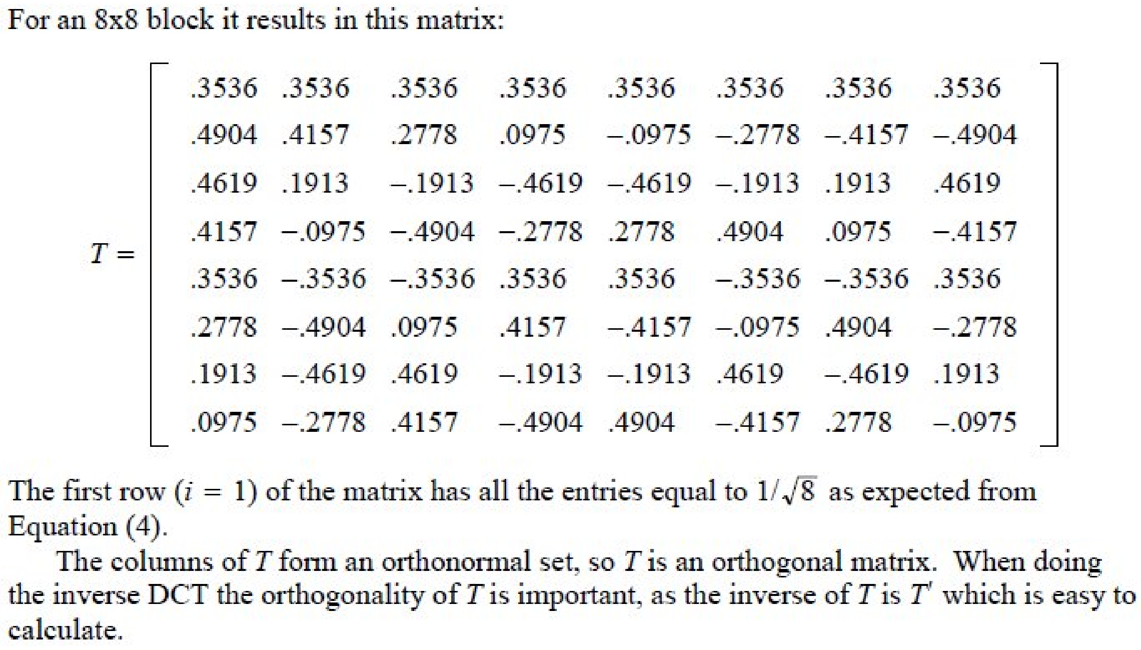
$$D(i,j) = \frac{1}{4} C(i)C(j) \sum_{x=0}^{7} \sum_{y=0}^{7} p(x,y) \cos\left[\frac{(2x+1)i\pi}{16} \right] \cos\left[\frac{(2y+1)j\pi}{16} \right] \tag{3}$$

Because of the fact that the DCT makes use of cosines functions, the resultant matrix depends on various frequencies that are horizontal, vertical and diagonal frequencies. Therefore a black image having numerous variations in frequency results in an array that looks quite arbitrary while a single color picture generates an output array in which first elements will be having large values and other elements have zero or small values.

3.4 The Discrete Cosine Transform Matrix

To transform equation (1) into a matrix form, we can use the equation (4)

$$T_{i,j} = \left\{ \begin{array}{ll} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2j+1)i\pi}{2N} \right] & \text{if } i > 0 \end{array} \right\} \qquad 4$$



For an 8x8 block it results in this matrix:

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

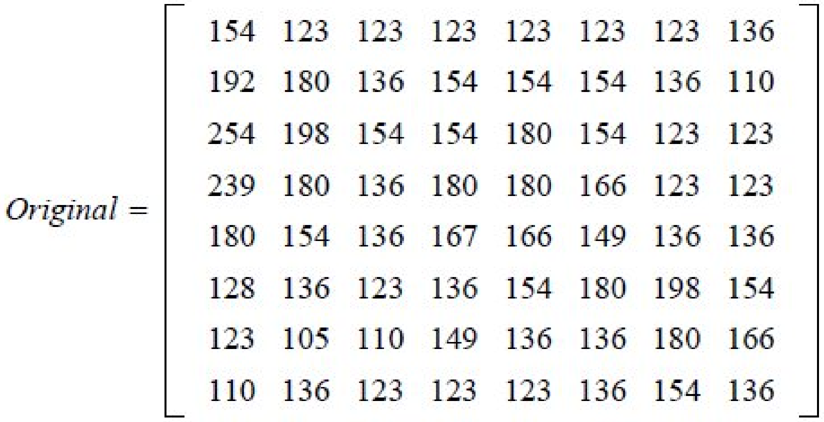
The first row ($i = 1$) of the matrix has all the entries equal to $1/\sqrt{8}$ as expected from Equation (4).

The columns of T form an orthonormal set, so T is an orthogonal matrix. When doing the inverse DCT the orthogonality of T is important, as the inverse of T is T' which is easy to calculate.

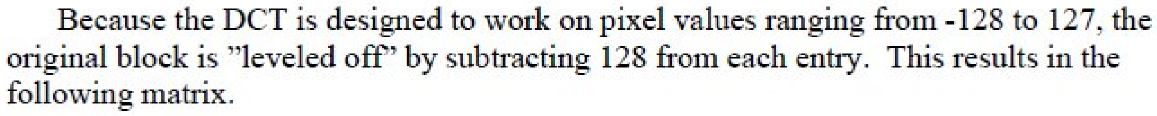
3.5 *Applying the Discrete Cosine Transform on an 8x8 block*

Before starting we should make note that in a greyscale picture the pixel values lie between 0 to 255. In that 255 symbolize plain white and 0 represents plain black. Thus these 256 shades of grey accurately represent a photo or illustrations.

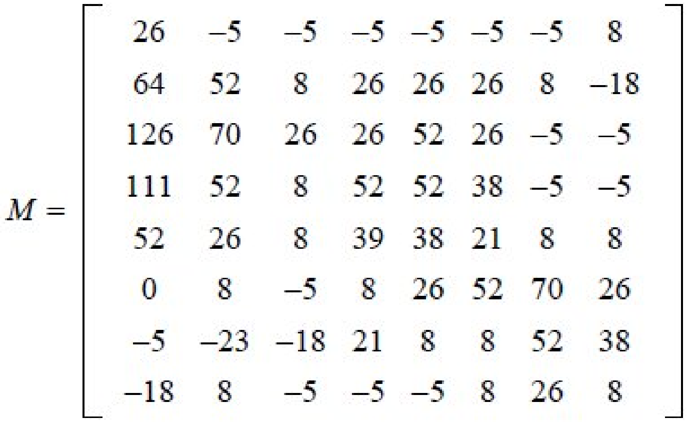
Since an image may contain a number of overlapping blocks, below is the description of how it works on a single 8x8 matrix. It is applied on the other blocks in a similar manner.



$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$



Because the DCT is designed to work on pixel values ranging from -128 to 127, the original block is ”leveled off” by subtracting 128 from each entry. This results in the following matrix.



$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

We can now perform DCT by multiplying matrices

$$D = TMT` \qquad (5)$$

The above equation (5) multiplies the matrix M with matrix T and its transpose T’. When M is multiplied by matrix T on the left, it transforms the rows of M and when the product is multiplied by T’ on the right, it transforms the columns as well.

$$D = \begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}$$

The derived matrix contains 64 DCT coefficients. The coefficient Coo corresponds to lowest frequency in the actual block. When we traverse in the D matrix in zig-zag order, the coefficients corresponds to higher frequencies of the block, thus C_{77} corresponds to highest frequency. It should be noted here that human eye is most sensitive to low frequency.

3.6 Quantization

Now the derived block containing DCT coefficients can be compressed by Quantization. One of the finest features in the DCT is that we can obtain varying level of compression and image quality by choosing from a range of quantization matrix. The Q_1 quantization matrix results in highest level of compression but poorest quality of reconstructed image whereas Q_{100} gives the lowest compression but high image quality.

These quantization matrixes are derived from a standard quantization matrix and the standard quantization matrix is a result of various subjective experiments on human visual system. To generate varying quantization matrix, we use the scalar multiple of the standard quantization matrix. E.g. to get a quantization matrix of quality level more than 50, we take a product of “standard Q matrix” and “(100 – qlty_lvl)/50”. Similarly to get a quantization matrix of lesser quality level, product of standard quantization matrix and 50/qlty-lvl is used.

Quantization is finally attained by dividing each component of D by correlated element of Q and approximating the values to closest digit.

$$C_{i,j} = round\left(\frac{D_{i,j}}{Q_{i,j}}\right) \tag*{6}$$

To illustrate, Q50 is used here

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3.7 Nature of Discrete Cosine Transform

The inherent nature of DCT is such that when we apply DCT on image the energy focuses only on the coefficients which correspond to low frequency, which means that all elements are not equally important but only the low frequency coefficients play a major role.

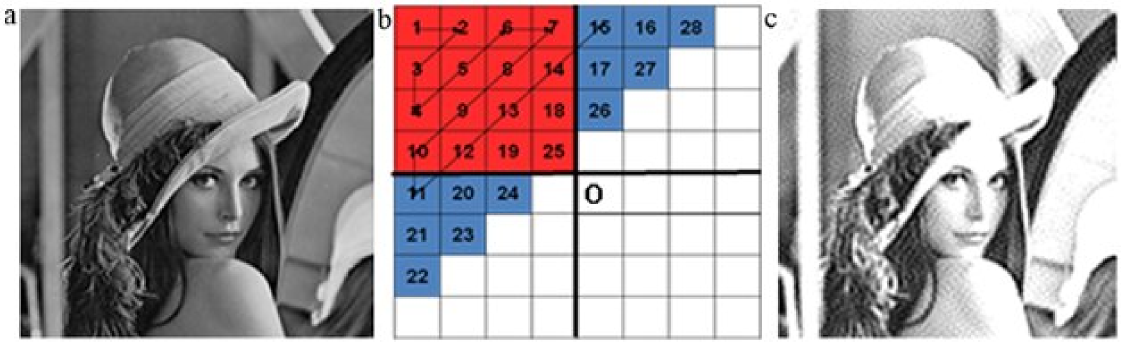


Fig. (a) the Lena image (b) Zigzag order scanning (c) the reconstruction image of Lena by using 1/4 DCT coefficients.

To prove the above statement, we made an illustration by using the Lena image of size 256x256 pixels. Fig (a) is the Lena image. We applied the discrete cosine transform to Fig (a) and extract a matrix of DCT coefficients. Subsequently, low frequency DCT coefficients are extracted in zig-zag sequence as shown in Fig (b) where red area is the part that corresponds to low frequency, which accounts to 1/4 of the entire DCT coefficients. Now with only these DCT coefficients we reconstruct the image using inverse of discrete cosine transform and generate Fig(c).

Thus, through the analysis of the given image we came to know that with the help of only 1/4th of DCT coefficients, one can retrieve the information without any significant loss.

Chapter 4

The Proposed Method

We propose an efficient and effective methodology to detect the cloning forgery within same image using the block based approach. The following figure illustrates the architecture of our approach.

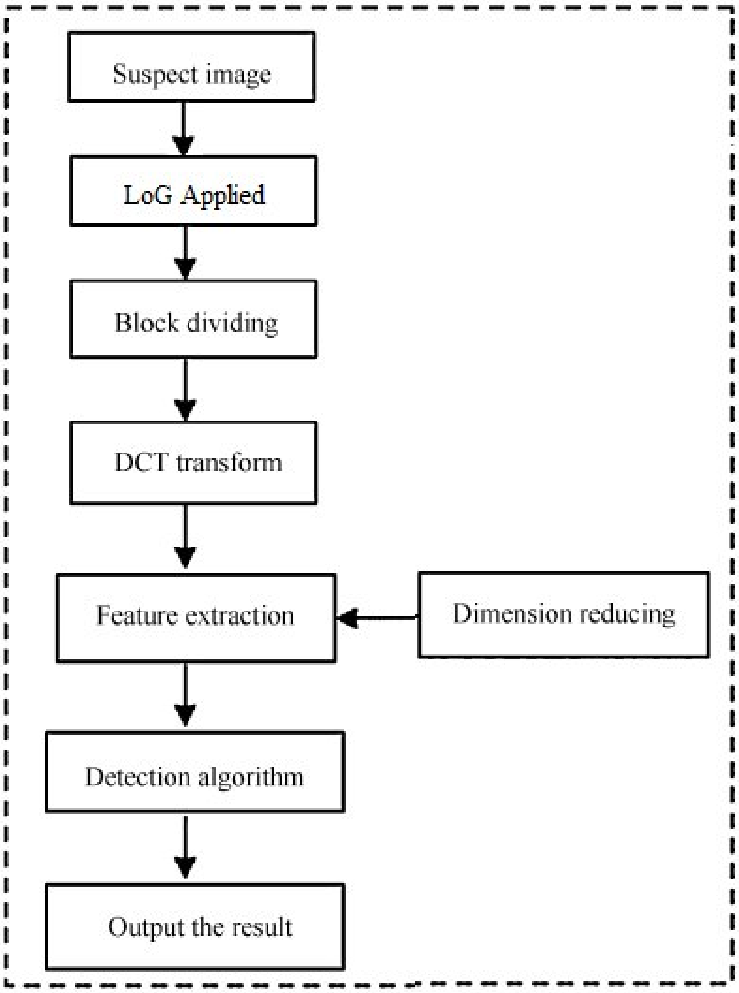


Fig. 1 Architecture of the detection algorithm.

Normally saying, a normal image is not likely to have two large similar regions (except in the case of image of nature where there is large similar areas such as blue sky or green grass). The task of detecting cloned forgery is to detect exactly same regions in the image. As we do not know the shape and size of the duplicated/cloned region, it would be computationally very expensive and impractical to compare pixel by pixel every possible pair. Hence, the more practical and feasible approach is to divide our input image into blocks. The block size is willingly kept much smaller than the cloned region so that cloned region contains a number of blocks which are duplicated. Now to further reduce the computation costs, a good feature extraction technique needs to be adopted through which the block can be represented appropriately. Thus with a precise and robust extraction technique, the dimensions of each block is reduced significantly thereby reducing the computational complexity. In our method we are using DCT coefficients to represent each block.

Steps

Going by the discussion above, the framework is given as below:

(1) Preprocessing of Input Image

Applying LoG, Rescaling, and RGB to grey conversion

(2) Dividing the image into Blocks

(3) DCT Transform of each block

(4) Feature extraction and Dimension Reduction

(5) Lexicographic Sorting of Feature Vector Matrix

(6) Detection and Decision

Further we shall see each step explained in detail

4.1 Pre-processing of Input Image

- The very first step that we do is take the input image as it is and create an ideal LoG filter and apply the filter to the input image so as to highlight the regions of interest such as edges, blobs, ridges, texture information, etc. These regions are important as they can provide an edge to our algorithm because of the fact that these do not change significantly when operations like rotation, scaling, adding noise, etc are applied on the image to make the forgery sophisticated and difficult to detect. Thus, extracting these features play a vital role in making our approach more robust to various kinds of manipulations. The feature Image so obtained is combined (subtracted, however the feature image is scaled first) with the original image to obtain an image where the above mentioned region of interest are mildly highlighted.
- The second sub-step is to convert the image so obtained into its grayscale equivalent. For this, one can either utilize the formula that is "I = 0.228R + 0.587G + 0.114B" or alternatively matlab has a dedicated function for this purpose.
- The third sub-step is to resize the image into the dimensions of our choice. Conventionally we chose to resize it into a square matrix. However the scale of resizing depends upon various factors such as processing and memory capabilities of the system. The algorithm works better without any downscaling.

4.2 Dividing the image into Blocks

The image so obtained after step 1 is divided into overlapping blocks of size *bxb* pixels in such a way that it replicates a sliding window of size *bxb* pixel. At each iteration, the window slides to its right by 1 pixel and when it cannot move any further to its right, the window slides down by 1pixel and begin from leftmost end. Thus there is a difference of only one row or column between two adjacent blocks.

Suppose we have an image of size M x N and we divide it into overlapping blocks of size b. Then number of blocks is NB = (M – b + 1) x (N – b + 1)

Bij is used to denote each block, where i and j indicates the starting location of B_{ij}.

$$B_{ij}(x, y) = f(x + j, y + i),$$

where $x,\ y \in \{0,\ \ldots,\ B-1\}$, $i \in \{1,\ \ldots,\ M-B+1\}$, and $j \in \{1,\ \ldots,\ N-B+1\}$

Concurrently we fill a matrix B of size (NB x 2) where row number indicates the number of block and it corresponding column elements indicates the starting position of the block that is, i and j respectively. This matrix shall come into use in the later stages after lexicographic sorting of the feature vector matrix, when it will be needed to find out the starting locations of the similar blocks in order to take a decision about the forgery.

4.3 DCT Transform of each block

Onto every block, DCT is applied so as to generate the quantized discrete coefficients transform matrix of size b x b, which is same as that of the block's size. Assuming our block size to be 8x8, the coefficients matrix that we get is also of the same size and can be used to represent the corresponding block. The nature of DCT is such that its energy gets focused on lower frequency coefficients which mean that each component is not equally significant and the coefficients with low frequency play relatively vital part. This is illustrated in Chapter 2 where DCT is explained.

4.4 Feature extraction and Dimension Reduction

Through analysis of the DCT and the illustration in Chapter 2, we know that if DCT is applied on an image, one is able to utilize only a fraction of the energy for representing the actual picture without loss of much significant data. This motivated us to extract only a few features from DCT coefficients matrix to represent each block thereby reducing the dimension of each block and hence the computational complexity.

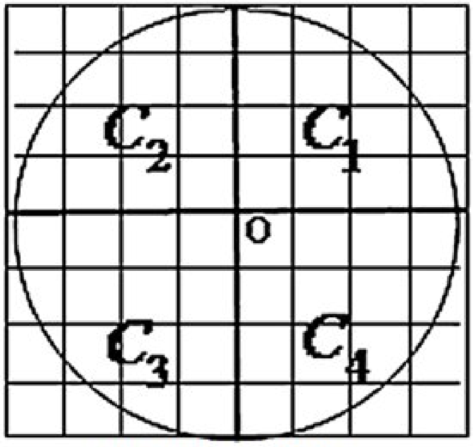


Fig: Feature extraction

The coefficients matrix is divided into four equal parts using a circle, as shown in the figure, inside coefficients matrix and dividing the circle in 4 divisions as shown in the figure. Our intention to divide the coefficients matrix into four parts is to extract four features that is, v_1, v_2, v_3 and v_4 from each part.

$$c\_area = pi * r^2$$

$$m\_area = 4r^2$$

where r denotes the radius of the circlular block, c_area denotes the circular block's area, m_area denotes the square matrix's area. Thus

$$p\_ratio = c\_area/m\_area$$

Based on the equation given above, p_ratio that we get is 0.79, which means circle block represents 79% of the elements of the DCT coefficients matrix and discard only a few. Moreover as we know the DCT's nature is s.t. energy gets focused on low frequencies. Hence efficiency of the detection is not affected when we do not use square block and use a circle block instead; on the other hand, it significantly reduces the computational complexity.

Going by the analysis given above, in order to extract the features, we designate v_1, v_2, v_3 and v_4 as the feature of quarters C_1, C_2, C_3, and C_4 in that order. We get v_i (i =1, 2, 3, 4) via Eq. given below

$$v_i = \frac{\sum f(x,y)}{c\_area_i}, \quad (f(x,y) \in c\_area_i, \; i = 1,2,3,4)$$

where v_i is the average of the coefficients value, with respect to each C_i. In view of the fact that each C_i contains divergent DCT coefficients, each v_i can as well be seen as quantized by c_area_i. Afterwards, we obtain a feature vector containing four features, which are pooled to form a feature vector of size 1 x 4, denoted as:

$$V = [v_1, v_2, v_3, v_4]$$

In this way a 1 x 4 feature vector represents an 8 x 8 matrix, weighed against with [1,2], which is a 1 x 64, 1 x 32 feature vector, our methods generates a representation with lower dimensions.

4.5 *Lexicographic Sorting of Feature Vector Matrix*

A feature vector matrix of size *NB* x *4* (denoted as FVM or F) is constructed in which each row F_i correspond to a block B_i and each row contains 4 feature vectors corresponding to the block B_i. Here *NB* is the no. of blocks i.e. (M – b + 1)(N – b + 1)

The FVM thus contains the feature vectors of each block and is illustrated in the following fig.

$$\text{FVM or F} = \begin{bmatrix} V_1 \\ \vdots \\ V_{(M-B+1)(N-B+1)} \end{bmatrix}$$

The matrix FVM is lexicographically sorted. The vectors V_i corresponding to similar blocks will appear adjacent to each other in the sorted matrix since similar blocks will have similar feature vectors.

4.6 Detection and Decision

The FVM obtained in the previous step contain the vectors of the blocks in such a way that the more similar two blocks are, nearer would be there feature vector in the FVM. Thus the vectors of same blocks would appear adjacent to each other in the FVM.

However this is not enough since in an image there can be many blocks which are similar to each other. Secondly the blocks adjacent to each other in an image would naturally have similar feature vectors since only a row or column is different in two adjacent blocks. Thirdly in natural images of blue sky or desert or grass, the blocks tend to appear similar over a large area. Because we deliberately kept our block size smaller than any viable duplicated section, we make out that any duplicated region will have a number of blocks in it. The judgment about tampering can only be taken, if there exist a sufficient number of similar blocks that are equidistant from each other and that are connected as well. In other words each of these blocks are equal distant apart from there source. This can be evidently understood from fig. below.

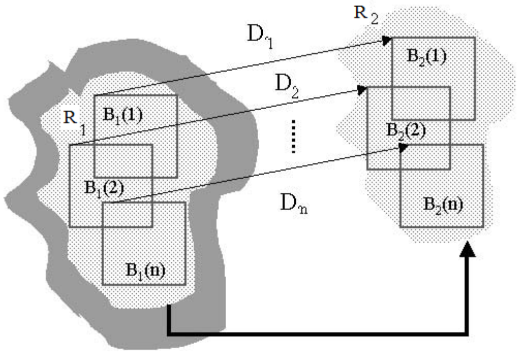


Fig : A diagram which shows the copied regions and alike blocks in it.

From the diagram, it is clear that in case of cloned region the distance between similar blocks is identical and the blocks are linked to each other.

Keeping the above factors in mind we need to have certain thresholds based on the type of image we are looking onto. These thresholds would help us determine if a match between the two feature vectors is an ideal match i.e.

i> The similarity threshold – which determines whether two feature vectors of blocks are similar to each other. In other words the normalized difference between two blocks should be less than similarity threshold.

ii> The distance threshold – Since many adjacent blocks would have similar feature vectors, they must not be included in our final result. Hence two similar blocks should be a certain distance apart in order.

iii> The connected block threshold – which determines the minimum number of similar blocks that must be connected with each other.

Suppose there are two similar feature vectors F_i and Fj of the blocks B_i and B_j. Because of the fact that the matrix FVM is lexicographically sorted, hence the feature vectors F_i and F_j will be adjacent/near to each other. For calculating the distance between two similar blocks, Euclidean distance is used. For example the block B_i and B_j have their starting position as x_i,y_i and x_j,y_j. Therefore distance between them is

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

But as we know two neighboring blocks can also be similar, hence this distance should be greater that a specified threshold to remove the neighboring blocks from detection.

Now, we create a distance vector D in which will contain all zero initially and will be incremented by 1, each time if same distance between two rows is found. $D(d_x,d_y) = D(d_x,d_y)+1$ where $d_x = |x_i – x_j|$ and $d_y = | y_i – y_j |$

If any of the element of the array $D(d_x, d_y)$ is found to be greater than a pre specified threshold value, we can take the decision about the forgery.

Chapter 5
Results and Analysis

The method proposed in the previous section is implemented in the Matlab software. It didn't require any special pre processing other than enhancing the edges and other details with the help of LoG filter described in previous sections. Several experiments are conducted on different images to gauze the efficacy of our approach. For the experiments, some were self doctored and a part of the image is copied on the other part to create a copy-move forgery. Besides, some tampered images made by researchers are also taken from the internet.

The experiments are designed carefully so that we can ascertain the behaviour of our proposed method. For this we take similar images of varying resolutions and sizes. Other than that we also used noisy and jpeg compressed image to gauze the detection ability and behaviour of the method. To measure the performance we observed four measures that are

5.1 Performance Measures

1. Accuracy of the detection: This means how precisely the algorithm is able to detect the forged region. One way of doing is to do the forgery by ourselves so that we know where exactly the duplicated region and source region lies in the image.
2. Effectiveness: This tells us about how well our method can detect when the forgeries are quite sophisticated or done over only a small portion image. To observe this, we used different images in which the duplicated regions were of varying sizes. In some there were multiple regions which were copied and pasted onto other region in the image.
3. Efficiency: Only detecting a forgery is not enough. A good method should be able to detect the forgery in reasonable amount of time. To measure this we used the timing clocks in our algorithm so that we can know the exact amount of time our method is taking.
4. Robustness: This performance measure tells us about the robustness or tolerability of a method. Sometimes forger uses a variety of techniques to make the detection

difficult. These techniques may be addition of noise or blur, compression, scaling, rotation and other similar methods. A robust method is one which is invariant to these.

5.2 The Parameters

We did our experiments on 32 bit AMD Phenom-II 3.20 GHz processor with 2GB memory. The code is executed using Matlab. For the sake of simplicity the images are converted to greyscale, since it is easier to handle.

Before beginning with detection process, one needs to specify a number of parameters. The very first being the image size itself. The selection of image size depends majorly on the computational and memory capacities of the system on which detection is to be carried out. For our experiments, we took two sets of images. In the first set, we resized the image to a size of 128 x 128 pixels and in the second we choose a size of 256 x 256 pixels. The second important parameter is the block size which determines the overall number of blocks in picture and hence size of feature matrix. It should be noted here that the size of block must be less that size of duplicated region. We choose varying block size ranging from 8 to 32 for our experiments depending on the size of image and size of duplicated region. The third parameter is the size of the forgery that is the approximate size of the duplicated portion. For most of the experiment we preferred it to be 32 x 32 pixels or 64 x64 pixels depending on the size of the image.

Other than the parameters described above there are certain thresholds that need to be specified. The first one is the N_D that is the minimum distance that should be there between two similar blocks so as to remove similar neighbouring blocks. For practical purposes we take it to be 30 pixels when block is 8 pixels and 60 pixels when block size is 32 pixels. The second threshold is S_{msr} that is the measure of similarity between two blocks. Normally the root mean square or the difference two blocks vectors is calculated and matched against $S_{msr.}$ For two blocks to be similar it should be less than $S_{msr.}$ The next threshold is shift threshold. When feature vector matrix is lexicographically sorted, vector corresponding to similar features come near to each other. The shift threshold gives us a window size. For example if shift threshold is 2, then in the FVM we compare each block’s vector to its 2 nearby vectors. In our experiment we mostly kept it 1. The last important threshold is N_c which tells us that a minimum of N_c similar and connected blocks must be there in order to take a decision about the forgery.

5.3 Experimental Results

Several images are taken from various sources, some of them are self forged to accurately calculate the detection effectiveness and efficiency. Some other images are also taken from the internet on which other researchers have also worked so that comparison can be done between both. On each image, detection is carried out several times by changing its parameters such as image size and block size.

5.3.1 Efficiency

In fig 5.1 and 5.2 two images are shown for illustration purpose. These are the same images used by previous researchers [15]. The images are resized and detection is carried out by varying the block size as can be seen in the figure.

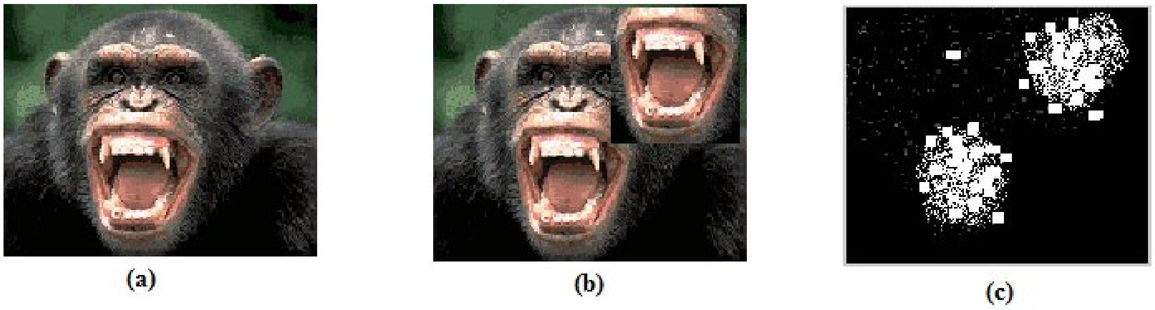


Fig- 5.1 : (a) The original image of gorilla taken from internet; (b) The portion of its mouth is taken and pasted onto upper right corner of the image; (c) The output image after applying the detection scheme.

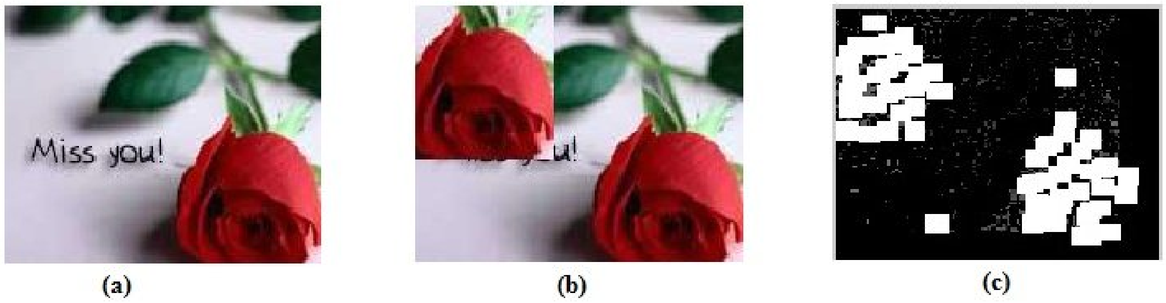


Fig- 5.2: (a) Novel image taken from [15]; (b) Tampered one, where bottom right portion copied and pasted onto upper right corner of the image; (c) The output image after applying the detection scheme

5.3.2 Effectiveness

In the second set of our experiments, we tested our method on some images from Casia database [27] and some images which were tested by other authors. In this set of experiments, to test the effectiveness of our method and its robustness towards additive Gaussian noise, we first tested the method on a forged image without any noise. Our method was able to accurately detect the duplicated region in the region duplication map. Subsequently we added Gaussian noise (with 0.01 variance) to check the robustness of the method. The proposed method was able to detect duplicated region although detection rate decreased and accuracy suffered a little bit. Further we raised the variance of noise to 0.05 to see if it could detect the regions. The method could detect the duplicated region but not in all cases and the accuracy decreased significantly. The illustration cab be seen in Fig 5.3

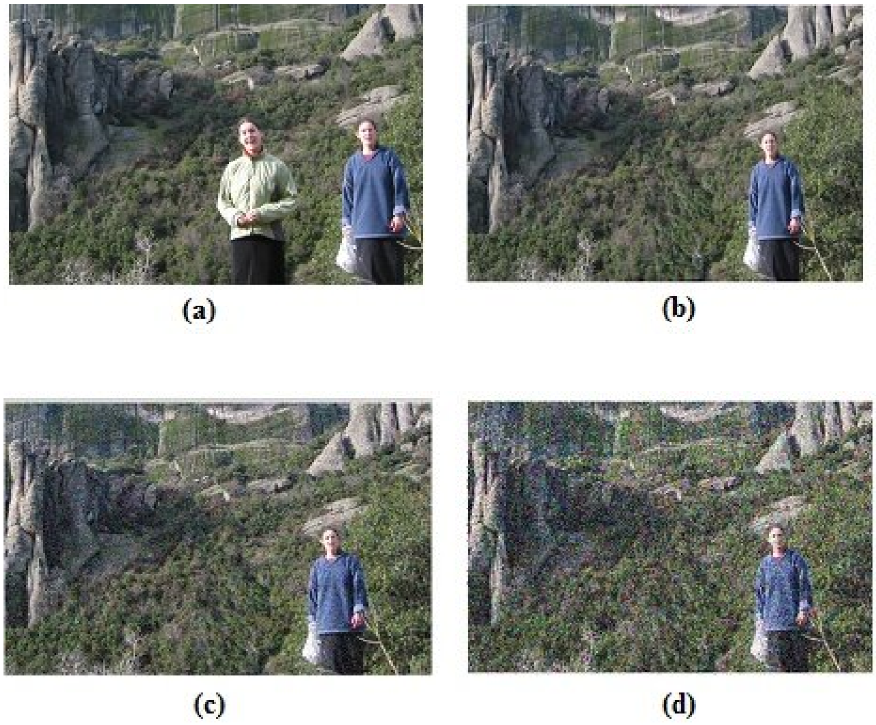


Figure 5.3: (a)The original image; (b)forged image; (c) forged image with additive Gaussian noise (variance =0.01); (d) forged image with additive Gaussian noise (variance = .05)

The results of the detection are illustrated in Figure 5.4 where we can see the detection was accurate (figure 5.4(a)) in the case where no modification was there that is the figure 5.3(b). When a small amount of noise was added to the forged image i.e., Figure 5.3(c), the method was robust enough to detect the duplicated region accurately (refer to Figure 5.4(b)). When we further increased the noise levels (Figure 5.3(d)), then detection rate suffered and detection accuracy was compromised as well as can be seen in Figure 5.4 (c)

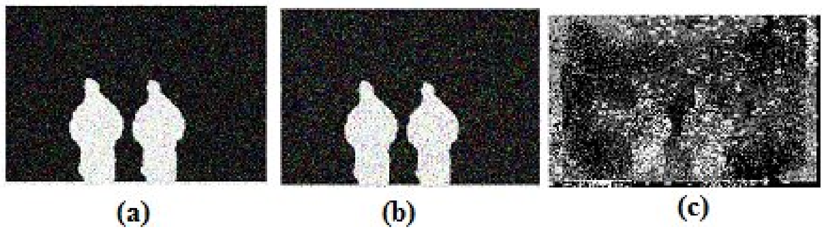


Figure 5.4 : Detection map (a) detection results of non manipulated forged image; (b) detectcion results when a Gaussian noise of low density is applied in the forged image; (c) detection result when high density noise is applied in forged image

5.4 Comparative Study and Result Analysis

To compare our method with that of previous, we calculated the execution time of our method and compared it with execution time of previous method [15]. It should be noted here that for fair comparison we used the same set of images as used in [15].

Size of image	Avg Detection Result(approx)	Exec Time to construct DCT feature matrix (sec)	Exec Time till final detection (sec)	Exec Time (sec) by similar approach in [15]
160 x 120	95%	22.5	22.81	67.2
174 x 132	95%	32.08	32.44	201.22
128 x 128	95%	15.57	15.84	26.21
208 x 144	65%	34.71	35.18	184.3

Table 5.I: Comparison of execution time when block size = 8×8.

Comparing the result of our method with that of [15] as can be seen in Table 5.1, we can ascertain that this method is better in terms of efficiency.

Next to evaluate the role of block size in detection efficiency; we used same set of images with different block sizes. As shown in Table 5.2, the results are shown when we used a block size of 16 x 16 whereas in Table 5.1, the results are obtained by using a block of size 8 x 8.

Size of image	Average Detection Result(approx)	Exec Time to construct DCT feature matrix (sec)	Exec Time till final detection (sec)
160 x 120	90%	26.92	27.20
174 x 132	90%	33.03	33.36
128 x 128	90%	22.32	22.56
208 x 144	65%	29.66	30.10

Table 5.2: Comparison of execution time when block size = 16×16.

Seeing the execution times of table 5.1 and table 5.2 and similar experiments performed with different image sizes and block sizes, we can derive a relation between image size, block size, detection and efficiency. This relationship is shown in Fig 5.5.

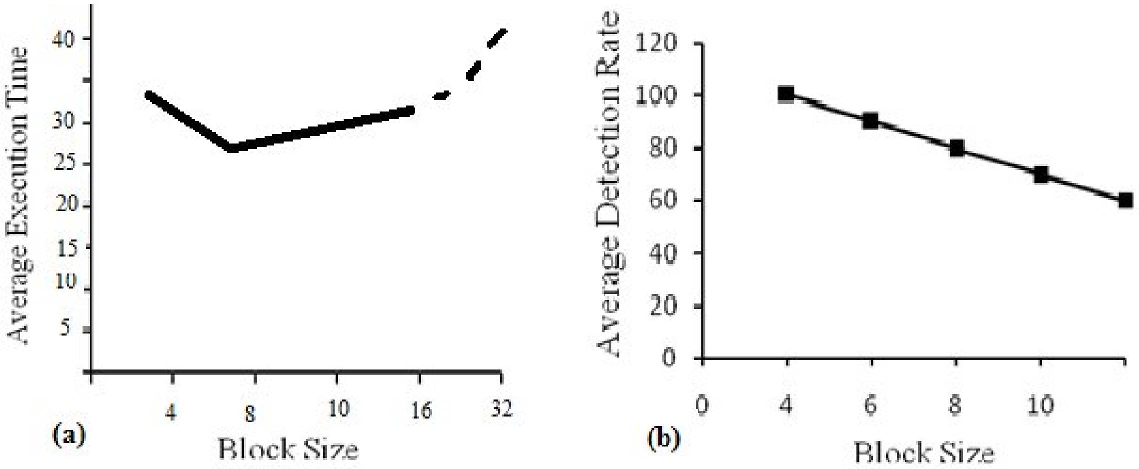


Fig 5.5: (a) Relationship between block size and avg execution time; (b) Relation between average detection rate and block size

In 5.5(a) we see that average execution time depends a great deal on block size. Intially when the block size is too small, it takes more time. The reason behind this is that when the block size is too small, although the DCT coefficients is generated fast but since total number of blocks increases, it requires more time to match each block with each other. When block size is 8 and 16, the algorithm performs best for the given image sizes. When block sizes are further increased, execution time increases as well. In fig 5.5(b), we see how size of the block affects rate of detection. When block size decreases, the detection rate increases. The reason behind this is that when block sizes increases, there is more number of elements in each block and hence more number of DCT coefficients. Since similarity between two blocks is measured by normalized difference between two feature vectors and now there are more feature vectors with respect to each block, thus their similarity automatically reduces as variance increases.

Chapter 6
Conclusion and Future Work

6.1 Conclusion

Copy move forgery is the most common forgery in digital images. A significant amount of research has been done in this field and thus a lot of methods have been developed. However a good approach must have high accuracy and low computational complexity. Besides that a good approach should be robust to various kinds of manipulations such as rotation, compression, scaling, etc.

We presented an improved approach to detect copy- move forgery using DCT coefficients [22] and then truncating the less important coefficients. We further used averaging to reduce the dimensionality of our feature vectors. Compared to other approaches in [1],[2],[7],[10] and[11], our method uses minimum number of features to detect forgery. The major achievement of our approach is reduced dimensionality of the feature vector. The DCT method in [1], PCA in [2] and improved DCT in [7] used a feature vector of size 1x64, 1x32 and 1x16 respectively; while our method just extract four features.

To do efficiency tests, we compared our results with that of an existing method [15] and found our results to be satisfactory. Further we analysed the relationship between block size, the detection rate and average execution time. The robustness towards additive noise is also analyzed and it was found that the method is robust to low density noise but suffers when noise level increases.

6.2 Future Work

There are some challenges that need to be addressed. The first one being the computational complexity that is the method runs fine on images up to 256 x256 but the computational time increases significantly after that. Though in our approach we resized the image but to our preferred size, but it can be enhanced to work fine with larger images as well.

The second challenge is method’s robustness. Although the method runs on unmodified image, its detection rate and accuracy decreases when manipulations like rotation or scaling is applied on the duplicated region. It is the major challenge that is faced by all the copy move forgery detection algorithms and need to be further investigated.

Another issue is that our method is not able to detect the forgery when the copied area is sufficiently small as compared to image size. The block size that we kept in most of the experiments is 8x 8 and 16x 16 and it could typically discover the forgery when the duplicated region was larger than 32 x 32. Additionally, sometimes the accuracy of the detection decreases when the background of the image and duplicated region matches significantly, for example in case of uniform blue sky or grass.

As the technology is growing exponentially and so do sophistication of the image forgeries, it is becoming a growing challenge to detect these forgeries. Hence the field of forgery detection need exhaustive study and research and scope of improvement is always there.

References

[1] J. Fridrich, D. Soukal, and J. Lukas, "Detection of Copy-move Forgery in Digital Images," Proceeding on Digital Forensic Research Workshop, Cleveland, OH, USA, August 2003.

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

[3] S. Bayram, H. T. Sencar, and N. Memon, "A survey of copy-move forgery detection techniques," in Proc. IEEE Western New York Image Processing Workshop, Rochester, NY, 2009.

[4] W. Li, Y. Yuan, N. Yu, Detecting copy-paste forgery of jpeg image via block artifact grid extraction, in: International Workshop on Local and Non-Local Approximation in Image Processing, 2008

[5] R. Duda and P. Hart. Pattern Classification and Scene Analysis. John Wiley and Sons, 1973.

[6] M. Sridevi, C. Mala and Siddhant Sanyam, "Comparative study of Image Forgery and Copy-Move Techniques" published in springer, advances in computer science, Eng. & Appl. AISC 166, pp. 715-723, 2012.

[7] Y. Huang, et al., Improved DCT-based detection of copy-move forgery in images, Forensic Science International 206 (1–3) (2011) 178–184.

[8] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, 1991.

[9] Qiong Tu Dan Sun Shaojie Li, Guohui Wu, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on dwt and svd," *ICME*, 2007.

[10] Weiqi Luo, Jiwu Huang, and Guoping Qiu, "Robust detection of region-duplication forgery in digital image," in *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, Washington, DC, USA, 2006, pp. 746–749, IEEE Computer Society.

[11] Sevinc Bayram, Taha Sencar, and Nasir Memon, “An efficient and robust method for detecting copy-move forgery,” *submitted to ICASSP 2009*, 2009.

[12] J. A. Bloom I. J. Cox M. L. Miller C. Y. Lin, M.Wu and Y. M. Lui, “Rotation, scale,and translation resilient watermarking for images,” *IEEE Trans. Image Processing*, vol. 10, pp. 767–782, 2001.

[13] S.-J. Lee and S.-H. Jung. A Survey of Watermarking Techniques Applied to Multimedia. *Proc. 2001 IEEE Int’l Symp. Industrial Electronics (ISIE2001)*, Vol. 1, pp. 272-277.

[14] D. G. Lowe, “Distinctive image features from scale-invariant key-points, ”IJCV, 60(2): 91–110, 2004.

[15] Ashima Gupta1, Nisheeth Saxena2, S.K Vasistha3 “ Detecting Copy move Forgery using DCT” International Journal of Scientific and Research Publications, Volume 3, Issue 5, May 2013 1 ISSN 2250-3153

[16] M. Chandra, S. Pandey and R. Chaudhary, “Digital Watermarking Technique for Protecting Digital Images” published in ICCSIT, IEEE conference vol. 7, pp. 226-233, july 2010.

[17] Niels Provosand Peter Honeyman,University of Michigan, “Hide and Seek: An Introduction to Stenography” published by IEEE computer society, 1540-7993/03.

[18] W. Lu and M. Wu, “Multimedia forensic hash based on visual words,”in Proc. IEEE Computer Soc. Int. Conf. Image Processing, 2010, pp.989–992.

[19] S. Kumar and P. K. Das, “Copy-Move Forgery Detection in digital Images: Progress and Challenges”, International Journal on Computer Science and Engineering, Vol. 3, No. 2, pp. 652-663, February 2011.

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

[21] B. Mahdian and S. Saic, “Detection of copy move forgery using a method based on blur moment invariants,” Forensic Sci. Int., 2007, vol. 171, pp. 180–189.

[22] S. Battiato and G. Messina, “Digital forgery estimation into DCT domain A critical analysis,” in *Proc. ACM Multimedia Workshop Multimedia in Forensics*, Oct. 2009, pp. 37–42.

[23] X. Guo, X. Cao, J. Zhang, and X. Li, “Mift: A mirror reflection invariant feature descriptor,” In Proc. ACCV, 2009.

[24] Weihai Li and Nenghai Yu "ROTATION ROBUST DETECTION OF COPY-MOVE FORGERY" Proceedings of 2010 IEEE 17th International Conference on Image Processing

[25] Vincent Christlein, Christian Riess, Elli Angelopoulou “A Study on Features for the Detection of Copy-Move Forgeries”

[26] X. Pan and S. Lyu, “Region duplication detection using image feature matching,” IEEE Transactions on Information Forensics and Security, vol. 5, no. 4, pp. 857– 867, 2010.

[27] CASIA Image Tampering Detection Evaluation Database, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science [Online]. Available: http://forensics.idealtest.org