IMAGE FORGERY DETECTION USING S.I.F.T METHOD

A PROJECT REPORT

Submitted by

Mr. Chaithanya L	**20171CSE0148**
Mr. Farhan Ahmed	**20171CSE0197**
Ms. Varsha B	**20171CSE0743**
Ms. Varsha N	**20171CSE0744**
Ms. Varsha R	**20171CSE0745**

Under the guidance of

Dr Madhusudhan MV

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Department of Computer Science and Engineering

School of Engineering

PRESIDENCY UNIVERSITY BANGALORE

MAY 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING

PRESIDENCY UNIVERSITY

CERTIFICATE

This is to certify that the Project report **"IMAGE FORGERY DETECTION USING S.I.F.T METHOD"** being submitted by

Ms. Chaithanya L	**20171CSE0148**
Ms. Farhan Ahmed	**20171CSE0197**
Ms. Varsha B	**20171CSE0743**
Ms. Varsha N	**20171CSE0744**
Ms. Varsha R	**20171CSE0745**

in partial fulfillment of requirement for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** is a bonafide work carried out under my supervision.

Dr. Dr. C. Kalaiarasan HOD Department of CSE Presidency University	**Prof. Dr Madhsudhan MV** Guide Asst.Professor Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING

PRESIDENCY UNIVERSITY

DECLARATION

I hereby declare that the work, which is being presented in the project report entitled "**IMAGE-FORGERY DETECTION USING S.I.F.T METHOD"** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Prof. Dr Madhsudhan MV - Asst Prof CSE, Professor, Department of Computer Science and Engineering, School of Engineering, Presidency University, Bangalore.**

Mr. Chaithanya L (20171CSE0148)
Mr. Farhan Ahmed (20171CSE0192)
Ms. Varsha B (20171CSE0702)
Ms. Varsha N (20171CSE0743)
Ms. Varsha R (20171CSE0744)

ABSTRACT

Due to the availability of higher resolution digital cameras, powerful hardware and software tools in the image editing and manipulating field, it become possible for someone to create, alter and modify the contents of a digital image and to violate its integrity. Image forensics is an important area of research used to indicate if a particular image is original or subjected to any kind of tampering. For forensic analysis, image forgery-detection techniques used to identify the forged images. An effective algorithm to indicate Copy Move Forgery in digital image presented. The Scale Invariant Feature Transform (SIFT) and Fuzzy C-means (FCM) for clustering are utilized in the algorithm. As a result performance rate will be high by using SIFT method and detection of images as original or forged will clearly checked. The S.I.F.T algorithm tested by number of experiments and results found satisfactory.

LIST OF FIGURES

TABLE OF CONTENTS

AKNOWLEDGEMENT

First of all, Weare indebted to the GOD ALMIGHTY for giving me an opportunity to excel in ourefforts to complete this project on time.

We are extremely grateful to **Dr.Abdul Sharief**, Dean, School of Engineering and**Dr.C. Kalaiarasan**, Head of the Department, Department of Computer Science and Engineering, for providing all the required resources for the successful completion of this project.

We sincerely thank our project guide, **Prof . Dr Madhsudhan MV**, for his guidance, help and motivation. Apart from the area of work, we learnt a lot from him, which we are sure will be useful in different stages of our life. We would like to express our gratitude to Faculty Coordinators and Faculty, for their review and many helpful comments.

We would like to acknowledge the support and encouragement of our friends.

Mr. Chaithanya L
Mr. Farhan Ahmed
Ms. Varsha B
Ms. Varsha N
Ms. Varsha R

CHAPTER 1

INTRODUCTION

In computer science and electrical engineering, Image processing is any form of processing for which the input is an image or a series of images or videos, such as photographs or frames of video. The output of image processing can be either an image or a set of characteristics or parameters related to the image. It also means "Analyzing and manipulating images with a computer".

Image processing is performed in three steps: First, import images with optical devices like a scanner or a camera or directly through digital processing. Second, manipulate or analyze the images in some way. This step can include image improvement and data summary, or the images are analyzed to find rules that aren't seen by the human eyes. For example, meteorologists use this processing to analyze satellite photographs. Last, output the result of image processing. The result might be the image changed by some way or it might be a report based on analysis or result of the images. Image processing usually means "digital" image processing, but optical and analog image processing are also included in image processing.

1.1 Image processing

Image processing is any form of signal processing for which the input is an image, such video frame the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing has extensive applications in many areas, including astronomy, medicine, industrial robotics, and remote sensing by satellites.

Image processing techniques are applied in the system. The goal of the manipulation can be divided into three categories:

- Image Processing (image in ->image out)
- Image Analysis (image in -> measurements out)
- Image Understanding (image in -> high-level description out)

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it.

It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

- Importing the image via image acquisition tools.
- Analyzing and manipulating the image.
- Output in which result can be altered image or report that is based on image analysis.

Photographs are never exactly the same. Even in the exact same environment, in the right spot, with the right angle, and in the same lighting conditions, the image would contain, at least, minute differences throughout the scene. In reality, images of the same object would have varying perspectives, lighting conditions, and amount of occlusion. A panorama is an image that is created from several smaller images.

This is done detecting and then making use of the similarities between images to join, or stitch, them in such a way that these images combine seamlessly. The study developed and tested an algorithm capable of combining two or more dissimilar images through existing copy-move methods. For the purpose of this study, images are classified as dissimilar when they have slightly or extremely altered perspective, have a different illumination or lighting conditions, and varying amount of occlusion.

1.1.1 Types of image processing

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

In Imaging science, image processing is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Images are also processed as three-dimensionalsignals where the third-dimension being time or the z-axis.

Closely related to image processing are computer graphics and computer vision. In computer graphics, images are manually *made* from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from *natural* scenes, as in most animated movies. Computer vision, on the other hand, is often considered*high-level* image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans).

Image processing pertains to the alteration and analysis of pictorial information. Common case of image processing is the adjustment of brightness and contrast controls on a television set by doing this As enhance the image until its subjective appearing to us is most appealing. The biological system (eye, brain) receives, enhances, and dissects analyzes and stores images at enormous rates of speed. Basically, there are two-methods for processing pictorial information.

They are:

I. Optical processing:

   Optical processing uses an arrangement of optics or lenses to carry out the process. An important form of optical image processing is found in the photographic dark room.

II. Electronic processing:

   Electronic image processing is further classified as:

   i. Analog processing.
   ii. Digital processing.

In analog image processing, the control of brightness and contrast of television image. The television signal is a voltage level that varies In amplitude to represent brightness throughout the image by electrically altering these signals, As correspondingly alter the final displayed image appearance.

In digital image processing**,** the digital images by means of digital computer refer to digital image processing. Digital images are composed of finite number of elements of which has a particular location value. Picture elements, image elements, and pixels are used as elements for digital image processing. Digital Image Processing is concerned with processing of an image. In simple words an image is a representation of a real scene, either in black and white 2 or in color, and either in print form or in a digital form i.e., technically an image is a two-dimensional light intensity function.

In other words, it is a data intensity values arranged in a two-dimensional form, the required property of an image can be extracted from processing an image. Image is typically by stochastic models. It is represented by AR model. Degradation is represented by MA model. Other form is orthogonal series expansion. Image processing system is typically non-casual system. Image processing is two-dimensional signal processing. Due to linearity property, as can operate on rows and columns separately. Image processing is vastly being implemented by "Vision Systems" in robotics. Robots are designed, and meant, to be controlled by a computer or similar devices. While "Vision Systems" are most sophisticated sensors used in Robotics. They relate the function of a robot to its environment as all other sensors do. "Vision Systems" may be used for a variety of applications, including manufacturing, navigation and surveillance. Some of the applications of Image Processing are: 1. Robotics, 2. Graphics and Animations, 3. Medical Field, 4. Satellite Imaging.

1.2 Image Forgery Detection

An image can be manipulated with a wide variety of manipulation techniques such as scaling, rotation, blurring, filtering, cropping, etc. They need image forgery detection technique in many fields for protecting copyright and preventing forgery. The verification of originality of images is required in variety of applications such as military, forensic, media, scientific, glamour, etc. Image tampering is a digital art which needs understanding of image properties and good visual creativity. Detection of image tampering deals with investigation on tampered images for possible correlations embedded due to tampering operations. Detecting forgery in digital images is a rising a field with important implications for ensuring the credibility of digital images.

1.2.1 Classification of Image Forgery Detection

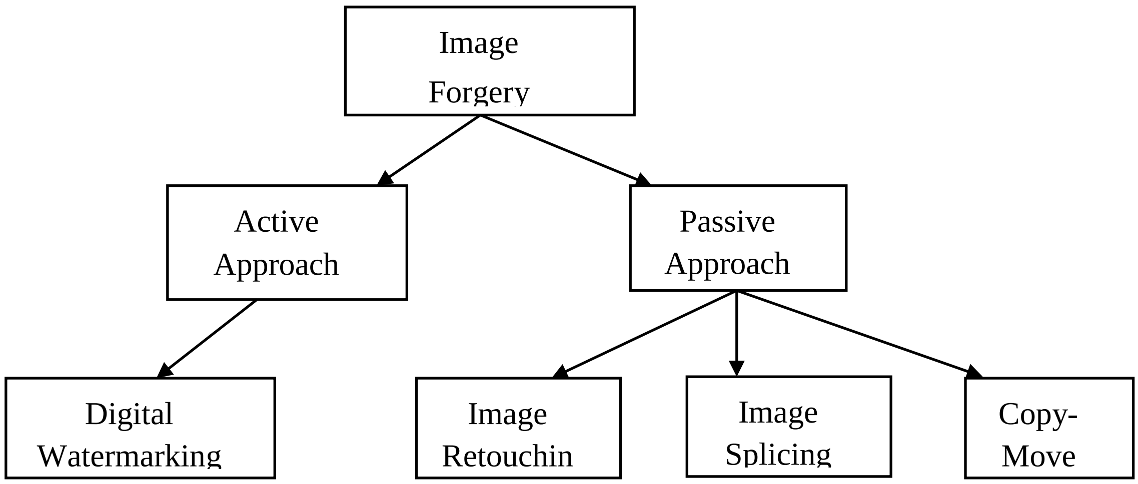


Figure 1.1: Classification of Forgery Detection Techniques.

There are many detection techniques are classified into two approaches as shown in Figure 1.1 active and passive techniques. For authenticating the digital image, watermarking of digital images and digital signature are introduced and they are known as active techniques. In an active approach it needs little preprocessing operations, like attaching watermark and signature when generating digital images, hence it limits their applications in practice. Unlike the technique of watermark and signature-based, the passive techniques does not require any digital signature to be created or to be put in any watermark. The passive authentication is the method of checking digital pictures lacking bringing in use any further data aside from the images themselves

1. **Image Retouching**

It can be treated to be the low dangerous moderately digital picture fake. Image retouching does not greatly transform or modify a picture, but a place of, improve the quality (or decreases) attribute e of a picture. Figure1.2 shows image retouching, and the difference between left image and right images (enhanced) clearly.

(a)Original Image (b) Image Retouching

Figure 1.2: Image Retouching

2. **Image Splicing**

This is second one type of forgery. Image splicing is an approach that includes a composition of multiple images which are fixed together to create a forgery as shown in Figure 1.3.This type of forgery is executed with attention; the boundary between the spliced areas can be barely observed.

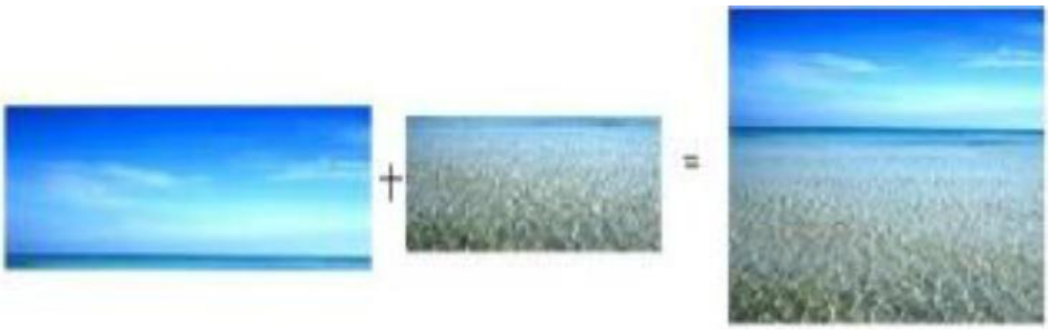


Figure 1.3: Image Splicing

3. Copy Move Forgery

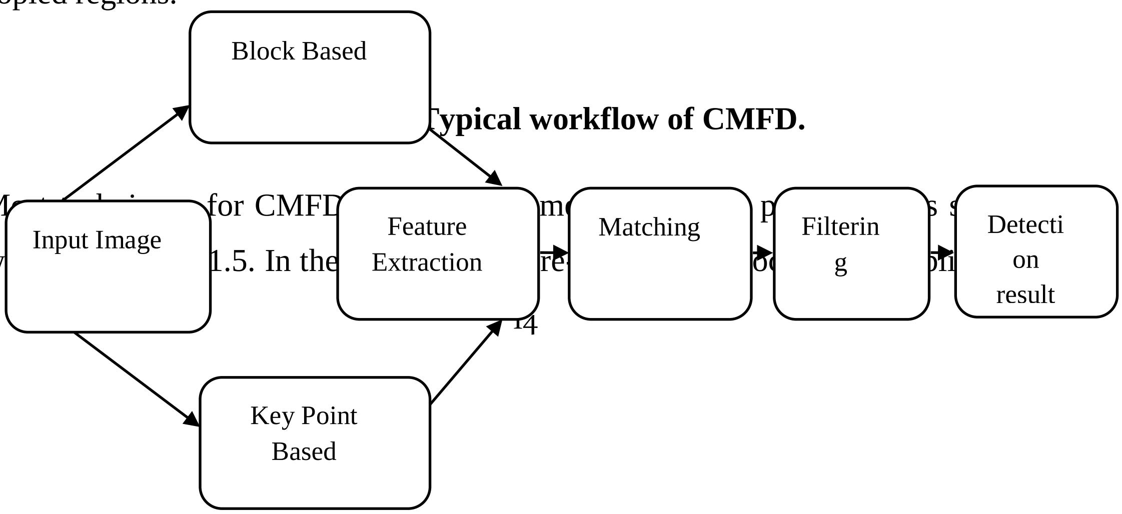
Copy move forgery is almost alike to image splicing. Here this kind of image fake a part of a picture itself is copied, moved to a desired place and pasted within the same picture. (Figure 1.4) shows a red pen has been removed from the original image in part (a), by covering some of the region by background of the same image to produce forged image (b). There are many types of copy move forgery as follows: 1) just Copy-move 2) Copy move with reflection 3) Copy-move with different scaling; and 4) Copy move with rotation.

(a)Original Image (b) Forged Image (Copy-Move)

Figure 1.4: Copy-Move of an Image

The copy move forgery is one of the difficult forgeries. This is the most common kind of image tampering technique used, where one needs to cover a part of the image in order to add or remove information. Copy-Move is a special type of image manipulation technique in which a part of the image itself is copied and pasted into another part of the same image. Image-splicing is defined as a paste-up produced by sticking together photographic images. In a copy-move attack, parts of the original image is copied, moved to a desired location, and pasted. Detecting copy-move in an image indulges broad search of local pattern or region matches.

The common CMFD workflow of feature extraction and feature extraction and matching process using a block or keypoint based approaches, Instead of listing the datasets and validations used in the literature and it is also categorize the types of the copied regions.



Typical workflow of CMFD.

M... for CMFD ... m ... p ... s s

w ... 1.5. In the ... re ... o ... n

input image. This step is very necessary for enhancing the picture data and the picture features and paves the way for more detection. The input image transformed into greyscale and another preprocessing can be optimized such as filtering or image resizing. After this procedure of preprocessing, the feature extraction to obtain feature of the picture is optimized. This procedure is classified into block based method, which split image into blocks and then obtain integral feature for each block such as discrete cosine transform DCT, singular value decomposition (SVD) , discrete wavelet transform DWT and histogram of oriented Gabor magnitude(HOGM). Key point-based technique that distinguish high-entropy image regions such as scale invariant feature transform (SIFT) and speeded up robust features (Surf). Hybrid technique which integrates both techniques that introduced a blended feature.

After the feature extraction procedure, it is very important to match identical features that mark double regions, and then use time filtering to diminish the fake matched features and finally decide if the image is forged or not.

1.3 Scale Invariant Feature Transform (SIFT)

An effective method based on Scale Invariant Feature Transform (SIFT) which was based on feature matching to detect copy-move forgery in digital images. To detect the original and the tampered region in the image SIFT descriptors were extracted and matched. Although the S.I.F.T method was found accurate and robust against different kinds of post processing operations such as JPEG compression, rotation, noise, scaling *etc.* and combination of all these operations. However, it still needs some improvements in detecting forgeries of small sized regions. In a SIFT, an objects key points are initially obtained from a set of source pictures and deposited in a database. And then an entity is identified in a new picture by separately differentiate each characteristic from the new picture to this database and using Euclidean distance of their attribute vectors to find applicant combining attributes. With that full set of duplicates, subsets of keypoints which accepts on the object and its place, scale, and direction in the new picture are discovered to filter out good equivalent. An effective hash table exertion of the generalized Hough transform is continuously applied to determination of consistent clusters. All cluster of 3 or more attributes which accepts on an entity and is then subject to foresee full design validation and eventually anomaly are removed.

SIFT algorithm converts an image data into local feature vectors named SIFT descriptors. Those features have the power to geometric transformations that are constant to scaling and rotation. This algorithm is divided into the following three main stages.

1. Scale Space Extrema Detection

The scale-space image is known as L(x,y,s) that is created by the convolution process between function and image. In this situation, convolution between Gaussian function, G(x,y,s) and an image I(x,y) is used

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

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

Optimizing a computable approximation of Gaussian's Laplacian is used to elicit the key points of the image named Difference of Gaussian (DoG), where DoG Image D is introduced as follows:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Where $L(x, y, k\sigma)$ is the convolution of the original image, I(x,y) with the Gaussian blur $G(x, y, k\sigma)$ at scale $k\sigma$.

2. Key point Localization

The image extrema contain the image main points. In order to select the main point from image extrema where the main points are unsettled over image variation have to be selected through rejecting the points over image edges and those which are characterized by low contrast. The Taylor expansion of scale-space function $D(x, y, \sigma)$ shifted such that the sample point is origin:

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

3. Key point Descriptor Generation

To ensure that The SIFT descriptors are constant in scaling and rotation, a canonical orientation is specified to each main point. In order to specify the descriptor orientation, a gradient orientation histogram is computed in the neighborhood of the key point. Particularly, for an image sample L(x, y, s) at scale s (the scale in which that key point was detected), the gradient magnitude m(x, y) and orientation q(x, y) are computed using below equation:

$$m(x,y) = \left(L(x+1,y) - L(x-1,y)\right)^2 + \left(\left(L(x,y+1) - L(x,y-1)\right)^2\right)^{\frac{1}{2}}$$

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

A feature vector with 128 elements is created for each descriptor. This vector is composed of the values of orientation histogram, in image plane and scale space form with 4X4 array of histograms and 8 orientation bins in each. The results obtained are 4X4X8 =128 element feature vector.

1.3.1Applications of SIFT features

- **Robot Localization and Mapping**

  This application a trinocular stereo system is used to determine 3D estimates for key point locations. Key points are used only when they appear in all 3 images with consistent disparities, resulting in very few outliers. As the robot moves, it localizes itself using feature matches to the existing 3D map, and then incrementally adds features to the map while updating their 3D positions using a Kalman filter. This provides a robust and accurate solution to the problem of robot localization in unknown environments.

- **Panorama Stitching**

  SIFT feature matching can be used in image stitching for fully automated panorama construction from non-panoramic images. The SIFT features extracted from the input images are matched against each other to find *k* nearest-neighbors for each feature. These correspondences are then used to find *m* candidate matching images for each image. Homo graphics between pairs of images are then computed using RANSAC and a probabilistic model is used for verification.

Because there is no restriction on the input images, graph search is applied to find connected components of image matches such that each connected component will correspond to a panorama. Finally for each connected component bundle adjustment is performed to solve for joint camera parameters, and the panorama is rendered using multi-band blending. Because of the SIFT-inspired object recognition approach to panorama stitching, the resulting system is insensitive to the ordering, orientation, scale and illumination of the images. The input images can contain multiple panoramas and noise images, and panoramic sequences are recognized and rendered as output.

- **3D Scene Modeling, Recognition and Tracking**

This application uses SIFT features for 3D object recognition and 3D modeling in context of augmented reality, in which synthetic objects with accurate pose are superimposed on real images. SIFT matching is done for a number of 2D images of a scene or object taken from different angles. This is used with bundle adjustment to build a sparse 3D model of the viewed scene and to simultaneously recover camera poses and calibration parameters. Then the position, orientation and size of the virtual object are defined relative to the coordinate frame of the recovered model. For online match moving, SIFT features again are extracted from the current video frame and matched to the features already computed for the world mode, resulting in a set of 2D-to-3D correspondences. These correspondences are then used to compute the current camera pose for the virtual projection and final rendering. A regularization technique is used to reduce the jitter in the virtual projection. 3D extensions of SIFT have also been evaluated for true 3D object recognition and retrieval.

- **3D SIFT-Like Descriptors for Human Action Recognition**

Extensions of the SIFT descriptor to 2+1-dimensional spatio-temporal data in context of human action recognition in video sequences have been studied. The computation of local position-dependent histograms in the 2D SIFT algorithm are extended from two to three dimensions to describe SIFT features in a spatio-temporal domain. For application to human action recognition in a video sequence, sampling of the training videos is carried out either at spatio-temporal interest points or at randomly determined locations, times and scales. The spatio-temporal regions around these interest points are then described using the 3D SIFT descriptor.

1.4 Fuzzy C-means clustering

Fuzzy C-means is defined as a technique of clustering that can be a section of data that is belong to two or more clusters, in order to decrease time complexity by clustering sift key point and the time consumed for matching key points. It depends on reducing the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^m ||x_j - c_j||^2, 1 \leq m < \infty$$

Where:

m is any real number greater than 1.

u_{ij} is the degree of member ship of xi in the cluster j.

x_i is the ith of d dimension measured data.

c_j is the d-dimension center of the cluster.

$|| * ||$ is any norm expressing the similarity between any measured data and the center.

A refined application of the objective function introduced above is executed Fuzzy partitioning which is created with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{||x_j - c_j||}{||x_j - c_k||} \right)^{\frac{2}{m-1}}} , c_j = \frac{\sum_{i=1}^{n} u_{ij}^m . c_j}{\sum_{i=1}^{n} u_{ij}^m}$$

This repetition will stop when$max_{ij}\{|u_{ij}^{k+1} - u_{ij}^{k}|\} < \varepsilon$ where $\varepsilon$is a termination criterion between 0 and 1, whereas kis the iteration steps. These steps assemble to a local minimum or a saddle point of jm.

The algorithm is performed by optimizing the following procedures:

1. Initialize $U = [u_{ij}]$ matrix, $U^{(0)}$
2. At k-step: compute the centers vectors$C^{(k)} = [c_j]$ with $U^{(K)} c_j = \frac{\sum_{i=1}^{n} u_{ij}^m . c_j}{\sum_{i=1}^{n} u_{ij}^m}$

3. Modify $U^{(k)}, U^{(k+1)} u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{||x_j - c_j||}{||x_j - c_k||}\right)^{\frac{2}{m-1}}}$

4. If $||U^{(k+1)} - U^{(k)}|| < \varepsilon$then STOP; otherwise return to step 2.

1.5 Problem Statement

Due to the availability of higher resolution digital cameras, powerful hardware & software tools in the image editing and manipulating field, it become possible for someone to create, alter and modify the contents of a digital image and to violate its integrity.

1.6 Objectives

The main objectives of our project are:

- To perform an in depth investigation of the available mitigation techniques of image forgery detection and there by resist copy-move attack.
- To develop a framework that can perform Gaussian filtering, KeypointLocalization and extraction of an SIFT features and for clustering FCM technique are utilized.
- To perform comparative performance analysis for the outcome of the results.

1.7 Organization of the Report

The rest of thesis is organized as follows: Chapter 2 discusses about the Requirement Analysis which includes an overview of functional requirements and software requirements. The Literature review is included in Chapter 3. The System design is included in Chapter 4. Implementation is included in Chapter 5. Testing and snapshots are discussed in Chapter 6. Finally the conclusion and future work is given in Chapter 7.

CHAPTER 2

REQUIREMENT ANALYSIS

The requirements means specifying the capabilities or conditions of the system and the functions that the system is intended that the system must obtain. There are different ways to gather requirements from different sources. For example, requirements can be gathered via surveys .The complete discretional behavior of system is developed. It includes the functional requirement for the software to be developed. The functional requirement includes what the software should do. Requirements must be measurable, testable, related to identified needs or opportunities, and defined to a level of detail sufficient for system design. What the software has to do is directly perceived by its users either human users or other software systems. The writing of software requirement specification reduces development effort.

Requirement analysis is critical to the success or failure of a system or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

2.1 Functional Requirements

In software engineering and system engineering, a functional requirement defines a function of a system or its component. Functional requirement may be calculations, technical details, data manipulation and processing and other specific functionality that defines what a system is supposed to accomplish.

In this study, Image forensics is an important area used to indicate if a particular image is original or subjected to any kind of tampering. Images are essential part of judgment in tribunals. For forensic analysis, Image forgery-detection techniques used to identify the forged images. Image forgery detection can be detected by several methods but, an effective method based on Scale Invariant Feature Transform (SIFT) which was based on feature matching to detect copy-move forgery in digital images. To specify the originality of digital image has become an important area to regain trust in digital image.

2.2 Software Requirements

In software engineering, A software requirements is a description of a software system to be developed. The software used in image forgery detection are MATLAB and MICC220 dataset. Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

2.2.1 MATLAB

MATLAB is a special-purpose computer program optimized to perform engineering and scientific calculations. It started life as a program designed to perform matrix mathematics, but over the years it has grown into a flexible computing system capable of solving essentially any technical problem. The MATLAB program implements the MATLAB programming language and provides a very extensive library of predefined functions to make technical programming tasks easier and more efficient. Here, the MATLAB language as it is implemented in MATLAB Version 9.4 (Release 2018a) and shows how to use it to solve typical technical problems. MATLAB is a huge program with an incredibly rich variety of functions. Even the basic version of MATLAB without any toolkits is much richer than other technical programming languages. There are more than 1000 functions in the basic MATLAB product alone, and the toolkits extend this capability with many more functions in various specialties. It makes no attempt to introduce the user to all of MATLAB's functions. Instead, it teaches a user the basics of how to write, debug, and optimize good MATLAB programs and presents a subset of the most important functions.

Figure 2.1: MATLAB LOGO

Just as importantly, it teaches the programmer how to use MATLAB's own tools to locate the right function for a specific purpose from the enormous number of choices available. Figure 2.1 shows the MATLAB LOGO. MATLAB has many advantages compared with conventional computer languages for technical problem solving. These include

1.Ease of Use

MATLAB is an interpreted language, like many versions of Basic. Like Basic, it is very easy to use. The program can be used as a scratch pad to evaluate expressions typed at the command line, or it can be used to execute large prewritten programs. Programs may be easily written and modified with the built-in integrated development environment, and debugged with the MATLAB debugger. Because the language is so easy to use, it is ideal for the rapid prototyping of new programs. Many program development tools are provided to make the program easy to use. They include an integrated editor/debugger, on-line documentation and manuals, a workspace browser, and extensive demos.

2. Platform Independence

MATLAB is supported on many different computer systems, providing a large measure of platform independence. At the time of this writing, the language is supported on Windows 2000/XP/Vista, Linux, several versions of Unix, and the Macintosh. Programs written on any platform will run on all of the other platforms, and data files written on any platform may be read transparently on any other platform. As a result, programs written in MATLAB can migrate to new platforms when the needs of the user change.

3.Predefined Functions

MATLAB comes complete with an extensive library of predefined functions

that provide tested and prepackaged solutions to many basic technical tasks. For example, suppose that you are writing a program that must calculate the statistics associated with an input data set. In most languages, you would need to write your own subroutines or functions to implement calculations such as the arithmetic mean, standard deviation, median, and so on. These and hundreds of other functions are built right into the MATLAB language, making your job much easier. In addition to the large library of functions built into the basic MATLAB language, there are many special-purpose toolboxes available to help solve complex problems in specific areas. For example, a user can buy standard toolboxes to solve problems in signal processing, control systems, communications, image processing, and neural networks, among many others. There is also an extensive collection of free user-contributed MATLAB programs that are shared through the MATLAB Web site.

4.Device-Independent Plotting

Unlike most other computer languages, MATLAB has many integral plotting

and imaging commands. The plots and images can be displayed on any graphical output device supported by the computer on which MATLAB is running. This capability makes MATLAB an outstanding tool for visualizing technical data.

5.Graphical User Interface

MATLAB includes tools that allow a programmer to interactively construct

a Graphical User Interface (GUI) for his or her program. With this capability, the programmer can design sophisticated data-analysis programs that can be operated by relatively inexperienced users.

6.MATLAB Compiler

MATLAB’s flexibility and platform independence is achieved by compilingMATLAB programs into a device-independent p-code and then interpreting the p-code instructions at runtime. This approach is similar to that used by Microsoft’s Visual Basic language. Unfortunately, the resulting programs can sometimes execute slowly because the MATLAB code is interpreted rather than compiled.

The MATLAB system consists of five main parts:

1. **Development Environment:** This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

2. **The MATLAB Mathematical Function Library:** This is a vast collection ofcomputational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

3. **The MATLAB Language:** This is a high-level matrix/array language withcontrol Flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.

4. **Graphics:** MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

5. **The MATLAB External Interfaces/API:** This is a library that allows you to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

MATLAB is the tool of choice for high-productivity development, and analysis. MATLAB features a family of add-on application-specific solutions called *toolboxes*. Very important to most users of MATLAB, toolboxes allow you to

learn and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

2.2.2 MICC-F220

A dataset is a collection of related sets of information that is composed of separate images. Here, MICC-F220 is the dataset which is composed by 220 images, 110 are tampered and 110 originals. A number of numerical experiments performed using the MICC-220 dataset.

2.3 Summary

This Chapter is about overview of functional requirement and software requirement. Functional requirement include functionality of the project. The software requirement includes brief discussion about the software used such as MATLAB, MICC-F220 dataset.

CHAPTER 3
LITERATURE REVIEW

Ref no	Paper Title	Author Names	Journal Name, Publisher and Year	Technique Used	Observation	Accuracy
1.	Digital Image Forgery Detection Using JPEG Features and Local Noise Discrepancies.	Bo Liu,1 Chi-Man Pun,1 and Xiao-Chen Yuan1	The Scientific World Journal in 16 Mar 2014	Block Artificial Grid and Noise Estimation, Integrated Method for Forgery Detection.	JPEG block artificial grids , local noise discrepancies were used to generate features	70%
2.	SIFT-Symmetry: A robust detection method for copy move forgery with reflection attack	Nor Bakiah Abd. Warif, Ainuddin Wahid Abdul Wahab, Rosli Salleh	Journal of Visual Communication and Image Representation in July 2017	SIFT based CMF detection	JPEG compression , Gaussian noise were used for reflection attacks.	80%
3.	A Systematic Study of Image Forgery Detection	Santhosh Kumar1 ∗,S.Karthi2,K .Karthika2, and Rajan Cristin1	August 2018Journal of Computational and Theoretical Nanoscience 15(8)	Format Based Technique,Camera Based Technique,Forgery Shadow Detection	Authenticity and integrity of the digital images, forensic, medi-cal imaging, e-commerce, industrial photography	80%
4.	Digital image forgery detection using deep learning approach	Andrey Kuznetsov Samara National Research University	November 2019 Journal of Physics Conference Series	Image Forgery Detection Using Jpeg Artifacts	The results obtained showed a high quality of image classification (97.8% accuracy)	97%

CHAPTER 4
SYSTEM DESIGN

The system design is the process of defining the architecture, modules, interfaces anddata for a system to specify specified requirements. Here, the methodology and an different stages in an S.I.F.T algorithm will be discussed.

4.1 System Design

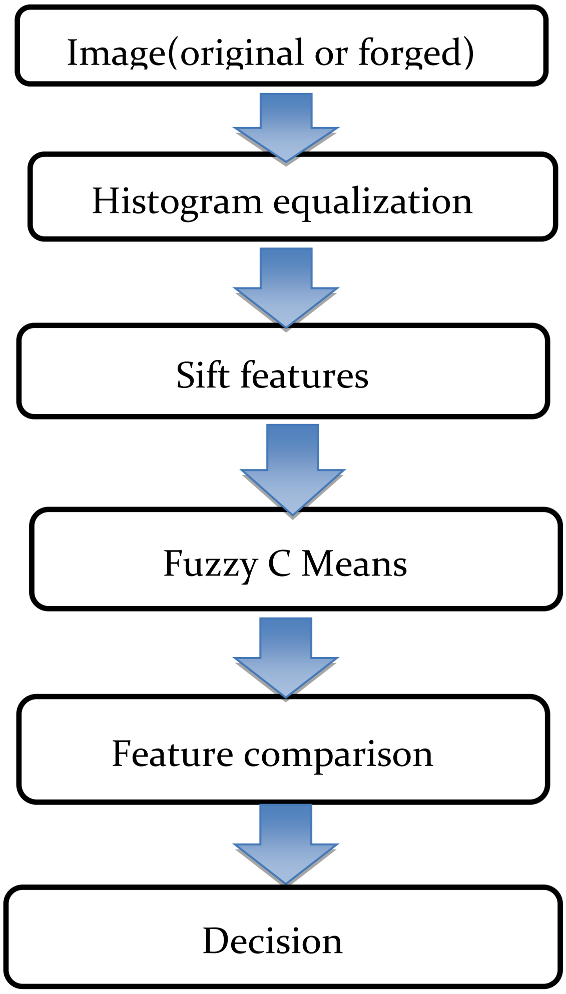


Figure 4.1: System design.

The introduced technique depends on the S.I.F.T algorithm to elicit solid features which enable it to specify if a region of an image was a copy-moved. The introduced technique decreases time complexity of S.I.F.T using FCM clustering method. The flow of a method is shown in the above block diagram Figure 3.1. In the S.I.F.T algorithm, S.I.F.T key points are clustered on the basis of their descriptors then, center key point and its neighbor are matched with other center key point and its neighbor clusters instead of identifying all keywords in the picture.

4.1.1 Image (Original or Forged)

Original image is a real visual representation of something or it is picture that has been created without any modification or alteration, where asforged images are the fake images which are altered from original image. Here they do image forgery means manipulation of the digital image and they are cases when it is difficult to identify the edited region from the original image. The detection of a forged image is driven by the need of authenticity and to maintain integrity of the image. In this stage checking the image is original or forged one.

4.1.2Histogram Equalization

Histogram equalization is a method to process images in order to adjust the contrast of an image by modifying the intensity distribution of the histogram. Itis a computer image processing technique used to improve contrast in images. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast. The objective of this technique is to give a linear trend to the cumulative probability function associated to the image.

4.1.3 S.I.F.T Features

Scale Invariant Feature Transform (SIFT) which was based on feature matching to detect copy-move forgery in digital images. To detect the original and the tampered region in the image SIFT descriptors were extracted and matched. The feature descriptor is elicited from every key point on the image including 128 dimensional. The resemblance between the descriptors is calculated to specify the matching among the descriptors for specifying the potential forgery on the image.

The S.I.F.T features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. They are also robust to changes in illumination, noise, and minor changes in viewpoint. In addition to these properties, they are highly distinctive, relatively easy to extract and allow for correct object identification with low probability of mismatch. Object description by set of S.I.F.T features is also robust to partial occlusion. As few as 3 S.I.F.T features from an object are enough to compute its location and pose. Recognition can be performed in close-to-real time, at least for small databases and on modern computer hardware.

4.1.4 FCM – Algorithm

Fuzzy C-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. Here we have considered three clusters. Fuzzy logic is a multi-valued logic derived from fuzzy set theory. FCM is popularly used for soft segmentations techniques.

This algorithm works by assigning membership to each data point corresponding to each cluster center on the basisof distance between the cluster center and the data point. More the data is near to the cluster center more is itsmembership towards the particular cluster center. Clearly, summation of membership of each data point should beequal to one. After each iteration membership and cluster centers are updated according to the formula :

$$\mu_{ij} = 1 / \sum_{k=1}^{c} (d_{ij} / d_{ik})^{(2/m-1)}$$

$$v_j = (\sum_{i=1}^{n} (\mu_{ij})^m x_i) / (\sum_{i=1}^{n} (\mu_{ij})^m), \forall j = 1, 2, \ldots.c$$

4.1.4 Gaussian Filter

The Gaussian blur is a type of image-blurring filter that uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image.It is a widely used effect in graphics software, typically to reduce image noise and reduce detail.A Gaussian blur effect is typically generated by convolving an image with an FIR kernel of Gaussian values.The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of an object under usual illumination.

4.1.5 Decision (Original or Forged)

The S.I.F.T algorithm along with usage of FCM Clustering and S.I.F.T feature comparison has the ability to reveal copy move forgery very fast without influencing the accuracy of matching process. This method decreases the detection time of appreciably same accuracy standards and minor enhancement in some cases.

Finally, we get one single image containing the detected forged elements in it.

CHAPTER 5

SYSTEM IMPLEMENTATION

System implementation is a complex process to execute. MATLAB which stands for Matrix Laboratory is a state-of-the-art mathematical software package, which is used extensively in both academia and industry. It is an interactive program for numerical computation and data which visualization, which along with its programming capabilities provides a very useful to for almost all areas of science and engineering. It remains however, one of the leading software packages for numerical computation. MATLAB program and script files always have file names ending with ".m"; the programming language is exceptionally straightforward since almost every data object is assumed to be an array. Graphical output is available to supplement numerical result.

5.1 Image (Forged or Original)

```
clc
clear
close all
%%
addpath 'FCM'
%% Select Test Image
[file,path]=uigetfile('*.jpg'),title('Select Test Image');
filename=[path,file];
%%
image = imread(filename);
rgbimage = imresize(image,[256 256]);
image = imresize(image,[256 256]);
```

The first part of the implementation is to select an image from the MICC220 dataset.

5.2 Implementation of Histogram Equalization

```
%% Histogram Equalization
for i=1:3
channelImage=image(:,:,i);
channelImage=histeq(channelImage);
    image(:,:,i)=channelImage;
end
```

Histogram Equalizationis a computer image processing technique used to improve contrast in images. This method usually increases the global contrast of images when its usable data is represented by close contrast values

5.3 Implementation of Scale Spaces Extrema Detection in S.I.F.T

The implementation of an S.I.F.T takes to create a scale space; you take the original image and generate progressively blurred out images.

Here's an example:

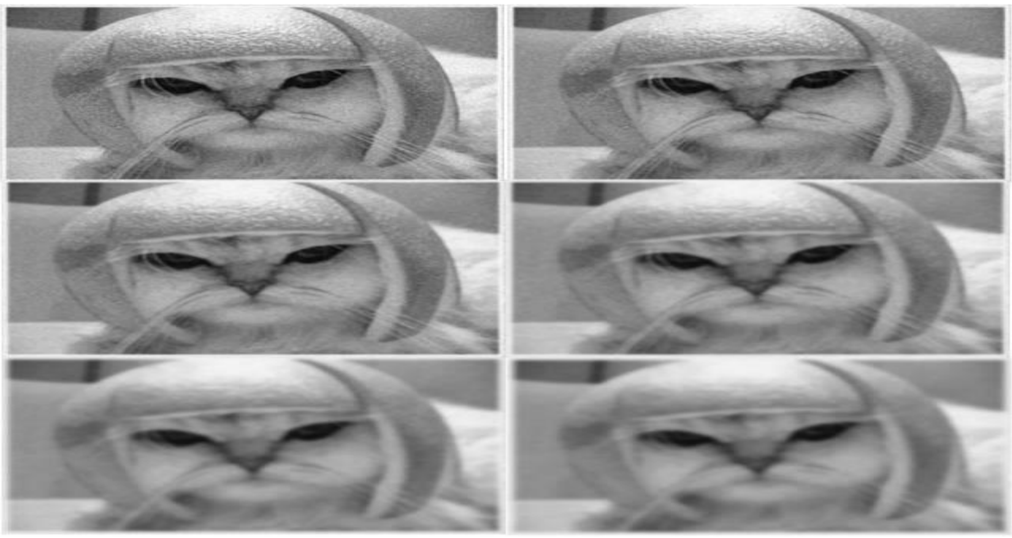


Figure 5.1: Gaussian Blur.

Figure 5.1 shows how the cat's helmet loses detail. So do it's whiskers. The S.I.F.T takes scale space to the next level by taking the original image and generates progressively blurred out images. Then, resize the original image to half size. Next, Generating blurred out images again and keep repeating.

Here's what it would look like in SIFT:

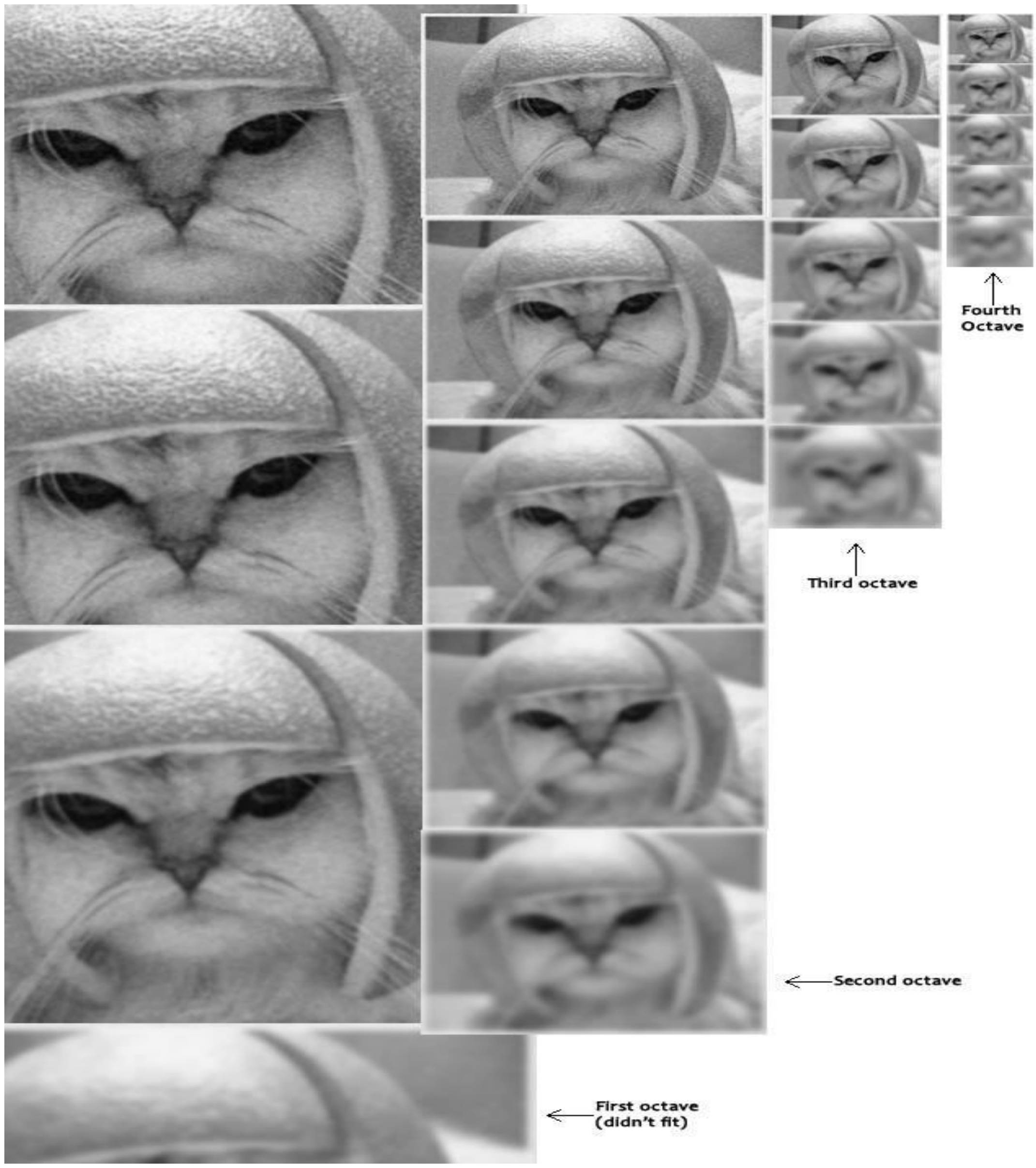


Figure 5.2: Resizing the original image into half size in scale space.

Figure 5.2 shows the image of the same size (vertical) form an octave. Above are four octaves. Each octave has 5 images. The individual images are formed because of theincreasing “scale” (the amount of blur).

Here is the code for Gaussian filter. In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail.

```
%% Scale-Space Extrema Detection

% original sigma and the number of actave can be modified. the larger
% sigma0, the more quickly-smooth images
sigma0=sqrt(2);
octave=3;%6*sigma*k^(octave*level)<=min(m,n)/(2^(octave-2))
level=3;
D=cell(1,octave);
fori=1:octave
D(i)=mat2cell(zeros(row*2^(2-i)+2,colum*2^(2-i)+2,level),row*2^(2-i)+2,colum*2^(2-i)+2,level);
end
% first image in first octave is created by interpolating the original one.
temp_img=kron(img,ones(2));
temp_img=padarray(temp_img,[1,1],'replicate');
```

The idea was to blur an image progressively, shrink it, blur the small image progressively and so on. Now to use those blurred images to generate another set of images, the Difference of Gaussians (DoG). These DoG images are a great for finding out interesting key points in the image.

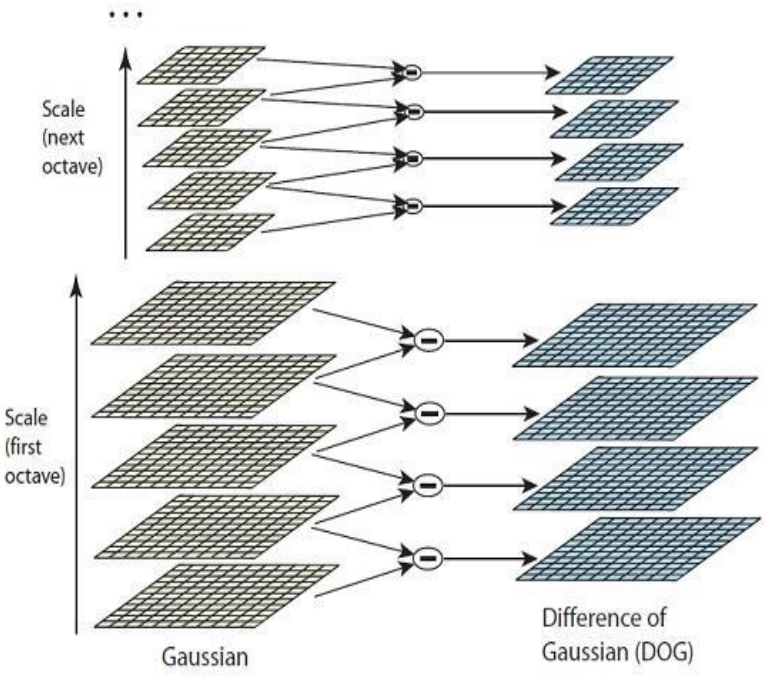


Figure 5.3: Difference of Gaussian.

Figure 5.3 shows the Difference of Gaussian images are approximately equivalent to the Laplacian of Gaussian and have replaced a computationally intensive process with a simple subtraction (fast and efficient).

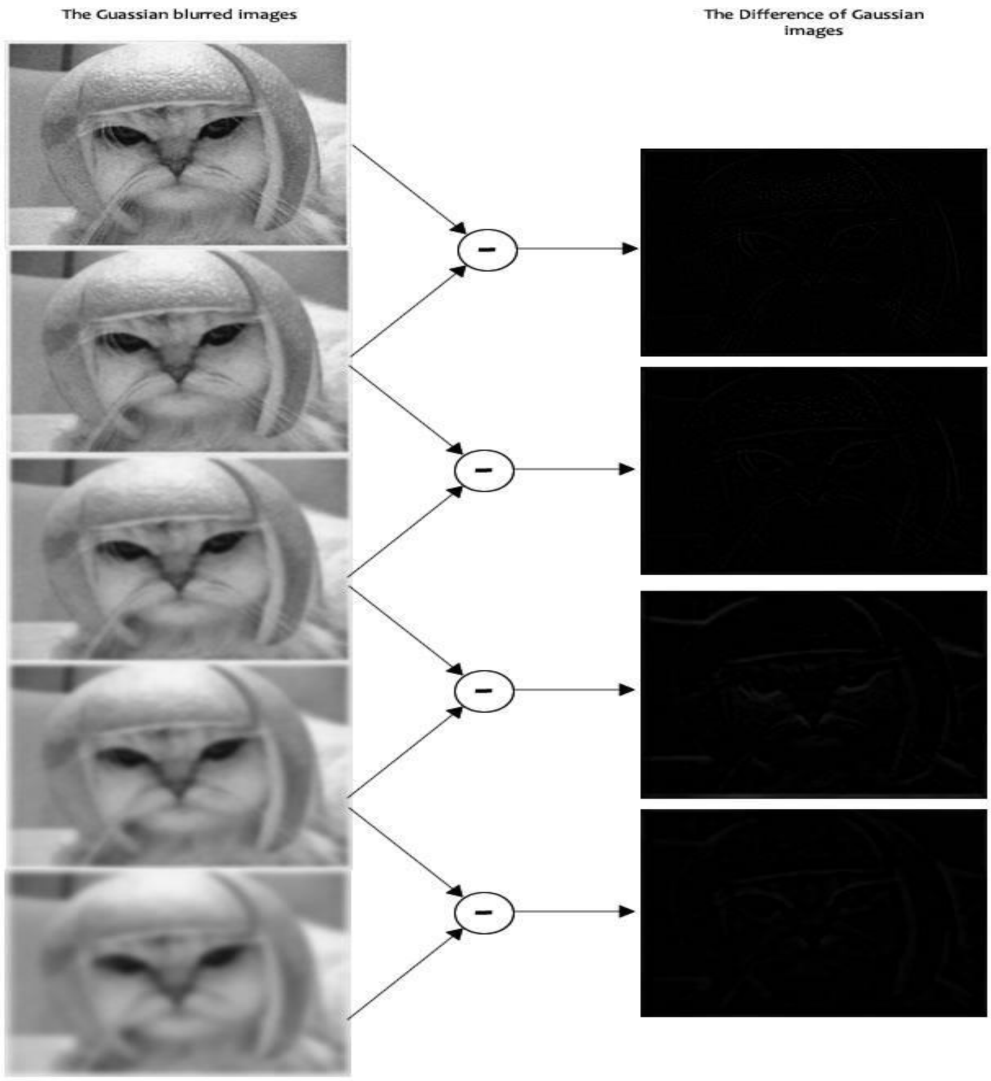


Figure 5.4: Difference of Gaussian example.

Figure 5.4 shows DoG image, here, it's done for the subtraction for just one octave. The same thing is done for all octaves. This generates DoG images of multiple sizes.

Here is the code for implementation of a Difference of Gaussian. Where difference of these two Gaussian smoothed images, called difference of Gaussian (DoG), can be used to detect edges in the image.

```
%create the DoG pyramid.
fori=1:octave
temp_D=D{i};
for j=1:level
scale=sigma0*sqrt(2)^(1/level)^((i-1)*level+j);
p=(level)*(i-1);
f=fspecial('gaussian',[1,floor(6*scale)],scale);
L1=temp_img;
if(i==1&&j==1)
L2=conv2(temp_img,f,'same');
L2=conv2(L2,f','same');
temp_D(:,:,j)=L2-L1;

L1=L2;
else
L2=conv2(temp_img,f,'same');
L2=conv2(L2,f','same');
temp_D(:,:,j)=L2-L1;
L1=L2;
if(j==level)
temp_img=L1(2:end-1,2:end-1);
end

end
end
D{i}=temp_D;
temp_img=temp_img(1:2:end,1:2:end);
temp_img=padarray(temp_img,[1,1],'both','replicate');
end
```

5.4 Implementation of Keypoint Localization in S.I.F.T

Finding key points is a two part process i.e., Locate maxima/minima in DoG images and Find sub pixel maxima/minima.To locate the maxima and minima you iterate through each pixel and check all its neighbors. The check is done within the current image, and also the one above and below it.

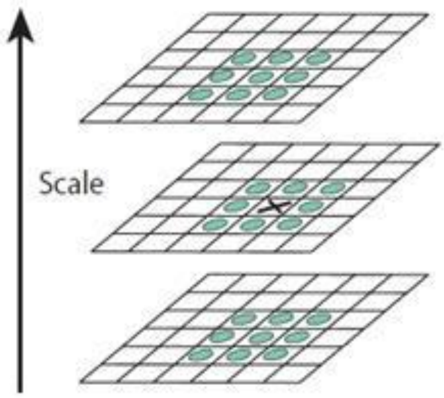


Figure 5.5: Locate maxima/minima in DoG.

Figure 5.5 shows the locate maxima/minima in DoG where X marks the current pixel. The green circles mark the neighbours. This way, a total of 26 checks are made. X is marked as a "key point" if it is the greatest or least of all 26 neighbours**.**

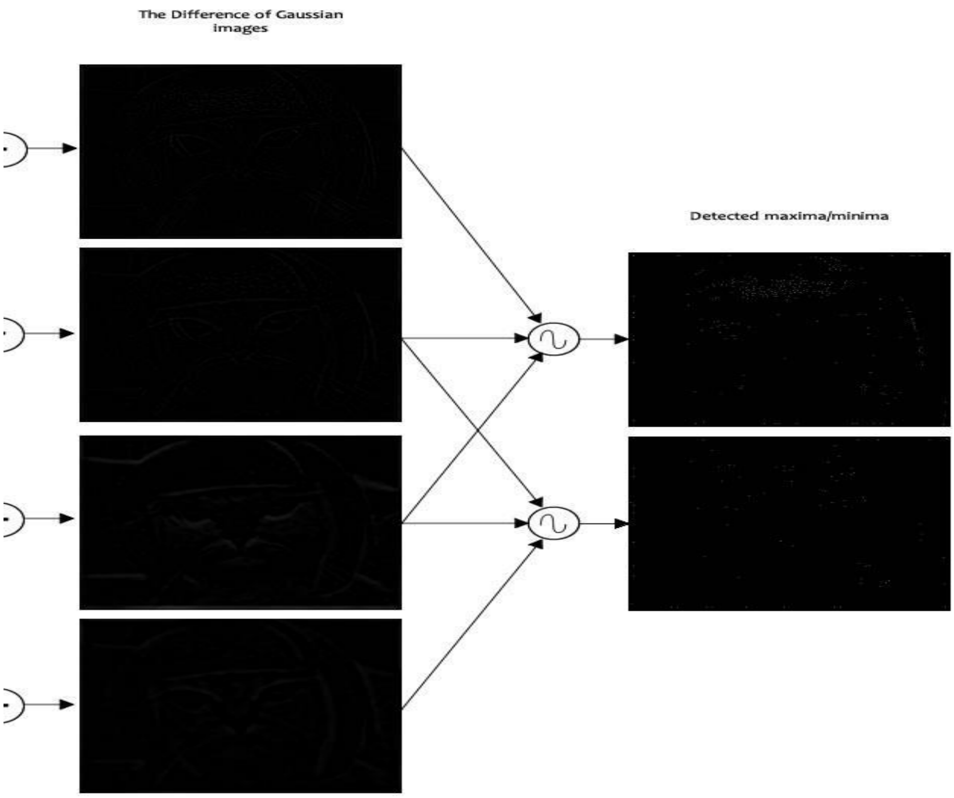


Fig 5.6: Detecting Maxima/ Minima.

Figure 5.6 shows the detection of maxima and minima in an DoG images where there are a bunch of white dots on those black images.It is done by comparing neighboring pixels in the current scale, the scale "above" and the scale "below".

In keypoint orientation, the idea is to collect gradient directions and magnitudes around each keypoint. Figure 3.8 shows the size of the "orientation collection region" around the keypoint depends on its scale.

Here is the code for implementation of a Keypoint Localization which searches each pixel in the DoG map to find the extreme point.

```
%% KeypointLocalistaion
% search each pixel in the DoG map to find the extreme point

interval=level-1;
number=0;
fori=2:octave+1
    number=number+(2^(i-octave)*colum)*(2*row)*interval;
end
extrema=zeros(1,4*number);
flag=1;
fori=1:octave
    [m,n,~]=size(D{i});
    m=m-2;
    n=n-2;
    volume=m*n/(4^(i-1));
for k=2:interval
for j=1:volume
% starter=D{i}(x+1,y+1,k);
            x=ceil(j/n);
            y=mod(j-1,m)+1;
            sub=D{i}(x:x+2,y:y+2,k-1:k+1);
            large=max(max(max(sub)));
            little=min(min(min(sub)));
if(large==D{i}(x+1,y+1,k))
                temp=[i,k,j,1];
                extrema(flag:(flag+3))=temp;
                flag=flag+4;
end
if(little==D{i}(x+1,y+1,k))
                temp=[i,k,j,-1];
                extrema(flag:(flag+3))=temp;
                flag=flag+4;
end
end
```

```
end
end
idx= extrema==0;
extrema(idx)=[];

[m,n]=size(img);
x=floor((extrema(3:4:end)-1)./(n./(2.^(extrema(1:4:end)-2))))+1;
y=mod((extrema(3:4:end)-1),m./(2.^(extrema(1:4:end)-2)))+1;
ry=y./2.^(octave-1-extrema(1:4:end));
rx=x./2.^(octave-1-extrema(1:4:end));
```

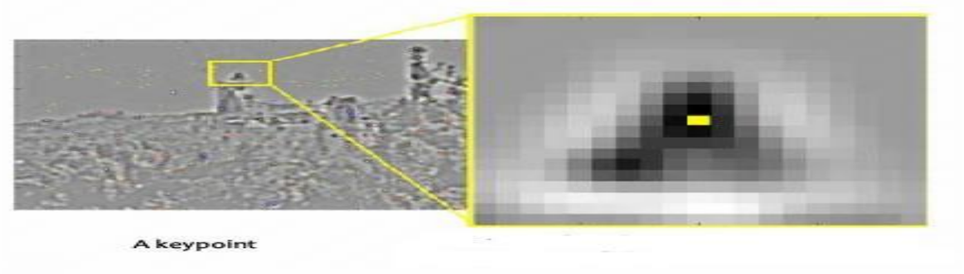


Figure 5.7: A Keypoint Orientation.

Here is the code for implementation of a KeypointOrientation which searches each pixel in the DoG map to find the extreme point.

%% Orientation Assignment(Multiple orientations assignment)

tic

kpori=zeros(1,36*extr_volume);

minor=zeros(1,36*extr_volume);

f=1;

flag=1;

for i=1:extr_volume

%search in the certain scale

scale=sigma0*sqrt(2)^(1/level)^((extrema(4*(i-1)+1)-1)level+(extrema(4(i-1)+2)));

width=2*round(3*1.5*scale);

count=1;

x=floor((extrema(4*(i-1)+3)-1)/(n/(2^(extrema(4*(i-1)+1)-2))))+1;

y=mod((extrema(4*(i-1)+3)-1),m/(2^(extrema(4*(i-1)+1)-2)))+1;

```
    %make sure the point in the searchable area
    if(x>(width/2)&&y>(width/2)&&x<(m/2^(extrema(4*(i-1)+1)-2)-width/2-
2)&&y<(n/2^(extrema(4*(i-1)+1)-2)-width/2-2))
rx=x+1;
ry=y+1;
rz=extrema(4*(i-1)+2);
reg_volume=width*width;%3? thereom
        % make weight matrix
        weight=fspecial('gaussian',width,1.5*scale);
        %calculate region pixels' magnitude and region orientation
reg_mag=zeros(1,count);
reg_theta=zeros(1,count);
    for l=(rx-width/2):(rx+width/2-1)
        for k=(ry-width/2):(ry+width/2-1)
            reg_mag(count)=sqrt((D{extrema(4*(i-1)+1)}(l+1,k,rz)-D{extrema(4*(i-
1)+1)}(l-1,k,rz))^2+(D{extrema(4*(i-1)+1)}(l,k+1,rz)-D{extrema(4*(i-1)+1)}(l,k-
1,rz))^2);
            reg_theta(count)=atan2((D{extrema(4*(i-1)+1)}(l,k+1,rz)-D{extrema(4*(i-
1)+1)}(l,k-1,rz)),(D{extrema(4*(i-1)+1)}(l+1,k,rz)-D{extrema(4*(i-1)+1)}(l-
1,k,rz)))*(180/pi);
            count=count+1;
        end
    end
    %make histogram
mag_counts=zeros(1,36);
    for x=0:10:359
mag_count=0;
       for j=1:reg_volume
           c1=-180+x;
           c2=-171+x;
```

```
        if(c1<0||c2<0)
        if(abs(reg_theta(j))<abs(c1)&&abs(reg_theta(j))>=abs(c2))
            mag_count=mag_count+reg_mag(j)*weight(ceil(j/width),mod(j-
1,width)+1);
        end
        else
            if(abs(reg_theta(j)>abs(c1)&&abs(reg_theta(j)<=abs(c2))))
                mag_count=mag_count+reg_mag(j)*weight(ceil(j/width),mod(j-
1,width)+1);
            end
        end
    end
mag_counts(x/10+1)=mag_count;
   end
   % find the max histogram bar and the ones higher than 80% max
   [maxvm,~]=max(mag_counts);
kori=find(mag_counts>=(0.8*maxvm));
kori=(kori*10+(kori-1)*10)./2-180;
kpori(f:(f+length(kori)-1))=kori;
    f=f+length(kori);
    temp_extrema=[extrema(4*(i-1)+1),extrema(4*(i-1)+2),extrema(4*(i-
1)+3),extrema(4*(i-1)+4)];
    temp_extrema=padarray(temp_extrema,[0,length(temp_extrema)*(length(kori)-
1)],'post','circular');
    long=length(temp_extrema);
    minor(flag:flag+long-1)=temp_extrema;
    flag=flag+long;
   end
end
idx= minor==0;
```

minor(idx)=[];

extrema=minor;

% delete unsearchable points and add minor orientation points

idx= kpori==0;

kpori(idx)=[];

extr_volume=length(extrema)/4;

toc

5.5 Implementation of Keypoint Descriptor Generation in S.I.F.T

Here is the code for implementation of a keypoint descriptor where a keypoints is defined by some particular image intensities "around" it, such as a corner. A keypoint can be used for deriving a descriptor.

```
%% keypoint descriptor

d=4;% In David G. Lowe experiment,divide the area into 4*4.
pixel=4;
feature=zeros(d*d*8,extr_volume);
fori=1:extr_volume
    descriptor=zeros(1,d*d*8);% feature dimension is 128=4*4*8;
    width=d*pixel;
%x,ycenteral point and prepare for location rotation
    x=floor((extrema(4*(i-1)+3)-1)/(n/(2^(extrema(4*(i-1)+1)-
2))))+1;
    y=mod((extrema(4*(i-1)+3)-1),m/(2^(extrema(4*(i-1)+1)-2)))+1;
    z=extrema(4*(i-1)+2);
if((m/2^(extrema(4*(i-1)+1)-2)-
pixel*d*sqrt(2)/2)>x&&x>(pixel*d/2*sqrt(2))&&(n/2^(extrema(4*(i-
1)+1)-2)-pixel*d/2*sqrt(2))>y&&y>(pixel*d/2*sqrt(2)))
sub_x=(x-d*pixel/2+1):(x+d*pixel/2);
sub_y=(y-d*pixel/2+1):(y+d*pixel/2);
        sub=zeros(2,length(sub_x)*length(sub_y));
        j=1;
for p=1:length(sub_x)
for q=1:length(sub_y)
                sub(:,j)=[sub_x(p)-x;sub_y(q)-y];
                j=j+1;
end
end
        distort=[cos(pi*kpori(i)/180),-
sin(pi*kpori(i)/180);sin(pi*kpori(i)/180),cos(pi*kpori(i)/180)];
```

```
%accordinate after distort
sub_dis=distort*sub;
fix_sub=ceil(sub_dis);
fix_sub=[fix_sub(1,:)+x;fix_sub(2,:)+y];
        patch=zeros(1,width*width);
for p=1:length(fix_sub)
        patch(p)=D{extrema(4*(i-
1)+1)}(fix_sub(1,p),fix_sub(2,p),z);
end
temp_D=(reshape(patch,[width,width]))';
%create weight matrix.
mag_sub=temp_D;
temp_D=padarray(temp_D,[1,1],'replicate','both');
        weight=fspecial('gaussian',width,width/1.5);
mag_sub=weight.*mag_sub;
        theta_sub=atan((temp_D(2:end-1,3:1:end)-temp_D(2:end-
1,1:1:end-2))./(temp_D(3:1:end,2:1:end-1)-temp_D(1:1:end-2,2:1:end-
1)))*(180/pi);
% create orientation histogram
for area=1:d*d
        cover=pixel*pixel;
ori=zeros(1,cover);
magcounts=zeros(1,8);
for angle=0:45:359
magcount=0;
for p=1:cover;
              x=(floor((p-1)/pixel)+1)+pixel*floor((area-1)/d);
              y=mod(p-1,pixel)+1+pixel*(mod(area-1,d));
              c1=-180+angle;
              c2=-180+45+angle;
if(c1<0||c2<0)
if (abs(theta_sub(x,y))<abs(c1)&&abs(theta_sub(x,y))>=abs(c2))

ori(p)=(c1+c2)/2;
magcount=magcount+mag_sub(x,y);
end
else
if(abs(theta_sub(x,y))>abs(c1)&&abs(theta_sub(x,y))<=abs(c2))
ori(p)=(c1+c2)/2;
magcount=magcount+mag_sub(x,y);
end
end
end
magcounts(angle/45+1)=magcount;
end
        descriptor((area-1)*8+1:area*8)=magcounts;
end
        descriptor=normr(descriptor);
% cap 0.2
```

```
for j=1:numel(descriptor)
if(abs(descriptor(j))>0.2)
            descriptor(j)=0.2;
end
end
        descriptor=normr(descriptor);
else
continue;
end
        feature(:,i)=descriptor';
end
index=find(sum(feature));
feature=feature(:,index);end
```

5.6 Implementation of Fuzzy C-means Clustering

Fuzzy clustering (also referred to as soft clustering or soft *k*-means) is a form of clustering in which each data point can belong to more than one cluster. Clustering or cluster analysis involves assigning data points to clusters such that items in the same cluster are as similar as possible, while items belonging to different clusters are as dissimilar as possible. Clusters are identified via similarity measures. These similarity measures include distance, connectivity, and intensity. Different similarity measures may be chosen based on the data or the application. Here is the code for implementation of FUZZY C-means clustering.

```
dist=zeros(cluster_n,n_r);dist2=zeros(cluster_n,data_n);
for w= 1:max_iter
mf = Num.*(U.^expo);
center = mf*data_u./((ones(size(data, 2), 1)*sum(mf'))');
for k=1: size(center, 1)
dist(k, :)=abs(center(k)-data_u)';
end
tmp=dist.^2;
h1=(tmp+eps).^(-1/(expo-1));
U=(h1)./(ones(cluster_n,1)*(sum(h1))+eps);
if w>2
sum_U{w}=double(U>0.5);
obj_U(w)=sum(sum(abs(sum_U{w}-sum_U{w-1})));
ifobj_U(w)==0,break; end,
end
end
iter_n = w; % Actual number of iterations
```

5.7 Gaussian Filter

The Gaussian blur is a type of image-blurring filter that uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image.

```
%% Gaussian filter
Filter_image = imgaussfilt(f_seg,.6);
% J = medfilt2(f_seg,[3 3]);
```

5.8 Feature Comparison

Once when both the images- Forged and Original are used from the MICC220 dataset, the features extracted are compared. This comparison yields us with the output of the different sections of the image where differences are appearing**.**

```
%% Feature comparison
Diffrence = Filter_image(1:R,1:C)-Feature_org(1:R,1:C);
Performance = perf(image,org_image);
Precision = Performance.precision
Recall = Performance.recall
F_Measure = Performance.Fmeasure
Resutl = sum(sum(Diffrence));
if abs(Resutl)<=10
disp('The Input Image Is Original');
msgbox('The Input Image Is Original');
else
disp('Input Image is Forged ');
msgbox(' Input Image is Forged')
tLabeled_Image = image - org_image;
Labeled_Image = tLabeled_Image>5;
    se = strel('disk',10);
Merged_Regions = imclose(Labeled_Image,se);
```

```
    boundaries = bwboundaries(Merged_Regions);
numberOfBoundaries = length(boundaries);
figure('Name','Detected Forged Region','NumberTitle','Off');
imshow(rgbimage);
    axis off;hold on;
    for i1 = 1:numberOfBoundaries
thisBoundary = boundaries{i1};
        plot(thisBoundary(:,2), thisBoundary(:,1), 'color' , 'r' , 'LineWidth', 2);
    end
title('Detected Forged Region','fontname','Times New Roman','fontsize',12);
end
```

5.9 Summary

Finally, after the feature comparison is done we get the desired output on difference, performance, precision and performance recall of the to images used.

Only one image, showing the forged or additional irrelevant cluster of the image is shown as output.

In the following chapters, testing and snapshots of the actual outputs are being shown for the reference.

CHAPTER 6

TESTING AND SNAPSHOTS

The final section of the concerns the implementation of results. A result is the final consequence of sequences of actions or events expressed qualitatively or quantitatively. A discussion is a collection of topics. In a discussion terms can converse and exchange ideas over an extended period.

The snapshots of our obtained result are shown below

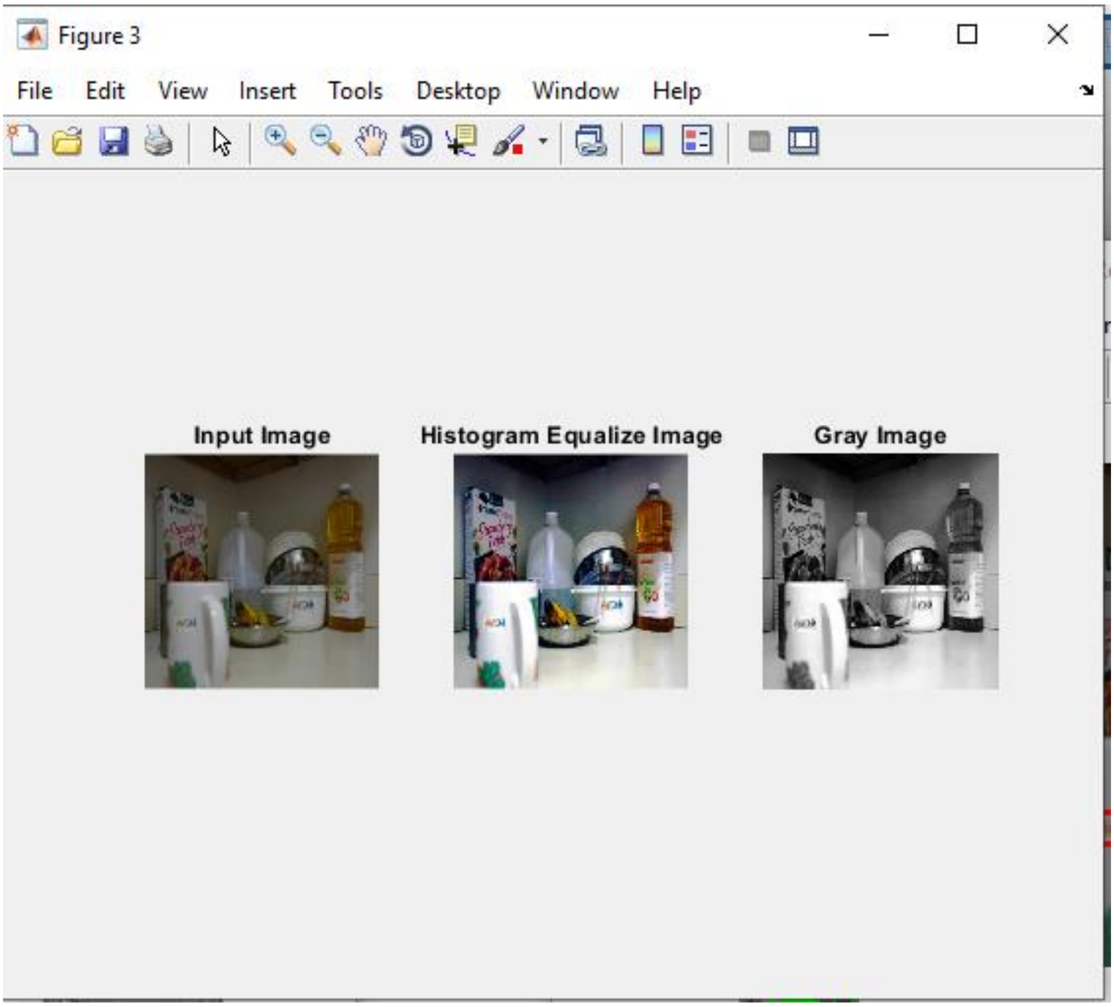


Figure 6.1: Snapshot of conversion of image

Figure 6.1 shows the conversion of the input image into gray scale image

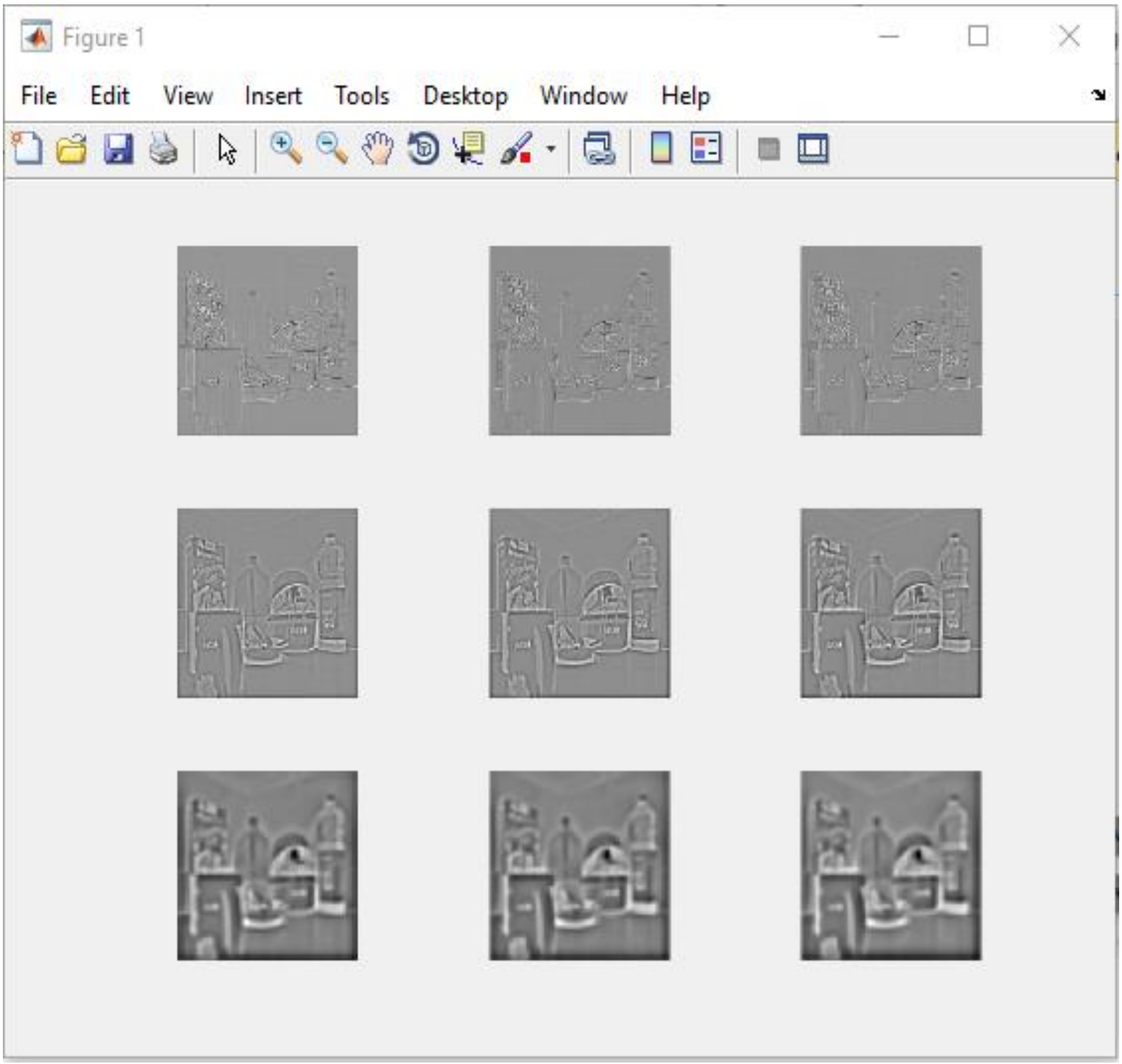


Figure 6.2: Snapshot of Difference of Gaussian.

Figure 6.2 shows the difference of Gaussian image, it's done for the subtraction for just one octave. The same thing is done for all octaves. This generates DoG images of multiple sizes.

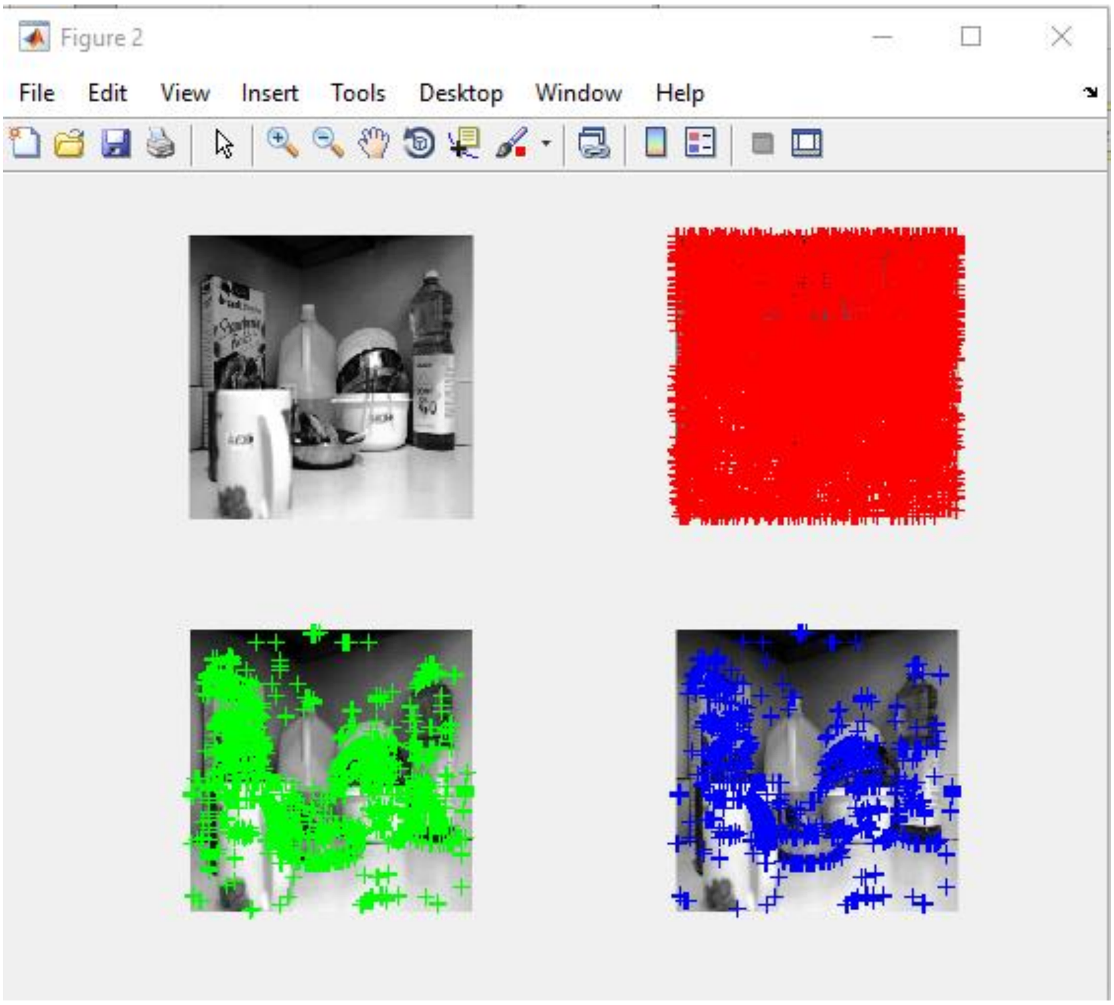


Figure 6.3: Displays the KeypointLocalization.

Figure 6.3 shows the keypoint localization which searches each pixel in the DoG map to find the extreme point.

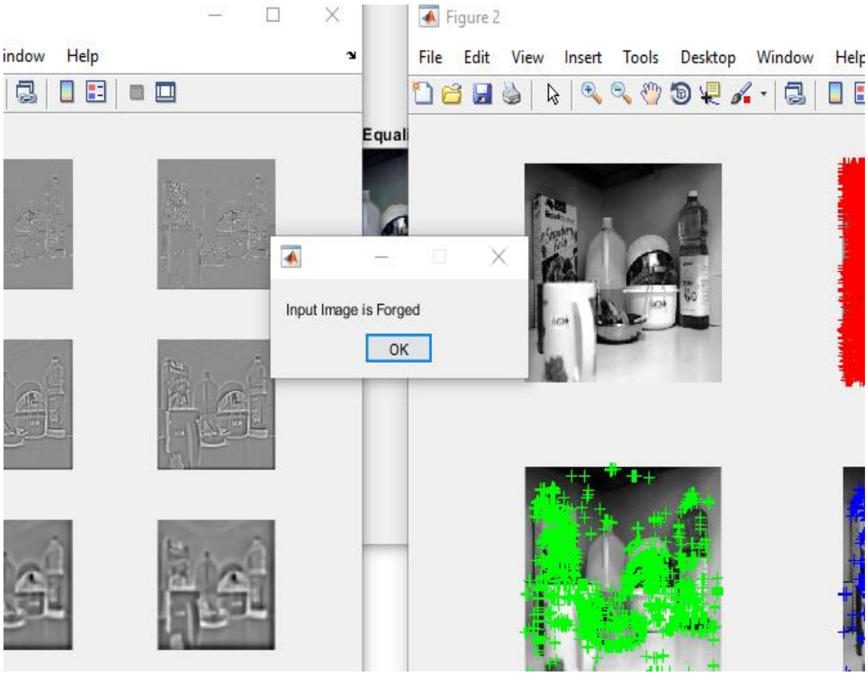


Figure 6.4: Result of Forged Image.

Figure 6.4 shows the displayed message i.e., the input image is forged

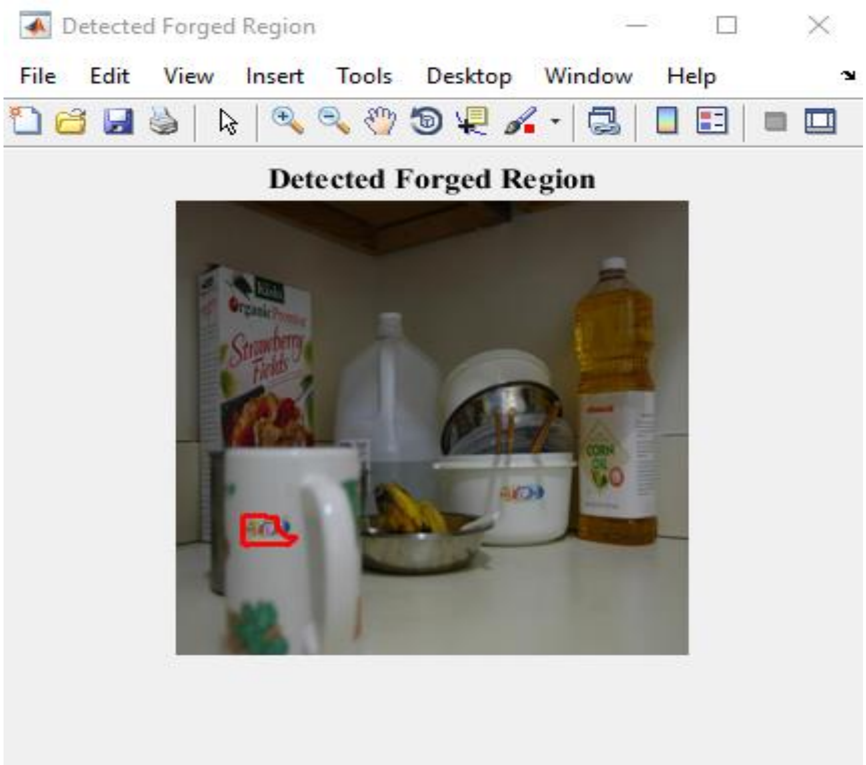


Figure 6.5: Detected Forged Region

Figure 6.5 shows the forged region which is detected after finding the input image is forged

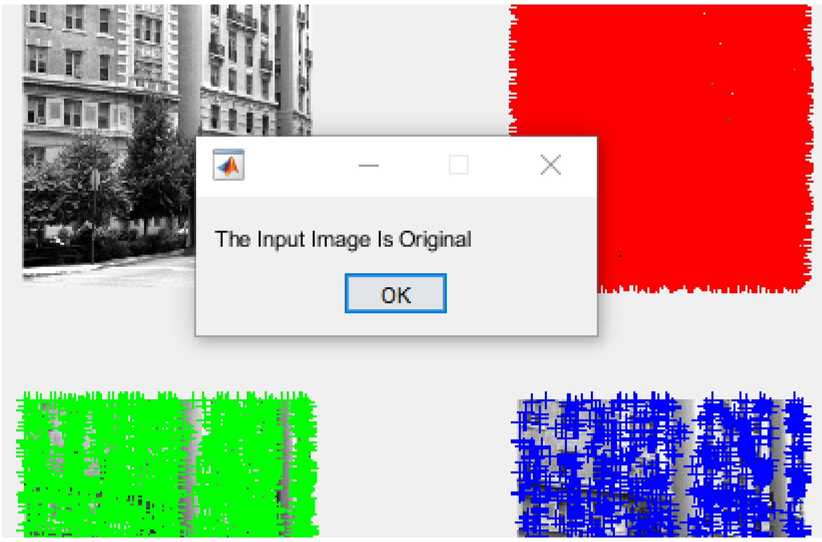


Figure 6.6: Result of Original Image

Figure 6.6 shows the displayed message i.e, the input image is original. Same process is done to check the input image is original.

6.1 Summary

This chapter shows the snapshots of the conversion to gray scale image, filtering of Gaussian, difference of Gaussian and identifying keypoints and the result of given input image is original or tampered one.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

The project on “Image Forgery Detection Using Fast Scale Invariant Feature Transform Method” has the essentials of image forgery detection technique in an image forensics has been discussed. As concluded copy move forgery detection is an important area of image processing for several security reasons. From the study it was evident that all such categories of image forgery technique have their potential adversarial feature depending upon the scale of vulnerability of the victim. The study has also discussed about the forgery detection technique i.e., scale invariant feature transform method. It optimizes the key point-based techniques for detection of Copy Move forgery. While raising the number of key points, the computational requirements will raise in these techniques, so minimal execution time will be needed. So, optimization of FCM technique for clustering the SIFT key points to decrease time complexity. It decreases the detection time of appreciably same accuracy standards and minor enhancement in some cases. It detects also in the status of rotation, scaling and multiple Copy Move attacks.

7.1 Future Scope

In this work, an image forgery detection is presented and it uses extraction of an SIFT features. There are many various available techniques of a Copy Move Forgery detection attacks, here SIFT technique were used but still it can be extended by several methods. A new data set can be created for CMFD that includes more manipulated pictures that were performed deliberately by professionals. The obtained data set is an open source and free to be optimized as benchmarking for more comparisons and it is highly recommended that optimizing multiple clustering algorithms or even using the FCM by matrix optimization rather than the sequential optimization done.

REFERENCES

[1]	Hesham A. Alberry, Abdelfatah A. Hegazy,Gouda I. Salama,et al; “A fast SIFT based method for copy move forgery detection”, future computing and informatics journal, volume 3, Issue 2, PP.159-165, December 2018.
[2]	AbdWarif NB, AbdulWahab AW, Idris MY, Ramli R, Salleh R, ShamshirbSh, et al; “Copy-Move Forgery Detection: Survey, Challenges and Future Directions”, Journal of Network Computer and Applications, PP. 75, September 2016.
[3]	Qasim, AsaadF, MezianeFarid, Aspin Rob et al; “Digital Watermarking: Applicability for developing trust in medical imaging workflows state of the art review”, Computer Science Review 27, PP.45-60, February 2018.
[4]	Kumar S, Desai J, Kumar S, Desai J et al; “A fast DCT based method for copy move forgery detection”, IEEE Second International Conference on Image Information Processing, December 2013.
[5]	BakiahAbdWarif, Nor, Abdul WahabAinuddin Wahid, IdrisMohd Yamani Idna, SallehRosli, Othman Fazidah; “SIFT-Symmetry: A Robust Detection Method for Copy-Move Forgery with Reflection Attack”, Journal of Visual Communication and Image Representation, April 2017.
[6]	Upase SG, Kuntawar SV; “Copy-Move Detection of Image Forgery by using DWT and SIFT Methodologies”, Computer Application, January 2016.
[7]	Raj R, Josephb N; “Keypoint Extraction Using SURF Algorithm for CMFD”, Procedia Computer Science, Volume 93, PP.375-381, 2016.