

PERCEPTUAL IMAGE HASHING USING SURF FOR FEATURE EXTRACTION AND ENSEMBLE CLASSIFIER

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Technology Degree in

Information Technology

By

PADMASHRI R (37120052)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University)**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMIL NADU**

MARCH 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with “A” grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai –
600 119
www.sathyabama.ac.in



SCHOOL OF COMPUTING
BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of PADMASHRI R (37120052) who underwent the Professional Training in “**Perceptual Image Hashing Using Surf for Feature Extraction and Ensemble Classifier**” under our supervision from December 2020 to March 2021.

Internal guide

Dr SENDURU SRINIVASULU, M.TECH., Ph.D.,

Head of the Department

Dr R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva Voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, PADMASHRI R (37120052) hereby declare that the Professional Training Report on **“Perceptual Image Hashing Using Surf for Feature Extraction and Ensemble Classifier”** done by me under the guidance of Dr Senduru Srinivasulu, M.TECH., Ph.D., at Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in **Information Technology**.

DATE:

PLACE:

SIGNATURE OF CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **the Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean, School of Computing , Dr.R.Subhashini M.E., Ph.D. , Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Senduru Srinivasulu, M.Tech., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Image hash regimes have been widely used for authenticating content, recovery of images and digital forensics. In this article we propose a new algorithm for image haunting (SSL) with the most stable key points and regional features, strong against various manipulation of content conservation, including multiple combinatorial manipulations. In order to extract most stable key point, the proposed algorithm combines the Speed Up Robust Features (SURF) with Saliency detection. The keyboards and characteristics of the local area are then combined in a hash vector. There is also a sperate secret key that is randomly given for the hash vector to prevent an attacker from shaping the image and the new hash value. The proposed hacking algorithm shows that similar or initial images, which have been individually manipulated, combined and even multiple manipulated contents, can be visently identified by experimental result. The probability of collision between hacks of various images is almost nil. Furthermore, the key-dependent security assessment shows the proposed regime safe to allow an attacker without knowing the secret key not to forge or estimate the right havoc value.

CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	4
	LIST OF FIGURES	7
	LIST OF ABBREVIATIONS	7
1	INTRODUCTION	8
2	LITERATURE SURVEY	9
3	THE IMAGE PROCESSING SYSTEM	12
	3.1 DIGITIZER	12
	3.2 IMAGE PROCESSOR	12
	3.2.1 DIGITAL COMPUTER	
	3.2.2 MASS STORAGE	
	3.2.3 HARD COPY DEVICE	
	3.2.4 OPERATOR CONSOLE	
	3.3 IMAGE PROCESSING FUNDAMENTAL	14
	3.3.1 IMAGE PROCESSING TECHNIQUES	
	3.3.2 IMAGE ENHANCEMENT	
	3.3.3 IMAGE RESTORATION	
	3.3.4 IMAGE ANALYSIS	
	3.3.5 IMAGE COMPRESSION	
	3.3.6 IMAGE SYNTHESIS	
	3.3.7 APPLICATIONS OF DIGITAL IMAGE PROCESSING	
	3.3.8 MEDICAL APPLICATIONS	
	3.3.9 SATELLITE IMAGING	
	3.3.10 COMMUNICATION	
	3.3.11 RADAR IMAGING SYSTEMS	
	3.3.12 DOCUMENT PROCESSING	
	3.3.13 DEFENSE/INTELLIGENCE	
	3.4 EXISTING SYSTEMS	17
	3.4.1 EXISTING SYSTEMS DISADVANTAGES	
	3.5 PROPOSED METHOD	18
	3.5.1 PROPOSED SYSTEM ADVANTAGES	
4	METHODOLOGY	19
	4.1 GENERAL	19
	4.1.1 TRAINING (DATABASE)	
	4.1.2 TESTING (QUERY INPUT IMAGE)	
	4.1.3 CLASSIFIER	
	4.2 SVM CLASSIFIER	21
	4.2.1 APPLICATIONS OF SVM	

	4.3 KNN CLASSIFIER	24
	4.4 IMAGE FORGERIES	25
	4.4.1 CLASSIFICATION OF DIFFERENT IMAGE FORGERIES DETECTION ALGORITHMS	
	4.4.2 PRINCIPAL COMPONENT ANALYSIS	
	4.4.3 IMAGE CLASSIFICATION	
	4.4.4 UNSUPERVISED CLASSIFICATION	
	4.4.5 SUPERVISED CLASSIFICATION	
	4.5 SYSTEM REQUIREMENTS	31
	4.5.1 HARDWARE REQUIREMENTS	
	4.5.2 SOFTWARE REQUIREMENTS	
	4.5.3 SYSTEM ARCHITECTURE	
	4.6 MATLAB SPECIFICATIONS	33
	4.6.1 FEATURES OF MATLAB	
	4.6.2 INTERFACING WITH OTHER LANGUAGES	
	4.6.3 THE MATLAB LANGUAGE	
	4.6.4 DEVELOPMENT TOOLS	
	4.6.5 MATLAB EDITOR	
	4.6.6 CODE ANALYZER	
	4.6.7 MATLAB PROFILER	
	4.6.8 DIRECTORY REPORTS	
	4.6.9 DESIGNING GRAPHICAL USER INTERFACES	
	4.7 ANALYSING AND ACCESSING DATA	36
	4.7.1 DATA ANALYSIS	
	4.7.2 DATA ACCESS	
	4.7.3 VISUALIZING DATA	
	4.7.4 2-D PLOTTING	
	4.7.5 3-D PLOTTING AND VOLUME VISUALIZATION	
	4.8 PERFORMING NUMERIC COMPUTATION	38
5	RESULTS AND DISCUSSION	39
6	CONCLUSION AND FUTUREWORK	41
	REFERENCES	42
	APENDDIX	
	A. SOURCE CODE	43
	B. SCREEN SHOTS	47
	C. PUBLICATION WITH PLAGIARISM REPORT	49

LIST OF FIGURES

FIGURES	TITLES	PAGE NO.
Fig 3.1	BLOCK DIAGRAM FOR IMAGE PROCESSING SYSTEM	12
Fig 3.2	BLOCK DIAGRAM OF FUNDAMENTAL SEQUENCE INVOLVED IN AN IMAGE PROCESSING SYSTEM	13
Fig 3.3	IMAGE PROCESSING TECHNIQUES	15
Fig. 3.4	EXAMPLE OF IMAGE SLICING	18
Fig.4.1	CLUSTERING ALGORITHM VECTOR	22
Fig.4.2	KERNEL MACHINE	23
Fig. 4.3	MODIFIED IMAGE	26
Fig. 4.4	ORIGINAL IMAGE	26
Fig.4.5	BASIC STEPS IN SUPERVISED	31
Fig.4.6	SYSTEM ARCHITECTURE	33
Fig.5.1	ORIGINAL IMAGE HASH INITIALIZING PROCESS	39
Fig 5.2	FORGERY IMAGE DETECTING PROCESS	40

LIST OF ABBREVIATIONS

NMR	: NUCLEAR MAGNETIC RESONANCE
MSD	: MEAN SQUARED DIFFERENCE
GPVE	: GENERALIZED PARTIAL VOLUME ESTIMATION
MI	: MUTUAL INFORMATION
RANSAC	: RANDOM SAMPLE CONSENSUS
GPU	: GRAPHICS PROCESSING UNIT
SURF	: SPEEDED-UP ROBUST FEATURE
SIFT	: SCALE INVARIANT FEATURE TRANSFORM
KNN	: K-NEAREST NEIGHBORS ALGORITHM
SVM	: SUPPORT VECTOR MACHINE

CHAPTER 1

INTRODUCTION

The term digital image refers to processing of a two-dimensional picture by a digital computer. In a broader context, it implies digital processing of any two-dimensional data. A digital image is an array of real or complex numbers represented by a finite number of bits. An image given in the form of a transparency, slide, photograph or an X-ray is first digitized and stored as a matrix of binary digits in computer memory. This digitized image can then be processed and/or displayed on a high-resolution television monitor. For display, the image is stored in a rapid-access buffer memory, which refreshes the monitor at a rate of 25 frames per second to produce a visually continuous display.

Digital images and videos have been increasingly exposed as important information or art carriers in our daily life. Despite easy and cheap to acquire, distribute and store, the threats of abuse are high, which if not carefully solved, will lead to great loss of property, fame, and even life. Images or videos can be cloned illegally, imperceptibly modified using image processing tools or even fabricated with the help of artificial intelligence (AI) to distort the truth to mislead people or clinch wrongful convictions in a court of law. However, lacking original data information makes these methods computationally intensive and very time consuming, often with low accuracy of detection. Perceptual Image Identification is mainly achieved using machine learning based methods, which basically follow three steps: image feature extraction, classifier training and image source class prediction.

Tamper detection and source camera identification are achieved simultaneously by using perceptual image hashing and PUF in a simple but effective way.

The reason that has motivated to work on this Project is to mainly focus towards the goal to achieve image tampering i.e., Image blackmailing crimes that has been leading to victim's trauma, mental health, self-hurt, losing money, bribes or more even suicide rates that have increased. Police and government have reported that these images tampered, fake product sale, private media leakage has been doubled over the years now. So, the Proposed Project helps to recover original data of an image and prevents Image Tampering crimes.

CHAPTER 2

LITERATURE SURVEY

TITLE: “Scene illumination as an indicator of image manipulation” IEEE,” *Inf. Hiding*, vol. 6387, pp. 66–80.

AUTHOR: C. Riess and E. Angelopoulou

YEAR: 2016

DESCRIPTION:

The goal of blind image forensics is to distinguish original and manipulated images. We propose illumination colour as a new indicator for the assessment of image authenticity. Many images exhibit a combination of multiple illuminances (ash photography, mixture of indoor and outdoor lighting, etc.). In the proposed method, the user selects illuminated areas for further investigation. The illuminated colours are locally estimated, electively decomposing the scene in a map of divergently illuminated regions. Inconsistencies in such a map suggest possible image tampering. Our method is physics-based, which implies that the outcome of the estimation can be further constrained if additional knowledge on the scene is available. Experiments show that these illumination maps provide a useful and very general forensics tool for the analysis of colour images.

TITLE: “Image forensic analyses that elude the human visual system”, IEEE in Proc. Symp. Electron. Imaging (*SPIE*), pp. 1–10.

AUTHOR: H. Farid and M. J. Bravo

DATE: 2010

DESCRIPTION:

When creating a digital composite of, for example, two people standing side-by-side, it is often difficult to match the lighting conditions from the individual photographs. Lighting inconsistencies can therefore be a useful tool for revealing traces of digital tampering. Borrowing and extending tools from the field of computer vision, we describe how the direction of a point light source can be estimated from only a single image. We show the secrecy of this approach in real-world settings.

TITLE: “Perceiving illumination inconsistencies in scenes”, IEEE, Perception, vol. 34, no. 11, pp. 1301–1314.

AUTHOR: Y. Ostrovsky, P. Cavanagh, and P. Sinha

DATE: 2015

DESCRIPTION:

We present a physics-based approach for illuminance colour estimation of arbitrary images, which is explicitly designed for handling images with multiple illuminances. The majority of techniques that extract the illuminate colour assume that the illumination is constant across the scene. This, however, is not often the case. We propose an illuminated-colour estimation method which is based on robust local illuminance estimates. There are no assumptions on the number or type of illuminatus. An illuminated colour estimate is obtained independently from distinct image mini-regions. From these mini-regions a robust local illumination colour is computed by consensus. These local estimates are then used in deriving the chromaticity of the dominant illuminance. Experiments on an established benchmark database of real-world images show that our technique performs comparably to uniform-illuminance estimation

methods. Furthermore, extensive tests on real-world images show that we can reliably process mixed illuminance scenes.

CHAPTER 3

THE IMAGE PROCESSING SYSTEM:

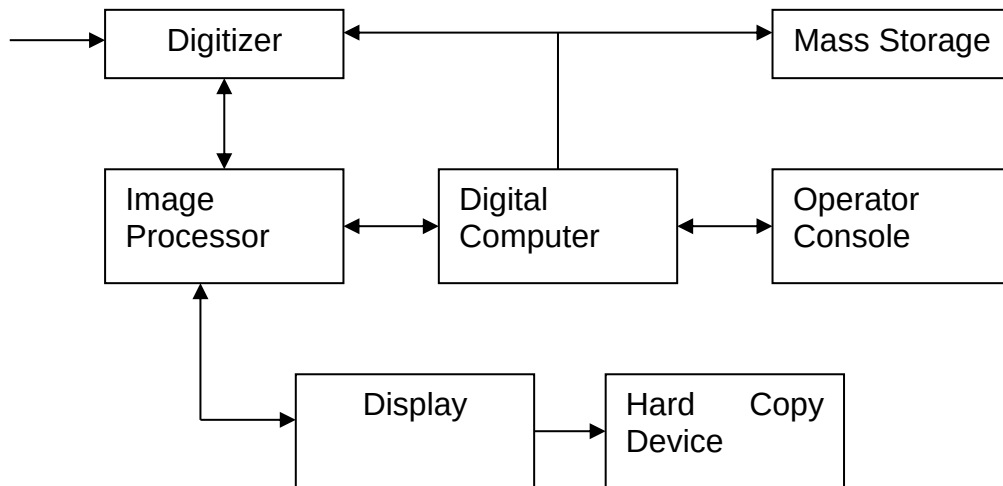


Fig 3.1 Block Diagram for Image Processing System

3.1 DIGITIZER

A digitizer converts an image into a numerical representation suitable for input into a digital computer. Some common digitizers are

1. Microdensitometer
2. Flying spot scanner
3. Image dissector
4. Videocon camera
5. Photosensitive solid- state arrays.

3.2 IMAGE PROCESSOR

An image processor does the functions of image acquisition, storage, pre-processing, segmentation, representation, recognition and interpretation and finally displays or records the resulting image. The following block diagram gives the fundamental sequence involved in an image processing system.

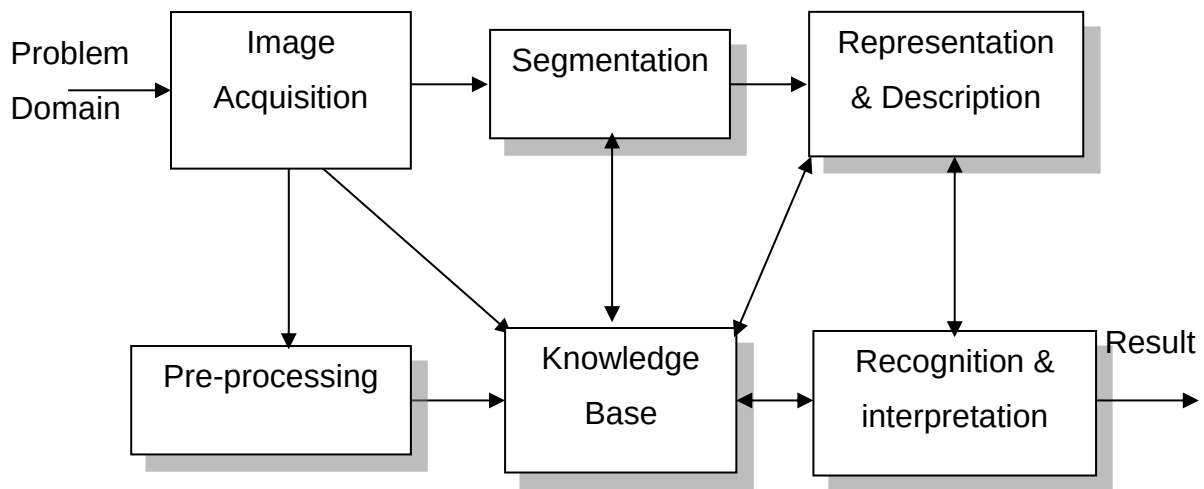


Fig 3.2 Block Diagram of Fundamental Sequence Involved in An Image Processing System

As detailed in the diagram, the first step in the process is image acquisition by an imaging sensor in conjunction with a digitizer to digitize the image. The next step is the pre-processing step where the image is improved being fed as an input to the other processes. Pre-processing typically deals with enhancing, removing noise, isolating regions, etc. Segmentation partitions an image into its constituent parts or objects. The output of segmentation is usually raw pixel data, which consists of either the boundary of the region or the pixels in the region themselves. Representation is the process of transforming the raw pixel data into a form useful for subsequent processing by the computer. Description deals with extracting features that are basic in differentiating one class of objects from another. Recognition assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects. The knowledge about a problem domain is incorporated into the knowledge base. The knowledge base guides the operation of each processing module and also controls the interaction between the modules. Not all modules need be necessarily present for a specific function. The composition of the image processing system depends on its application. The frame rate of the image processor is normally around 25 frames per second.

3.2.1 DIGITAL COMPUTER

Mathematical processing of the digitized image such as convolution, averaging, addition, subtraction, etc. are done by the computer.

3.2.2 MASS STORAGE

The secondary storage devices normally used are floppy disks, CD ROMs etc.

3.2.3 HARD COPY DEVICE

The hard copy device is used to produce a permanent copy of the image and for the storage of the software involved.

3.2.4 OPERATOR CONSOLE

The operator console consists of equipment and arrangements for verification of intermediate results and for alterations in the software as and when require.

The operator is also capable of checking for any resulting errors and for the entry of requisite data.

3.3 IMAGE PROCESSING FUNDAMENTAL

Digital image processing refers processing of the image in digital form. Modern cameras may directly take the image in digital form but generally images are originated in optical form. They are captured by video cameras and digitalized. The digitalization process includes sampling, quantization. Then these images are processed by the five fundamental processes, at least any one of them, not necessarily all of them.

3.3.1 IMAGE PROCESSING TECHNIQUES

This section gives various image processing techniques

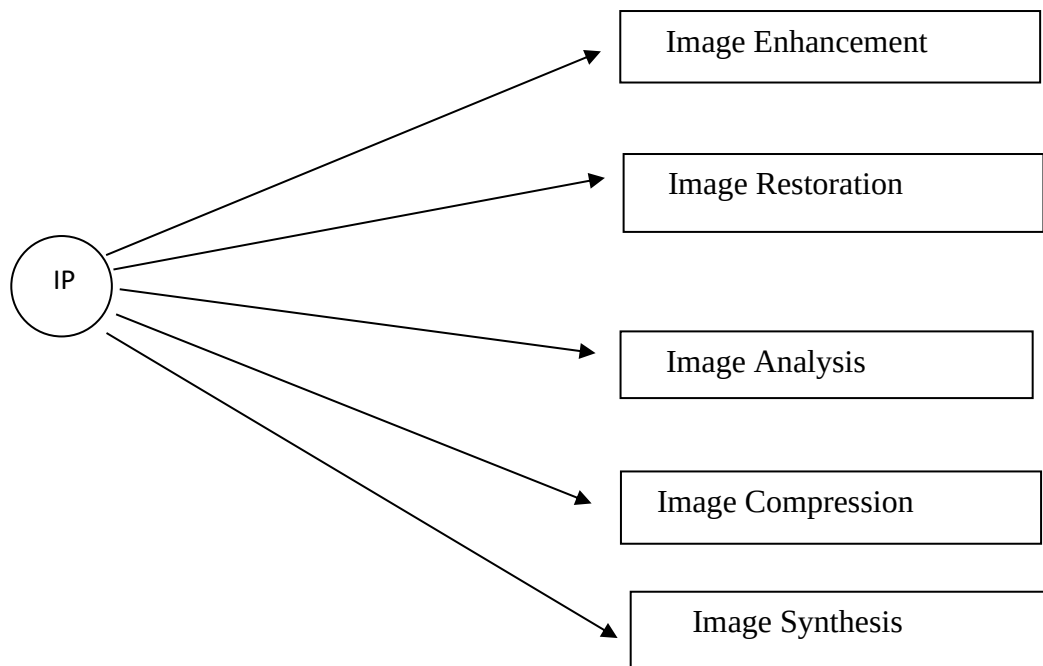


Fig 3.3 Image Processing Techniques

3.3.2 IMAGE ENHANCEMENT

Image enhancement operations improve the qualities of an image like improving the image's contrast and brightness characteristics, reducing its noise content, or sharpen the details. This just enhances the image and reveals the same information in more understandable image. It does not add any information to it.

3.3.3 IMAGE RESTORATION

Image restoration like enhancement improves the qualities of image but all the operations are mainly based on known, measured, or degradations of the original image. Image restorations are used to restore images with problems such as geometric distortion, improper focus, repetitive noise, and camera motion. It is used to correct images for known degradations.

3.3.4 IMAGE ANALYSIS

Image analysis operations produce numerical or graphical information based on characteristics of the original image. They break into objects and then classify them. They depend on the image statistics. Common operations are extraction and

description of scene and image features, automated measurements, and object classification. Image analyze are mainly used in machine vision applications.

3.3.5 IMAGE COMPRESSION

Image compression and decompression reduce the data content necessary to describe the image. Most of the images contain lot of redundant information, compression removes all the redundancies. Because of the compression the size is reduced, so efficiently stored or transported. The compressed image is decompressed when displayed. Lossless compression preserves the exact data in the original image, but Lossy compression does not represent the original image but provide excellent compression.

3.3.6 IMAGE SYNTHESIS

Image synthesis operations create images from other images or non-image data. Image synthesis operations generally create images that are either physically impossible or impractical to acquire.

3.3.7 APPLICATIONS OF DIGITAL IMAGE PROCESSING

Digital image processing has a broad spectrum of applications, such as remote sensing via satellites and other spacecrafts, image transmission and storage for business applications, medical processing, radar, sonar and acoustic image processing, robotics and automated inspection of industrial parts.

3.3.8 MEDICAL APPLICATIONS

In medical applications, one is concerned with processing of chest X-rays, cine angiograms, projection images of trans axial tomography and other medical images that occur in radiology, nuclear magnetic resonance (NMR) and ultrasonic scanning. These images may be used for patient screening and monitoring or for detection of tumors or other disease in patients.

3.3.9 SATELLITE IMAGING

Images acquired by satellites are useful in tracking of earth resources; geographical mapping; prediction of agricultural crops, urban growth and weather;

flood and fire control; and many other environmental applications. Space image applications include recognition and analysis of objects contained in image obtained from deep space-probe missions.

3.3.10 COMMUNICATION

Image transmission and storage applications occur in broadcast television, teleconferencing, and transmission of facsimile images for office automation, communication of computer networks, closed-circuit television-based security monitoring systems and in military communications.

3.3.11 RADAR IMAGING SYSTEMS

Radar and sonar images are used for detection and recognition of various types of targets or in guidance and manoeuvring of aircraft or missile systems.

3.3.12 DOCUMENT PROCESSING

It is used in scanning, and transmission for converting paper documents to a digital image form, compressing the image, and storing it on magnetic tape. It is also used in document reading for automatically detecting and recognizing printed characteristics.

3.3.13 DEFENSE/INTELLIGENCE

It is used in reconnaissance photo-interpretation for automatic interpretation of earth satellite imagery to look for sensitive targets or military threats and target acquisition and guidance for recognizing and tracking targets in real-time smart-bomb and missile-guidance systems.

3.4 EXISTING SYSTEMS

This technique for making forgery images is more aggressive than image retouching. Image splicing is fundamentally simple process and can be done as crops and pastes regions from the same or separate sources. This method refers to a paste-up produced by sticking together images using digital tools available such as Photoshop. In Image Splicing technique there is composition of two or more images, which are combined to create a fake image.

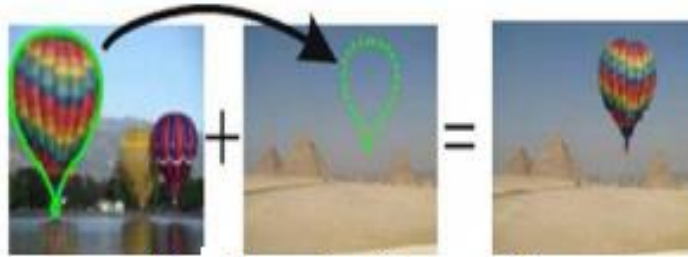


Fig. 3.4 Example of Image Slicing

3.4.1 EXISTING SYSTEMS DISADVANTAGES

- Less accurate in gray
- restricted to Image certain format
- Higher computational complexity and detect small size and multiple copy-move forgery.

3.5 PROPOSED METHOD

- In Preprocessing, image intensity, grayscale conversion, binary form & if image is affected by noise, it can be removed.
- Gathering the key points from the corresponding image by using SIFT
- Segment the object by using the segmentation technique. it's done by using SVM clustering method.
- Image is kept taken for feature extraction, in that wavelet, PCA & GLCM features values are gathered.
- These feature data are combined in form of single matrix & applied to input for the classifier as test feature data, train feature data.

3.5.1 PROPOSED SYSTEM ADVANTAGES

- Faster and Accurate Approach.
- Can be working for multiple images at the time.
- The system enables to work with all format and types of images with requires less time to execute it.

CHAPTER 4

METHODOLOGY

4.1 GENERAL

4.1.1 Training (Database)

- Dataset image's is taken for reading, after that those images are pre-processed.
- In Preprocessing, image intensity, grayscale conversion, binary form & if image is affected by noise, it can be removed.
- After that Gathering the key points from the corresponding image by using SIFT (Scale-invariant feature transform) - [to detect and describe local features in images]
- We need to segment the object by using the segmentation technique. it's done by using FCM- fuzzy c means clustering method.
- After segmented, image is kept taken for feature extraction segmented image is applied to DWT (first level) in that image is decomposed into 4 forms such as approximation, horizontal, vertical, diagonal/detail coefficients. In that approximation coefficients will have a complete detail about the entire image. so we take that particular part for further processing in second level DWT, approximation coefficient of first level DWT as input.
- In that image is decomposed into 4 forms such as approximation, horizontal, vertical, diagonal/detail coefficients. In that approximation coefficients will have a complete detail about the entire image. So, we take that particular part for further processing in third level DWT, approximation coefficient of second level DWT as input. In that image is decomposed into 4 forms such as approximation, horizontal, vertical, diagonal/detail coefficients. In that approximation coefficients will have a complete detail about the entire image.

- So, we take that particular part for further processing Output of third level DWT is applied to input for PCA. Main purpose of using PCA is for dimension reduction.
- PCA output is applied to GLCM for calculating the parameters like energy, correlation, skewness, homogeneity, mean, variance, standard deviation etc... like we calculate around 12 or 13 parameters for feature extraction for each image in data set. These feature data are combined in form of single matrix ----- called as trained feature (dataset)

4.1.2 Testing (Query Input Image)

- Query input image is taken for reading, after that those images are pre-processed.
- In Preprocessing, image intensity, grayscale conversion, binary form & if image is affected by noise, it can be removed.
- After that Gathering the key points from the corresponding image by using SIFT (Scale-invariant feature transform) - [to detect and describe local features in images]
- We need to segment the object by using the segmentation technique. its done by using FCM- fuzzy c means clustering method.
- After segmented, image is kept taken for feature extraction segmented, image is applied to DWT (first level) in that image is decomposed into 4 forms such as approximation, horizontal, vertical, diagonal/detail coefficients. In that approximation coefficients will have a complete detail about the entire image. so we take that particular part for further processing in second level DWT, approximation coefficient of first level DWT as input. In that image is decomposed into 4 forms such as approximation, horizontal, vertical, diagonal/detail coefficients. In that approximation coefficients will have a complete detail about the entire image. so, we take that particular part for further processing in third level DWT, approximation coefficient of second level DWT as input. In that image is decomposed into 4 forms such as approximation, horizontal, vertical, diagonal/detail coefficients. In that approximation coefficients will have a complete

detail about the entire image. So, we take that particular part for further processing. Output of third level DWT is applied to input for PCA. Main purpose of using PCA is for dimension reduction.

- PCA output is applied to GLCM for calculating the parameters like energy, correlation, skewness, homogeneity, mean, variance, standard deviation etc... like we calculate around 12 or 13 parameters for feature extraction for each image in data set. These feature data are combined in form of single matrix ----- called as test feature (dataset)

4.1.3 Classifier

While initiating the classifier, it needs 3 inputs. They are test feature data, train (dataset) feature, train labels (i.e., categories) are applied to the classifier as input. Classifier will compare the test feature data, train (dataset) feature along with train labels. After that Classifier will predict the input query image is an edited image or normal image by SVM / KNN Classifier.

There are 2 classifiers that are beneficial for this process which are:

- *SVM Classifier*
- *KNN Classifier*

4.2 SVM CLASSIFIER

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use **SVM** in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as

possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are not labelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support vector clustering algorithm created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabelled data, and is one of the most widely used clustering algorithms in industrial applications.

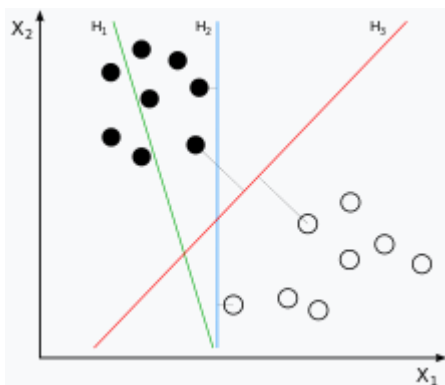


Fig.4.1 Clustering Algorithm Vector

H_1 does not separate the classes. H_2 does, but only with a small margin. H_3 separates them with the maximum margin.

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

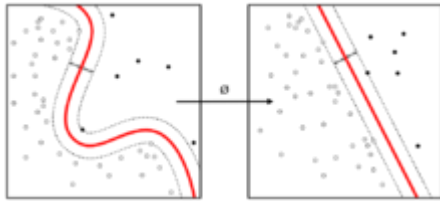


Fig.4.2 Kernel machine

Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel unction $k(x, y)$ selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters (α) of images of feature vectors X_i that occur in the data base.

4.2.1 Applications of SVM:

- SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labelled training instances in both the standard inductive and transudative settings.
- Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback. This is also true of image segmentation systems, including those using a modified version SVM that uses the privileged approach as suggested by Vapnik.
- Hand-written characters can be recognized using SVM.

4.3 KNN CLASSIFIER

In pattern recognition, the k-nearest neighbour's algorithm (k-NN) is a *non-parametric* method used for **classification** and **regression**. In both cases, the input consists of the k closest training examples in the *feature space*. The output depends on whether k-NN is used for classification or regression:

- In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbour's, with the object being assigned to the class most common among its k nearest neighbour's (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbour.
- In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbours.

k-NN is a type of **instance-based learning**, or **lazy learning**, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all *machine learning* algorithms.

Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbours, so that the nearer neighbours contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbour a weight of $1/d$, where d is the distance to the neighbour.

The neighbours are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data. The algorithm is not to be confused with *k-means*, another popular *machine learning* technique.

Description: -

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the *feature vectors* and class labels of the training samples.

In the classification phase, k is a user-defined constant, and an unlabelled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point.

A commonly used distance metric for **continuous variables** is *Euclidean distance*. For discrete variables, such as for text classification, another metric can be used, such as the overlap metric (or *Hamming distance*). In the context of gene expression microarray data, for example, k -NN has also been employed with correlation coefficients such as Pearson and Spearman. Often, the classification accuracy of k -NN can be improved significantly if the distance metric is learned with specialized algorithms such as *Large Margin Nearest Neighbor* or *Neighbourhood components analysis*.

A drawback of the basic "majority voting" classification occurs when the class distribution is skewed. That is, examples of a more frequent class tend to dominate the prediction of the new example, because they tend to be common among the k nearest neighbour's due to their large number. One way to overcome this problem is to weight the classification, taking into account the distance from the test point to each of its k nearest neighbours. The class (or value, in regression problems) of each of the k nearest points is multiplied by a weight proportional to the inverse of the distance from that point to the test point. Another way to overcome skew is by abstraction in data representation. For example, in a *self-organizing map* (SOM), each node is a representative (a centre) of a cluster of similar points, regardless of their density in the original training data. K -NN can then be applied to the SOM.

4.4 IMAGE FORGERIES

Faked images are created from the original image or from the original image itself some modifications are done by using high graphical software.

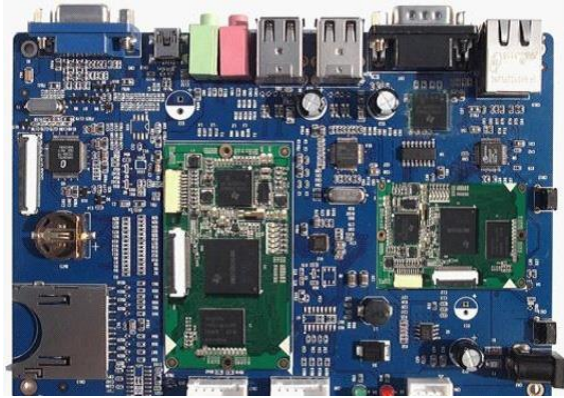


Fig. 4.3 Modified image

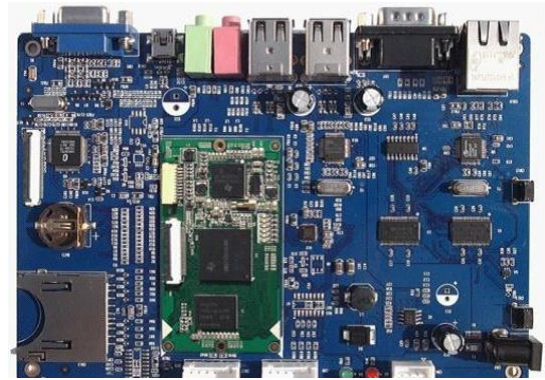


Fig. 4.4 Original image

4.4.1 CLASSIFICATION OF DIFFERENT IMAGE FORGERIES DETECTION ALGORITHMS

Some manipulated image detection techniques are already explained in literature survey, here we discussing about different algorithms.

FCM algorithm: -

- Choose the number of clusters.
- FCM starts with an initial guess for the cluster centres, which are intended to mark the mean location of each cluster.
- FCM assigns every data point with a membership grade for each cluster.
- The membership function represents the probability that a pixel belongs to a specific cluster. The pixels close to the centroid of their clusters are assigned high membership values, and low membership values are assigned to pixels with data far from the centroid.
- Convergence can be detected by comparing the changes in the membership function or the cluster centre at two successive iteration steps.

SIFT:

The scale-invariant feature transform (SIFT) is a *feature detection* algorithm in *computer vision* to detect and describe local features in images. Applications include object recognition, robotic mapping and navigation, image stitching, 3D

modelling, gesture recognition, video tracking, individual identification of wildlife and match moving.

SIFT key points of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on *Euclidean distance* of their feature vectors. From the full set of matches, subsets of key points that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient *hash table* implementation of the generalised *Hough transform*. Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally, the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

4.4.2 PRINCIPAL COMPONENT ANALYSIS

The multispectral image data is usually strongly correlated from one band to the other. The level of a given picture element on one band can to some extent be predicted from the level of that same pixel in another band.

Principal component analysis is a pre-processing transformation that creates new images from the uncorrelated values of different images. This is accomplished by a linear transformation of variables that corresponds to a rotation and translation of the original coordinate system.

Principal component analysis operates on all bands together. Thus, it alleviates the difficulty of selecting appropriate bands associated with the band ratioing operation. Principal components describe the data more efficiently than the original band reflectance values. The first principal component accounts for a maximum portion of the variance in the data set, often as high as 98%. Subsequent principal components account for successively smaller portions of the remaining variance.

Principal component transformations are used for spectral pattern recognition as well as image enhancement. When used before pattern recognition, the least important principal components are dropped altogether. This permits us to omit the

insignificant portion of our data set and thus avoids the additional computer time. The transformation functions are determined during the training stage. Principal component images may be analysed as separate black and white images, or any three component images may be colour coded to form a colour composite.

Principal component enhancement techniques are particularly appropriate in areas where little a priori information concerning the region is available.

4.4.3 IMAGE CLASSIFICATION

The overall objective of image classification is to automatically categorize all pixels in an image into land cover classes or themes. Normally, multispectral data are used to perform the classification, and the spectral pattern present within the data for each pixel is used as numerical basis for categorization. That is, different feature types manifest different combination of DN's based on their inherent spectral reflectance and emittance properties.

The term classifier refers loosely to a computer program that implements a specific procedure for image classification. Over the years scientists have devised many classification strategies. From these alternatives the analyst must select the classifier that will best accomplish a specific task. At present it is not possible to state that a given classifier is "best" for all situations because characteristics of each image and the circumstances for each study vary so greatly. Therefore, it is essential that the analyst understands the alternative strategies for image classification.

The traditional methods of classification mainly follow two approaches: unsupervised and supervised. The unsupervised approach attempts spectral grouping that may have an unclear meaning from the user's point of view. Having established these, the analyst then tries to associate an information class with each group. The unsupervised approach is often referred to as clustering and results in statistics that are for spectral, statistical clusters. In the supervised approach to classification, the image analyst supervises the pixel categorization process by specifying to the computer algorithm; numerical descriptors of the various land cover types present in the scene.

To do this, representative sample sites of known cover types, called training areas or training sites, are used to compile a numerical interpretation key that describes the spectral attributes for each feature type of interest. Each pixel in the data set is then compared numerically to each category in the interpretation key and labeled with the

name of the category it looks most like. In the supervised approach the user defines useful information categories and then examines their spectral separability whereas in the unsupervised approach he first determines spectrally separable classes and then defines their informational utility.

It has been found that in areas of complex terrain, the unsupervised approach is preferable to the supervised one. In such conditions if the supervised approach is used, the user will have difficulty in selecting training sites because of the variability of spectral response within each class. Consequently, a prior ground data collection can be very time consuming. Also, the supervised approach is subjective in the sense that the analyst tries to classify information categories, which are often composed of several spectral classes whereas spectrally distinguishable classes will be revealed by the unsupervised approach, and hence ground data collection requirements may be reduced. Additionally, the unsupervised approach has the potential advantage of revealing discriminable classes unknown from previous work. However, when definition of representative training areas is possible and statistical information classes show a close correspondence, the results of supervised classification will be superior to unsupervised classification.

4.4.4 UNSUPERVISED CLASSIFICATION

Unsupervised classifiers do not utilize training data as the basis for classification. Rather, this family of classifiers involves algorithms that examine the unknown pixels in an image and aggregate them into a number of classes based on the natural groupings or clusters present in the image values. It performs very well in cases where the values within a given cover type are close together in the measurement space, data in different classes are comparatively well separated.

The classes that result from unsupervised classification are spectral classes because they are based solely on the natural groupings in the image values the identity of the spectral classes will not be initially known. The analyst must compare the classified data with some form of reference data (such as larger scale imagery or maps) to determine the identity and informational value of the spectral classes. In the supervised approach we define useful information categories and then examine their spectral separability; in the unsupervised approach we determine spectrally separable classes and then define their informational utility.

There are numerous clustering algorithms that can be used to determine the natural spectral groupings present in data set. One common form of clustering, called the “K-means” approach also called as ISODATA (Interaction Self-Organizing Data Analysis Technique) accepts from the analyst the number of clusters to be located in the data. The algorithm then arbitrarily “seeds”, or locates, that number of cluster centers in the multidimensional measurement space. Each pixel in the image is then assigned to the cluster whose arbitrary mean vector is closest.

After all pixels have been classified in this manner, revised mean vectors for each of the clusters are computed. The revised means are then used as the basis of reclassification of the image data. The procedure continues until there is no significant change in the location of class mean vectors between successive iterations of the algorithm. Once this point is reached, the analyst determines the land cover identity of each spectral class. Because the K-means approach is iterative, it is computationally intensive. Therefore, it is often applied only to image sub-areas rather than to full scenes.

4.4.5 SUPERVISED CLASSIFICATION

Supervised classification can be defined normally as the process of samples of known identity to classify pixels of unknown identity. Samples of known identity are those pixels located within training areas. Pixels located within these areas term the training samples used to guide the classification algorithm to assigning specific spectral values to appropriate informational class

The basic steps involved in a typical supervised classification procedure are illustrated on Fig. The training stage Feature selection Selection of appropriate classification algorithm Post classification smoothing Accuracy assessment.

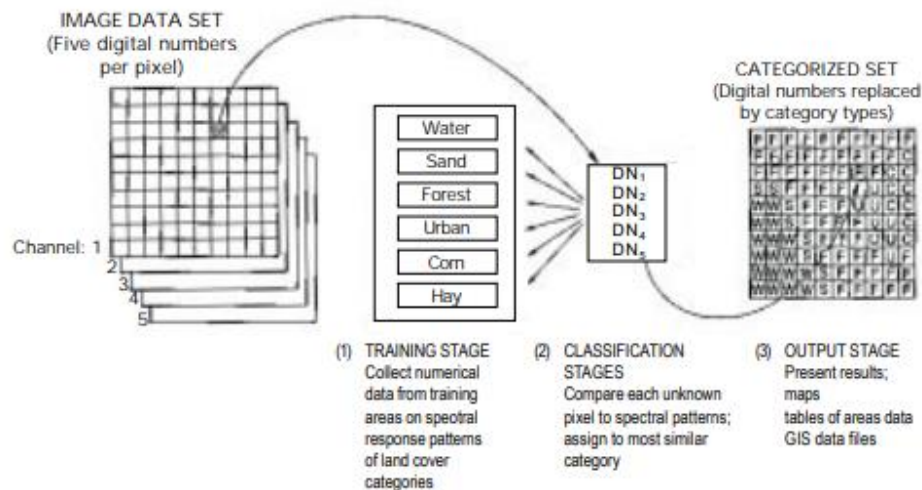


Fig.4.5 Basic Steps in Supervised

4.5 SYSTEM REQUIREMENTS: -

4.5.1 Hardware requirements

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and how it should be implemented.

Processor	:	Pentium Iv 2.6 GHz, Intel Core 2duo.
Ram	:	512 Mb Dd Ram
Monitor	:	15" Colour
Hard Disk	:	240 Gb
Cd drive	:	Lg 52x
Keyboard	:	Standard 102 Keys
Mouse	:	3 Button

4.5.2 Software Requirements

Operating system	:	Windows 7 or 8 or 10.
------------------	---	-----------------------

Coding Language : MATC
Tool : MATLAB R2014a

Predicting how MATLAB will perform while running an application on a particular computer is difficult. So, Math Works offers this general guidance on platform selection criteria and emphasizes that it is not a substitute for testing your application on a particular computer such as Operating Systems(32bit (or) 64 bit),Hardware considerations like central processing unit, hard disk, memory, Graphics Processing Unit (GPU) for display ,Graphics Processing Unit (GPU) for computation.

Typical uses include: -

- Math and computation.
- Algorithm development.
- Modelling, simulation, and prototyping.
- Data analysis, exploration, and visualization.
- Scientific and engineering graphics.
- Application development, including Graphical User Interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN (Formula translation).

4.5.3 System Architecture

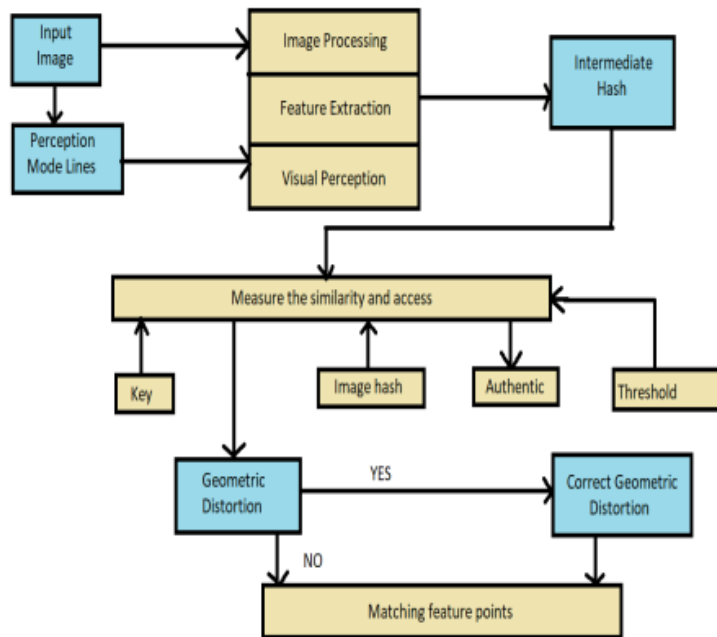


Fig.4.6 System Architecture

4.6 MATLAB SPECIFICATIONS

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

4.6.1 Features of MATLAB

- High-level language for technical computing.
- Development environment for managing code, files, and data.
- Interactive tools for iterative exploration, design, and problem solving.
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration.
- 2-D and 3-D graphics functions for visualizing data.
- Tools for building custom graphical user interfaces.
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java™, COM, and Microsoft Excel.

MATLAB is used in vast area, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational. Add-on toolboxes (collections of special-purpose MATLAB functions) extend the MATLAB environment to solve particular classes of problems in these application areas.

4.6.2 Interfacing with other Languages

MATLAB can call functions and subroutines written in the C programming language or FORTRAN. A wrapper function is created allowing MATLAB data types to be passed and returned. The dynamically loadable object files created by compiling such functions are termed "MEX-files" (for MATLAB executable).

Libraries written in Java, ActiveX or .NET can be directly called from MATLAB and many MATLAB libraries (for example XML or SQL support) are implemented as wrappers around Java or ActiveX libraries. Calling MATLAB from Java is more complicated, but can be done with MATLAB extension, which is sold separately by Math Works, or using an undocumented mechanism called JMI (Java-to-Mat lab Interface), which should not be confused with the unrelated Java that is also called JMI.

As alternatives to the MuPAD based Symbolic Math Toolbox available from Math Works, MATLAB can be connected to Maple or Mathematica.

Libraries also exist to import and export MathML.

Development Environment

- Startup Accelerator for faster MATLAB startup on Windows, especially on Windows XP, and for network installations.
- Spreadsheet Import Tool that provides more options for selecting and loading mixed textual and numeric data.
- Readability and navigation improvements to warning and error messages in the MATLAB command window.
- Automatic variable and function renaming in the MATLAB Editor.

Developing Algorithms and Applications: -

MATLAB provides a high-level language and development tools that let you quickly develop and analyze your algorithms and applications.

4.6.3 The MATLAB Language

The MATLAB language supports the vector and matrix operations that are fundamental to engineering and scientific problems. It enables fast development and execution. With the MATLAB language, you can program and develop algorithms faster than with traditional languages because you do not need to perform low-level administrative tasks, such as declaring variables, specifying data types, and allocating memory. In many cases, MATLAB eliminates the need for 'for' loops. As a result, one line of MATLAB code can often replace several lines of C or C++ code.

At the same time, MATLAB provides all the features of a traditional programming language, including arithmetic operators, flow control, data structures, data types, object-oriented programming (OOP), and debugging features.

MATLAB lets you execute commands or groups of commands one at a time, without compiling and linking, enabling you to quickly iterate to the optimal solution. For fast execution of heavy matrix and vector computations, MATLAB uses processor-optimized libraries. For general-purpose scalar computations, MATLAB generates machine-code instructions using its JIT (Just-In-Time) compilation technology.

This technology, which is available on most platforms, provides execution speeds that rival those of traditional programming languages.

4.6.4 Development Tools

MATLAB includes development tools that help you implement your algorithm efficiently. These include the following:

4.6.5 MATLAB Editor

Provides standard editing and debugging features, such as setting breakpoints and single stepping

4.6.6 Code Analyzer

Checks your code for problems and recommends modifications to maximize performance and maintainability.

4.6.7 MATLAB Profiler

Records the time spent executing each line of code.

4.6.8 Directory Reports

Scan all the files in a directory and report on code efficiency, file differences, file dependencies, and code coverage.

4.6.9 Designing Graphical User Interfaces

By using the interactive tool GUIDE (Graphical User Interface Development Environment) to layout, design, and edit user interfaces. GUIDE lets you include list boxes, pull-down menus, push buttons, radio buttons, and sliders, as well as MATLAB plots and Microsoft ActiveX® controls. Alternatively, you can create GUIs programmatically using MATLAB functions.

4.7 ANALYSING AND ACCESSING DATA

MATLAB supports the entire data analysis process, from acquiring data from external devices and databases, through pre-processing, visualization, and numerical analysis, to producing presentation-quality output.

4.7.1 Data Analysis

MATLAB provides interactive tools and command-line functions for data analysis operations, including:

- Interpolating and decimating
- Extracting sections of data, scaling, and averaging
- Thresholding and smoothing
- Correlation, Fourier analysis, and filtering
- 1-D peak, valley, and zero finding
- Basic statistics and curve fitting
- Matrix analysis

4.7.2 Data Access

MATLAB is an efficient platform for accessing data from files, other applications, databases, and external devices. You can read data from popular file formats, such as

Microsoft Excel; ASCII text or binary files; image, sound, and video files; and scientific files, such as HDF and HDF5. Low-level binary file I/O functions let you work with data files in any format. Additional functions let you read data from Web pages and XML.

4.7.3 Visualizing Data

All the graphics features that are required to visualize engineering and scientific data are available in MATLAB. These include 2-D and 3-D plotting functions, 3-D volume visualization functions, tools for interactively creating plots, and the ability to export results to all popular graphics formats. You can customize plots by adding multiple axes; changing line colours and markers; adding annotation, Latex equations, and legends; and drawing shapes

4.7.4 2-D Plotting

Visualizing vectors of data with 2-D plotting functions that create:

- Line, area, bar, and pie charts.
- Direction and velocity plots.
- Histograms.
- Polygons and surfaces.
- Scatter/bubble plots.
- Animations.

4.7.5 3-D Plotting and Volume Visualization

MATLAB provides functions for visualizing 2-D matrices, 3-D scalar, and 3-D vector data. You can use these functions to visualize and understand large, often complex, multidimensional data. Specifying plot characteristics, such as camera viewing angle, perspective, lighting effect, light source locations, and transparency.

3-D plotting functions include: -

- Surface, contour, and mesh.
- Image plots.
- Cone, slice, stream, and iso surface.

4.8 PERFORMING NUMERIC COMPUTATION

MATLAB contains mathematical, statistical, and engineering functions to support all common engineering and science operations. These functions, developed by experts in mathematics, are the foundation of the MATLAB language. The core math functions use the LAPACK and BLAS linear algebra subroutine libraries and the FFTW Discrete Fourier Transform library. Because these processor-dependent libraries are optimized to the different platforms that MATLAB supports, they execute faster than the equivalent C or C++ code.

MATLAB provides the following types of functions for performing mathematical operations and analyzing data:

- Matrix manipulation and linear algebra.
- Polynomials and interpolation.
- Fourier analysis and filtering.
- Data analysis and statistics.
- Optimization and numerical integration.
- Ordinary differential equations (ODEs).
- Partial differential equations (PDEs).

Sparse matrix operations.

CHAPTER 5

RESULTS AND DISCUSSION

The database has 6-7 images and the hashing takes place beforehand, the images are processed to initialize and train the data for the extraction of ORIGINAL and FORGERY detection from the database.

- 1st part of the project is to hash the image in database and we get the output for the initial image as ORIGINAL using sift feature extraction.

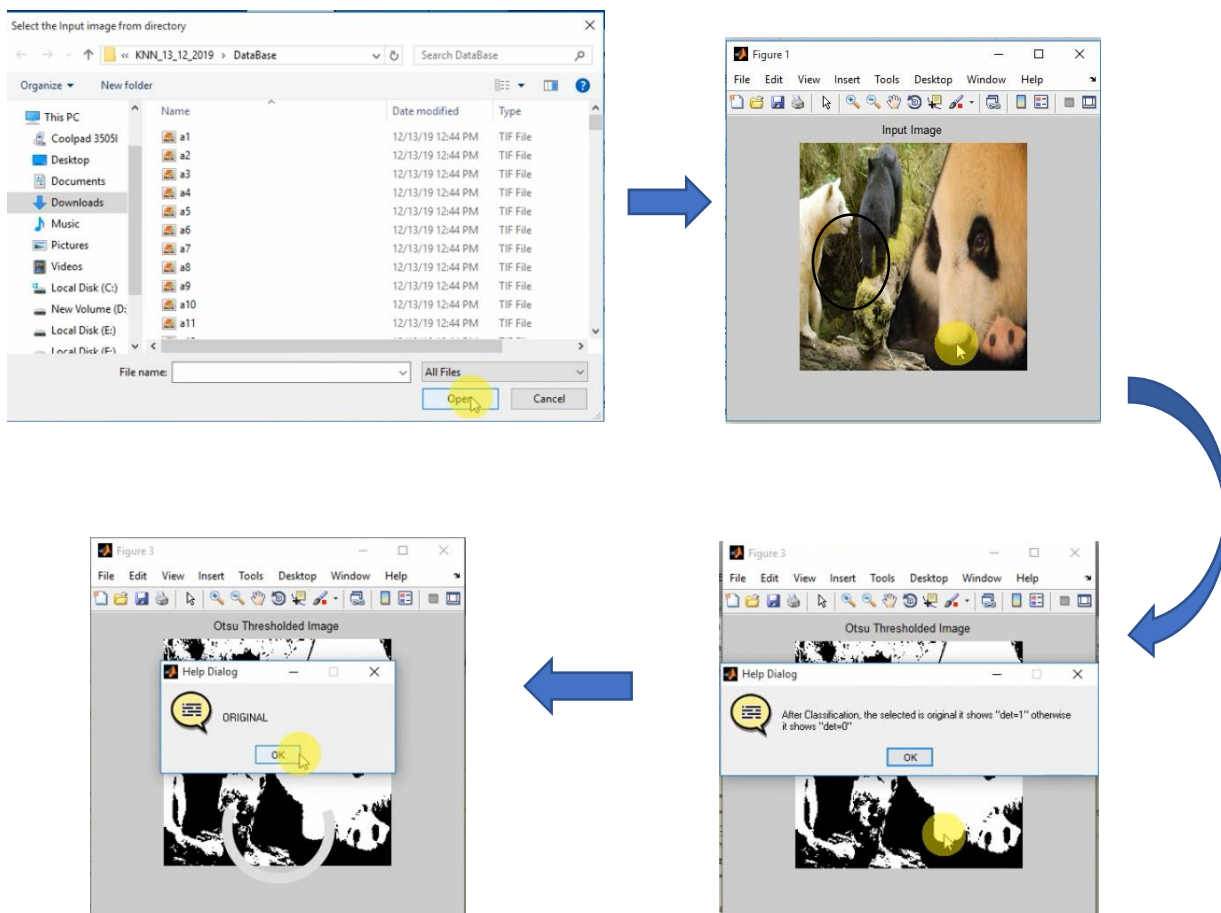


Fig.5.1 Original Image Hash Initializing Process

- 2ND part of the project is to identify the tampered image in database similar to the initial image as FORGERY using sift feature extraction.

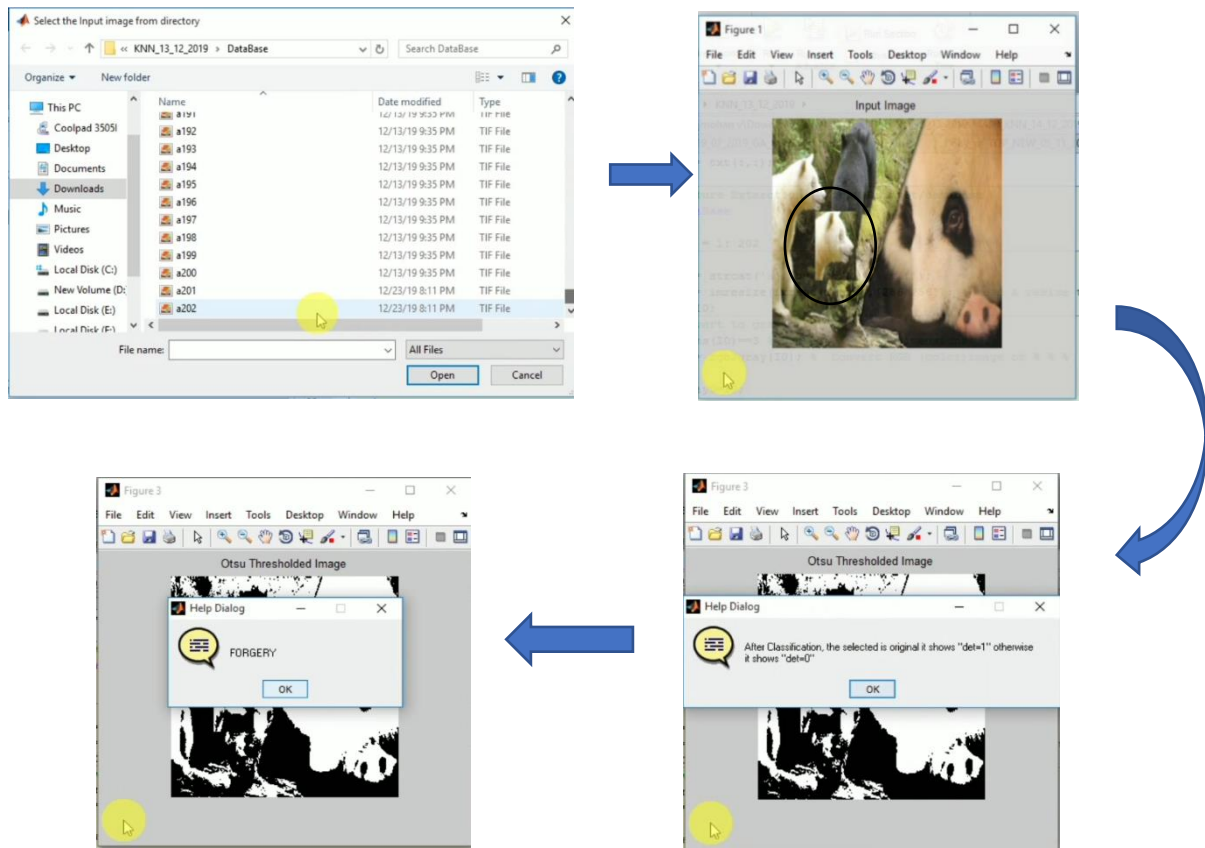


Fig 5.2 Forgery Image Detecting Process

- The black-circled area in each process reveals the tampered differences from the (fig 5.1) original image and (fig 5.2) the forgery image.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION:

The proposed hash carries both time, data and device information. Not only can it detect and precisely locate image forgeries, but also identify the camera of the source image with high accuracy. Besides, the hash is robust against normal content-preserving manipulations such as noising, filtering, JPEG compression, Gamma correction, rotation, scaling, etc. The proposed work is more secure than existing image hashes that rely on a locally stored secret key for the generation and validation, as the random space used for mapping the feature points is generated only on demand by a tamper-resistant PUF. As the first robust rotation-/scaling-invariant PUF-based perceptual image hash, the proposed data-device hash has introduced an added attribute of birth certification, which is essential in digital forensics to prove the authenticity of a visual evidence conveyed by the image content. This is made possible using the idea of random projection, i.e., projecting the content-based image features into an image sensor of Perceptual Image Hashing with the help of SURF or SIFT for feature Extraction to reach the expected output under the category Image processing.

6.2 FUTURE WORK:

For the Future work I would like to further improve the models, perhaps with more training data or advanced feature extraction if possible, both. Also, I could focus on applying the proposed forgery detection scheme based on adaptive over-segmentation and feature-point matching on other types of forgery, such as splicing or other types of media, for example, video and audio data with the software compatible support.

REFERENCES

- S. Cole, “Reddit just shut down the deepfakes subreddit,” Feb. 2018. [Online]. Available: https://motherboard.vice.com/en_us/article/neqb98/reddit-shutdown-deepfakes
- R. Davarzani, S. Mozaffari, and K. Yaghmaie, “Perceptual image hashing using center-symmetric local binary patterns,” *Multimedia Tools Appl.*, vol. 75, no. 8, pp. 4639–4667, Mar. 2016.
- Y. Guo, O. C. Au, R. Wang, L. Fang, and X. Cao, “Halftone image watermarking by content aware double-sided embedding error diffusion,” *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3387–3402, Jul. 2018
- X. L. Liu, C. C. Lin, and S. M. Yuan, “Blind dual watermarking for color images authentication and copyright protection,” *IEEE Trans. Circuits Syst. Video Techno.*, vol. 28, no. 5, pp. 1047–1055, May 2018.
- T. H. Thai, R. Cogranne, F. Retraint et al., “JPEG quantization step estimation and its applications to digital image forensics,” *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 123–133, Jan. 2017.
- L. Wen, H. Qi, and S. Lyu, “Contrast enhancement estimation for digital image forensics,” *ACM Trans. Multimedia Comput., Commun., Appl. (TOMM)*, vol. 14, no. 2, p. 49, May 2018.
- P. Korus and J. Huang, “Improved tampering localization in digital image forensics based on maximal entropy random walk,” *IEEE Signal Process. Letters*, vol. 23, no. 1, pp. 169–173, Dec. 2016.
- J. A. Redi, W. Taktak, and J.-L. Dugelay, “Digital image forensics: a booklet for beginners,” *Multimedia Tools Appl.*, vol. 51, no. 1, pp. 133–162, 2011.
- X. Lv and Z. J. Wang, “Perceptual image hashing based on shape contexts and local feature points,” *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1081–1093, Jun. 2012.
- X. Wang et al., “A visual model-based perceptual image hash for content authentication,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1336–1349, Jul. 2015.

APPENDIX

A. SOURCE CODE

KNN_MAIN: -

```
clc
clear all
close all
%%
[fname pname]=uigetfile('*.jpg','Select the Input image from directory');
img1 = imresize(imread(fname),[256 256]);
figure,image(img1);
colormap(gray(256));
title 'Input Image'
I=img1;
%% Convert to grayscale
if ndims(I)==3
gray = rgb2gray(I);
else
    gray=I;
end
figure,imshow(gray);
title 'Input Image in GrayScale form'
%% Otsu Binarization for segmentation
level = graythresh(I);
img = im2bw(I,level);
figure, imshow(img);
title('Otsu Thresholded Image');
drawnow;
%% Detect the SIFT features:
fprintf(1,'Computing the SIFT features for ima1.jpg...\n')
[features1,pyr1,imp1,keys1] = detect_features(img1);
%%
figure,showfeatures(features1,img1);
title('SIFT features of Input Image');
```

```

drawnow;
%%
figure,plotpoints(features1,img1,0);
title('Points of Input Image');
drawnow;
%%
[M N v]=size(l);
l=gray;
data=double(reshape(l,[size(l,1)*size(l,2) 1]));
n=3;
[center, U, obj_fcn] =fcm(data, n);
maxU = max(U);
data = data';

wholeD = zeros(size(data));
for k = 1:n
    indexk = (U(k,:) == maxU);
    lk = indexk.*data;
    lk = reshape(lk,M,N);
    result{k} = lk;
    wholeG(indexk) = k-1;

end
mean_cluster_val = zeros(3,1);
for k = 1:n
    mean_cluster_val(k) = mean(center(k));
end

[mean_cluster_val,idx] = sort(mean_cluster_val);
imgr=result{idx(end)};
imgr=medfilt2(imgr,[1 1]);
figure,imshow(imgr);

```

```

hh=imgr;
title(['Segmented image with ',num2str(n),' bins']);
imwrite(hh, strcat('seg_img.bmp'))
seg_img=hh;
% Extract features using DWT
x = double(seg_img);
m = size(seg_img,1);
n = size(seg_img,2);
signal1 = seg_img(:,:);
[cA1,cH1,cV1,cD1] = dwt2(signal1,'db4');
[cA2,cH2,cV2,cD2] = dwt2(cA1,'db4');
[cA3,cH3,cV3,cD3] = dwt2(cA2,'db4');

DWT_feat = [cA3,cH3,cV3,cD3];
G = pca(DWT_feat);
whos DWT_feat
whos G
g = graycomatrix(G);
stats = graycoprops(g,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(G);
Standard_Deviation = std2(G);
Entropy = entropy(G);
RMS = mean2(rms(G));
%Skewness = skewness(img)
Variance = mean2(var(double(G)));
a = sum(double(G(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(G(:)));
Skewness = skewness(double(G(:)));

```

```
% Inverse Difference Movement
```

```
m = size(G,1);
```

```
n = size(G,2);
```

```
in_diff = 0;
```

```
for i = 1:m
```

```
    for j = 1:n
```

```
        temp = G(i,j)./(1+(i-j).^2);
```

```
        in_diff = in_diff+temp;
```

```
    end
```

```
end
```

```
IDM = double(in_diff);
```

```
feat = [Contrast,Correlation,Energy,Homogeneity, Mean, Standard_Deviation,  
Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness, IDM];
```

```
load Features
```

```
xdata = meas;
```

```
group = label;
```

```
species= label;
```

```
g1='ORIGINAL';
```

```
g2='FORGERY';
```

```
X=meas;Y=label;
```

```
mdl = fitcknn(X,Y)
```

```
c = knnclassify(feat,meas,label);
```

```
hyu=cell2mat(c);
```

```
det=[];
```

```
disp('After Classification, the selected is original it shows "det=1" otherwise it shows  
"det=0"')
```

```
helpdlg('After Classification, the selected is original it shows "det=1" otherwise it  
shows "det=0"')
```

```
pause(2)
```

```
if strcmpi(hyu,g1)
```

```

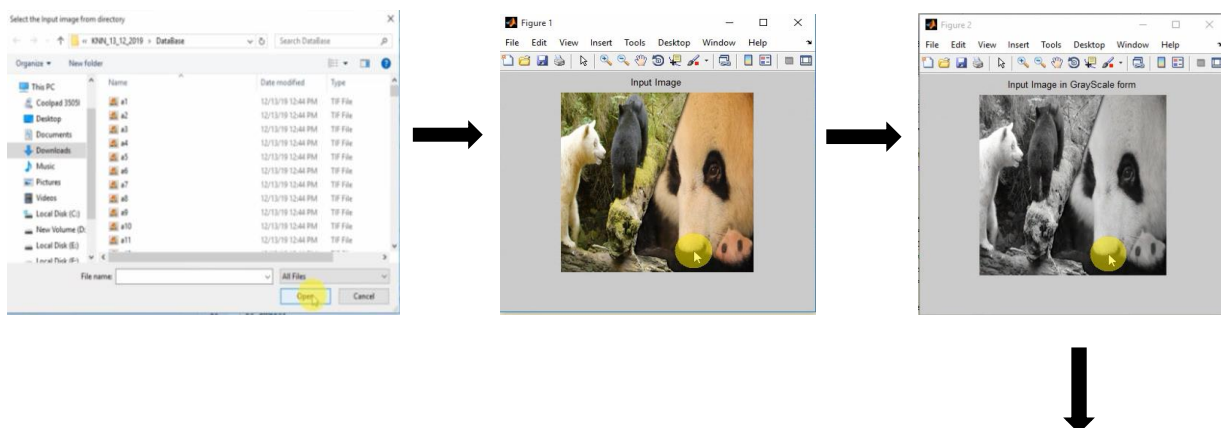
helpdlg('ORIGINAL');
disp('ORIGINAL');
det=1
elseif strcmpi(hyu,g2)
    helpdlg('FORGERY');
    disp('FORGERY');
    det=0
else
    return
end

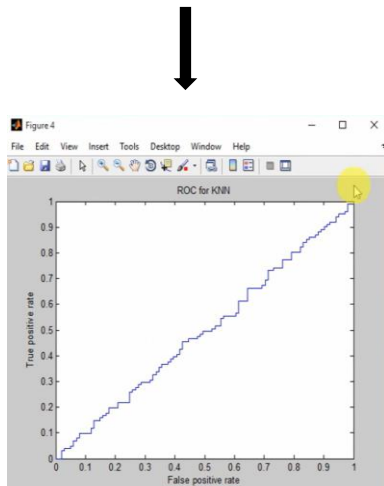
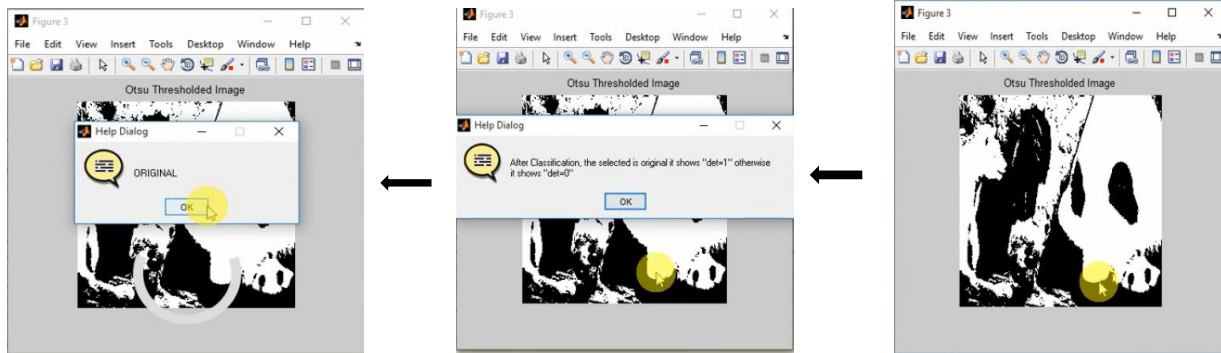
rloss = resubLoss mdl
cvmdl = crossval(mdl);
kloss = kfoldLoss(cvmdl)

resp = strcmp(label,'ORIGINAL'); % resp = 1, if Y = 'b', or 0 if Y = 'g'
pred = meas;
mdl = fitglm(pred,resp,'Distribution','binomial','Link','logit');
scores = mdl.Fitted.Probability;
[X,Y,T,AUC] = perfcurve(label,scores,'ORIGINAL');
figure,plot(X,Y)
xlabel('False positive rate')
ylabel('True positive rate')
title('ROC for KNN')

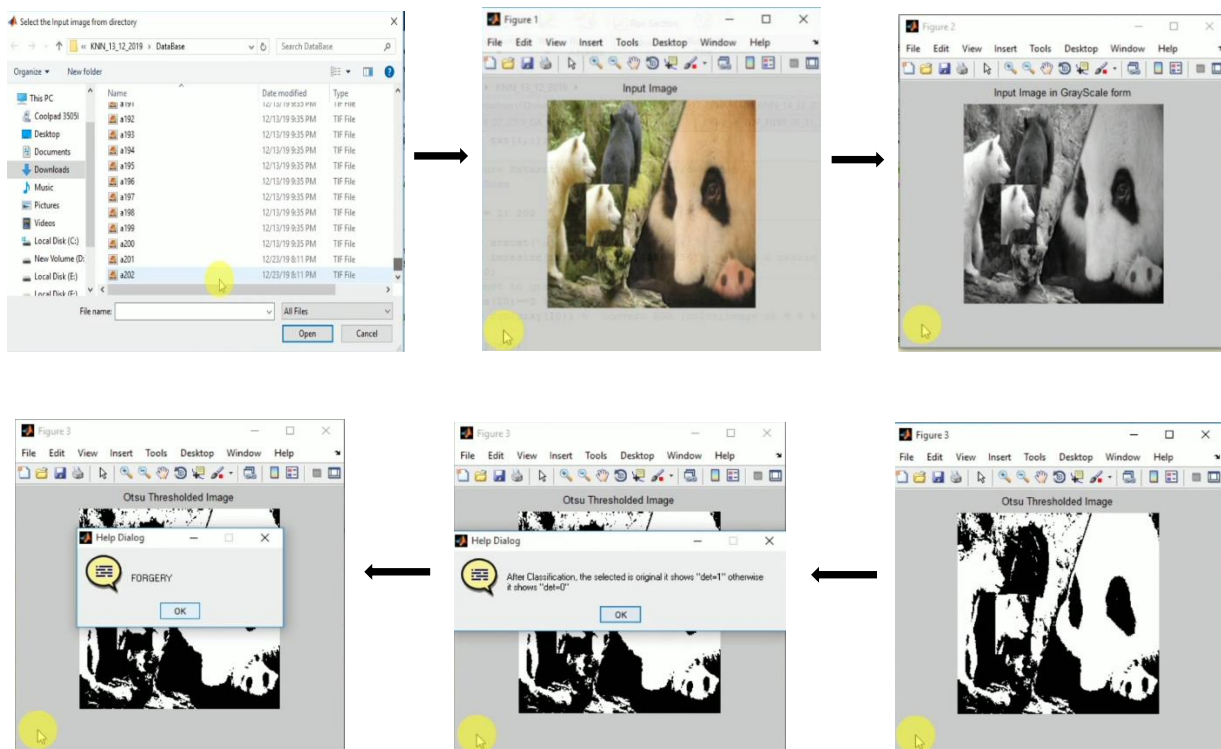
```

B. SCREEN SHOTS





- The process above indicates the trained-data image to initialize it as ORIGINAL image.
- The process below detects the tampered data to identify that it is a FORGERY image. (refer fig 5.1 & 5.2 pgs. 39, 40.,)



C. PUBLICATION WITH PLAGIARISM REPORT