# FACIAL EXPRESSION RECOGNITION USING MACHINE LEARNING

A dissertation submitted in the partial fulfillment of the academic requirements for the award of degree of

### Bachelor of Engineering

In

### COMPUTER SCIENCE AND ENGINEERING

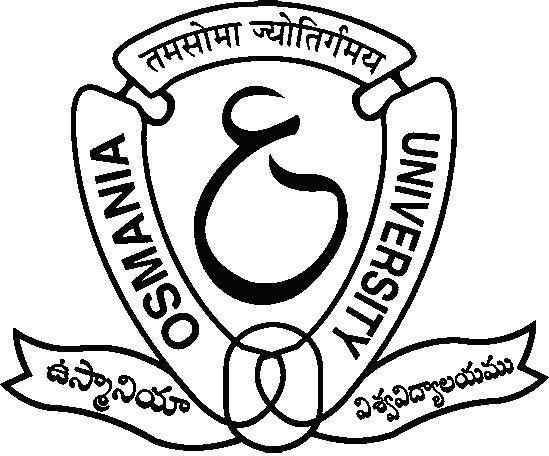
By

SYED OMER FAROOQ 1005-15-735058

SRIJAY POOSA 1005-15-735017

SUSHANTH KRISHNA 1005-15-735036

Under the guidance of **Dr. S.RAMBABU** Associate Professor



### Department of Computer Science and Engineering

**University College of Engineering (Autonomous)**

**Osmania University, Hyderabad-500007**

**2018-2019**

#### Department of Computer Science and Engineering

#### University College of Engineering (Autonomous)

#### Osmania University, Hyderabad- 500007



**Certificate**

This is to certify that this project work entitled **“FACIAL EXPRESSION RECOGITION USING MACHINE LEARNING”** is a bonafide work carried out by Syed Omer Farooq(1005-15-733058),Srijay Poosa(1005-15-733041) and Sushanth Krishna (1005-15-733050) from the department of “**Computer Science and Engineering”** in the partial fulfillment of the requirements for the award of “**Bachelor of Engineering**” degree in “**Computer Science and Engineering**” from “**University College of Engineering(A), Osmania University**”, Hyderabad, during the period 2018- 2019.

Project guide

###### DR. S.RAMBABU, Dr. K.SHYAMALA

Associate Professor, Head of the Department,

Department of CSE, Department of CSE,

Osmania University Osmania University

**DECLARATION**

We hereby declare that the results in this dissertation work entitled **“FACIAL EXPRESSION DETECTION USING MACHINE LEARNING”** is the bonafide work done and carried out by us during the year 2018-19 in partial fulfillment of the academic requirements for the award of **Bachelor of Engineering** in Computer Science and Engineering from **University College of Engineering**, **OSMANIA UNIVERSITY,** Hyderabad. This project was done under the supervision of **Prof. S.RAMBABU**.

Further, we declare that the report has not been submitted by us to any other institute or university for award of any other degree.

**SYED OMER FAROOQ (1005-15-735058)**

**SRIJAY POOSA (1005-15-735017)**

**SUSHANTH KRISHNA (1005-15-735036)**

# ACKNOWLEDGEMENT

This is an acknowledgment of the intensive drive and technical competence of many individuals who have contributed to the success of our project. The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course project.

We would like to express our deep sense of gratitude and whole-hearted thanks to our project guide, **DR. S. RAMBABU**, **Associate Professor, Department** **of Computer Science and Engineering,** University College of Engineering Osmania University, for giving us privilege of working under his guidance, with tremendous support and cogent discussion, constructive criticism and encouragement throughout this dissertation work carrying out project work.

We also thank **Dr. K. SHYAMALA**, Head Department of Computer Science and Engineering for her support from the Department and allowing us to make all the resources available to us students.

We also extend our thanks to the entire faculty of the Department of Computer Science and Engineering, University College of Engineering ,Osmania University who encourages us throughout the course of our Bachelor degree and allowing us to use the many resources present in the department.

Our sincere thanks to our parents and friends for their valuable suggestions, morals, strength and support for the completion of our project.

**Syed Omer Farooq (1005-15-735058)**

**Srijay Poosa (1005-15-735017)**

**Sushanth Krishna (1005-15-735036)**

# ABSTRACT

### Facial expressions are one of the more important aspects of human communication. The face is responsible for communicating not only thoughts or ideas, but also emotions. What makes the communication of emotions interesting is that it appears as if some of these expressions of emotion may be biologically hardwired, and are expressed the same way by all peoples of all cultures. This contrasts with other views that all facial expressions are a product of social learning and culture. Facial Expressions are one of the most simple and basic ways through which people can showcase their feelings. A single expression can mean a lot if taken as feedback or for understanding a particular problem.

### The project aims at developing will be a Facial Expression Recognition model that will be based on the following basic expression like anger, disgust, fear, happiness, sadness, surprise and neutral. To improve accuracy we will be using deeper Convolution Neural Network. This network had 4 convolutional layers and with 2 Fully Connected layer. This trained model will give as output the probability of the particular expression to fit to each of the 7 expressions. The model is said to be accurate when the label having the highest probability is actually how the person feels.

### There are many applications for this project. This can be used to study the patients that are suffering from different mental disorders, in the medical field to understand autism patients, for marketing purposes also the output of this model can be used to gain important feedback from the user about the product just by understanding the first expressions of the customer.

### CONTENTS

[LIST OF FIGURES 9](#_TOC_250003)

[CHAPTER 1](#_bookmark2) [INTRODUCTION](#_bookmark3)

* 1. [INTRODUCTION](#_bookmark4) 11
  2. [MOTIVATION](#_bookmark5) 12

[1.3](#_bookmark6) AIM… 12

* 1. [OBJECTIVE 13](#_TOC_250002)
  2. [THESIS ORGANIZATION](#_bookmark7) 13

[CHAPTER 2](#_bookmark0) [LITERATURE SURVEY](#_bookmark1)

* 1. [HISTORY OF WATERMARKING 14](#_TOC_250001)
  2. [LIMITATIONS OF EXISTING TECHNIQUES](#_bookmark8) 16

[CHAPTER 3](#_bookmark9)

[BASICS OF IMAGE PROCESSING](#_bookmark10)

* 1. [INTRODUCTION](#_bookmark11) 19
  2. [CONCEPT OF PIXEL](#_bookmark12) 20
  3. [TYPES OF IMAGES](#_bookmark13) 21
  4. [COLOR CODES CONVERSION](#_bookmark14) 24
  5. [GRAYSCALE TO RGB CONVERSION](#_bookmark15) 27
  6. [HISTOGRAM](#_bookmark16) 30
  7. [EDGE DETECTION](#_bookmark17) 32
  8. [BASICS OF VIDEO](#_bookmark18) 34

CHAPTER 4

[DIGITAL WATERMARKING](#_bookmark19)

* 1. [INTRODUCTION](#_bookmark20) 35
  2. [TYPES OF WATERMARKING](#_bookmark21) 35
  3. [WATERMARKING VS CRYPTOGRAPHY VS STEGANOGRAPHY](#_bookmark22) 36
  4. [WATERMARKING APPLICATIONS](#_bookmark23) 37
  5. [WATERMARKING PROPERTIES](#_bookmark24) 38
  6. [WATERMARKING LIFECYCLE PHASES](#_bookmark25) 39

[CHAPTER 5](#_bookmark26)

[WATERMARKING TECHNIQUES](#_bookmark27)

* 1. [BIT PLANE SLICING](#_bookmark28) 41
  2. [ARNOLD TRANSFORMATION](#_bookmark29) 43
  3. [SINGULAR VALUE DECOMPOSITION](#_bookmark30) 45
  4. [NON NEGATIVE MATRIX FACTORIZATION](#_bookmark31) 48

[CHAPTER 6](#_bookmark32)

[IMPLEMENTATION OF WATERMARKING TECHNIQUES](#_bookmark33)

* 1. [IMAGE PRE-PROCESSING](#_bookmark34) 50
  2. [VIDEO PRE PROCESSING](#_bookmark35) 52
  3. [EMBEDDING](#_bookmark36) 53
  4. [EXTRACTION](#_bookmark37) 55

[CHAPTER 7](#_bookmark38) [RESULTS](#_bookmark39)

* 1. [RESULTS](#_bookmark40) OF EMBEDDED VIDEO 57
  2. [EXTRACTION RESULTS](#_bookmark41) 58
     1. [Test case 1](#_bookmark42) 58
     2. [Test case 2](#_bookmark43) 59
  3. [Original Bit plane slices](#_bookmark44) 60
     1. [Red](#_bookmark45) 60
     2. [Green](#_bookmark46) 60
     3. [Blue](#_bookmark47) 61
  4. [Scrambled slices](#_bookmark48) 62
     1. [Red](#_bookmark49) 62
     2. [Green](#_bookmark50) 62
     3. [Blue](#_bookmark51) 63
  5. [Descrambled slices](#_bookmark52) 63
     1. [Red](#_bookmark53) 63
     2. [Green](#_bookmark54) 64
     3. [Blue](#_bookmark55) 64
  6. [ORIGINAL 24 VIDEO FRAMES](#_bookmark56) 65
  7. [EMBEDDED 24 VIDEO FRAMES](#_bookmark57) 66

CHAPTER 8

[CONCLUSION AND FUTURE SCOPE](#_bookmark58)

* 1. [CONCLUSION](#_bookmark59) 67
  2. [FUTURE SCOPE](#_bookmark60) 67

[REFERENCES 68](#_TOC_250000)

### LIST OF FIGURES

Fig 1: Digital image 20

Fig 2: Binary image object outline… 22

Fig 3: Examples of gray-scale images… 22

Fig 4: Representation of a typical RGB color image 23

Fig 5: color image 28

Fig 6: grayscale image 28

Fig 7: Original Image 29

Fig 8: Grayscale Image 30

Fig 9: experimental image for histogram… 31

Fig 10: histogram… 31

Fig 11: Original Image 34

Fig 12: Sharpen Image 34

Fig13: types of watermarking 35

Fig14: watermarking phases… 39

Fig 15: bit plane slicing… 41

Fig 16: pre processing 51

Fig 17: embedding 53

Fig 18: extraction… 55

Fig 19: Y-component of original video frame… 57

Fig 20: Y-component of embedded video frame… 57

Fig 21: Histogram of original frame… 57

Fig 22: Histogram of embedded frame… 57

Fig 23: Original Watermark Image 58

Fig 24: Extracted watermark Image 58

Fig 25: Histogram of original image 58

Fig 26: Histogram of extracted image 58

Fig 27: Original Image 2 59

Fig 28: Extracted Image 2… 59

Fig 29: Histogram of Original Image 2… 59

Fig 30: Histogram of Extracted Image 2… 59

Fig31: red component slices… 60

Fig32: green component slices… 60

Fig33: blue component slices… 61

Fig34: red component scrambled slices… 62

Fig35: red component scrambled slices… 62

Fig36: blue component scrambled slices… 63

Fig37: red component descrambled slices… 63

Fig38: green component descrambled slices… 64

Fig39: blue component descrambled slices… 64

Fig40: original 24 video frames… 65

Fig41: embedded 24 video frames 66

# CHAPTER 1

## INTRODUCTION

##### INTRODUCTION

Automatic facial expression recognition has attracted much attention since the early nineties, especially in human-computer interaction. As computers start becoming a part of our life, they need to become more and more intelligent. Expression recognition systems will enhance this intelligent interaction between the human and the computer.

Facial Expression Recognition is an Image Classification problem located within the wider field of Computer Vision. Image Classification problems are ones in which images must be algorithmically assigned a label from a discrete set of categories. In FER systems specifically, the images are of human faces and the categories are a set of emotions.

Machine learning approaches to FER all require a set of training image examples, each labeled with a single emotion category. A standard set of seven emotion classifications are often used:

1. Anger
2. Disgust
3. Fear
4. Happiness
5. Sadness
6. Surprise
7. Neutral

Companies have also been taking advantage of emotion recognition to drive business outcomes. For the upcoming release of Toy Story 5, Disney plans to use facial recognition to judge the emotional responses of the audience. Apple even released a new feature on the iPhone X called Animoji, where you can get a computer simulated emoji to mimic your facial expressions. It’s not so far off to assume they’ll use those capabilities in other applications soon.

##### MOTIVATION

With the constant expansion of the computer field in every aspect of our lives especially machine learning , we considered building a proper facial expression detection model. Considering the various application of the project we have selected we tried to incorporate the idea of machine learning into the field.

Research has shown that over 90% of our communication can be non-verbal, but technology has struggled to keep up, and traditional code is generally bad at understanding our intonations and intentions. But emotion recognition – also called Affective Computing – is becoming accessible to more types of developers.

Understanding contextual emotion has widespread consequences for society and business. In the public sphere, governmental organizations could make good use of the ability to detect emotions like guilt, fear, and uncertainty. It’s not hard to imagine the TSA auto-scanning airline passengers for signs of terrorism, and in the process making the world a safer place. These reasons prompted us to start working on a project that had some real-world application in the world.

##### AIM

To create a facial expression detection model that will correctly predict the facial emotion of a person that may be passed as a picture or real time photo can be taken as the input

##### OBJECTIVE

1. To create a proper training model that has a good prediction percentage
2. Convert a picture into the correct input format
3. Performing de-noising and converting the image into grayscale format

##### 

## CHAPTER 2

LITERATURE SURVEY

##### HISTORY OF WATERMARKING

Watermarking has been around for several centuries, found initially in plain paper and subsequently in paper bills. However, the field of digital watermarking was only developed during the last 15 years and it is now being used for many different applications.

Over the last 25 years, there has been much work on multimedia digital watermarking. In this domain, the primary limitation to watermark strength has been in its visibility. For multimedia watermarks, invisibility is defined in human terms (that is, in terms of human sensory limitations).

Through history, digital watermarking was widely used for hiding some information within a given content. This leads us to a discussion of information hiding in general and steganography in particular. Steganography and digital watermarking are two related fields that share many technical approaches but there are fundamental differences between them. While steganography aims at hiding information in such a way that it should not be even possible to detect the existence of a secret message, in digital watermarking, the effective coupling of a message to the digital content is of value and therefore the watermark embedding process is usually known.

Moreover, in steganography, the host data can be a mere ‘decoy’ and may have no relationship to the secret message. In contrast, a watermark usually carries supplementary information about the host data such as sender identifier or data provenance information. One could consider both techniques as a way for making embedded information ‘disappear’ within the host data. This perspective raises the possibility of encryption techniques as a solution for hiding information.

However, encryption is not successful in diverting attention away from suspected data, but also after decryption all the protection is lost which means there is no guarantee that the legitimate data decoder does not distribute the information. In fact, watermarking and encryption are more

complementary than competitive approaches and can be used in conjunction to provide maximum security.

Generally a watermarking system consists of two components: watermark encoder or embedder and watermark decoder/extractor or detector. The watermark encoder embeds a watermark message M, possibly utilizing a secret key K, within the data set D also referred to as cover data or host data. Occasionally, we might want K to depend on D in order to provide side-information to the decoder or use asymmetric keys to boost the security.

For over two decades, watermarking theory has been applied for intellectual property protection to multimedia communication applications, including audio, still images, single view video and three-dimensional (3-D) objects (Joshi et al., 2015). While from the application point of view the same set of criteria, i.e., transparency, robustness and watermark payload, are considered for performance evaluation, each type of the aforementioned multimedia contents comes with its own methodological framework which may or may not be applicable for other types of contents.

For example, in audio watermarking, there are six main frameworks commonly used, i.e., phase coding, spread spectrum, echo data hiding, patchwork coding, low bit encoding and noise gate watermarking (Mat Kiah et al., 2011). Each technique is trying to achieve the best trade-off between inaudibility, robustness and watermark data payload. Two other embedding frameworks introduced to 3-D mesh watermarking include modifying the mesh geometry map or the connectivity in the spatial domain, and modifying spectral-like coefficients in the spectral domain (Bouzidi et al., 2013a; Wang et al., 2007a). For(single view) video watermarking,

spatial-domain techniques embed watermark signals by modifying video pixels directly, whereas transform-domain video watermarking techniques apply domain transformation to video and then modify video frames’ coefficients according to predetermined embedding scheme (Hartung et al., 1997; Paul, 2011).

Watermark embedded by a robust watermarking scheme for copyright protection can survive and be detectable even if the carrier medium has undergone a certain level of transformations (or attacks) (Deguillaume et al., 1999; Wang et al., 2007b). Imperceptible watermarking refers to those techniques which make the watermarked signal perceptually indistinguishable from the un- watermarked medium, and is therefore considered as transparent (Bensaied et al., 2014; Hartung

et al., 1996). There are several robust imperceptible watermarking methods which have been developed for protecting copyright information in 2-D images and single view video (Bi et al., 2014; Hartung et al., 1997). There have been relatively few reports, however, on protecting 3-D visual signals or data (Lee et al., 2011). Today, 3-D multimedia are more frequently transmitted over the Internet than ever before using mobile devices, exposing the data shared to greater likelihood of piracy and attacks on watermark protected contents (Chen et al., 2013).

In most of the state-of-the-art watermarking schemes, the original signal is not required for detection of the watermark (Asikuzzaman et al., 2014). This scheme is referred to as blind watermarking (Franco-Contreras et al., 2011; Xie et al., 1998). In contrast, if the original signal is required, the scheme is called non-blind watermarking (Bovik, 2013; Garcia et al., 2003).Whereas there is no need for the original (unwatermarked) content for watermark detection, some extra information may be required to assist the detector to synchronize the input signal sequence with possible distortion, and such a scheme is called semi-blind watermarking (Cayre et al., 2003; Gao et al., 2012). Compared with single view video, 3-D video raises new challenges for watermarking in terms of both perceptual fidelity and robustness to combat illegal distributions of this new form of visual media, since the end-user experience becomes increasingly important in terms of perceived picture quality and visual comfort (Zhou et al., 2014). To protect 3-D video contents from copyright violation, a 3-D video watermarking system can be used with robust performance to withstand various attacks on the watermark protection (Franco-Contreras et al., 2011) so that any attempt to infringe on copyright by manipulating digital media to remove watermark will lead to noticeable degradation of quality of service and/or user experience (Chammem et al., 2013). This paper examines and compares different watermarking techniques now available in the published literature for protection of 3-D or stereoscopic videos to identify a benchmark for future development and performance evaluation of new 3-D video watermarking techniques and to highlight associated challenges.

##### LIMITATIONS OF EXISTING TECHNIQUES

* + 1. **Limitations of LSB based watermarking**

The earliest work of digital image watermarking schemes embeds watermarks in the LSB of the pi pixels. Given an image with pixels, and each pixel being represented by an 8-bit sequence, the

watermarks are embedded in the last (i.e., least significant) bit, of selected pixels of the image. This method is easy to implement and does not generate serious distortion to the image; however, it is not very robust against attacks. For instance, an attacker could simply randomize all LSBs, which effectively destroys the hidden information. Such an approach is very sensitive to noise and common signal processing and cannot be used in practical applications.

* + 1. **Limitations of HASH function based watermarking**

The hash function depends on a parameter K (a secret key) in a sensitive manner and on the image in a robust manner. The hash function is designed to return N = 50 bits from a 64×64 image block. The bits obtained from two different images or for two different keys K will generally be different (uncorrelated). However, for the same key K, two images that can be matched after applying gray scale operations, such as lossy compression, re-coloring, filtering, noise adding, gamma correction, and simple geometrical operations including rotation and scaling, the extracted N-tuple will be almost the same except for a few bits. It is explained how the extracted N-tuple can be further utilized for synthesizing a Gaussian sequence that gradually changes with increasing number of errors in the extracted bits. Thus the robust hash function can be used for generating pseudo-random watermark sequences that depend sensitively on a secret key yet continuously on the image. This robustness enables us to construct watermarks that depend on the original un-watermarked image in a non-trivial manner while making it possible to recover the watermark without having to access any information about the original image (oblivious watermarking).

#### Texture Mapping Coding Technique

This method is useful in only those images which have some texture part in it. This method hides the watermark in the texture part of the image. This algorithm is only suitable for those areas with large number of arbitrary texture images (disadvantage) , and cannot be done automatically. This method hides data within the continuous random texture patterns of a picture

#### Limitations of DCT based watermarking

The watermark is embedded into the coefficients of the middle frequency, so the visibility of image will not get affected and the watermark will not be removed by any kind of attack.

However, Block wise DCT destroys the invariance properties of the system. Certain higher frequency components tend to be suppressed during the quantization step.

#### Limitations of DWT based watermarking

It allows good localization both in time and spatial frequency domain and higher compression ratio which is relevant to human perception. However, cost of computing may be higher and longer compression time is required. Noise/blur near edges of images or video frames are observed.

#### Limitations of DFT based watermarking

DFT is rotation, scaling and translation (RST) invariant. Hence it can be used to recover from geometric distortions .Complex implementation is its disadvantage and cost of computing may be higher.

## CHAPTER 3

CONCEPTS INVOLVED

##### **IMAGE PROCESSING**

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 research area within engineering and computer science disciplines too.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

* 1. **WHAT IS AN IMAGE?**

An image is defined as a two-dimensional function,F(x,y), where x and y are spatial coordinates, and the amplitude of F at any pair of coordinates (x,y) is called the intensity of that image at that point. When x,y, and amplitude values of F are finite, we call it a digital image.

In other words, an image can be defined by a two-dimensional array specifically arranged in rows and columns.

Digital Image is composed of a finite number of elements, each of which elements have a particular value at a particular location. These elements are referred to as picture elements, image elements, and pixels. A Pixel is most widely used to denote the elements of a Digital Image.

* + 1. **TYPES OF IMAGES**

**3.2.1.1** **BINARY IMAGE**

The binary image as its name suggests, contain only two pixel elements i.e 0 & 1,where 0 refers to black and 1 refers to white. This image is also known as Monochrome.

**3.2.1.2 BLACK AND WHITE IMAGE**

The image which consist of only black and white color is called BLACK AND WHITE IMAGE.

**3.2.1.3 8 bit COLOR FORMAT**

It is the most famous image format.It has 256 different shades of colors in it and commonly known as Grayscale Image. In this format, 0 stands for Black, and 255 stands for white, and 127 stands for gray.

**3.2.1.4 16 bit COLOR FORMAT**

It is a color image format. It has 65,536 different colors in it.It is also known as High Color Format. In this format the distribution of color is not as same as Grayscale image.

#### IMAGE AS A MATRIX

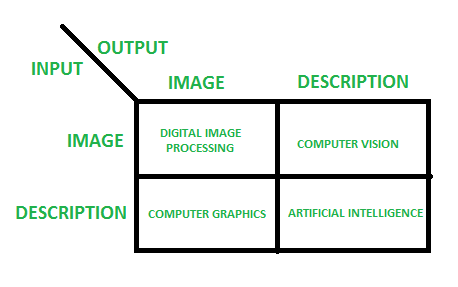
#### As we know, images are represented in rows and columns we have the following syntax in which images are represented:

#### https://cdncontribute.geeksforgeeks.org/wp-content/uploads/gfg2-5.png

#### The right side of this equation is digital image by definition. Every element of this matrix is called image element, picture element, or pixel.

* 1. **PHASES OF IMAGE PROCESSING**

1. **ACQUISITION**– It could be as simple as being given an image which is in digital form. The main work involves:
   1. Scaling
   2. Color conversion(RGB to Gray or vice-versa)
2. **IMAGE** **ENHANCEMENT**– It is amongst the simplest and most appealing in areas of Image Processing it is also used to extract some hidden details from an image and is subjective.
3. **IMAGE** **RESTORATION**– It also deals with appealing of an image but it is objective(Restoration is based on mathematical or probabilistic model or image degradation).
4. **COLOR** **IMAGE** **PROCESSING**– It deals with pseudocolor and full color image processing color models are applicable to digital image processing.
5. **WAVELETS** **AND** **MULTI-RESOLUTION PROCESSING**– It is foundation of representing images in various degrees.
6. **IMAGE COMPRESSION**-It involves in developing some functions to perform this operation. It mainly deals with image size or resolution.
7. **MORPHOLOGICAL PROCESSING**-It deals with tools for extracting image components that are useful in the representation & description of shape.
8. **SEGMENTATION PROCEDURE**-It includes partitioning an image into its constituent parts or objects. Autonomous segmentation is the most difficult task in Image Processing.
9. **REPRESENTATION & DESCRIPTION**-It follows output of segmentation stage, choosing a representation is only the part of solution for transforming raw data into processed data.
10. **OBJECT DETECTION AND RECOGNITION**-It is a process that assigns a label to an object based on its descriptor.
    1. **OVERLAPPING FIELDS WITH IMAGE PROCESSING**

****

* According to block 1, if input is an image and we get out image as a output, then it is termed as **Digital Image Processing.**
* According to block 2, if input is an image and we get some kind of information or description as a output, then it is termed as **Computer Vision**.
* According to block 3, if input is some description or code and we get image as an output, then it is termed as **Computer** **Graphics**.
* According to block 4, if input is description or some keywords or some code and we get description or some keywords as a output, then it is termed as **Artificial Intelligence.**
  1. **WHAT IS COMPUTER VISION?**

Computer vision is an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do.

Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that can interface with other thought processes and elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems.

* + 1. **Typical Tasks for Computer Vision**

Each of the application areas described above employ a range of computer vision tasks; more or less well-defined measurement problems or processing problems, which can be solved using a variety of methods. Some examples of typical computer vision tasks are presented below.

Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that can interface with other thought processes and elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

* **Recognition**

The classical problem in computer vision, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. Different varieties of the recognition problem are described in the literature:

1. Object Recognition
2. Identification
3. Detection

* **Screen Reconstruction**

Given one or (typically) more images of a scene, or a video, scene reconstruction aims at computing a 3D model of the scene. In the simplest case the model can be a set of 3D points. More sophisticated methods produce a complete 3D surface model. The advent of 3D imaging not requiring motion or scanning, and related processing algorithms is enabling rapid advances in this field. Grid-based 3D sensing can be used to acquire 3D images from multiple angles. Algorithms are now available to stitch multiple 3D images together into point clouds and 3D models.

* **Image Resolution**

The aim of image restoration is the removal of noise (sensor noise, motion blur, etc.) from images. The simplest possible approach for noise removal is various types of filters such as low-pass filters or median filters. More sophisticated methods assume a model of how the local image structures look, to distinguish them from noise. By first analyzing the image data in terms of the local image structures, such as lines or edges, and then controlling the filtering based on local information from the analysis step, a better level of noise removal is usually obtained compared to the simpler approaches.

* 1. **WHAT IS MACHINE LEARNING?**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

* 1. **Some Machine Learning Methods**
* **Supervised machine learning** algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
* In contrast, **unsupervised machine learning** algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
* **Semi-supervised machine learning** algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn’t require additional resources.
* **Reinforcement machine learning** algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

CHAPTER 4

TECHNIQUES INVOLVED

**4.1 APPROACHES TOWARDS THE PROBLEM**

Facial Expression Recognition is an Image Classification problem located within the wider field of Computer Vision. Image Classification problems are ones in which images must be algorithmically assigned a label from a discrete set of categories. In FER systems specifically, the images are of human faces and the categories are a set of emotions.

Machine learning approaches to FER all require a set of training image examples, each labeled with a single emotion category. A standard set of seven emotion classifications are often used:

* Anger
* Disgust
* Fear
* Happiness
* Sadness
* Surprise
* Neutral

These emotion classifications are illustrated in the image below, showing representative sample images taken from this 2014 paper on expression recognition.

Classifying an image based on its depiction can be a complicated task for machines. It is straightforward for humans to look at an image of a bicycle and know that it is a bicycle, or to look at a person’s face and know that they are smiling and happy.



A selection of labeled images for expression analysis

When computers look at an image, what they ‘see’ is simply a matrix of pixel values. In order to classify an image, the computer has to discover and classify numerical patterns within the image matrix.

These patterns can be variable, and hard to pin down for multiple reasons. Several human emotions can be distinguished only by subtle differences in facial patterns, with emotions like anger and disgust often expressed in very similar ways. Each person’s expressions of emotions can be highly idiosyncratic, with particular quirks and facial cues. There can be a wide variety of divergent orientations and positions of people’s heads in the photographs to be classified.

For these types of reasons, FER is more difficult than most other Image Classification tasks. However, well-designed systems can achieve accurate results when constraints are taken into account during development.

For example, higher accuracy can be achieved when classifying a smaller subset of highly distinguishable expressions, such as anger, happiness, and fear. Lower accuracy is achieved when classifying larger subsets, or small subsets with less distinguishable expressions, such as anger and disgust.

Like most image classification systems, FER systems typically use image preprocessing and feature extraction followed by training on selected training architectures. The end result of training is the generation of a model capable of assigning emotion categories to newly provided image examples.



**4.2 COMPARISON OF TRAINING ALGORITHMS**

Once any feature extraction or image preprocessing stages are complete, the training algorithm produces a trained prediction model. A number of options exist for training FER models, each of which has strengths and weaknesses making them more or less suitable for particular situations.

We will compare some of the most commonly use algorithms:

* **Multiclass Support Vector Machines (SVM)**

It is a supervised learning algorithms that analyze and classify data, and they perform well when classifying human facial expressions. However, they only do so when the images are created in a controlled lab setting with consistent head poses and illumination. SVMs perform less well when classifying images captured “in the wild,” or in spontaneous, uncontrolled settings. Therefore, the latest training architectures being explored are all deep neural networks which perform better under those circumstances. Convolutional Neural Networks (CNN) are currently considered the go-to neural networks for image classification, because they pick up on patterns in small parts of an image, such as the curve of an eyebrow.

* **Convolution Neural Network(CNN)**

CNNs apply kernels, which are matrices smaller than the image, to chunks of the input image. By applying kernels to inputs, new activation matrices, sometimes referred to as feature maps, are generated and passed as inputs to the next layer of the network. In this way, CNNs process more granular elements within an image, making them better at distinguishing between two similar emotion classifications.

* **Recurrent Neural Network(RNN)**

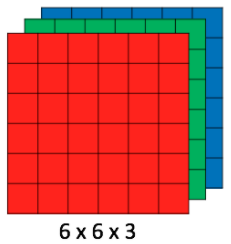
Recurrent Neural Networks (RNN) use dynamic temporal behavior when classifying an image. This means that when an RNN processes an input example, it doesn’t just look at the data from that example — it also looks at the data from previous inputs, which are used to provide further context. In FER, the context could be previous image frames of a video clip.

The idea of this approach is to capture the transitions between facial patterns over time, allowing these changes to become additional data points supporting classification. For example, it is possible to capture the changes in the edges of the lips as an expression goes from neutral to happy by smiling, rather than just the edges of a smile from an individual image frame.

**4.3 CNN ALGORITHM**

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d( h = Height, w = Width, d = Dimension ). Eg., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image.



**Figure 1 : Array of RGB Matrix**

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

****

**Figure 2 : Neural network with many convolutional layers**

For our project we will be using 4 convolution layers and 2 fully connected layers. The final step that is the SOFTMAX step for our project involves the 7 emotions .

**Convolution Layer**

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

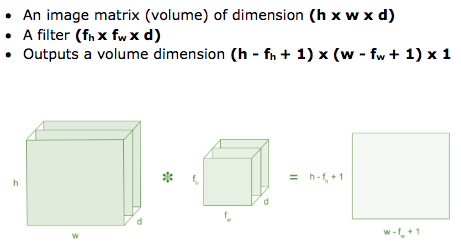
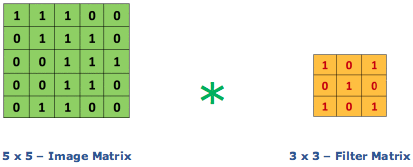


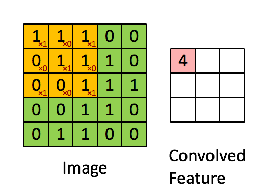
Figure 3: Image matrix multiplies kernel or filter matrix

Consider a 5 x 5 whose image pixel values are 0, 1 and filter matrix 3 x 3 as shown in below



**Figure 4: Image matrix multiplies kernel or filter matrix**

Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called “Feature Map” as output shown in below.

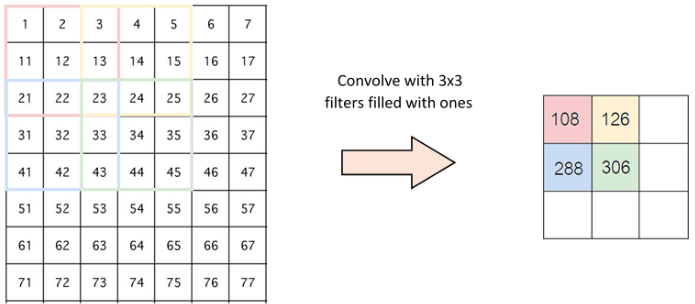


**Figure 5: 3 x 3 Output matrix**

Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters. The below example shows various convolution image after applying different types of filters (Kernels).

**Strides**

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2.



**Figure 6 : Stride of 2 pixels**

**Padding**

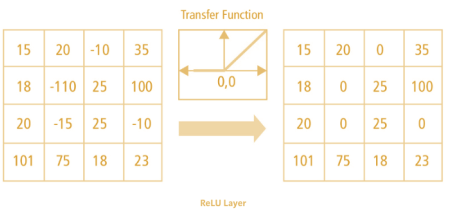
Sometimes filter does not fit perfectly fit the input image. We have two options:

* Pad the picture with zeros (zero-padding) so that it fits
* Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

**Non Linearity (ReLU)**

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is ƒ(x) = max(0,x).

ReLU’s purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values.



**Figure 7 : ReLU operation**

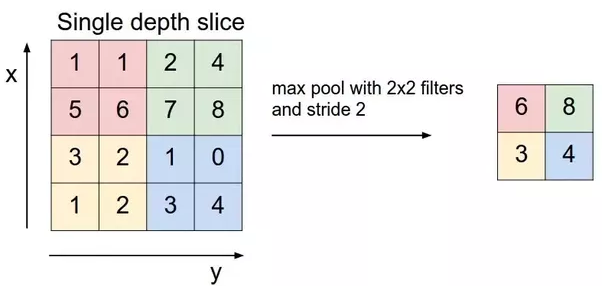
There are other non linear functions such as tanh or sigmoid can also be used instead of ReLU. Most of the data scientists uses ReLU since performance wise ReLU is better than other two.

**Pooling Layer**

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains the important information. Spatial pooling can be of different types:

* Max Pooling
* Average Pooling
* Sum Pooling

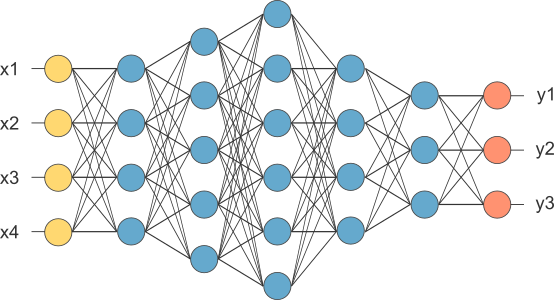
Max pooling take the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.



**Figure 8 : Max Pooling**

**Fully Connected Layer**

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like neural network.



**Figure 9 : After pooling layer, flattened as FC layer**

In the above diagram, feature map matrix will be converted as vector (x1, x2, x3, …). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax or sigmoid to classify the outputs as cat, dog, car, truck etc.,

Steps involved in CNN:

* Provide input image into convolution layer
* Choose parameters, apply filters with strides, padding if requires. Perform convolution on the image and apply ReLU activation to the matrix.
* Perform pooling to reduce dimensionality size
* Add as many convolutional layers until satisfied
* Flatten the output and feed into a fully connected layer (FC Layer)
* Output the class using an activation function (Logistic Regression with cost functions) and classifies images.

**4.4 CONVERSION TO GRAYSCALE**

##### 

##### COLOR CODES CONVERSION

All the colors here are of the 24 bit format, which means each color has 8 bits of red, 8 bits of green, 8 bits of blue, in it. Or we can say each color has three different portions. We just have to change the quantity of these three portions to make any color.

Binary color format Color: Black Image:



Decimal Code: (0,0,0)

Explanation: As it has been explained in the previous tutorials, that in an 8-bit format, 0 refers to black. So if we have to make a pure black color, we have to make all the three portion of R, G, B to 0.

Color: White Image:



Decimal Code: (255,255,255)

Explanation: Since each portion of R, G, B is an 8 bit portion. So in 8-bit, the white color is formed by 255. It is explained in the tutorial of pixel. So in order to make a white color we set each portion to 255 and that’s how we got a white color. By setting each of the value to 255, we get overall value of 255, that’s making the color white.

RGB color model: Color: Red Image:



Decimal Code: (255,0,0)

Explanation: Since we need only red color, so we zero out the rest of the two portions which are green and blue, and we set the red portion to its maximum which is 255.

Color: Green Image:



Decimal Code: (0,255,0)

Explanation: Since we need only green color, so we zero out the rest of the two portions which are red and blue, and we set the green portion to its maximum which is 255.

Color: Blue Image:



Decimal Code: (0,0,255)

Explanation: Since we need only blue color, so we zero out the rest of the two portions which are red and green, and we set the blue portion to its maximum which is 255

Gray color: Color: Gray Image:



Decimal Code: (128,128,128)

Explanation: As we have already defined in our tutorial of pixel, that gray color Is actually the mid point. In an 8-bit format, the mid point is 128 or 127. In this case we choose 128. So we set each of the portion to its mid point which is 128, and that results in overall mid value and we got gray color.

CMYK color model:

CMYK is another color model where c stands for cyan, m stands for magenta, y stands for yellow, and k for black. CMYK model is commonly used in color printers in which there are two carters of color is used. One consist of CMY and other consist of black color.

The colors of CMY can also made from changing the quantity or portion of red, green and blue.

Color: Cyan Image:



Decimal Code: (0,255,255)

Explanation: Cyan color is formed from the combination of two different colors which are Green and blue. So we set those two to maximum and we zero out the portion of red. And we get cyan color.

Color: Magenta Image:



Decimal Code: (255,0,255)

Explanation: Magenta color is formed from the combination of two different colors which are Red and Blue. So we set those two to maximum and we zero out the portion of green. And we get magenta color.

Color: Yellow Image:



Decimal Code: (255,255,0)

Explanation: Yellow color is formed from the combination of two different colors which are Red and Green. So we set those two to maximum and we zero out the portion of blue. And we get yellow color.

##### GRAYSCALE TO RGB CONVERSION

There are two methods to convert a GRAYSCALE image to an RGB image. Both have their own merits and demerits. The methods are:

* + - Average method
    - Weighted method or luminosity method Average method

Average method is the simplest one. You just have to take the average of three colors. Since it

is an RGB image, so it means that you have add r with g with b and then divide it by 3 to get your desired grayscale image.

It’s done in this way. Grayscale = (R + G + B / 3)

For example:



fig5: color image

If you have a color image like the image shown above and you want to convert it into grayscale using average method. The following result would appear.



fig6: grayscale image

Explanation

There is one thing to be sure; that something happens to the original works. It means that our average method works. But the results were not as expected. We wanted to convert the image into a grayscale, but this turned out to be a rather black image.

Problem

This problem arises due to the fact, that we take average of the three colors. Since the three different colors have three different wavelengths and have their own contribution in the

formation of image, so we have to take average according to their contribution, not done it averagely using average method. Right now what we are doing is this, 33% of Red, 33% of Green, 33% of Blue.

We are taking 33% of each, which means, each of the portion has same contribution in the image. But in reality that’s not the case. The solution to this has been given by luminosity method.

Weighted method or luminosity method

You have seen the problem that occurs in the average method. Weighted method has a solution to that problem. Since red color has more wavelengths of all the three colors, and green is the color that has not only less wavelength then red color but also green is the color that gives more soothing effect to the eyes.

It means that we have to decrease the contribution of red color, and increase the contribution of the green color, and put blue color contribution in between these two.the new equation that forms is:

New grayscale image = ((0.3 \* R) + (0.59 \* G) + (0.11 \* B)).

According to this equation, Red has contributed 30%, Green has contributed 59% which is greater in all three colors and Blue has contributed 11%..Applying this equation , we get this

Fig 7. Original Image

**Fig 8:Grayscale Image**



Explanation

As you can see here, that the image has now been properly converted to grayscale using weighted method. As compare to the result of average method, this image is brighter.

##### HISTOGRAM

A histogram is a graph. A graph that shows frequency of anything. Usually histograms have bars that represent frequency of occurring of data in the whole data set.

A Histogram has two axes the x axis and the y axis. The x axis contains event whose frequency you have to count. The y axis contains frequency.

The different heights of the bars show different frequency of occurrence of data. Usually a histogram looks like this.

Histogram of an image:

Histogram of an image, like other histograms also shows frequency. But an image histogram, shows frequency of pixels intensity values. In an image histogram, the x axis shows the gray level intensities and the y axis shows the frequency of these intensities.

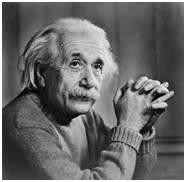


fig 9: experimental image for histogram

The histogram of the above picture of the Einstein would be something like this

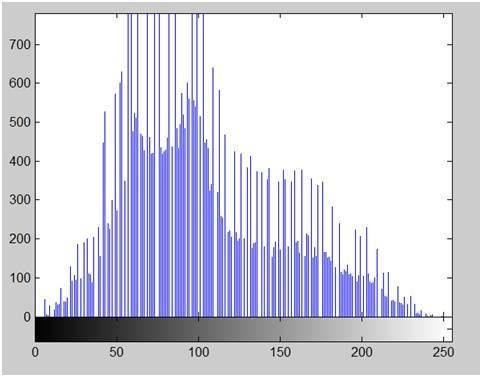


fig 10: histogram

The x axis of the histogram shows the range of pixel values. Since it’s an 8 bpp image, which means it has 256 levels of gray or shades of gray in it. That’s why the range of x axis starts from 0 and end at 255 with a gap of 50. Where as, on the y axis, is the count of these intensities. As you can see from the graph, that most of the bars that have high frequency lies in the first half portion which is the darker portion. That means that the image we have got is darker. And this can be proved from the image too.

Histograms have many uses in image processing. The first use as it has also been discussed above is the analysis of the image. We can predict about an image by just looking at its histogram. Its like looking an x ray of a bone of a body. The second use of histogram is for

brightness purposes. The histograms have wide application in image brightness. Not only in brightness, but histograms are also used in adjusting contrast of an image.

##### EDGE DETECTION

What are edges

We can also say that sudden changes of discontinuities in an image are called as edges. Significant transitions in an image are called as edges.

Types of edges

Generally edges are of three types:

* + - Horizontal edges
    - Vertical Edges
    - Diagonal Edges Why detect edges

Most of the shape information of an image is enclosed in edges. So first we detect these edges

in an image and by using these filters and then by enhancing those areas of image which contains edges, sharpness of the image will increase and image will become clearer.

Here are some of the masks for edge detection that we will discuss in the upcoming tutorials.

* + - Prewitt Operator
    - Sobel Operator
    - Robinson Compass Masks
    - Krisch Compass Masks
    - Laplacian Operator.

Above mentioned all the filters are Linear filters or smoothing filters.

Prewitt Operator

Prewitt operator is used for detecting edges horizontally and vertically.

Sobel Operator

The sobel operator is very similar to Prewitt operator. It is also a derivate mask and is used for edge detection. It also calculates edges in both horizontal and vertical direction.

Robinson Compass Masks

This operator is also known as direction mask. In this operator we take one mask and rotate it in all the 8 compass major directions to calculate edges of each direction.

Kirsch Compass Masks

Kirsch Compass Mask is also a derivative mask which is used for finding edges. Kirsch mask is also used for calculating edges in all the directions.

Laplacian Operator

Laplacian Operator is also a derivative operator which is used to find edges in an image. Laplacian is a second order derivative mask. It can be further divided into positive laplacian and negative laplacian.

All these masks find edges. Some find horizontally and vertically, some find in one direction only and some find in all the directions. The next concept that comes after this is sharpening which can be done once the edges are extracted from the image

Sharpening

Sharpening is opposite to the blurring. In blurring, we reduce the edge content and in Sharpening, we increase the edge content. So in order to increase the edge content in an image, we have to find edges first. Edges can be find by one of the any method described above by using any operator. After finding edges, we will add those edges on an image and thus the image would have more edges, and it would look sharpen.

The sharpen image is shown below.



Fig 11:Original Image

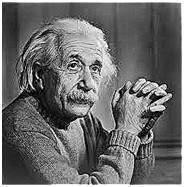


Fig 12:Sharpen Image

##### BASICS OF VIDEO

A video is formed by combining a number of images which are referred to as frames and the quality of each frame determines the overall quality of the video. Another important term in a video is **Frame Rate**. It is expressed in **frames/second** and refers to the number of frames that appear as a whole in a video per second. Based on this frame rate, we can obtain either a slow- motion video with more frame rate or a fast-motion video with a smaller frame rate.

Video processing indirectly applies processing on the frames of a video rather than the video itself. Audio processing is a totally different branch of video processing.

## CHAPTER 4 DIGITAL WATERMARKING

##### INTRODUCTION

Digital watermarking is the act of hiding a message related to a digital signal (i.e. an image, song, video) within the signal itself. It is a concept closely related to steganography, in that they both hide a message inside a digital signal. However, what separates them is their goal.

Watermarking tries to hide a message related to the actual content of the digital signal, while in steganography the digital signal has no relation to the message, and it is merely used as a cover to hide its existence.

##### TYPES OF WATERMARKING

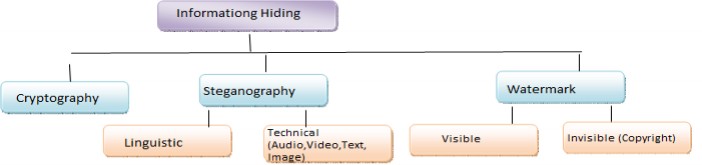


fig13: types of watermarking

A **visible** watermark is a visible semi-transparent text or image overlaid on the original image. It allows the original image to be viewed, but it still provides copyright protection by marking the image as its owner’s property. Visible watermarks are more robust against image transformation (especially if you use a semi-transparent watermark placed over whole image). Thus they are preferable for strong copyright protection of intellectual property that’s in digital format.

An **invisible** watermark is an embedded image which cannot be perceived with human’s eyes. Only electronic devices (or specialized software) can extract the hidden information to identify the copyright owner. Invisible watermarks are used to mark a specialized digital content (text,

35

images or even audio content) to prove its authenticity.

Although the copyright protection is the main field of using digital watermarks, they can also be used for such purposes as advertising (adding company’s name and logo as a watermark for promotion rather than for protection) or even adding memo titles to digital photos. It’s obvious that only visible watermarks can satisfy these requirements.

##### WATERMARKING VS CRYPTOGRAPHY VS STEGANOGRAPHY

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Steganography** | **Cryptography** | **Watermarking** |
| Techniques | LSB, Spatial domain, JSTEG, outguess | Transposition, substitution, RSA | Compensated prediction, DCT |
| Naked eye identification | No, as message is hidden within other carrier | Yes, as message is converted in other way which sough something is hidden | Yes, as actual message is hidden by some watermark |
| Capacity | Differs as different technology usually low hiding capacity | Capacity is so high, but as message is long, it has chances to be  decrypted | Capacity depends on size of hidden data |
| Detection | Not easy to detect because finding steganographic  image is hard | Not easy to detect , depend on technology used to  generate | Not easy to detect |
| Strength | Hide message without altering the message, it conceals  information | Hide message by altering the message by assigning key | Extend information and become an attribute of the  cover image |
| Imperceptibility | high | High | High |
| Applicability | universally | Universally | Universally |
| Robust | yes | Yes | Yes |

##### WATERMARKING APPLICATIONS

The increasing amount of research on watermarking over the past decade has been largely driven by its important applications in digital copyrights management and protection.

One of the first applications for watermarking was broadcast monitoring. It is often crucially important that we are able to track when a specific video is being broadcast by a TV station. This is important to advertising agencies that want to ensure that their commercials are getting the air time they paid for. Watermarking can be used for this purpose. Information used to identify individual videos could be embedded in the videos themselves using watermarking, making broadcast monitoring easier.

Another very important application is owner identification. Being able to identify the owner of a specific digital work of art, such as a video or image can be quite difficult. Nevertheless, it is a very important task, especially in cases related to copyright infringement. So, instead of including copyright notices with every image or song, we could use watermarking to embed the copyright in the image or the song itself.

Transaction tracking is another interesting application of watermarking. In this case the watermark embedded in a digital work can be used to record one or more transactions taking place in the history of a copy of this work. For example, watermarking could be used to record the recipient of every legal copy of a movie by embedding a different watermark in each copy. If the movie is then leaked to the Internet, the movie producers could identify which recipient of the movie was the source of the leak.

Finally, copy control is a very promising application for watermarking. In this application, watermarking can be used to prevent the illegal copying of songs, images of movies, by embedding a watermark in them that would instruct a watermarking compatible DVD or CD writer to not write the song or movie because it is an illegal copy.

##### WATERMARKING PROPERTIES

Every watermarking system has some very important desirable properties. Some of these properties are often conflicting and we are often forced to accept some tradeoffs between these properties depending on the application of the watermarking system. The first and perhaps most important property is effectiveness. This is the probability that the message in a watermarked image will be correctly detected. We ideally need this probability to be 1.

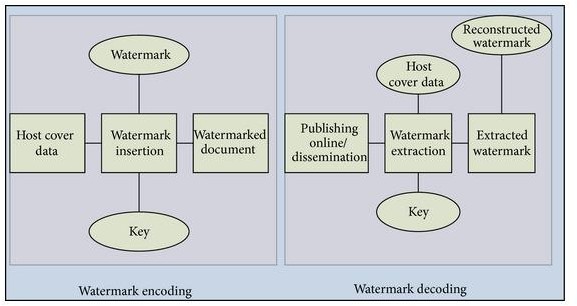
Another important property is the image fidelity. Watermarking is a process that alters an original image to add a message to it, therefore it inevitably affects the image’s quality. We want to keep this degradation of the image’s quality to a minimum, so no obvious difference in the image’s fidelity can be noticed. The third property is the payload size. Every watermarked work is used to carry a message.

The size of this message is often important as many systems require a relatively big payload to be embedded in a cover work. There are of course applications that only need a single bit to be embedded. The false positive rate is also very important to watermarking systems. This is the number of digital works that are identified to have a watermark embedded when in fact they have no watermark embedded. This should be kept very low for watermarking systems. Lastly, robustness is crucial for most watermarking systems.

There are many cases in which a watermarked work is altered during its lifetime, either by transmission over a lossy channel or several malicious attacks that try to remove the watermark or make it undetectable. A robust watermark should be able to withstand additive Gaussian noise, compression, printing and scanning, rotation, scaling, cropping and many other operations.

##### WATERMARKING LIFECYCLE PHASES

The advice to be anchored in a arresting is alleged a agenda watermark, although in some contexts the byword agenda watermark agency the aberration amid the watermarked arresting and the awning signal. The arresting area the watermark is to be anchored is alleged the host signal. A watermarking arrangement is usually disconnected into three audible steps, embedding, attack, and detection. In embedding, an algorithm accepts the host and the abstracts to be embedded, and produces a watermarked signal.



**fig14: watermarking phases**

Then the watermarked agenda arresting is transmitted or stored, usually transmitted to addition person. If this being makes a modification, this is alleged an attack. While the modification may not be malicious, the appellation advance arises from absorb aegis application, area pirates attack to abolish the agenda watermark through modification. There are abounding accessible modifications, for example, lossy compression of the abstracts (in which resolution is diminished), agriculture an angel or video, or carefully abacus noise.

Detection (often alleged extraction) is an algorithm which is activated to the attacked arresting to

attack to abstract the watermark from it. If the arresting was blunt during transmission, again the watermark still is present and it may be extracted. In able-bodied agenda watermarking applications, the abstraction algorithm should be able to aftermath the watermark correctly, alike if the modifications were strong. In brittle agenda watermarking, the abstraction algorithm should abort if any change is fabricated to the signal.

## CHAPTER 5 WATERMARKING TECHNIQUES

In this chapter, we would like to combine the four major watermarking techniques which were not integrated so far. The four techniques are as follows:

##### BIT PLANE SLICING

Instead of highlighting gray level images, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that each pixel in an image is represented by 8 bits. Imagine the image is composed of 8, 1-bit planes ranging from bit plane1-0 (LSB)to bit plane 7 (MSB).

In terms of 8-bits bytes, plane 0 contains all lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all high order bits.

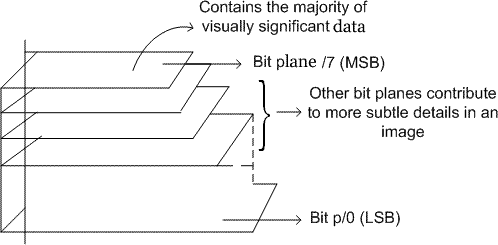


fig 15:bit plane slicing

Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, implying, it determines the adequacy of numbers of bits used to quantize each pixel, useful for image compression.

In terms of bit-plane extraction for a 8-bit image, it is seen that binary image for bit plane 7 is obtained by proceeding the input image with a thresholding gray-level transformation function that maps all levels between 0 and 127 to one level (e.g 0)and maps all levels from 129 to 253 to another (eg. 255).

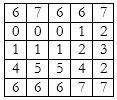
The gray level of each pixel in a digital image is stored as one or more bytes in a computer. For an 8-bit image, 0 is encoded as 00000000 and 255 is encoded as 11111111. Any number between 0 t0 255 is encoded as one byte. The bit in the far left side is referred as the most significant bit (MSB) because a change in that bit would significantly change the value encoded by the byte.

The bit in the far right is referred as the least significant bit (LSB), because a change in this bit does not change the encoded gray value much. The bit plane representation of an eight-bit digital image is given by:



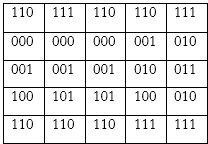
Bit plane slicing is a method of representing an image with one or more bits of the byte used for each pixel. One can use only MSB to represent the pixel, which reduces the original gray level to a binary image. The three main goals of bit plane slicing is:

* + - Converting a gray level image to a binary image.
    - Representing an image with fewer bits and corresponding the image to a smaller size
    - Enhancing the image by focussing.

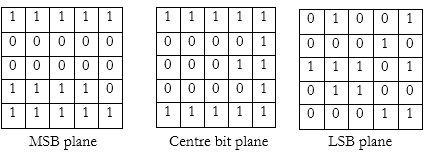


**Bit plane slicing:**

Since the given image has a maximum grey level of 7, it is a 3-bit image. We convert the image to binary and separate the bit planes.



Separating the bit planes, we obtain



##### ARNOLD TRANSFORMATION

Arnold’s Cat Map algorithm (ACM) is one of the cryptographic algorithm used to encrypt the image. The concept of the algorithm is continuously rotate the image so that it becomes a form that is not visible and random so that the image cannot be seen by the naked eye but can still be recognized by the system for image file (image) of the same. Arnold’s Cat Map Algorithm Chaos is a common technique used in the random number generator, it's happening because this

technique is faster and easier to use in the process stream object both in terms of storage and process objects.

Only a few functions (chaotic maps) and some parameters (initial conditions) were quite good used if the process takes quite a long time. Arnold's Cat Maps are chaotic two dimensions that can be used to change the position of the pixel of the image without removing any information from the image, pixel image can be assumed by

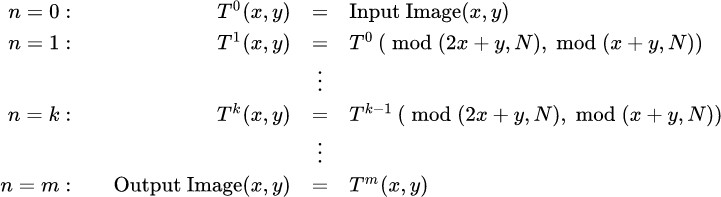
##### S = {(x, y) | x, y = 0, 1, 2 ... N-1}

2-dimensional image of Arnold's Cat Map can be written by the following equation:



where N is the dimension of the square matrix. When N is set to an integer value, the position and momentum variables can be restricted to integers and the mapping becomes a mapping of a toroidial square grid of points onto itself. The number of iterations needed to restore the image can be shown never to exceed 3N.

For an image, the relationship between iterations could be expressed as follows:



##### SINGULAR VALUE DECOMPOSITION

Singular value decomposition takes a rectangular matrix of gene expression data (defined as A, where A is a *n* x *p* matrix) in which the *n* rows represents the genes, and the *p* columns represents the experimental conditions. The SVD theorem states:

**A*nxp*= U*nxn* S*nxp* VT**

***pxp***

Where

**U**T**U** = **I**nxn

**V**T**V** = **I**pxp (i.e. U and V are orthogonal)

Where the columns of U are the left singular vectors (*gene coefficient vectors*); S (the same dimensions as *A*) has singular values and is diagonal (*mode amplitudes*); and VT has rows that are the right singular vectors (*expression level vectors*). The SVD represents an expansion of the original data in a coordinate system where the covariance matrix is diagonal.

Calculating the SVD consists of finding the eigenvalues and eigenvectors of *AAT* and *ATA*. The eigenvectors of *ATA* make up the columns of *V* , the eigenvectors of *AAT* make up the columns

of *U*. Also, the singular values in **S** are square roots of eigenvalues from *AAT* or *ATA*. The singular values are the diagonal entries of the *S* matrix and are arranged in descending order. The singular values are always real numbers. If the matrix *A* is a real matrix, then *U* and *V* are also real.

To understand how to solve for SVD, let’s take the example of the matrix that was provided in Kuruvilla *et al*:

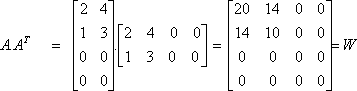


In this example the matrix is a 4x2 matrix. We know that for an n x n matrix W, then a nonzero vector **x** is the eigenvector of W if:

W **x** = l **x**

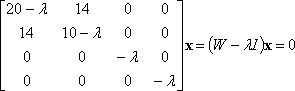
For some scalar l. Then the scalar l is called an eigenvalue of A, and **x** is said to be an eigenvector of A corresponding to l.

So to find the eigenvalues of the above entity we compute matrices *AAT* and *ATA*. As previously stated , the eigenvectors of *AAT* make up the columns of *U* so we can do the following analysis to find U.



Now that we have a n x n matrix we can determine the eigenvalues of the matrix W.

Since W **x** = l **x** then (W- lI) **x** = 0



For a unique set of eigenvalues to determinant of the matrix (W-lI) must be equal to zero. Thus from the solution of the characteristic equation, |W-lI|=0 we obtain:

l=0, l=0; l = 15+Ö221.5 ~ 29.883; l = 15-Ö221.5 ~ 0.117 (four eigenvalues since it is

a fourth degree polynomial). This value can be used to determine the eigenvector that can be placed in the columns of U. Thus we obtain the following equations:

19.883 x1 + 14 x2 = 0

14 x1 + 9.883 x2 = 0

x3 = 0

x4 = 0

Upon simplifying the first two equations we obtain a ratio which relates the value of x1 to

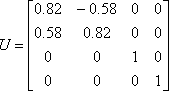
x2. The values of x1 and x2 are chosen such that the elements of the S are the square roots of the eigenvalues. Thus a solution that satisfies the above equation x1 = -0.58 and x2 = 0.82 and x3 = x4 = 0 (this is the second column of the U matrix).

Substituting the other eigenvalue we obtain:

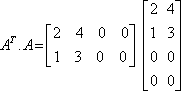
-9.883 x1 + 14 x2 = 0

14 x1 - 19.883 x2 = 0 x3 = 0; x4=0

Thus a solution that satisfies this set of equations is x1 = 0.82 and x2 = -0.58 and x3 = x4 = 0 (this is the first column of the U matrix). Combining these we obtain:



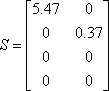
Similarly *ATA* makes up the columns of *V* so we can do a similar analysis to find the value of V.



and similarly we obtain the expression:



Finally as mentioned previously the S is the square root of the eigenvalues from *AAT* or *ATA.* and can be obtained directly giving us:



Note that: s1 > s2 > s3 > … which is what the paper was indicating by the figure 4 of

the Kuruvilla paper. In that paper the values were computed and normalized such that the highest singular value was equal to 1.

Proof:

**A**=**USV**T and **A**T=**VSU**T **A**T**A** = **VSU**T**USV**T

**A**T**A** = **VS**2**V**T **A**T**AV** = **VS**2

##### NON NEGATIVE MATRIX FACTORIZATION

One major drawback of SVD is that the basis vectors may have both positive and negative components and the data are represented as linear combinations of basis vectors of positive and negative coefficients. In many applications the negative components contradict physical realities

and to address this problem, NMF approach was proposed to search for a representative basis with only negative vectors.

Nonnegative matrix factorization *(*NMF*)* is a dimension-reduction technique based on a low-rank approximation of the feature space. Besides providing a reduction in the number of features, NMF guarantees that the features are nonnegative, producing additive models that respect, for example, the non negativity of physical quantities.

Given a nonnegative m-by-n matrix X and a positive integer k < min (m,n), NMF finds nonnegative m-by-k and k-by-n matrices W and H, respectively, that minimize the norm of the difference X – WH. W and H are thus approximate nonnegative factors of X.

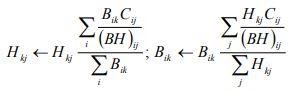
The k columns of W represent transformations of the variables in X; the k rows of H represent the coefficients of the linear combinations of the original n variables in X that produce the transformed variables in W. Since k is generally smaller than the rank of X, the

product WH provides a compressed approximation of the data in X. A range of possible values for k is often suggested by the modeling context.

The NMF can be formulated as follows. Given a cover image C of size m × m, we can approximately factorize C into the product of two non negative matrices B and H with sizes m × r and r × m respectively, that is C = BH, where r ≤ m. The non negative matrix B contains the NMF basis vectors and the nonnegative weight matrix H contains the associated coefficients (nonnegative weights). To measure the quality of approximation factorization C = BH, a cost function between C and BH needs to be optimized subject to nonnegative constraints on B and

H. This is done by minimizing the I–information divergence given by



This yields the following multiplicative update rules

## CHAPTER 6

IMPLEMENTATION OF WATERMARKING TECHNIQUES

##### IMAGE PRE-PROCESSING

Before embedding the watermark image into the cover video, few pre-processing steps are performed on the image to add robustness. These steps are as follows

###### STEP 1:

Divide the colour watermark image (wm) into red, green and blue component images.

###### STEP 2:

Bit plane slice each component image. So we get eight bit plane slices for each image and totally we have 24 bit plane slices.

###### STEP 3:

Apply Arnold transform to these 24 bit plane slices so that they get scrambled to add more strength to the embedding methodology used.

###### STEP 4:

Apply non negative matrix factorization (NMF)on each image to get weight matrix (W) and coefficient matrix (H). To this coefficient matrix, apply singular value decomposition (SVD). On applying SVD we get 3 more components namely unitary matrix (U), rectangular diagonal matrix (S) and complex unitary matrix (V) where S matrix has the singular values to be embedded.

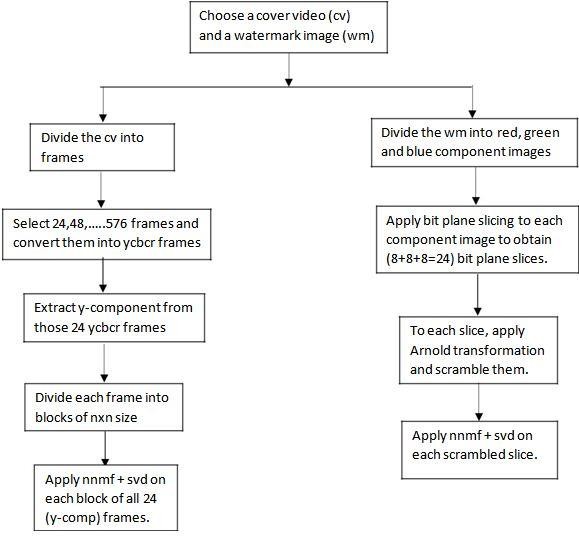


Fig 16: preprocessing

##### VIDEO PRE PROCESSING

To embed an image into a cover video, the video must be pre processed to match the required constraints. These pre processing steps makes it easier to embed the image. The following are the pre processing steps.

###### STEP 1:

Split the cover video into frames which are generally images.

###### STEP 2:

Select few frames to embed the water mark. Here we select the multiples of 24 frames (such as 24,48,72…576). So these 24 video frames are modified as follows.

###### STEP 3:

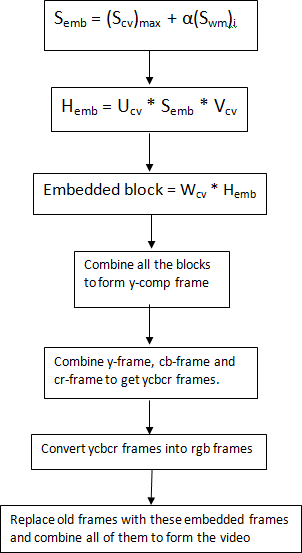
Convert these 24 rgb frames into ycbcr frames and extract y component for further implementation and preserve the cb,cr frames for further use.

###### STEP 4:

Divide these 24 y component frames into blocks of nxn size .

###### STEP 5:

Now apply NMF and SVD on these blocks of all 24 frames.



###### Fig 17: embedding

##### EMBEDDING

After pre processing, the image and video become eligible for embedding procedure. These pre processing procedures ensure that any image of any format can be embedded into any video of any format. The steps involved in embedding are

###### STEP 1:

Modify the singular values of cover video frame blocks by adding the singular values of watermark which are scaled by a factor α.

###### STEP 2:

These modified singular values are then used to form the modified coefficient matrix by performing inverse SVD.

###### STEP 3:

The above modified coefficient matrix is then combined with the original weigh matrix to get an embedded block.

###### STEP 4:

These embedded blocks are then combined to form an embedded image.

###### STEP 5:

The above obtained image is the y component which is yet to be combined with the respective cb,cr components.

###### STEP 6:

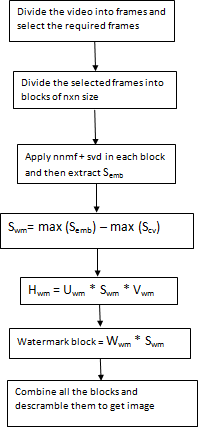
The obtained ycbcr image is then converted back into rgb image.

###### STEP 7:

The above steps should be repeated for remaining 23 frames and replace the original frames of the video with these frames.

###### STEP 8:

Combine all the frames back into a video which is the embedded video**.**



###### Fig 18: extraction

##### EXTRACTION

The following extraction procedure is suggested to extract the embedded watermark from cover video.

###### STEP 1:

Divide the video into frames and select the required frames (which were used embed the watermark in previous concept.

###### STEP 2:

Divide the above selected frames into blocks of nxn size.

###### STEP 3:

Apply NMF and SVD on these blocks to obtain the required singular values.

###### STEP 4:

Using the reverse algebra of step 1 in embedding procedure, we obtain the singular values of the required watermark.

###### STEP 5:

Obtain the coefficient matrix of watermark image by multiplying the above singular values with the preserved unitary matrices.

###### STEP 6:

Now obtain the image by multiplying the coefficient matrix with its respective weight matrix. This is the extracted watermark image.

## CHAPTER 7 RESULTS

#### [7.1 RESULTS](#_bookmark40) OF EMBEDDED VIDEO



Fig 19. Y-component of original video frame Fig 20. Y-component of embedded video frame

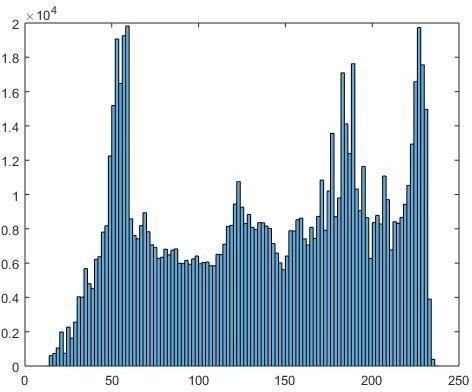
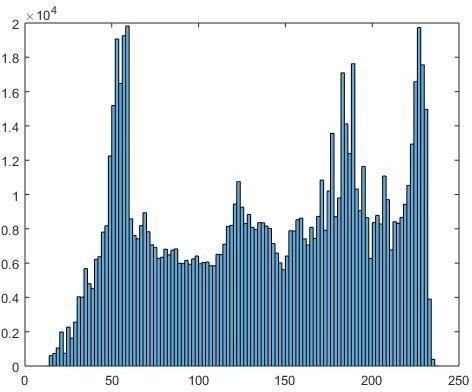


Fig 21. Histogram of original frame Fig 22. Histogram of embedded frame

**Inference:** The embedded frame and the original frame are exactly the same and so are their histograms. Thus, the original video and the embedded video match each other in every aspect. This shows the accuracy and the efficiency of the embedding process.

##### 6.2 EXTRACTION RESULTS

TEST CASE 1:

Fig 23: Original Watermark Image Fig 24: Extracted watermark Image

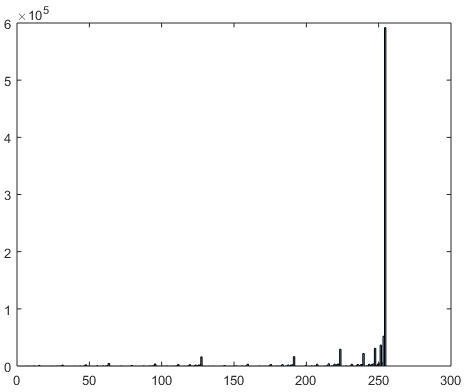
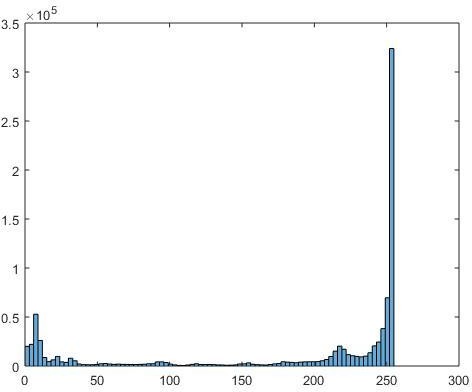


Fig25.Histogram of original image Fig 26. Histogram of extracted image

**Inference:** The image is selected so as to examine how the embedding and extraction is performed on blacks and whites in an image. Upon observing the corresponding histograms, we can infer that the black areas are degraded while the white areas are enhanced.

##### TEST CASE 2:

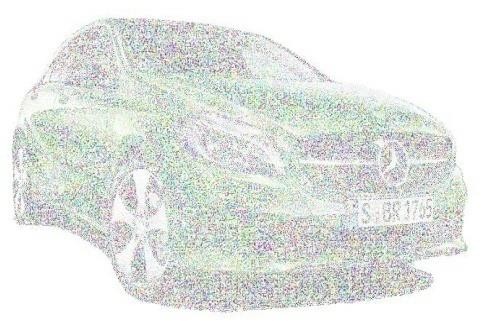


Fig 27. Original Image 2 Fig 28. Extracted Image 2

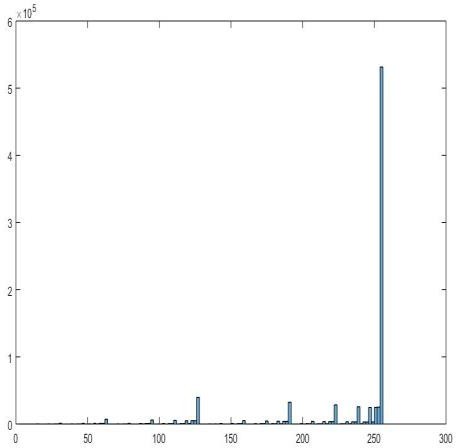
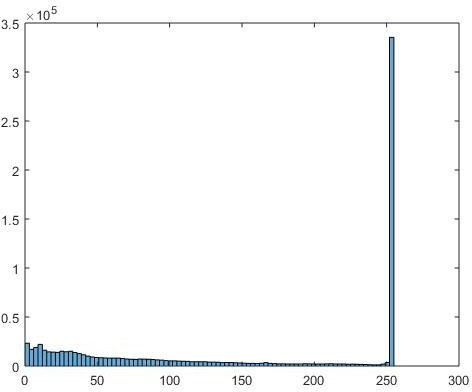


Fig29. Histogram of Original Image 2 Fig 30. Histogram of Extracted Image 2

**Inference:** The color processing capabilities are tested in this case. The histogram of the original image contains intensities even in the mid-range whereas they are faded away in the extracted image. The reason behind this is that the ‘integer’ data type evaluates any value that is in the ‘double’ form, to the nearest integer, thereby removing the mid values.

##### ORIGINAL BIT PLANE SLICES RED COMPONENT SLICES

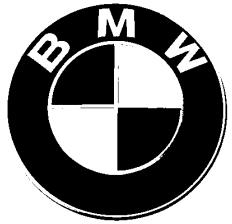
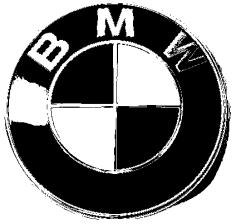
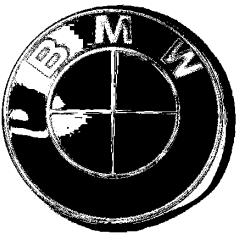
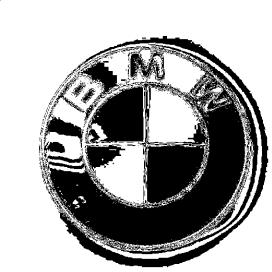


fig31:red component slices

##### GREEN COMPONENT SLICES

fig32: green component slices

##### BLUE COMPONENT SLICES

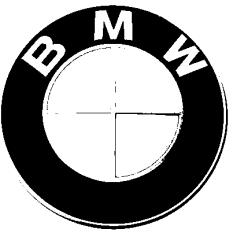
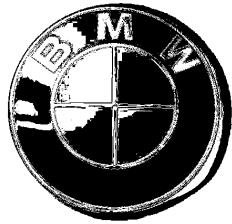
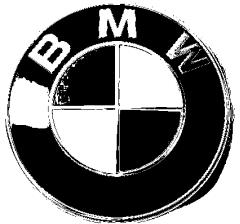
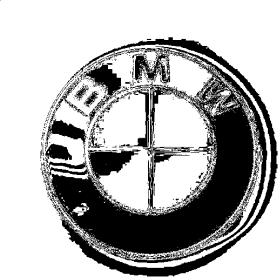


fig33: blue component slices

##### SCRAMBLED BIT PLANE SLICES RED

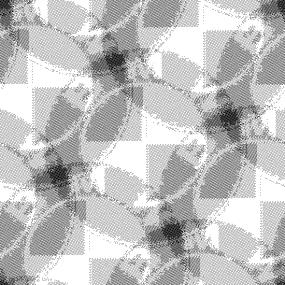
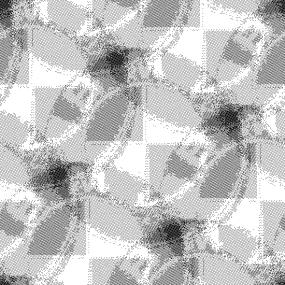
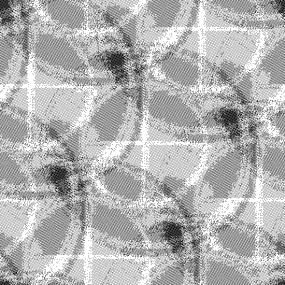
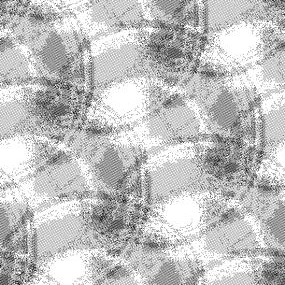
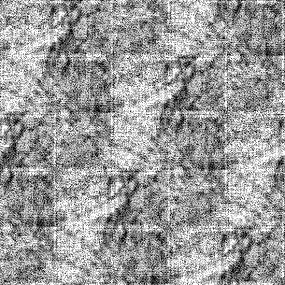


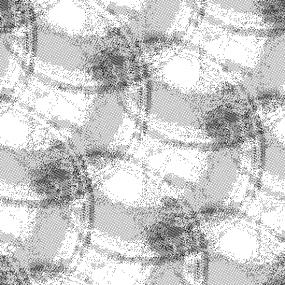
fig34: red component scrambled slices

##### GREEN

fig35: green component scrambled slices

##### BLUE





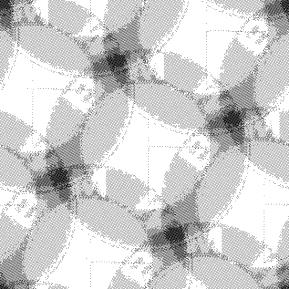
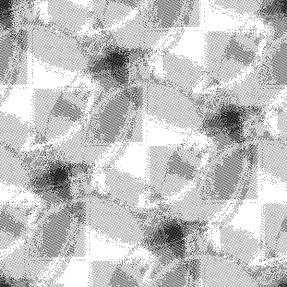
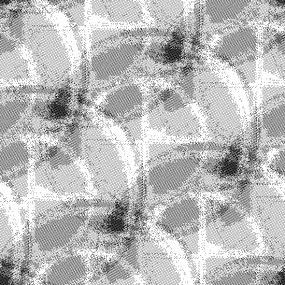
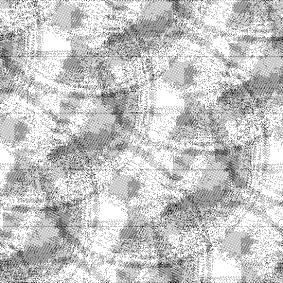
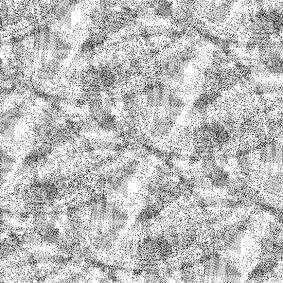
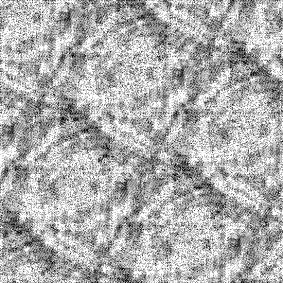
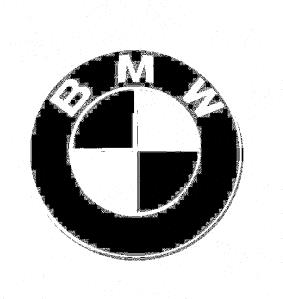
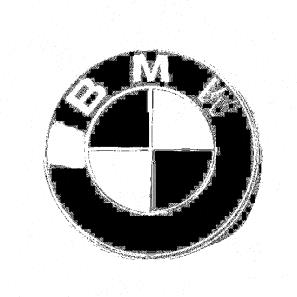
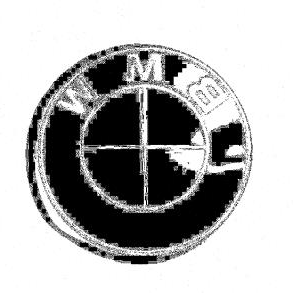
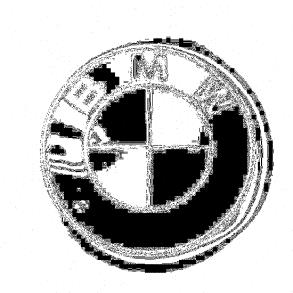


fig36: blue component scrambled slices

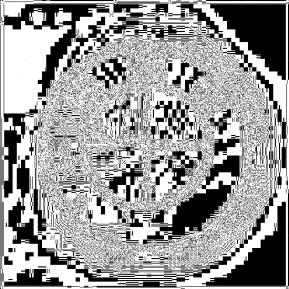
##### DESCRAMBLED SLICES RED

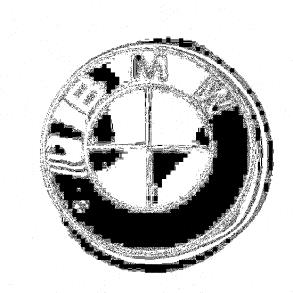




**fig37: red component descrambled slices**

##### GREEN





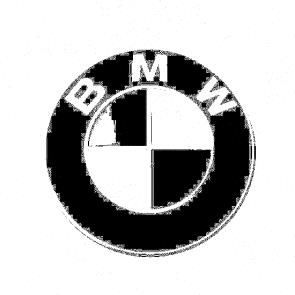
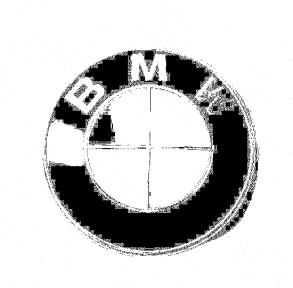
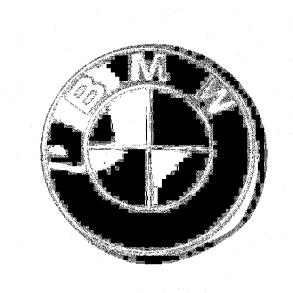
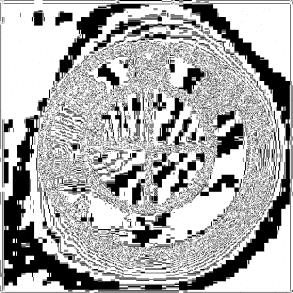
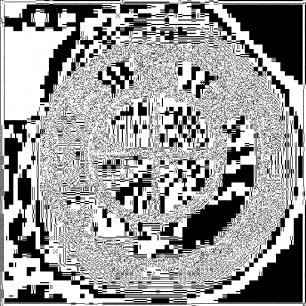
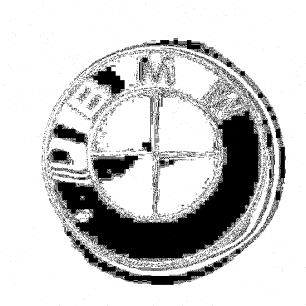


fig38: green component scrambled slices

##### BLUE





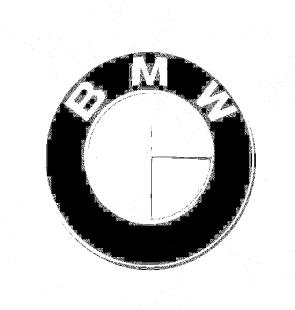
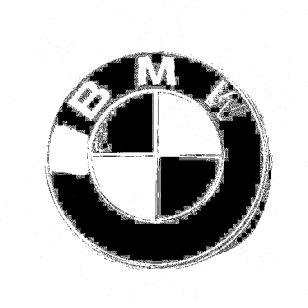
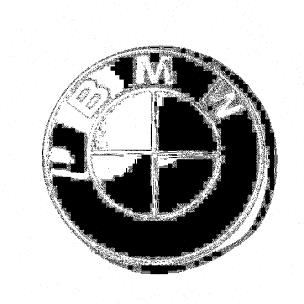
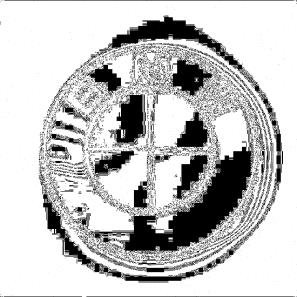
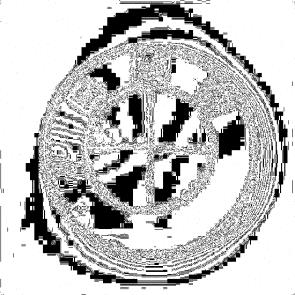
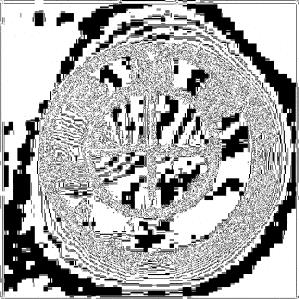


fig39: blue component scrambled slices

##### ORIGINAL 24 VIDEO FRAMES

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Fig40: original 24 video frames

##### EMBEDDED 24 VIDEO FRAMES

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Fig 41: embedded 24 video frames

## CHAPTER 8 CONCLUSION AND FUTURE SCOPE

##### CONCLUSION

In this thesis, Non-negative matrix factorization (NNMF) based color video watermarking is discussed. It is implemented with the help of Singular Value Decomposition (SVD) method. To add a layer of imperceptibility, Arnold transform is applied. The embedding process was up to the mark as there is no distortion in the cover video and is one of our achievements. The image is sliced and these slices are embedded into the video. This adds complexity in extraction. The original image and the extracted image, in two test cases, along with their histograms are given in the results section. The extraction of the watermark was not exact as the colors were not accurate but is reasonable as the image content can be recognized. The reason for this degradation in colors is found to be data type mismatch between the input image and the output image, without which we couldn’t perform the intermediate processing.

##### FUTURE SCOPE

The proposed technique is robust and gives reasonably efficient results. However, improvement can be made to this method by reducing the processing time taken by the video to convert into frames. The extracted watermark image can be further improved by taking care of data types without undergoing any mismatch. The proposed combination of nmf+svd gave better results compared to the existing embedding techniques. However, different combinations of frequency domain and spatial domain transform techniques can be tried other than the proposed nmf + svd technique which may improve our efficiency and robustness.

## REFERENCES

* + 1. Mohammadreza Ghaderpanah and A. Ben Hamza, “*A Non-Negative Matrix Factorization scheme for digital image watermarking*”.
    2. Lingling Wu, Jianwei Zhang, Weitao Deng, Dongyan He, “*Arnold Transformation Algorithm andante-Arnold Transformation Algorithm*”, 2009 1st International Conference Information Science and Engineering( ICISE)
    3. K.Kishore Kumar, Movva Pavani, V.Seshu Babu, “*Adaptive Hbrid Video Watermarking using WBCT and NMF*”, International Journal of Computer Applications (0975 – 8887) Volume 56– No.16, October 2012
    4. https://in.mathworks.com/matlabcentral/fileexchange/48844-bit-plane-slicing
    5. Tomas Kanocz, peter Goc-Matis, Patrik Gallo, “*Real-time digital video watermarking based on SVD*”, 21st International Conference Radio elecktronika 2011.
    6. Jian Liu, Xiangjian He, “ *A Review Study on Digital Watermarking*”, 2005 International Conference on Information and Communication Technologies.