PG CA S4-05 (DSE)

TWO YEAR MASTER'S DEGREE COURSE

(CBCS in Accordance with NEP 2020)

SEMESTER IV

IMAGE PROCESSING

MSc IT

(Discipline Specific Elective Course)

Hiranya Chandra Bhuyan School of Science and Technology

KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

Subject Experts

Prof. Anjana Kakoti Mahanta, Gauhati University

Prof. Hiren Kumar Deva Sarma,Gauhati University

Dr. Lipi B Mahanta, IASST, Guwahati

Dr. Sanjib Kr. Kalita, Gauhati University

Course Co-ordinator: Dr. Ridip Dev Choudhury, KKHSOU

SLM Preparation Team

UNITS	CONTRIBUTORS
1, 2	Dr. Antara Malakar, Assam Engineering College
3	Ms Jonalee Barman Kakati, NERIM
4,15	Mr. Kaustav Pratim Kalita, Barak Valley Engineering College
5,6,7,8	Dr. Pranab Das, Assam Don Bosco University
9, 11	Dr. Brajen Deka, NERIM
10, 12	Dr. Sonia Sarma, Assam Don Bosco University
13, 14	Dr. Manashi Hazarika, Assam Don Bosco University

Editorial Team

Content : Dr. Utpal Barman, Assam Skill University

Structure, Format & Graphics : Dr. Nabankur Pathak, KKHSOU

March, 2025

This Self Learning Material (SLM) of the Krishna Kanta Handiqui State Open University is made available under a Creative Commons Attribution-Non Commercial-Share Alike 4.0 License (international): http://creativecommons.org/licenses/by-nc-sa/4.0/

Printed and published by Registrar on behalf of the Krishna Kanta Handiqui State Open University

Headquarter : Patgaon, Rani Gate, Guwahati - 781017

City Office : Resham Nagar, Khanapara, Guwahati-781022; Website: www.kkhsou.in

The University acknowledges with thanks the financial support provided by the **Distance Education Bureau**, **UGC** for the preparation of this study material.

CONTENTS

		Pages
UNIT 1:	**FUNDAMENTAL OF IMAGE** Definition of image, Analog image, digital image, Advantages & disadvantages of digital image, digital image representation, pixel, neighbours of pixel	**7–13**
UNIT 2:	**IMAGE SAMPLING** Introduction, Theory of 2D sampling, violation of sampling criterion, quantization, Resolution, Anatomy of the human visual system	**14–28**
UNIT 3:	**DIGITAL IMAGE** Classification of digital image : Raster image, vector image, Image types : Binary image, grayscale image, color image, multispectral image, hyper spectral image, Elements of image processing system : Image sensor, acquisition, CCD sensor, CMOS image sensor comparison of CCD and CMOS sensors, Digital camera, Image file format, GIF file format, JPEG, PNG, TIFF, PSD, EPS, BMP, Application of digital image processing	**29–49**
UNIT 4:	**IMAGE TRANSFORM** Introduction, need for transform, Image transform, classification of image transform ,Fourier transform, 2D discrete Fourier transform, Properties of 2D DFT: Separable, spatial shift property, periodicity, convolution, correlation properties, scaling property, rotation property	**50–63**
UNIT 5:	**IMAGE ENHANCEMENT IN SPATIAL DOMAIN** Introduction, Image enhancement in spatial domain: point operation, mask operation, Types of point operation: Brightness modification, contrast adjustment	**64–76**
UNIT 6:	**IMAGE HISTOGRAM** Introduction, histogram equalization, procedure to perform histogram equalization	**77–90**
UNIT 7:	**FILTERING** Introduction, Spatial filtering, linear filtering, low pass filter, limitations of low pass filter, weighted average filter, Bartlett filter, Gaussian filter, Median filter, High pass filtering	**91–107**

UNIT 8: IMAGE ENHANCEMENT IN FREQUENCY DOMAIN **108–120**
Introduction, Low pass filtering in frequency domain, High pass filter in frequency domain, Butterworth high pass filter, Homomorphic filter

UNIT 9: IMAGE RESTORATION **121–147**
Introduction, Image degradation, Types of image Blur, Classification of Image Restoration technique, Image restoration model, Linear Image Restoration techniques, Pseudo Inverse filter, Wiener filter, Non Linear Image Restoration techniques : Iterative method, Maximum likelihood Method, Stochastic image restoration technique

UNIT 10: IMAGE NOISE **148–161**
Introduction, classification of noise image: adaptive noise, Multiplicative noise, impulse noise, Median filtering, trimmed average filter

UNIT 11: IMAGE SEGMENTATION **162–215**
Introduction, Classification of Image segmentation technique, Local and global segmentation, region approach to image segmentation, clustering technique, Image segmentation based on thresholding, Limitation, edge detection techniques, edge linking, Watershed transformation

UNIT 12: IMAGE COMPRESSION **216–229**
Introduction, Need for Image compression, Redundancy in images, Image compression scheme Huffman coding, Arithmetic coding, transform base coding, Vector quantization

UNIT 13: BINARY IMAGE PROCESSING **230–247**
Introduction, Binarisation, Mathematical morphology, structuring element, Morphological image processing, Basic set theory, Logical operations, Standard binary morphological operations :Dialation, Erosion, Opening, Closing, Properties of Morphological operations

UNIT 14: COLOR IMAGE PROCESSING **248–265**
Introduction, Color formation, Human perception of color, Color model: RGB model, CMY color model, HIS color model, YIQ color model, Histogram equalization of color image, Color Image filtering

UNIT 15: WAVELET-BASED IMAGE PROCESSING **266–278**
Introduction, Wavelet, Wavelet transform, 2D continuous wavelet transform, Discrete wave lettrans form, Haar Wavelet, Filter bank, Embedded Image coding

COURSE INTRODUCTION

The field of Image Processing lies at the intersection of computer science, mathematics, and engineering, enabling machines to analyze, enhance, and interpret visual information. This course introduces postgraduate learners to the foundational and advanced concepts of image processing, guiding them through the journey from understanding basic image representations to sophisticated techniques like wavelet-based processing and color image analysis. Covering key areas such as image sampling, transformation, enhancement, restoration, segmentation, and compression, the syllabus is designed to provide a holistic understanding of both theoretical and practical aspects.

Learners will explore a wide range of topics including digital image acquisition, pixel operations, spatial and frequency domain filtering, noise reduction, morphological operations, and image histogram techniques. Advanced concepts such as image restoration using Wiener and pseudo-inverse filters, binary and colour image processing, and wavelet-based techniques for image coding will also be covered. By the end of the course, learners will gain hands-on experience in implementing various algorithms and techniques, preparing them for research or industry roles in fields like computer vision, multimedia systems, and AI-driven image analysis.

While going through the units, you will find that unit is further divided into certain sections and sub-sections, wherever necessary, for your better understanding. Another category that has been included at the end of each section of a particular unit is “CHECK YOUR PROGRESS”. The purpose of this category is to help you to asses for yourself as to how thoroughly you have understood a particular section. You may then match your answers with “ANSWERS TO CHECK YOUR PROGRESS” given at the end of each unit. The section “FURTHER READING” in each unit contains the names of a few books which you might want to consult if you are intereted in learning more elaborately about the concepts discussed in a particular unit. Furthermore, the category called “MODEL QUESTIONS” is intended to give you a hint of the type of questions you are likely to get in the examination.

UNIT 1: FUNDAMENTALS OF IMAGE

UNIT STRUCTURE

1.1 Learning Objectives
1.2 Introduction
1.3 Definition of Image
1.4 Analog Image
1.5 Digital Image
1.6 Advantages and Disadvantages of Digital Image
1.7 Digital Image Representation
1.8 Pixel
1.9 Neighbours of Pixel
1.10 Let Us Sum Up
1.11 Further Reading
1.12 Answers to Check Your Progress
1.13 Possible Questions

1.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- understanding the concept of image and its forms
- difference between analog and digital image.
- role of pixel in formation of digital image
- understanding the idea of pixel resolution in image formation.

1.2 INTRODUCTION

The study of images forms the foundation of many modern technologies and applications. The applications ranges from photography and video to medical imaging and computer vision. Images are visual representations that capture and convey information in a two-dimensional form.

The fundamentals of images will explore key concepts such as the nature and types of images, the differences between analog and digital

images, and the basic principles of image formation. It will also delve into the critical role of pixels, the smallest units of a digital image, and how they influence image resolution and quality. By understanding these foundational elements, one can better appreciate the processes and technologies that enable the creation, manipulation, and interpretation of images in various domains.

1.3 DEFINITION OF IMAGE

An **image** is a two-dimensional representation of a scene, object, or person, captured using devices like cameras, scanners, or generated digitally. Mathematically, an image can be defined as a two-dimensional function f(x,y), where x and y are spatial coordinates, and the amplitude f at any pair of coordinates represents the intensity or grayscale level of the image at that point.

1.4 ANALOG IMAGE

An **analog image** is a continuous image that represents visual information in a continuous manner. The color or brightness varies smoothly over the image space. Examples include traditional film photographs or images on analog television. The intensity and color of every point in an analog image can theoretically have infinite values within a given range.

Characteristics:

I) Continuous in both spatial domain and amplitude.

II) Real-world scenes are inherently analog.

III) Requires conversion (digitization) for digital processing.

1.5 DIGITAL IMAGE

A **digital image** is a discretized version of an analog image. It is represented in a digital form where both spatial and intensity values are quantized. That means:

- The image is divided into a finite number of **pixels (picture** elements).

- Each pixel has a fixed position and a fixed numerical value representing intensity (grayscale) or color.

Mathematically, A digital image is represented as a 2D matrix or array.

Image [i][j] = Pixel intensity at position (i,j)

1.6 ADVANTAGES AND DISADVANTAGES OF DIGITAL IMAGE

Advantages:

I) **Storage & Transmission**: Easy to store, copy, and transmit.

II) **Processing**: Enables complex image processing algorithms.

III) **Compression**: Efficient image compression techniques are available.

IV) **Enhancement**: Images can be enhanced, filtered, and restored.

V) **Integration**: Easily integrated with multimedia and computing systems.

Disadvantages:

I) **Memory Usage**: High-resolution images require significant storage.

II) **Processing Power**: Needs computation power for complex operations.

III) **Quantization Errors**: Loss of detail may occur during digitization.

IV) **Initial Cost**: Digital imaging devices can be expensive.



CHECK YOUR PROGRESS

Q1: What are the advantages of digital images?

.....

.....

Q2: What are the disadvantages of digital images?

.....

.....

1.7 DIGITAL IMAGE REPRESENTATION

A **digital image** is represented as a 2D matrix (array) of **pixels**:

Each pixel stores a value:

I) **Grayscale image**: Single intensity value (e.g., 0–255 for 8-bit).

II) **Color image**: Usually stored in RGB format—three channels (Red, Green, Blue), each with its own intensity.

Example for a grayscale image (3x3 pixels):

[[125, 130, 140],
[122, 128, 135],
[120, 123, 130]]

In memory, the image is stored row-by-row (raster scan order).

1.8 PIXEL (PICTURE ELEMENT)

A **pixel** is the smallest unit of a digital image. Each pixel carries intensity information (grayscale or color). The quality and resolution of the image depend on:

I) **Pixel size**: Smaller pixels give higher resolution.

II) **Bit depth**: More bits per pixel = more intensity/color levels.

Example:

I) 8-bit pixel → 256 intensity levels (0–255).

II) 24-bit RGB → 8 bits per channel, over 16 million color combinations.

1.9 NEIGHBOUR OF A PIXEL

In image processing, a pixel's **neighbours** are the pixels adjacent to it. Types of neighbourhoods:

a) 4-Neighbours (N4):

Pixels directly above, below, left, and right:

P2
P3 P P4
P1

- P2: (x, y–1)
- P3: (x–1, y)
- P4: (x+1, y)
- P1: (x, y+1)

b) Diagonal Neighbours (D):

Pixels that are diagonally adjacent:

P5 P6

P

P7 P8

- P5: (x–1, y–1)
- P6: (x+1, y–1)
- P7: (x–1, y+1)
- P8: (x+1, y+1)

c) 8-Neighbours (N8):

Combines 4-neighbours and diagonal neighbours:

P5 P2 P6

P3 P P4

P7 P1 P8

It is used in operations like edge detection, region growing, filtering, etc.



CHECK YOUR PROGRESS

Q3: What is the thinest unit of a digital image that you can't see untill you zoom in?

.....

.....

Q4: What are the neighbours of a pixel?

.....

.....



1.10 LET US SUM UP

- An image is a visual representation of objects, scenes, or persons, typically two-dimensional.
- Images can be of two main types: analog and digital.

- An analog image is continuous in nature and found in formats like printed photographs or film. In contrast, a digital image is composed of discrete units called pixels, and it represents an image using numeric values stored in a grid format.
- Digital image representation involves storing an image in a computer as a matrix of pixel values, where each pixel carries color or intensity information.
- A pixel (short for picture element) is the smallest unit of a digital image, and it plays a vital role in the clarity and quality of the image.
- The neighbours of a pixel refer to the surrounding pixels. These can be 4-neighbours (up, down, left, right) or 8-neighbours (including diagonals), and they are important in image processing tasks like edge detection and filtering.
- Digital images offer several advantages such as easy editing, sharing, storage, and compatibility with modern devices. However, they also have disadvantages like large file sizes, potential quality loss from compression, and dependence on electronic devices for viewing.



1.11 FURTHER READING

1) Rafael C. Gonzalez & Richard E. Woods. *Digital Image Processing*. Pearson.



1.12 ANSWERS TO CHECK YOUR PROGRESS

Answer to Q No. 1: Easy storage, editing, transmission, and processing; better noise resistance and reproducibility.

Answer to Q No. 2: Requires more storage space, may lose quality due to compression, and needs digitization hardware.

Answer to Q No. 3: A pixel (picture element) is the smallest unit of a digital image that holds intensity or color information

Answer to Q No. 4: Neighbours of a pixel are adjacent pixels, which can be 4-neighbours (up, down, left, right) or 8-neighbours (including diagonals).



1.13 MODEL QUESTIONS

Q1: Define an image. Explain the differences between analog and digital images with suitable examples.

Q2: Discuss in detail the advantages and disadvantages of digital images compared to analog images.

Q3: What is meant by digital image representation? Explain how an image is represented in a computer using pixels.

Q4: What is a pixel? How does pixel size and resolution affect the quality of a digital image?

Q5: Define pixel neighbours. Describe 4-neighbours, diagonal neighbours, and 8-neighbours with a diagram.

*****

UNIT 2: IMAGE SAMPLING

UNIT STRUCTURE

2.1 Learning Objectives

2.2 Introduction

2.3 Theory of 2D sampling

- 2.3.1 Continuous vs. Discrete Images
- 2.3.2 Sampling in Two Dimensions
- 2.3.3 Sampling Frequency and Sampling Interval
- 2.3.4 Nyquist Sampling Theorem (for 2D Signals)
- 2.3.5 Band-limited Images
- 2.3.6 Aliasing and Violation of Sampling Criterion

2.4 Violation of Sampling Criterion

2.5 Quantization

2.6 Resolution

2.7 Anatomy of Human Visual System

2.8 Let Us Sum Up

2.9 Further Reading

2.10 Answers to Check Your Progress

2.11 Possible Questions

2.1 LEARNING OBJECTIVES

After going through this unit, you will be able to have knowledge about:

- understanding the concept of image sampling in two dimensions
- define and explain the sampling criterion in image processing
- understand the process of quantization in digital image representation
- describe the structure and function of the human eye relevant to image perception.

2.2 INTRODUCTION

Digital image processing begins with the crucial step of converting real-world visual information into a digital format that computers can

understand and manipulate. This unit introduces the foundational concepts required to understand this transformation, starting with the theory of 2D sampling.

2D sampling is the process of converting a continuous image into discrete data points by capturing its intensity values at regular spatial intervals. This step is vital for digital representation and lays the groundwork for further image processing tasks. However, improper sampling—especially at a rate lower than required—can lead to a violation of the sampling criterion, causing unwanted distortions known as aliasing. Understanding how to prevent aliasing is key to maintaining image integrity.

Once an image is sampled, it must undergo quantization, where continuous intensity values are mapped into a finite number of gray or colour levels. This step enables image storage and display using binary data, and also introduces quantization errors if not carefully managed.

2.3 THEORY OF 2D SAMPLING

In digital image processing, the first step in converting a real-world continuous image into a digital form is sampling. Sampling is the process of selecting discrete points from a continuous image to represent it in a form that a computer can process.

While one-dimensional (1D) sampling is familiar in signals like audio, images require two-dimensional (2D) sampling because they have both horizontal (x-axis) and vertical (y-axis) spatial dimensions.

2.3.1 Continuous vs. Discrete Images

1. Continuous image : A continuous image can be represented as a function f(x,y), where x and y represent continuous spatial coordinates, and f(x,y) gives the intensity or color at that point. Such images have infinite resolution and contain information at every point.
2. Discrete Image : A discrete image is a sampled version of the continuous image where intensity values are available only at a

finite number of points arranged in a grid. These points are called pixels (picture elements).

2.3.2 Sampling in Two Dimensions

Sampling involves measuring the continuous image f(x,y) at discrete locations spaced by intervals Δx and Δy along the x and y directions respectively.

- The sampled image can be written as:
  $f_s(m,n)=f(m\Delta x,n\Delta y)$

where m,n are integers indicating the pixel positions in the sampled grid.

- Each sampled value $f_s(m,n)$ corresponds to the intensity of one pixel in the digital image.

2.3.3 Sampling Frequency and Sampling Interval

- Sampling Interval (Δx,Δy) is the physical distance between adjacent samples (pixels). Smaller intervals mean more samples and higher resolution.
- Sampling Frequency (f_x,f_y) is the reciprocal of the sampling interval:
  $f_x=1/\Delta x, f_y=1/\Delta y$
  It represents the number of samples per unit distance.

2.3.4 Nyquist Sampling Theorem (for 2D Signals)

The Nyquist theorem states:

To avoid loss of information, the sampling frequency must be at least twice the highest frequency present in the signal.

For 2D images, this means the sampling frequencies f_x and f_y must satisfy:

$f_x \geq 2f_{max,x}, fy \geq f_{max,y}$

where $f_{max,x}$ and $f_{max,y}$ are the highest spatial frequencies in the x and y directions, respectively.

2.3.5 Band-limited Images

1. An image is said to be band-limited if it contains no frequency components above a certain maximum frequency f_{max} .
2. Most natural images are approximately band-limited because very fine detail (very high frequency) is limited by the physical properties of the scene and the imaging system.
3. Band-limiting is important because it determines the minimum sampling frequency required to avoid loss of detail.

2.3.6 Aliasing and Violation of Sampling Criterion

1. Aliasing occurs when the sampling frequency is less than twice the maximum frequency in the image (i.e., Nyquist criterion is violated).
2. Aliasing causes overlapping of frequency components in the sampled image spectrum, resulting in distortion.
3. Visually, aliasing appears as jagged edges, moiré patterns, or false textures.
4. To prevent aliasing, images are often pre-filtered (low-pass filtered) to remove high-frequency components before sampling. This is called anti-aliasing filtering.

2.3.7 Summary

Aspect	**Explanation**
Sampling	Converting continuous image to discrete pixel values.
Sampling Interval	Distance between samples, inversely related to sampling frequency.
Sampling Frequency	Number of samples per unit distance.
Nyquist Theorem	Sampling frequency must be at least twice the max frequency to avoid aliasing.
Aliasing	Distortion caused by under-sampling.

Band-limited image	Image with frequency content limited below a max value.
Anti-aliasing	Filtering to remove high frequencies before sampling.



CHECK YOUR PROGRESS

Q1: What process captures value from a flat image at regular interval in both direction?

.....

.....

Q2: What is a discrete image? What kind of image is made up it separate, countable values?

.....

.....

Q3: What is sampling frequency?

.....

.....

Q4: What is the Nyquist Sampling Theorem in 2D?

.....

.....

Q5: How can aliasing be prevented?

.....

.....

2.4 VIOLATION OF SAMPLING CRITERION

In digital image processing, sampling is the process of capturing a continuous image at discrete intervals to convert it into digital form. The sampling criterion ensures that this conversion preserves all the necessary information in the original image. When this criterion is not met, it leads to distortions and errors in the digital image.

The sampling criterion is based on the Nyquist Sampling Theorem, which states:"To accurately reconstruct a continuous image from its samples, the sampling frequency must be at least twice the highest frequency present in the image."This minimum required frequency is known as the Nyquist rate.

A **violation** occurs when the image is sampled at a frequency lower than the Nyquist rate. This results in aliasing, where different signal frequencies become indistinguishable from each other in the sampled image.

Effects of Violation (Aliasing)

When the sampling criterion is violated:

I) High-frequency details in the image are lost.

II) False patterns (alias patterns) may appear.

III) Edges and textures may look distorted or jagged.

IV) Moiré patterns or unusual wavy patterns may appear.

Causes of Violation:

I) Under sampling: Sampling at too low a rate.

II) High-frequency content in the image that exceeds the sampling capacity.

III) Lack of pre-processing (anti-aliasing filters) before sampling.

Preventing Violation :

To avoid violating the sampling criterion the following must be done:

I) Use sufficient sampling frequency (at least twice the maximum frequency in the image).

II) Apply anti-aliasing filters (low-pass filters) to remove high-frequency components before sampling.

III) Analyze the frequency content of the image to determine the appropriate sampling rate.



CHECK YOUR PROGRESS

Q6: What is the sampling criterion in image processing?

.....

.....

Q7: What happens when the sampling criterion is violated?

.....

.....

Q8: What causes the violation of the sampling criterion?

.....

.....

Q9: What is an anti-aliasing filter?

.....

.....

2.5 QUANTIZATION

Quantization is a fundamental step in converting an analog or continuous image into a digital image. After sampling an image to obtain discrete spatial points (pixels), **quantization** assigns each sampled pixel intensity a finite set of values, typically represented by integers. This step converts the continuous range of intensity values into discrete levels, enabling the image data to be stored and processed digitally.The image is divided into a finite number of **pixels** (picture elements).

Quantization is the process of mapping a large (often infinite) set of input values to a smaller set of output values.In image processing, it refers to converting the continuous amplitude (intensity) values of sampled pixels into a finite number of discrete levels.

Quantization is required due to the following :

I) Computers store and manipulate data in binary form.

II) Continuous intensity values must be approximated by discrete levels so they can be represented with a finite number of bits.

III) Quantization reduces data size and makes digital processing feasible.

2.5.1 Some Key Aspects

A. Quantization levels: The number of discrete intensity levels depends on the **bit depth** (number of bits used per pixel). For example, an 8-bit image can represent 2^8=256 intensity levels (0 to 255). More bits mean finer intensity resolution and better image quality.

B. Quantization Error: Since continuous values are approximated by discrete levels, there is an inevitable error called **quantization error** or **quantization noise.** This error is the difference between the actual continuous value and the quantized value. Larger quantization intervals lead to more noticeable errors, such as banding in images.

C. Uniform vs Non-Uniform Quantization:

I) Uniform Quantization: Equal-sized intervals for all quantization levels. Simpler but may not always match human perception.

II) Non-Uniform Quantization: Intervals vary in size, often designed to match the sensitivity of the human visual system, providing better perceived image quality at lower bit depths.

D. Impact on Image Quality

I) Higher quantization levels preserve more detail and subtle intensity differences.

II) Lower quantization levels reduce data size but may cause visible artifacts like contouring or banding.

III) Choosing an appropriate quantization level balances image quality and storage requirements.



CHECK YOUR PROGRESS

Q10: Why is quantization necessary?

.....

.....

Q11: What determines the number of quantization levels?

.....

.....

Q12: How many levels can an 8-bit image represent?

.....

.....

Q13: What is quantization error?

.....

.....

2.6 RESOLUTION

Resolution refers to the amount of detail that an image holds. It is a critical factor in both capturing and displaying images, affecting how sharp, clear, and detailed an image appears.

2.6.1 Some Key Aspects

A. Types of Resolution

In digital imaging, there are mainly two types of resolution:

1. Spatial Resolution
2. Intensity (or Gray-level) Resolution

Spatial Resolution: Spatial resolution defines the number of pixels used to represent an image. It indicates how finely the image is sampled in space (both horizontally and vertically).Higher spatial resolution means more pixels per unit area, resulting in finer detail and sharper images. Example: An image of 1024 x 768 pixels has higher spatial resolution than an image of 640 x 480 pixels.It is measured in pixels per inch (PPI) or dots per inch (DPI) in printing.

Intensity Resolution: Intensity resolution refers to the number of distinct gray levels or colors that each pixel can represent. It is determined by the bit depth of the image.

Example: An 8-bit gray scale image can represent 256 different intensity levels. While a 24-bit color image can represent over 16 million colors.

B. Relationship Between Resolution and Image Quality: Higher spatial resolution improves the sharpness and detail of the image but increases file size. Higher intensity resolution improves the quality of color or gray scale representation. Both types of resolution contribute to overall image quality.

C. Resolution in Different Contexts:

I) Display Resolution: The number of pixels a display device can show.

II) Image Resolution: The pixel dimensions of an image file.

III) Print Resolution: The number of ink dots per inch (DPI) when printing.

D. Importance of Resolution

I) Determines the clarity and usability of images in applications like medical imaging, satellite imaging, digital photography, and more.

II) Insufficient resolution can lead to loss of detail and poor image interpretation.



CHECK YOUR PROGRESS

Q14: What are the two main types of resolution?

.....

.....

Q15: What effect does higher spatial resolution have on an image?

.....

.....

Q16: What effect does higher intensity resolution have?

.....

.....

2.7 ANATOMY OF HUMAN VISUAL SYSTEM

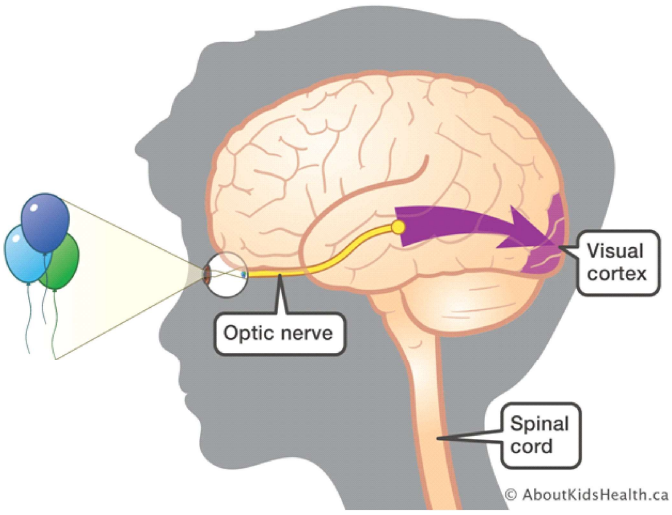


Fig 1: The human visual anatomy

[Source : https://www.aboutkidshealth.ca/optic-neuritis]

The human visual system is a complex biological system that allows us to perceive and interpret the visual world. Understanding its anatomy helps in designing image processing techniques that align with human perception. The key parts of the human visual system include:

I) Eye

II) Optic Nerve

III) Visual Cortex

The Eye : The eye functions like a camera, capturing light and converting it into neural signals.

I) Cornea and Lens: Focus light rays onto the retina.

II) Retina: The light-sensitive layer at the back of the eye. It contains photoreceptor cells that detect light.

Photoreceptors: Two main types.

I) Rods:

a. Responsible for vision in low light (night vision).

b. Do not detect color.

c. Highly sensitive to light.

II) Cones:

a. Responsible for color vision and detail in bright light (daylight).

b. Three types, sensitive to red, green, and blue wavelengths (trichromatic vision).

Retina to Brain: Photoreceptors convert light into electrical signals.Signals travel through the **optic nerve** to the **visual cortex** in the brain.The visual cortex processes these signals to form images and recognize patterns.

Visual Acuity and Perception: Visual acuity refers to the eye's ability to distinguish fine details. The human eye is more sensitive to luminance (brightness) variations than color.The eye's sensitivity varies with spatial frequency — it can detect certain sizes of patterns better than others.

Importance in Image Processing

I) Understanding human visual anatomy helps optimize image compression, enhancement, and display.

II) For example, image compression algorithms often reduce color information more than brightness because the eye is less sensitive to color changes.

Summary Table

Component	**Function**
Cornea & Lens	Focus light onto the retina
Retina	Detects light via rods and cones
Rods	Night vision, no color sensitivity
Cones	Color vision (red, green, blue), daylight
Optic Nerve	Transmits signals from retina to brain
Visual Cortex	Processes visual signals into images



2.8 LET US SUM UP

- In digital image processing, 2D sampling involves converting a continuous image into a discrete one by taking samples at regular intervals in both horizontal and vertical directions.

- According to the Nyquist sampling criterion, an image must be sampled at a rate at least twice the highest frequency present.
- If this criterion is violated (under-sampling), it causes aliasing, where different signals become indistinguishable, resulting in distortion and loss of image detail.
- After sampling, quantization is the process of mapping continuous amplitude values to discrete levels.
- More quantization levels (higher bit depth) lead to better image quality.
- Resolution refers to the amount of detail in an image.
- The human visual system includes the eye, optic nerve, and visual cortex.
- The eye captures light, and the retina contains rods (low-light vision) and cones (color vision).
- Understanding how the human eye perceives brightness and color helps in designing image processing systems aligned with human perception.



2.9 FURTHER READING

1) Rafael C. Gonzalez & Richard E. Woods. *Digital Image Processing*. Pearson.



2.10 ANSWERS TO CHECK YOUR PROGRESS

Answer to Q No. 1: 2D sampling is the process of converting a continuous image into a discrete image by measuring intensity values at regular spatial intervals in two dimensions.

Answer to Q No. 2: A discrete image has intensity values only at specific points arranged in a grid, representing pixels.

Answer to Q No. 3: It is the number of samples taken per unit distance, calculated as $f_x=1/\Delta x$ and $f_y=1/\Delta y$.

Answer to Q No. 4: It states that the sampling frequency must be at least twice the highest frequency in the image to avoid aliasing.

Answer to Q No. 5: By sampling at or above the Nyquist rate and applying a low-pass (anti-aliasing) filter before sampling.

Answer to Q No. 6: It states that the sampling frequency must be at least twice the highest frequency in the image (Nyquist rate).

Answer to Q No. 7: Aliasing occurs, leading to distortion and false patterns in the image.

Answer to Q No. 8: Sampling at a frequency lower than the Nyquist rate or not filtering out high frequencies before sampling.

Answer to Q No.9: A low-pass filter that removes high-frequency components before sampling to prevent aliasing.

Answer to Q No.10: Because computers store data in discrete binary form, so continuous values must be approximated to finite levels.

Answer to Q No. 11: The bit depth, or number of bits used per pixel.

Answer to Q No. 12: 256 levels (from 0 to 255).

Answer to Q No. 13: The difference between the actual continuous value and the quantized discrete value.

Answer to Q No. 14: Spatial resolution and intensity (gray-level) resolution.

Answer to Q No.15: It increases image sharpness and detail.

Answer to Q No.16: It improves the smoothness of color or brightness gradations.



2.11 MODEL QUESTIONS

Q1: Explain the theory of 2D sampling in digital image processing. Describe how an analog image is converted into a digital image through sampling. Discuss the role of sampling frequency and its importance.

Q2: Explain what happens when this criterion is violated in image processing. Discuss the phenomenon of aliasing with suitable examples and how it affects image quality.

Q3: Describe the difference between uniform and non-uniform quantization. Discuss the effect of quantization on image quality and explain quantization error with examples.

Q4: Differentiate between spatial resolution and intensity (gray-level) resolution. How does resolution affect image quality and storage requirements? Give examples to support your answer.

Q5: Describe the role of the retina, rods, and cones in visual perception. How does understanding human vision help in the design of effective image processing algorithms?

*****

UNIT 3: FUNDAMENTALS OF IMAGE

UNIT STRUCTURE

3.1 Learning Objectives

3.2 Introduction

3.3 Classification of Digital Image

- 3.3.1 Raster Image
- 3.3.2 Vector Image

3.4 Image Types

- 3.4.1 Binary Image
- 3.4.2 Grayscale Image
- 3.4.3 Color Image
- 3.4.4 Multispectral Image
- 3.4.5 Hyper Spectral Image

3.5 Let Us Sum Up

3.6 Further Reading

3.7 Answers to Check Your Progress

3.8 Model Questions

3.1 LEARNING OBJECTIVES

After going through this unit, you will be able to have knowledge about:

- define digital images and identify their basic components such as pixels and resolution
- classify digital images into raster and vector types, and explain their differences
- describe various image types including binary, grayscale, color, multispectral, and hyperspectral images
- analyze the suitability of different image formats for applications in design, medical imaging, and remote sensing
- compare the advantages and limitations of raster and vector images in practical use cases

3.2 INTRODUCTION

In the previous unit, we learnt about sampling of images where a continuous image (analog) is converted into a discrete representation, usually for the purpose of digital processing or analysis. The process involved selecting a finite number of points from the continuous image to create a pixel-based representation. Image sampling is a critical step in digital imaging that affects the quality and fidelity of the resultant image. In this unit, we will learn about what digital images are and what the different types of digital images are.

Digital images are representations of visual information in a format that can be processed, stored, and transmitted by computers and electronic devices. Unlike traditional photographs, which are captured on film, digital images are composed of pixels (picture elements) arranged in a grid, where each pixel represents a specific color or shade. The resolution of a digital image, often measured in pixels per inch (PPI) or dots per inch (DPI), determines its detail and clarity. Digital images come in various formats, each serving different purposes and offering distinct features.

3.3 CLASSIFICATION OF DIGTAL IMAGE

A digital image represents a two-dimensional picture as a finite collection of digital values which are commonly referred to as picture elements or pixels. Each pixel in an image possesses a specific value that dictates its color and brightness. These pixels together create the entire image. Digital images underpin a wide range of technologies and applications across various fields, including photography, medical imaging, remote sensing, and computer vision.

Digital images can be classified based on several criteria, each emphasizing different aspects of the images. Based on the image format, they can be classified in to raster and vector images. Understanding the classification of digital images is crucial for selecting appropriate technologies and techniques for processing, storing, and analyzing them. Each type of image serves specific needs across various industries and

applications, enabling advances in fields such as medicine, entertainment, and technology. Digital cameras capture images by converting light into electrical signals through a sensor, such as a CCD (charge-coupled device) or CMOS (complementary metal-oxide-semiconductor). These signals are subsequently processed and saved as digital image files. Similarly, scanners digitize printed images by moving them over a light source and capturing the reflected light with a sensor.

Digital images are essential to modern technology, finding applications across various fields. Grasping the technical aspects, categories, creation methods, and uses of digital images offers a thorough understanding of their significance and adaptability. As technology progresses, the role of digital images will continue to grow, unveiling new opportunities for innovation and creativity.

3.3.1 Raster Images

Raster images are made up of tiny dots of colour called pixels. These images consist of rectangular arrays of pixels. A pixel is the smallest individual unit of a picture displayed on a screen. Raster images, which are the most commonly used format for storing and processing images, are particularly well-suited for detailed and vibrant visuals like photographs. When you think of images, raster (or bitmap) images are the ones that typically what comes to mind. They are created through scanning or photographing objects. These images are made up of pixels—tiny dots that contain specific color and tonal information—coming together to form the complete picture.

Because raster images are based on pixels, they are dependent on resolution. The quality of an image is affected by both the number of pixels it contains and the pixel density (the number of pixels displayed per inch). As you might expect, a higher number of pixels and greater resolution result in superior image quality. Like many other graphical elements, raster images can be copied, moved, or cropped. You have the ability to modify an image using grip modes, adjust its contrast, crop it into a rectangle or polygon,

or utilize an image as an edge for trimming. A raster graphics image maps bits directly to a display area, commonly known as a bitmap. Image types can be 8-bit grayscale, 8-bit color, or 24-bit color. However, images with a 16-bit color depth are not supported. One of the primary benefits of raster graphics is their ability to create detailed and realistic images, such as photographs, paintings, textures, and visual effects. Raster graphics effectively capture the nuances and variations of color, light, and shadow that vector graphics may struggle with. Additionally, raster graphics are widely supported by various image formats and software, making them easy to view and edit across multiple tools and platforms. For instance, most digital cameras and scanners generate raster graphics, and the majority of image editing software is compatible with them.

3.3.2 Vector Image

Vector images are graphics created using mathematical equations to define shapes, lines, and colors. Unlike raster images, which are composed of a grid of pixels and can lose quality when resized, vector images can be scaled infinitely without losing clarity or detail. This makes them ideal for logos, icons, and illustrations that require versatility in size and resolution. Common vector file formats include SVG, AI, and EPS. Vector graphics are widely used in graphic design, print media, and digital applications due to their flexibility and scalability. Vector images consist of paths or line art that are infinitely scalable because they are defined by algorithms instead of pixels. One of the key advantages of vector images is their ability to be resized both larger and smaller without losing clarity, while also maintaining a consistent file size. Most computer font files are based on vector representations of letters, allowing them to be scaled up or down while remaining sharp. All clipart in Microsoft Office is vector-based, as are many charts and graphs produced by Office and statistical analysis software. Typically, vector art is

created using illustration software like Adobe Illustrator or Corel DRAW. Vector illustrations are particularly well-suited for logos, artwork, animations, and text.

Raster graphics consist of a fixed number of pixels, which limits their scalability compared to vector graphics. When a raster image is enlarged beyond a certain point, its edges can become jagged and appear pixelated, as the individual pixels become visible. As a result, raster graphics cannot be enlarged without compromising image quality.

One of the key benefits of vector graphics is their resolution independence, allowing them to be resized both larger and smaller without impacting their quality or visual integrity. Another benefit of vector graphics is that they typically have smaller file sizes compared to raster graphics, enabling faster loading times and more efficient storage usage. Vector images are easier to edit. Users can quickly alter vector relationships to change colors or modify line shapes. Vector files are easily portable and can be loaded onto various devices and applications without difficulty.



CHECK YOUR POGRESS 1

Q1: Fill In the blanks

1. Digital images can be _____, stored, and _____ by computers and electronic devices.
2. The resolution of a digital image is measured in _____ or _____.
3. Based on the format, the images can be classified in to _____ and _____ images
4. A _____ is the smallest individual unit of a picture displayed on a screen.
5. The quality of an image is affected by both the _____ it contains and the _____.

6. A key benefit of vector graphics is their_____, allowing them to be resized both larger and smaller without impacting their quality or visual integrity.
7. Vector graphics are widely used in graphic _____, _____, and _____.

3.4 IMAGE TYPES

In the realm of digital imaging, various image types exist to cater to different needs, applications, and qualities. Understanding these image types is essential for anyone involved in photography, graphic design, web development, or digital media. Below are some of the most common image formats along with their characteristics. Choosing the right image type depends on the intended use, desired quality, and file size considerations. Understanding the strengths and weaknesses of each format can help to make informed decisions in the digital projects. Whether for web graphics, professional printing, or digital art, the right image type can enhance the effectiveness and quality of the work.

3.4.1 Binary Image

A binary image is a type of digital image that consists of only two colors, typically represented as black and white. Each pixel in a binary image can take on one of two values, usually encoded as 0 and 1, where,

- 0 represents one color (often black)
- 1 represents the other color (often white)

They are often created from grayscale images through a process called thresholding. In this process, a threshold value is selected, and pixels in the grayscale image are converted to black or white based on whether their intensity values are below or above the threshold. An example of a binary image is a simple silhouette, where the object of interest is in black against a white background. In contrast, a complex scene with multiple colors and shades would

require a grayscale or color image to capture the necessary details.

The characteristics of binary image are as follows:

1. Pixel representation

In a binary image, each pixel can be represented by a single bit of information. This allows for very efficient storage and processing, as each pixel requires only one bit compared to multiple bits for grayscale or color images.

2. Image size

Binary images are typically smaller in file size compared to grayscale or color images due to the reduced amount of data per pixel. This makes them suitable for applications where storage space is a concern.

There are different applications of binary images:

- ✓ Document scanning: Binary images are often used in scanned documents and OCR (Optical Character Recognition) systems, where text is represented clearly against a white background.
- ✓ Image processing: They are used in image processing tasks such as edge detection, object recognition, and segmentation, as the simplicity of binary images makes it easier to analyze shapes.
- ✓ Graphics and icons: Simple graphics, icons, and certain types of logos are often created as binary images for clarity and scalability.

While binary images are efficient for certain applications, they lack the detail and nuance of grayscale or color images. This limitation makes them unsuitable for applications requiring detailed visual information, such as photographs.

3.4.2 Grayscale Image

A grayscale is a type of digital image that contains varying shades of gray, ranging from black to white. Unlike binary images that consist of only two colors, grayscale images can represent a wide spectrum of intensities, allowing for more detailed and nuanced visual

information. Such images require less storage space than color images since they only contain intensity information, not color data. The RGB can be converted to grayscale using various methods, such as averaging the red, green, and blue channels or applying specific weights to each channel based on human perception. The characteristics of grayscale image are as follows:

1. **Pixel representation**

In a grayscale image, each pixel is represented by a single value that corresponds to a shade of gray. Commonly, this value ranges from 0 to 255 in an 8-bit image, where

- 0 represents black
- 255 represents white
- Values in between represent different shades of gray.

2. **Bit depth**

Grayscale images can have different bit depths, affecting the number of possible gray levels.

- 8-bit grayscale: 256 levels (0-255)
- 16-bit grayscale: 65,536 levels (0-65,535), allowing for finer gradations and more details
- Higher bit depths (e.g., 32-bit) can also be used for specialized applications.

3. **Lack of color information**

Grayscale images do not contain color information; they solely represent intensity. This can simplify the analysis and processing of images since there is only one channel to consider.

4. **Contrast and detail**

They can effectively capture and represent contrast and detail, making them suitable for applications like photography, medical imaging (X-rays, MRIs, and CT scans etc), and computer vision (edge detection, segmentation). The absence of color allows viewers to focus on texture, patterns, and shapes.

5. **Storage and processing efficiency**

These images generally require less storage space compared to color images, as they contain less data. The grayscale images provide more detail than binary images, however they lack the richness and depth of color images, which can convey additional information. A classic example of a grayscale image is a black-and-white photograph, where the various shades of gray emphasize contrast and texture, allowing viewers to appreciate the composition without the influence of color. Grayscale images are a versatile and widely used type of digital image that effectively capture variations in light intensity. They are essential in numerous fields, including photography, medical imaging, and computer vision, providing a balance between detail and simplicity. Understanding grayscale images is fundamental for anyone working in digital media or image processing.

3.4.3 Color image

Color images are a fundamental aspect of visual media, representing images in a way that resembles how humans perceive colors. Unlike grayscale images, which contain only shades of gray, color images are composed of multiple colors that contribute to a richer representation of a scene or object. Color images are typically made up of pixels, which are the smallest units of the image. Each pixel contains color information that is represented in various ways, depending on the color model used. The most common representation comprises three primary colors: Red (R), Green (G) and Blue (B). Color images play a crucial role in the way we perceive and interpret visual information. They enrich our experiences across various media, from digital photography to film and graphic design. Understanding the fundamentals of color representation, models, and applications is essential for professionals in fields related to design, art, technology, and science. With technology advancement,

the methods and techniques for creating, processing, and utilizing color images continue to evolve, offering new possibilities for enhancing visual communication.

Color images can be represented in several color models, each suited for different purposes and applications:

1. **RGB Color Model:**

   - The RGB model is the most widely used color model for digital displays.

   - It is an additive color model where different intensities of red, green, and blue light are combined to produce a broad spectrum of colors.

   - Each color channel (R, G, B) usually contains values ranging from 0 to 255 in an 8-bit representation, allowing for over 16 million possible colors.

2. **CMY and CMYK Color Model:**

   - The CMY (Cyan, Magenta, Yellow) model is a subtractive color model primarily used in color printing.

   - CMYK (Cyan, Magenta, Yellow, Key/Black) extends the CMY model by adding a black component, which enhances depth and detail in printed images.

   - In this model, colors are created by subtracting varying percentages of the primary colors from white, the natural background color of paper.

3. **HSV Color Model:**

   - The HSV model stands for Hue, Saturation, and Value. It represents colors in a cylindrical coordinate system.

   - Hue refers to the color type (e.g., red, blue, green).

   - Saturation indicates the intensity or purity of the color.

   - Value (or Brightness) represents the brightness of the color.

   - This model is often favored in image processing and editing because it aligns more closely with human perception of colors.

Color depth refers to the amount of color information stored in each pixel. Common color depths include:

- 8-bit color: Typically allows 256 colors (often used in GIF images).
- 16-bit color: Allows for 65,536 colors, providing greater detail.
- 24-bit color (True Color): Each of the RGB channels has 8 bits, allowing for over 16 million colors.
- 32-bit color: Extends 24-bit color with an additional 8 bits for alpha (transparency) which allows for blending pixels with different levels of transparency.

Applications of Color Images

Color images are used in various fields, including:

1. Digital Photography: Capturing scenes in full color, enhancing aesthetics and visual impact.
2. Television and Film: Color grading and editing to convey moods or themes.
3. Computer Graphics: Used extensively in video games, simulations, and animations.
4. Medical Imaging: Color images in medical fields assist in analyzing scans and X-rays for better diagnosis.
5. Remote Sensing: In satellite imagery, color images help in land use classification and environmental monitoring.

3.4.4 Multispectral image

Multispectral imaging is a powerful remote sensing technique that captures image data at specific frequency ranges across the electromagnetic spectrum. Unlike traditional cameras that operate in the visible range (typically red, green, and blue channels), multispectral sensors can capture information from multiple bands of light, allowing for a more comprehensive analysis of objects and landscapes. This technology has numerous applications in various fields, including agriculture, environmental monitoring, and resource management.

Key Features of Multispectral Imaging:

1. **Spectral Bands:**

- Multispectral images are captured using sensors that can record data across various wavelengths. These bands often include visible light as well as infrared and sometimes ultraviolet, depending on the application.

- **Common bands in multispectral imaging include:**

- Visible bands: Red (approximately 620-750 nm), Green (approximately 495-570 nm), and Blue (approximately 450-495 nm).
- Near Infrared (NIR): Generally ranges from about 750 nm to 1400 nm; useful for vegetation health analysis.
- Shortwave Infrared (SWIR): Ranges from about 1400 nm to 3000 nm; useful for mineral and soil composition studies.

2. **Data Cubes:**

- Multispectral images are often represented as 3D data cubes, where two dimensions correspond to pixel locations (spatial dimensions) and the third dimension corresponds to spectral information (wavelengths captured).

3. **Calibration:**

- Multispectral sensors must be carefully calibrated to ensure that the data collected is accurate and comparable across different imaging sessions or instruments.
- Radiometric calibration is important to account for variations in lighting conditions and atmospheric effects.

Applications of Multispectral Imaging

1. **Agriculture:**

- Crop Monitoring: Farmers use multispectral imaging to assess crop health, determine water stress, and optimize fertilization through techniques like Normalized Difference Vegetation Index (NDVI), which analyzes the difference between near-infrared and red reflectance.

- Precision Agriculture: By combining multispectral data with geographic information systems (GIS), farmers can apply resources more efficiently, minimizing costs and maximizing yields.

2. **Environmental Monitoring:**

- Land Use and Land Cover Mapping: Multispectral images help monitor changes in land use, deforestation, urban expansion, and habitat alterations.
- Water Quality Assessment: Sensors can detect pollutants and other water quality parameters by analyzing spectral signatures in water bodies.

3. **Forestry:**

- Forest health assessments can be conducted using multispectral imaging to analyze canopy conditions, biodiversity, and biomass estimation.

4. **Mining and Geology:**

- Multispectral imaging can detect minerals and geological formations, helping in mineral exploration and assessing the impacts of mining activities.

5. **Disaster Management:**

- Multispectral data can aid in monitoring the effects of natural disasters, such as floods, wildfires, and landslides, providing critical information for emergency response and recovery planning.

6. **Archaeology:**

- By utilizing multispectral imaging, archaeologists can identify and record features in landscapes that are not visible in standard photographs, such as ancient structures or buried artifacts.

Advantages of Multispectral Imaging

- Detailed Analysis: The ability to capture data across multiple wavelengths enhances the ability to analyze scenes beyond what is visible to the naked eye.
- Non-Destructive: Multispectral imaging can be applied to monitor areas without physical interference or damage to the subject.

- Remote Sensing Capabilities: With the use of satellites and drones, multispectral images can cover large areas quickly and efficiently.

Challenges

- Data Complexity: The amount of data generated can be substantial, requiring advanced processing and analytical techniques.
- Atmospheric Interference: Factors like clouds, humidity, and atmospheric particles can affect the quality of multispectral images and may require correction techniques.
- Cost: High-quality multispectral sensors and the necessary data analysis tools can be expensive, which may limit access for some users.

Multispectral imaging represents a significant advancement in the field of remote sensing, providing detailed and diverse data that is critical for various scientific and practical applications. By capturing information across a wide range of wavelengths, multispectral imagery enables professionals to gain insights into agricultural health, environmental changes, geological features, and more. As technology continues to evolve, the capabilities and accessibility of multispectral imaging are likely to expand, opening new frontiers for research and application in multiple disciplines.

3.4.5 Hyper spectral image

Hyper spectral imaging is an advanced technique used to capture and analyze images across a broad range of wavelengths of the electromagnetic spectrum. Unlike traditional imaging methods, which typically capture data in three (RGB) or a few discrete bands (multispectral), hyper spectral imaging collects detailed information from hundreds of contiguous spectral bands. This allows for a more comprehensive analysis of materials and surfaces, facilitating applications in various fields such as agriculture, environmental monitoring, mineralogy, and medical diagnostics.

Key Features of Hyper spectral Imaging

1. **Spectral Resolution:**

- Hyper spectral images are characterized by their high spectral resolution, typically capturing data across 100 to 300 spectral bands. Each band corresponds to a narrow wavelength range, allowing for finer distinctions between materials based on their spectral signatures.

2. **Data Cubes:**

- Hyper spectral images are structured as 3D data cubes, where two dimensions represent spatial coordinates (x and y), and the third dimension corresponds to spectral information (wavelengths). This 3D representation allows for sophisticated analysis and classification of materials based on their spectral characteristics.

3. **Spectral Signatures:**

- Each material has unique reflectance properties across various wavelengths, forming a spectral signature. By comparing these signatures in hyper spectral data, users can identify and differentiate between different materials.

4. **Calibration:**

- Like multispectral imaging, hyper spectral sensors require careful calibration to ensure accurate and reliable data collection. Radiometric calibration compensates for atmospheric effects, sensor noise, and other variables affecting the spectral readings.

Applications of Hyper spectral Imaging

1. **Agriculture:**

- Crop Monitoring: Hyper spectral imaging can analyze plant health, nutrient levels, and water stress at a much finer resolution compared to multispectral methods. This enables more precise interventions.
- Soil Analysis: The spectral information can also help determine soil composition and moisture levels.

2. **Environmental Monitoring:**

- Water Quality Assessment: Hyper spectral imaging can detect contaminants, algae blooms, and sediment in water bodies through their unique spectral signatures.
- Land Use and Habitat Mapping: It aids in mapping habitats and understanding biodiversity by providing detailed information about vegetation and land cover types.

3. **Mineralogy and Geology:**

- Mineral Identification: Hyper spectral imaging is widely used in mining and geological surveys to identify different minerals based on their spectral signatures, which helps in exploration and resource management.
- Soil and Rock Classification: It can be used to classify soil and rock types based on their surface reflectance properties.

4. **Food Quality and Safety:**

- Quality Assessment: In the food industry, hyper spectral imaging can assess the quality and ripeness of fruits and vegetables and detect contaminants or spoilage.
- Safety Monitoring: It can be utilized to identify foreign materials or pathogens in food products.

5. **Medical Diagnostics:**

- Tissue Analysis: Hyper spectral imaging can evaluate biological tissues to differentiate between healthy and diseased tissues in non-invasive medical diagnostics.
- Cancer Detection: It has potential applications in identifying cancerous tissues based on their unique spectral properties.

6. **Defense and Security:**

- Hyper spectral imaging is employed in surveillance and reconnaissance for material identification, environmental analysis, and signature recognition.

Advantages of Hyper spectral Imaging

- Detailed Spectral Information: The fine spectral resolution allows for precise differentiation of materials that may appear similar in multiband images.
- Comprehensive Analysis: Hyper spectral data provide a wealth of information that can be analyzed for various applications, from identifying minerals to assessing plant health.
- Non-Destructive: The technique allows for analysis without damaging the subject, making it suitable for sensitive applications.

Challenges

- Data Volume and Processing: The considerable amount of data generated requires extensive storage and sophisticated analytical tools, potentially leading to challenges in data management and processing.
- Cost: High-quality hyper spectral sensors can be expensive, limiting access for some researchers and organizations.
- Atmospheric Effects: Hyper spectral imaging can be sensitive to atmospheric conditions, necessitating correction algorithms to account for these variations.

Hyper spectral imaging represents a significant advancement in the capability to analyze and interpret materials based on their spectral properties. With applications spanning agriculture, environmental monitoring, healthcare, and beyond, this technology provides valuable insights that can drive decision-making and innovation across various domains. As hyper spectral sensors continue to improve and data analysis techniques advance, the applications and accessibility of hyper spectral imaging are expected to grow, offering new opportunities in research and industry.



CHECK YOUR POGRESS 2

Q2: Fill In the blanks

1. A _____ image is a type of digital image that consists of only two colors - black and white.
2. In a grayscale image, each pixel is represented by a single value that corresponds to the range _____ in an 8-bit image.
3. The _____ model is a subtractive color model primarily used in color printing.
4. The HSV model aligns more closely with human perception of _____.
5. Multispectral images are captured using _____ that can record data across various wavelengths.



3.5 LET US SUM UP

- A digital image represents a two-dimensional picture as a finite collection of digital values which are commonly referred to as picture elements or pixels.
- Digital images can be classified based on several criteria, each emphasizing different aspects of the images. Based on the image format, they can be classified in to raster and vector images.
- Digital cameras capture images by converting light into electrical signals through a sensor, such as a CCD (charge-coupled device) or CMOS (complementary metal-oxide-semiconductor).
- A pixel is the smallest individual unit of a picture displayed on a screen.
- The quality of an image is affected by both the number of pixels it contains and the pixel density (the number of pixels displayed per inch).
- A raster graphics image maps bits directly to a display area, commonly known as a bitmap. Image types can be 8-bit grayscale, 8-bit color, or 24-bit color.

- One of the primary benefits of raster graphics is their ability to create detailed and realistic images, such as photographs, paintings, textures, and visual effects.
- Vector images are graphics created using mathematical equations to define shapes, lines, and colors.
- One of the key advantages of vector images is their ability to be resized both larger and smaller without losing clarity, while also maintaining a consistent file size.
- One of the key benefits of vector graphics is their resolution independence, allowing them to be resized both larger and smaller without impacting their quality or visual integrity.
- A binary image is a type of digital image that consists of only two colors, typically represented as black and white. Each pixel in a binary image can take on one of two values, usually encoded as 0 and 1
- A grayscale is a type of digital image that contains varying shades of gray, ranging from black to white. Unlike binary images that consist of only two colors, grayscale images can represent a wide spectrum of intensities, allowing for more detailed and nuanced visual information.
- Color images are a fundamental aspect of visual media, representing images in a way that resembles how humans perceive colors. Unlike grayscale images, which contain only shades of gray, color images are composed of multiple colors that contribute to a richer representation of a scene or object.
- Multispectral imaging is a powerful remote sensing technique that captures image data at specific frequency ranges across the electromagnetic spectrum.
- Multispectral imaging represents a significant advancement in the field of remote sensing, providing detailed and diverse data that is critical for various scientific and practical applications.
- Hyper spectral imaging is an advanced technique used to capture and analyze images across a broad range of wavelengths of the electromagnetic spectrum. Unlike traditional imaging methods, which

typically capture data in three (RGB) or a few discrete bands (multispectral), hyper spectral imaging collects detailed information from hundreds of contiguous spectral bands.



3.6 FURTHER READING

1. “Digital image processing and pattern recognition”, Pakhira malay K, PHI
2. “Digital image processing”, Sundaramurthy Sridhar, Oxford
3. “Fundamentals of Digital Image Processing”, S Annadurai, R Shanmugalakshmi, Pearson



3.7 ANSWERS TO CHECK YOUR PROGRESS

ANSWERS TO QUESTION NUMBER 1

1. Processed, transmitted
2. pixels per inch (PPI), dots per inch (DPI).
3. Raster, vector
4. Pixel
5. number of pixels, pixel density
6. resolution independence
7. design, print media, digital applications

ANSWERS TO QUESTION NUMBER 2

1. Binary
2. 0 to 255
3. Cyan, Magenta, Yellow
4. Colors
5. sensors



3.8 MODEL QUESTIONS

Q1: Explain digital images and its different types.

Q2: What are the differences between raster and vector images.

Q3: Why are different formats used in image processing?

*****

UNIT 4: IMAGE TRANSFORM

UNIT STRUCTURE

4.1 Learning Objectives

4.2 Introduction

4.3 Need for Transformation

4.4 Image Transformation

4.5 Classification of Image Transformation

4.6 Fourier Transform

4.7 2D Discrete Fourier Transform

4.8 Properties of 2D DFT

4.9 Let Us Sum Up

4.10 Answers to Check Your Progress

4.11 Further Reading

4.12 Model Questions

4.1 LEARNING OBJECTIVES

After going through this unit, you will be able to have knowledge about:

- list the usefulness of image transformation
- list the classes under which different transformation techniques are categorized
- explain the concept of Fourier Transform
- describe the theoretical background of 2D DFT and Inverse 2D DFT
- explain the various properties of 2D DFT

4.2 INTRODUCTION

Image processing is a pivotal discipline within the realms of computer science and engineering, dedicated to the manipulation and analysis of digital imagery. This dynamic field encompasses a diverse array of techniques, all geared towards enhancing image quality, extracting valuable information, and preparing images for targeted applications. At the core of image processing lies the fundamental concept of image transformation,

which occupies a crucial role across various image processing tasks and applications. Through the application of mathematical operations, image transformation allows researchers and practitioners to examine digital images from multiple perspectives, uncovering hidden patterns and characteristics that are not immediately apparent in the original form. This versatile technique serves as a powerful tool for gaining deeper insights and facilitating more robust understanding of visual data.

4.3 NEED FOR TRANSFORMATION

Image transforms are essential for several reasons:

- *Feature Extraction*: Converting images to alternative representations can accentuate and highlight distinctive features and characteristics that are not readily apparent or easily discernible when examining the original image in the spatial domain. For instance, the application of the Fourier transform can uncover and reveal the underlying frequency components present within the image, thereby facilitating more in-depth analysis and understanding of periodic patterns, textures, and other visual elements that may not be immediately evident in the original spatial representation.
- *Compression*: Transforms such as the Discrete Cosine Transform consolidate and compactify the image data, facilitating more efficient storage and transmission by selectively discarding or prioritizing the less significant frequency components. This process optimizes the representation of the image information, allowing for reduced file sizes and faster data transfer while preserving the essential visual features.
- *Restoration*: Image transforms enable noise reduction and image restoration by separating and isolating the noise components from the actual meaningful image data within the transformed domain. This allows for more effective filtering and processing to recover the original image signal, while mitigating the detrimental effects of unwanted noise and distortions.

- *Analysis*: Image transforms facilitate the analysis and examination of key visual characteristics such as edges, corners, and shapes, which are crucial for various image processing tasks, including object recognition and enhancing overall image understanding. These transforms enable researchers and practitioners to extract and analyze essential visual features and patterns that are integral to interpreting and comprehending the content and structure of digital images.
- *Watermarking*: Image transforms can be employed to embed digital watermarks into images, which can serve as a valuable tool for copyright protection, content authentication, and ensuring the integrity of digital visual assets. This technique allows for the invisible incorporation of identifying information or digital signatures into the image data, which can be later extracted and utilized to verify the provenance and authenticity of the visual content. By embedding these watermarks, image owners and creators can effectively safeguard their intellectual property and prevent unauthorized usage or distribution of their work.

4.4 IMAGE TRANSFORMATION

In the field of image processing, image transformation refers to the process of modifying a digital image by applying a mathematical operation. This process alters the representation of the image data, often transitioning it between different domains to reveal diverse features or attributes. The analogy can be made to observing an object through various lenses, where each lens provides a unique perspective. Image transformation allows researchers and practitioners to examine an image from multiple angles, uncovering different characteristics and patterns that may not be immediately apparent when viewing the image in its original form. By applying various mathematical functions, the image data can be manipulated and analyzed in ways that facilitate a deeper understanding of the visual information contained within the image.

Image transformation can be utilized multiple ways to analyze, manipulate, and interpret image data. The key areas of utilization are highlighted as follows:

1. **Image Enhancement**

   Smoothing/Blurring: Transforms allow for easy manipulation of frequency components. By attenuating high-frequency components (which correspond to sharp edges and details), you can achieve smoothing and blurring effects.

   Sharpening: Conversely, boosting high-frequency components can sharpen images, enhancing edges and fine details.

   Noise Reduction: Noise often manifests as high-frequency components. Transform domain filtering can isolate and suppress these components, effectively reducing noise while preserving important image features.

   Contrast Enhancement: Transforms can be used to modify the distribution of intensity levels, leading to improved contrast.

2. **Image Compression**

   Transform Coding: Transforms like the DCT are the foundation of many image compression standards (e.g., JPEG). They compact image energy into a smaller number of coefficients, allowing for significant data reduction.

3. **Image Analysis and Feature Extraction**

   Edge Detection: Transforms can highlight edges and boundaries within an image by emphasizing high-frequency components.

   Texture Analysis: Different textures exhibit distinct patterns in the transform domain, enabling texture classification and segmentation.

   Shape Description: Transforms like the Hough transform can detect specific shapes (lines, circles, etc.) within an image.

4. **Image Restoration**

   Deblurring: If the blurring function is known or can be estimated, transforms can be used to deblur images by reversing the blurring operation in the frequency domain.

Inpainting: Transforms can assist in filling in missing or damaged parts of an image.

5. **Pattern Recognition**

   Template Matching: Transforms can be used to efficiently search for specific patterns or objects within an image.

   Object Recognition: Transform domain features can be used as input to machine learning algorithms for object recognition.

6. **Image Watermarking**

   Robust Watermarking: Transforms can be used to embed watermarks in the frequency domain, making them resistant to common image manipulations.

4.5 CLASSIFICATION OF IMAGE TRANSFORMATION

Image transforms can be classified, categorized, and organized based on a variety of criteria, such as the specific mathematical domain in which they operate, their linearity properties, and the spatial locality or global nature of their application. This diverse range of classification schemes allows users to select and utilize the most appropriate transform techniques for their specific image processing tasks and requirements. The widely popular classifications of image transforms are as follows:

1. **Based on Domain**:

   Spatial Domain Transforms: These transforms operate directly on the pixel values of the image. Examples include point operations (e.g., contrast stretching, histogram equalization) and spatial filtering (e.g., smoothing, sharpening).

   Frequency Domain Transforms: These transforms decompose the image into its frequency components. Examples include the Discrete Fourier Transform, Discrete Cosine Transform, and Walsh-Hadamard Transform.

2. **Based on Linearity**:

   Linear Transforms: These transforms satisfy the superposition principle, meaning that the transform of a weighted sum of images

is equal to the weighted sum of their individual transforms. Examples include the DFT, DCT, and Wavelet Transform.

Non-linear Transforms: These transforms do not adhere to the superposition principle. Examples include logarithmic transformations and homomorphic filtering.

3. **Based on Locality**:

   Global Transforms: These transforms operate on the entire image at once, considering all pixel values simultaneously. Examples include the DFT and DCT.

   Local Transforms: These transforms operate on smaller regions or blocks of the image, focusing on local features. Examples include the Wavelet Transform and Gabor Transform.

4.6 FOURIER TRANSFORM

The Fourier Transform is a foundational mathematical technique that decomposes a function, such as an image or signal, into a summation of sine and cosine waves with varying frequencies, amplitudes, and phases. This powerful transform shifts the function from its original spatial or time domain to the frequency domain, exposing the underlying frequency components that comprise the function. This capability is essential for a wide range of applications, including analyzing periodic patterns, mitigating noise, and compressing data in an efficient manner. By decomposing the function into its frequency constituents, the Fourier Transform enables the analysis, manipulation, and optimization of the function's spectral characteristics, which is crucial for numerous image and signal processing tasks. The Fourier Transform is a versatile and widely-used tool that allows researchers and practitioners to gain deeper insights into the structural and frequency-based properties of digital signals and images, empowering them to develop more effective techniques for tasks such as feature extraction, pattern recognition, and data compression.

The *Discrete Fourier Transform* is the discrete counterpart of the Fourier Transform, designed specifically for processing digital data such

as images. It takes a finite sequence of data points and transforms them into a finite sequence of complex numbers representing the amplitudes and phases of the different frequency components present in the input. This process enables the detailed analysis and precise manipulation of the frequency spectra of digital images, which is crucial for a wide range of essential image processing tasks, including noise reduction, image enhancement, and data compression. The ability to work in the frequency domain provided by the Discrete Fourier Transform is a powerful tool that allows development of sophisticated and effective techniques for various image processing applications.

4.7 2D DISCRETE FOURIER TRANSFORM

The 2D Discrete Fourier Transform extends the concept of frequency domain analysis to two spatial dimensions, making it a powerful and versatile tool for a wide range of image processing applications. This transform decomposes a digital image into a summation of 2D sinusoidal waves with varying frequencies, amplitudes, and phases along both the horizontal and vertical axes. It is computed by sequentially applying the 1D Discrete Fourier Transform to each row and then each column of the image data, resulting in a 2D array of complex numbers. Each element in this array represents the amplitude and phase of a specific 2D frequency component present in the input image.

For an image $f(x, y)$ of size $M X N$, where x and y represent spatial coordinates (rows and columns), the 2D DFT, $F(u, v)$, is defined as:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

where:

- $f(x, y)$ is the image in the spatial domain.
- $F(u, v)$ is the DFT of the image in the frequency domain.
- u and v are frequency variables representing the horizontal and vertical frequencies, respectively.
- M and N are the dimensions of the image.
- J is the imaginary unit ($\sqrt{-1}$)

The *Inverse 2D DFT* is essential because it allows us to reconstruct the original image after processing it in the frequency domain. Many image processing techniques are more easily or efficiently performed in the frequency domain using the 2D DFT. However, the results of these operations are also in the frequency domain. To see the effects of these operations in the spatial domain (i.e., on the actual image), we must apply the Inverse 2D DFT. The inverse 2D DFT reconstructs the original image from its frequency domain representation:

$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})}$$

where:

- $f(x, y)$ is the image in the spatial domain.
- $F(u, v)$ is the DFT of the image in the frequency domain.
- u and v are frequency variables representing the horizontal and vertical frequencies, respectively.
- M and N are the dimensions of the image.
- J is the imaginary unit ($\sqrt{-1}$)

Implementation of Discrete Fourier Transform using Python's OpenCV package:

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

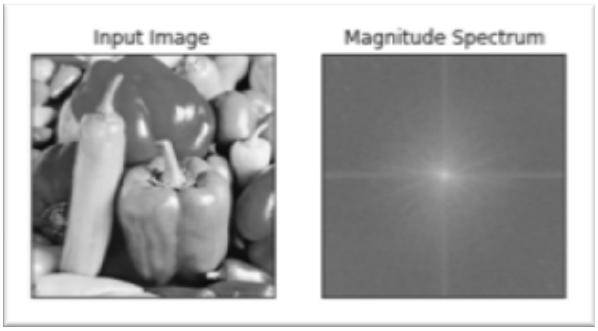
img = cv2.imread('peppers_gray.tif',0)

dft = cv2.dft(np.float32(img),flags = cv2.DFT_COMPLEX_OUTPUT)
dft_shift = np.fft.fftshift(dft)

magnitude_spectrum = 20*np.log(cv2.magnitude(dft_shift[:,:,0],dft_shift[:,:,1]))

plt.subplot(121),plt.imshow(img, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(magnitude_spectrum, cmap = 'gray')
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
plt.show()
```

Output:



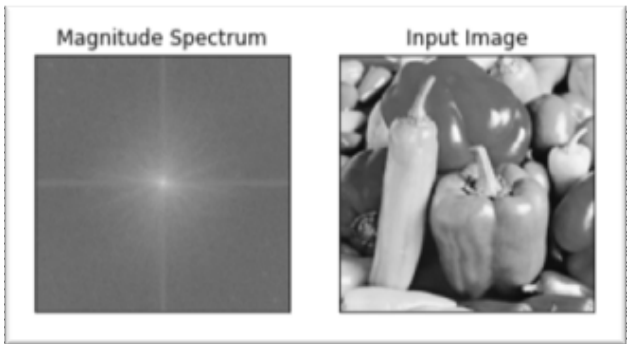
Implementation of Inverse Discrete Fourier Transform using Python's OpenCV package:

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

fshift = dft_shift
f_ishift = np.fft.ifftshift(fshift)
img_back = cv2.idft(f_ishift)
img_back = cv2.magnitude(img_back[:,:,0],img_back[:,:,1])

plt.subplot(121),plt.imshow(magnitude_spectrum, cmap = 'gray')
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(img_back, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.show()
```

Output:



4.8 PROPERTIES OF 2D DFT

The following are the key properties of 2D DFT:

Separable: The separable property of the 2D DFT means that it can be computed by applying the 1D DFT separately along each dimension of the 2D input data (like an image). Instead of directly computing the 2D DFT, you can first apply the 1D DFT to each row of the image, and then apply the 1D DFT to each column of the resulting transformed data. This is equivalent to applying the 2D DFT directly but is often computationally more efficient.

Spatial Shift Property: The spatial shift property of the 2D DFT describes how a shift in the spatial domain (the image itself) affects the transform in the frequency domain. Specifically, shifting an image circularly by a certain amount results in a linear phase shift in the frequency domain. The magnitude of the DFT remains unchanged.

Periodicity:

The periodicity property of the 2D DFT states that the transformed function $F(u, v)$ is periodic in both the u and v directions with periods M and N respectively, where M and N are the dimensions of the image $f(x, y)$.

Mathematically, this means:

$$F(u + aM, v + bN) = F(u, v)$$

for any integers a and b.

This property arises from the discrete nature of the DFT. Because the input image is a finite grid of pixels, the DFT treats it as one period of an infinitely repeating 2D signal. Therefore, the transformed representation in the frequency domain also repeats infinitely with the same periods M and N.

Convolution: The convolution property of the 2D DFT indicates that convolution in the spatial domain corresponds to multiplication in the frequency domain, and vice versa. This fundamental relationship renders the DFT highly beneficial for image processing applications.

Specifically, if you have two images, $f(x, y)$ and $h(x, y)$, and their convolution is denoted by $g(x, y) = f(x, y) * h(x, y)$, then the 2D DFT of the convolution is simply the product of the individual DFTs:

$$G(u,v) = F(u,v)H(u,v)$$

where:

- $G(u, v)$ is the 2D DFT of $g(x, y)$
- $F(u, v)$ is the 2D DFT of $f(x, y)$
- $H(u, v)$ is the 2D DFT of $h(x, y)$

Correlation Properties: The correlation property of the 2D DFT describes the relationship between correlation in the spatial domain and the frequency domain. Similar to the convolution property, correlation in the spatial domain can be computed as multiplication in the frequency domain, with a slight modification.

The 2D DFT of the correlation, $R(u, v)$, is related to the DFTs of the individual images as follows:

$$R(u,v) = F(u,v)H^*(u,v)$$

where:

- $R(u, v)$ is the 2D DFT of $r(x, y)$ [here, $r(x, y)$ is the correlation of the two images $f(x, y)$ and $h(x, y)$]
- $F(u, v)$ is the 2D DFT of $f(x, y)$
- $H^*(u, v)$ is the complex conjugate of $H(u, v)$, the 2D DFT of $h(x, y)$

Scaling Property: The scaling property of the 2D DFT describes how scaling the spatial coordinates of an image affects its frequency domain representation. Specifically, scaling the spatial domain leads to an inverse scaling in the frequency domain, along with an amplitude adjustment.

Rotation Property: The rotation property of the 2D DFT states that rotating an image in the spatial domain corresponds to an equal rotation of its DFT in the frequency domain.



CHECK YOUR PROGRESS

Q1: What is the main purpose of applying image transformation in image processing?

.....

.....

Q2: Name two common types of image transforms based on the domain in which they operate.

.....

.....

Q3: How does the Fourier Transform help in analyzing digital images?

.....

.....

Q4: What is the difference between the 2D DFT and the Inverse 2D DFT?

.....

.....

Q5: Mention any two properties of the 2D Discrete Fourier Transform.

.....

.....

Q6: Why is image transformation useful in tasks like noise reduction or image restoration?

.....

.....



4.9 LET US SUM UP

- **Image transformation** is a fundamental concept in image processing used to manipulate and analyze digital images through mathematical operations.
- Image transforms serve multiple purposes such as **enhancement**, **compression**, **restoration**, **analysis**, and **watermarking**.
- Transformations can help in **feature extraction** by making hidden patterns and frequency components in images more visible.
- **Image compression**is achieved using transforms like the Discrete Cosine Transform (DCT), which compact image data for efficient storage.

- Transforms aid in **image restoration**, such as noise removal and deblurring, by separating noise from the main signal in the frequency domain.
- **Contrast enhancement**, **smoothing**, and **sharpening** are common operations facilitated by image transforms.
- Transforms are also vital in **pattern recognition**, including tasks like template matching and object detection.
- Image transformations are classified based on the **domain (spatial or frequency)**, **linearity (linear or nonlinear)**, and **locality (global or local)**.
- The **Fourier Transform** is a powerful tool that breaks down image data into frequency components using sine and cosine functions.
- The **2D Discrete Fourier Transform (2D DFT)** extends frequency analysis to two dimensions and is crucial for analyzing image patterns.
- The **Inverse 2D DFT** allows reconstruction of the original image after frequency-domain processing.
- Key **properties of 2D DFT** include separability, periodicity, spatial shift, convolution, correlation, scaling, and rotation.



4.10 FURTHER READING

- "Digital image processing and pattern recognition", Pakhiramalay K, PHI
- "Digital image processing", Sundaramurthy Sridhar, Oxford
- "Fundamentals of Digital Image Processing", S Annadurai, R Shanmugalakshmi, Pearson



4.11 ANSWERS TO CHECK YOUR POGRESS

Ans. to Q.No 1: Image transformation helps to analyze and manipulate digital images by converting them into different domains to uncover hidden patterns and enhance visual data interpretation.

Ans. to Q.No 2: Spatial Domain Transforms, Frequency Domain Transforms

Ans. to Q.No 3: The Fourier Transform decomposes an image into its frequency components, allowing detailed analysis of textures, patterns, and structures not visible in the spatial domain.

Ans. to Q.No 4: The 2D DFT transforms a spatial image into its frequency domain representation, while the Inverse 2D DFT converts the processed frequency data back into the spatial image.

Ans. to Q.No 5: Separable, Convolution property

Ans. to Q.No 6: Transformations allow separation of noise from meaningful data in the frequency domain, enabling more effective filtering and accurate image restoration.



4.12 MODEL QUESTIONS

Q 1: Explain the concept of image transformation in detail. Discuss its significance in various image processing tasks with suitable examples.

Q 2: Describe the classification of image transformation techniques. Illustrate the basis of classification with examples for each type.

Q 3: What is the Fourier Transform? How is the 2D Discrete Fourier Transform (2D DFT) applied in digital image processing? Explain with mathematical formulation.

Q 4: Discuss the key properties of the 2D DFT. How do these properties support efficient image analysis and processing?

Q 5: Explain the steps involved in computing the 2D DFT and its inverse using OpenCV in Python. What are the practical implications of these implementations?

Q 6: Illustrate the role of image transforms in feature extraction and pattern recognition. Provide examples of specific transforms used in these applications.

Q 7: How does image transformation contribute to image compression and watermarking? Explain with examples of commonly used transforms.

*****

UNIT 5 : IMAGE ENHANCEMENT IN SPATIAL DOMAIN

UNIT STRUCTURE

5.1 Learning Objectives

5.2 Introduction

5.3 Image Enhancement in Spatial Domain: Point operation, Mask operation

5.4 Types of Point Operation: Brightness Modification, Contrast Adjustment

5.5 Let Us Sum Up

5.6 Further Reading

5.7 Answers to Check Your Progress

5.8 Model Questionss

5.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- understand the role of point operations in modifying pixel values to adjust image features like brightness and contrast.
- explore brightness modification techniques, including the use of linear and nonlinear functions to adjust the overall intensity of an image.
- explore contrast adjustment methods to enhance the visibility of details in an image.
- compare and contrast different methods of point operation, evaluating their effectiveness in improving image clarity and quality.
- apply brightness and contrast adjustment techniques in practical scenarios, adjusting images for specific requirements in fields like medical imaging, computer vision, and photography.

5.2 INTRODUCTION

Image enhancement in the spatial domain refers to improving the visual quality or interpretability of an image by directly manipulating its pixel

intensities. This domain primarily focuses on spatial characteristics such as edges, textures, and brightness levels, enabling more effective visualization or further processing of the image.

Spatial domain techniques operate on the raw pixel values of an image and are typically straightforward to implement. These methods can be broadly categorized into point processing and mask operation. Point processing involves manipulating individual pixels independently. Common techniques in this category include contrast stretching, intensity transformations such as logarithmic and power-law transformations, and histogram equalization, which enhances the overall contrast of an image by redistributing pixel intensity values.

Mask operations, also known as neighbourhood orSpatial filtering operations, on the other hand, considers the relationship between a pixel and its neighbours. Filters like smoothing (low-pass) and sharpening (high-pass) are commonly applied to reduce noise, enhance edges, or emphasize specific features. For instance, a Gaussian blur reduces random noise, while Laplacian and Sobel filters enhance edges by highlighting areas of rapid intensity change.

The spatial domain is particularly useful for applications requiring localized enhancements, such as medical imaging, satellite image processing, and document analysis. However, the limitations of these methods include potential artifacts, sensitivity to noise, and their inability to address global image properties comprehensively.

Despite these challenges, spatial domain techniques remain a fundamental aspect of image processing due to their simplicity and efficiency. They often serve as a preliminary step in more complex workflows, laying the groundwork for further analysis in other domains, such as the frequency domain.

By employing spatial domain techniques judiciously, one can significantly enhance image quality, making it more suitable for visualization and interpretation.

5.3 IMAGE ENHANCEMENT IN SPATIAL DOMAIN: POINT OPERATION, MASK OPERATION

Point operations: Point operations are fundamental techniques in image enhancement, performed directly on individual pixel intensities in an image. These methods are simple yet powerful, manipulating the intensity of each pixel independently based on predefined mathematical functions. The primary advantage of point operations is their computational efficiency, as they do not require interaction with neighbouring pixels.

If (x) represents the intensity of a pixel at location (x, y) in the input image, and $g\ (x, y)$ is the intensity of the corresponding pixel in the output image, point operations can be expressed as:

$$g(x,y)=T[f(x,y)]$$

Here, T is the transformation function applied to each pixel.

The Fig. 5.1 demonstrates the Point operations in image processing which involve modifying pixel values based on certain rules or functions. As show in fig 5.1, brightness Adjustmentadds a constant value to each pixel to make the image brighter or darker. Contrast adjustment multiplies the pixel values by a factor to increase or decrease the intensity differences. And thresholding converts the image to binary by setting pixel values above a threshold to 255 and below to 0.

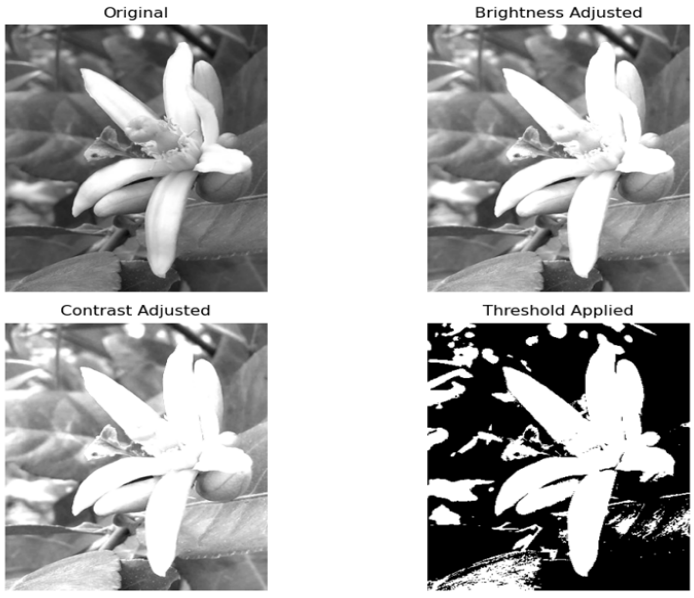


Figure 5.1. Image with adjusted brightness, Contrast and Threshold

Mask operations: In image processing, mask operations are a key technique in the spatial domain used to improve image quality by applying filters or kernels to a local region of the image. Unlike point operations, which adjust individual pixel values independently, mask operations (often referred to as neighbourhood operations) operate on a pixel and its surrounding neighbourhood. This makes them particularly useful for tasks like noise reduction, sharpening, edge detection, and smoothing.

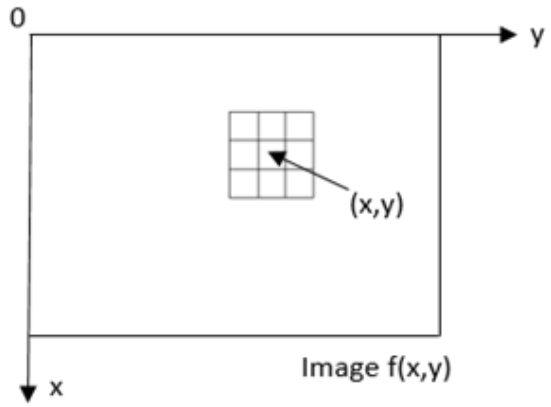


Figure 5.2: 3*3 neighbourhood about a point (x, y) in an image.

Mask operations involve the use of a small matrix, known as a mask or kernel, which is applied to an image to compute a new value for each pixel based on its neighbours. The principal approach in defining a neighbourhood about a point (x, y) is to use a squareor rectangular sub image area centered at (x, y), as shown in below figure 5.2.The kernel is typically a 3x3, 5x5, or larger matrix, but smaller kernels like 3x3 are the most commonly used. The mask is moved over the image in a sliding window fashion, and at each position, it performs a mathematical operation (usually a weighted sum) on the pixel values of the image.

The output pixel value is calculated as follows:

$$I'(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} i(x+i, y+j).M(i,j)$$

Where,

I(x,y) is the pixel value at location (x, y),

$(i,)$ is the mask value at position $(5\text{ØVÜ}, 5\text{ØWÜ})$,

k represents the radius of the mask (for a 3x3 kernel, k =1),

$I'(x,y)'$ is the new pixel value after the mask operation.

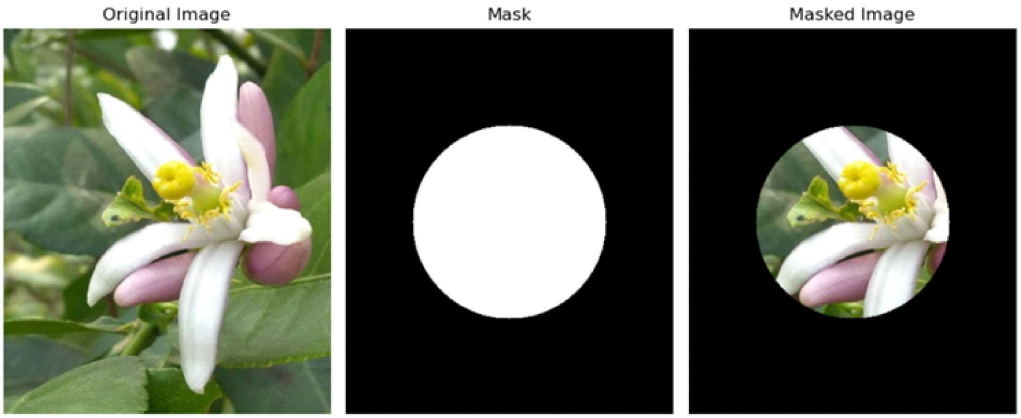


Figure 5.3. Original Image & the masked image

Fig. 5.3. demonstates a binary mask with the same dimensions as the image. Here, a circular region is defined as the area to retain.The operation retains only the parts of the image where the mask has non-zero values.

Types of Mask Operations

Smoothing Filters (Low-pass filters): These filters reduce high-frequency noise by averaging the pixel values in a neighbourhood. A common smoothing filter is the mean filter.

A 3x3 mean filter is applied by averaging the pixel values in a 3x3 region around each pixel. The filter looks like this:

Mean Filter Kernel: $\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$

This filter computes the average of all the pixels in the 3x3 neighbourhood and assigns the result to the centre pixel.

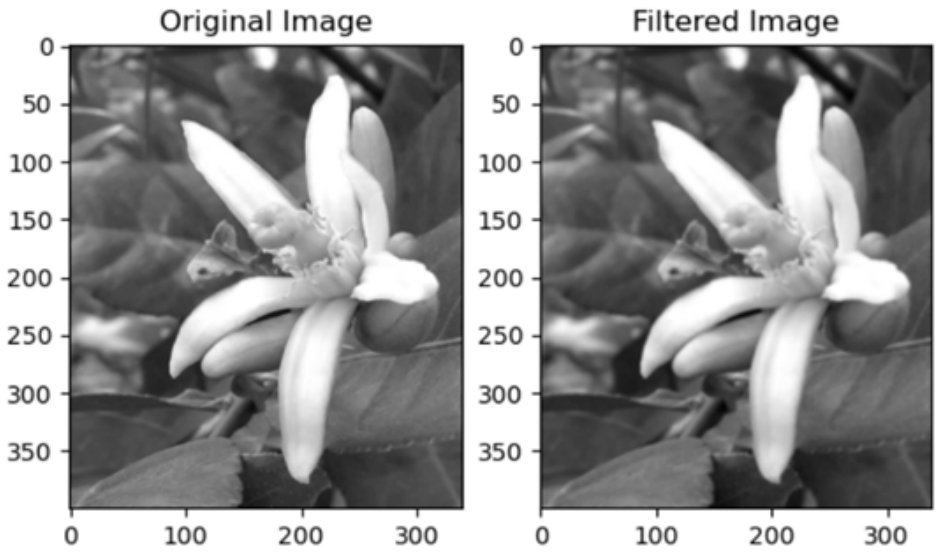


Figure 5.4. Original Image and Filtered Image

In Fig 5.4 a 3x3 kernel filled with equal values (1/9), which ensures that every pixel in the region is averaged equally. The function applies the kernel over the image by sliding it across each pixel, performing convolution. The original and filtered images are displayed side by side.

Sharpening Filters (High-pass filters): Sharpening filters enhance high-frequency components, such as edges, by subtracting low-frequency components. A common sharpening filter is the Laplacian filter.

A 3x3 Laplacian kernel emphasizes edges by computing the difference between the central pixel and its neighbours.

Laplacian Kernel: $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

This filter highlights areas with high intensity changes (edges) in the image.

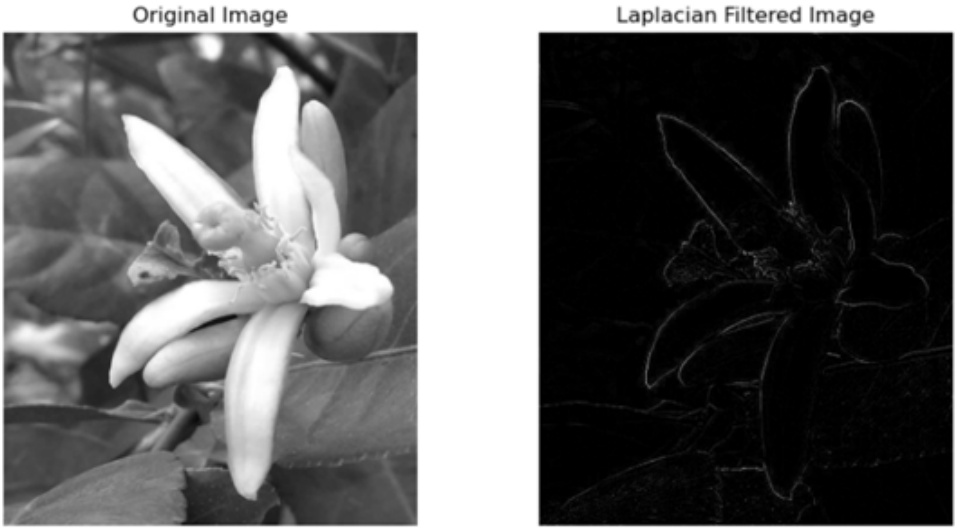


Figure 5.5. Original Image and Laplacian Filtered Inage

In the Fig. 5.5, a Laplacian Kernel emphasizes the differences between neighboring pixels and the central pixel. The value 4 at the center of the kernel creates a higher weight for the center pixel, and -1 values for the neighboring pixels allow the kernel to highlight edges. This function convolves the image with the kernel, producing the edge-enhanced result. The display the original and processed images side by side.

Edge Detection Filters: Edge detection filters identify boundaries in an image where significant changes in intensity occur. The Sobel operator is a popular edge detection filter used for detecting edges in both horizontal and vertical directions.

Example: A Sobel filter uses two kernels: one for detecting horizontal edges and one for vertical edges.

Sobel Vertical Kernel: $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

Sobel Vertical Kernel: $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

The Sobel filter detects edges by emphasizing changes in pixel values along the horizontal and vertical directions.

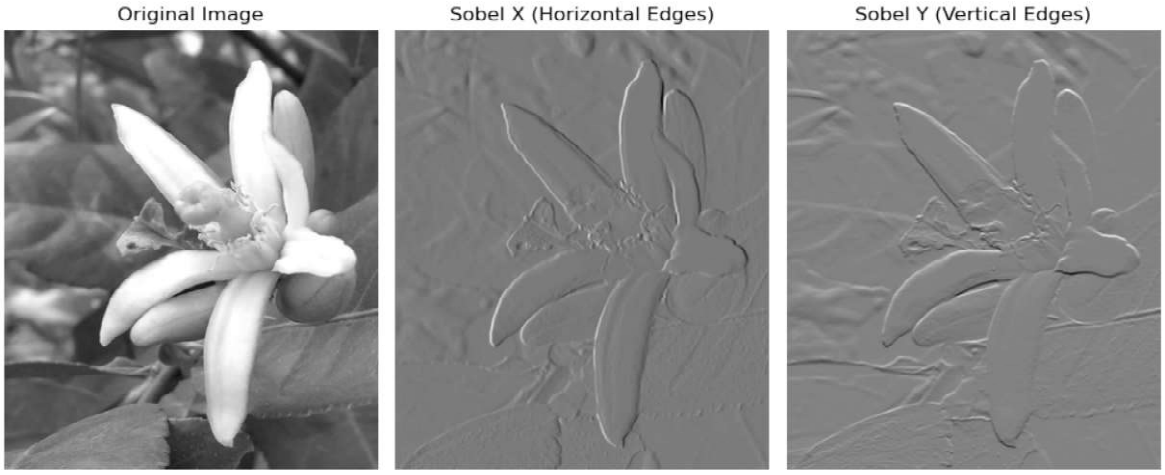


Figure 5.5. Original Image and Image after applying Sobel Filter

5.4 TYPES OF POINT OPERATION: BRIGHTNESS MODIFICATION, CONTRAST ADJUSTMENT

Brightness Modification

Brightness modification involves adjusting the intensity of every pixel by adding or subtracting a constant value. It is often used to make an image appear lighter or darker.

Let $(x,)$ be the input pixel intensity, (x, y) be the output pixel intensity, y and b be the brightness offset. The transformation is:

$$g(x,y)=f(x,y)+b$$

If b>0: The image becomes brighter.

If b<0: The image becomes darker.

Let us look at an example of an input grayscale image with intensity values ranging from 50 to 200.

The Brightness offset (b)can be set to +50

After applying b =+50, pixel intensities range from 100 to 255 in the output image.

Contrast Adjustment

Contrast adjustment modifies the difference between the darkest and brightest regions of an image, enhancing its dynamic range. It involves scaling pixel intensities around a central value.

Let a be the contrast scaling factor. The transformation is:

$(x,y)=a\cdot(f(x,y)-f_{\text{mean}})+f_{\text{mean}}$

Where f_{mean} is the average intensity of the image.

If a>1: Contrast increases.

If 0<a<1: Contrast decreases.

Simplified Linear Adjustment:

$g(x,y)=a\cdot f(x,y)+b$

Here, a controls the contrast, and b adjusts brightness.

Let us look at an example Use Case where the scenario is toenhancing a poorly lit image.

We can apply Brightness Modification b =30 to shift all pixel values upward by +30, making the image brighter. And Contrast Adjustment by a factor of$a$=1.2 to stretch the intensity range, increasing visual distinction.

By combining brightness modification and contrast adjustment, a wide range of visual enhancements can be achieved, a comparison of effects is shown in table 5.1.

Table 5.1

Operation	Impact on Pixel Values	Visual Effect
Brightness Increase	Adds a constant to all pixel intensities	Lighter overall appearance
Contrast Adjustment	Scales pixel intensities	Sharper differences between regions



CHECK YOUR PROGRESS

i. Which operation is performed directly on the pixels of an image in the spatial domain?

a) Fourier Transform
b) Point Operation
c) Wavelet Transform
d) Histogram Equalization

ii. Point operations modify the pixel value based on which of the following?
   a) Neighboring pixels
   b) A single pixel
   c) Frequency domain coefficients
   d) Entire image

iii. What is the primary use of mask operations in spatial domain?
   a) Compressing images
   b) Highlighting edges and textures
   c) Enhancing colors
   d) Reducing image size

iv. In spatial domain, a mask is often applied using which mathematical operation?
   a) Addition
   b) Convolution
   c) Multiplication
   d) Subtraction

v. Which operation smooths an image in spatial domain?
   a) High-pass filtering
   b) Band-pass filtering
   c) Low-pass filtering
   d) Thresholding

vi. Which of the following is a type of point operation?
   a) Smoothing
   b) Edge Detection
   c) Segmentation
   d) Brightness Modification

vii. Contrast adjustment involves which of the following?
   a) Multiplying pixel values by a scaling factor
   b) Adding a constant to pixel values
   c) Applying a Fourier Transform
   d) Histogram normalization

viii. What happens if a very high scaling factor is used in contrast adjustment?
   a) The image becomes darker
   b) The image becomes too bright or saturated
   c) The image loses sharpness
   d) No change in the image

ix. Which of the following is a possible outcome of brightness adjustment?
   a) Loss of spatial details
   b) Change in the dynamic range of the image
   c) Change in edges only
   d) Introduction of noise

x. Contrast stretching is used to achieve what?
   a) Reduce image size
   b) Increase the dynamic range of pixel values
   c) Smooth the image
   d) Highlight edges



5.5 LET US SUM UP

- Image enhancement in the spatial domain refers to improving the visual quality or interpretability of an image by directly manipulating its pixel intensities.
- Image enhancement in the spatial domain can be broadly categorized into point processing and mask operation.
- Point processing involves manipulating individual pixels independently. Common techniques in this category include contrast stretching, intensity transformations.
- Mask operations, also known as neighbourhood or Spatial filtering operations, considers the relationship between a pixel and its neighbours. Filters like smoothing (low-pass) and sharpening (high-pass) are commonly applied to reduce noise, enhance edges, or emphasize specific features.

- Smoothing Filters (Low-pass filters): These filters reduce high-frequency noise by averaging the pixel values in a neighbourhood. A common smoothing filter is the mean filter.
- Sharpening filters enhance high-frequency components, such as edges, by subtracting low-frequency components.
- A common sharpening filter is the Laplacian filter.
- Edge detection filters identify boundaries in an image where significant changes in intensity occur.
- The Sobel operator is a popular edge detection filter used for detecting edges in both horizontal and vertical directions.
- Brightness modification involves adjusting the intensity of every pixel by adding or subtracting a constant value. It is often used to make an image appear lighter or darker.
- Contrast adjustment modifies the difference between the darkest and brightest regions of an image, enhancing its dynamic range. It involves scaling pixel intensities around a central value.



5.6 FURTHER READING

- Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods, Publisher: Pearson, 4th Edition (2018).
- Computer Vision: Algorithms and Applications by Richard Szeliski, Publisher: Springer, 1st Edition (2010).
- Digital Image Processing and Analysis by J. R. Parker, Publisher: Society of Photo-Optical Instrumentation Engineers, 1st Edition (1997).
- Image Processing: The Fundamentals by Maria M. P. van Loon and Johan G. S. van der Meer, Publisher: CRC Press, 1st Edition (2017).



5.7 ANSWERS TO CHECK YOUR PROGRESS

i. b	ii. b	iii. b	iv. d	v. c	vi. d
vii. a	viii. b	ix. b	x. b		



5.8 MODEL QUESTIONS

Q1: Define image enhancement and explain its importance in digital image processing.

Q2: What is the spatial domain in image processing? How does it differ from the frequency domain?

Q3: Explain the concept of point operations in the spatial domain. Provide examples.

Q4: What are mask operations in image processing? How are they implemented?

Q5: Compare and contrast point operations and mask operations in image enhancement.

Q6: Describe the process of brightness modification as a point operation.

Q7: What is contrast adjustment, and why is it significant in image enhancement?

Q8: Explain how point operations are applied pixel-by-pixel to an image.

Q9: Provide examples of mask operations used in edge detection.

Q10: How does adding a constant value to each pixel affect the brightness of an image?

Q11: Describe the mathematical formula for contrast adjustment. How does it affect pixel intensity values?

Q12: Illustrate with examples how brightness modification and contrast adjustment can be combined to enhance an image.

*****

UNIT 6: IMAGE HISTOGRAM

UNIT STRUCTURE

6.1 Learning Objectives

6.2 Introduction

6.3 Histogram Equalization

6.4 Procedure to Perform Histogram Equalization

6.5 Let Us Sum Up

6.6 Further Reading

6.7 Answers to Check Your Progress

6.8 Model Questions

6.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- learn what an image histogram represents: the frequency distribution of pixel intensity values in an image.
- recognize underexposed, overexposed, and well-balanced images based on histogram shapes.
- understand the principle of histogram equalization as a contrast enhancement technique.
- perform histogram equalization to redistribute intensity values and improve image contrast.
- analyse how histogram equalization affects the image histogram and overall appearance.

6.2 INTRODUCTION

An image histogram is a graphical representation of the tonal distribution in a digital image. It provides a visual summary of the intensity levels, or pixel values, across the image, offering insight into its brightness, contrast, and dynamic range. This tool is widely used in image processing, photography, and computer vision to analyze and enhance images.

The x-axis of an image histogram represents the intensity levels, typically ranging from 0 to 255 for an 8-bit grayscale image. Here, 0 corresponds to black, and 255 represents white, with intermediate values indicating shades of gray. For color images, histograms are often generated separately for each channel—red, green, and blue (RGB)—to illustrate the distribution of intensities in each color component. The y-axis shows the number of pixels corresponding to each intensity level.

An image histogram serves multiple purposes. First, it helps assess the exposure of an image. A histogram skewed to the left indicates an image that is predominantly dark (underexposed), while one skewed to the right suggests a lighter image (overexposed). A balanced histogram, spreading across the entire range, often suggests a well-exposed image with a good tonal range.

Histograms are also used in contrast enhancement techniques. For example, histogram equalization redistributes the pixel intensity values to improve contrast. This method stretches and flattens the histogram so that intensity levels are more uniformly distributed, enhancing details in both dark and bright areas. Similarly, histogram stretching adjusts the range of pixel intensities to use the full dynamic range of the image, improving visual clarity.

In addition to contrast enhancement, histograms are utilized in tasks like thresholding and segmentation. For instance, a bimodal histogram, which exhibits two distinct peaks, can indicate the presence of two primary regions in an image (e.g., foreground and background). In such cases, determining an appropriate threshold value to separate these regions becomes straightforward.

In color image processing, separate histograms for each RGB channel are analyzed to balance and correct colors. For example, an image with a dominant blue channel might require adjustments to achieve a more natural or aesthetically pleasing appearance.

While histograms offer valuable insights, they have limitations. They do not consider the spatial arrangement of pixels, meaning two images with the same histogram might look entirely different. Therefore, while

histograms are powerful for global image analysis, they are often combined with other tools for comprehensive image processing.

6.3 HISTOGRAM EQUALIZATION

Histogram equalization is a fundamental image processing technique used to improve the contrast of digital images. It operates by redistributing the pixel intensity values of an image to achieve a uniform intensity distribution. This technique is especially beneficial in images where details are obscured due to poor contrast, often caused by lighting conditions or the limitations of imaging devices.

In an image, the histogram is a graphical representation showing the frequency of occurrence of each intensity level. For a grayscale image, the intensity values range from 0 (black) to 255 (white). A histogram with a narrow range indicates low contrast, where pixel values are clustered in specific intensity levels. Histogram equalization addresses this by spreading the pixel intensity values more evenly across the entire range.

The process of histogram equalization involves two key steps: computing the cumulative distribution function (CDF) and remapping the pixel intensities. First, the image's histogram is computed to identify the frequency of each intensity level. Then, the CDF, which accumulates these frequencies, is calculated. This cumulative function is normalized to map the pixel values into a new range, ensuring a more uniform intensity distribution. The result is an image with enhanced contrast where details in both dark and bright areas become more visible.

Fig. 6.1 shows original Image, underexposed image and overexposed image. The results are obtained by simulating exposure effects. In underexposed image the pixel values are multiplied by a factor (less than 1) to darken the image. In overexposed Image the pixel values are multiplied by a factor (greater than 1) to brighten the image.



Figure: 6.1 Original Image, Underexposed image & Overexposed image

Figure: 6.2 shows the histogram of the original image and equalized well balanced image.The histograms showing the pixel intensity distributions before and after equalization. The equalized histograms typically have a more uniform distribution, improving contrast.

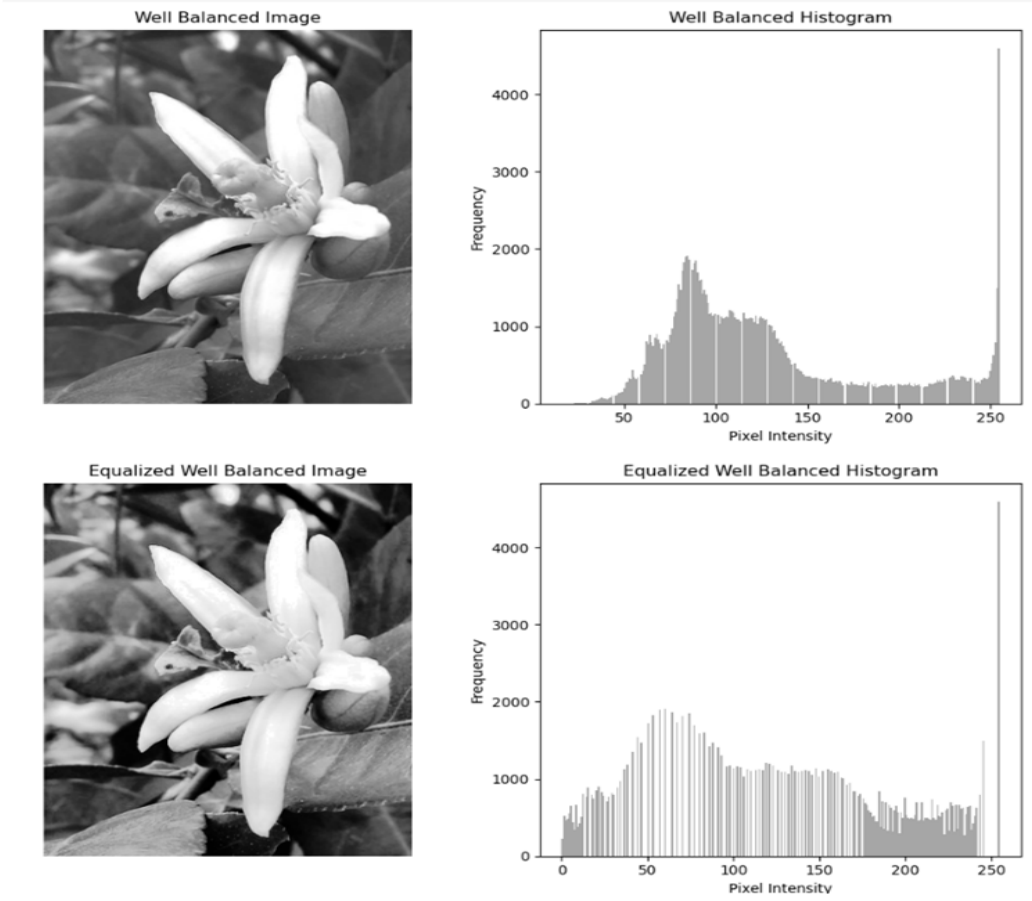


Figure: 6.2 Histogram of Original Image and Equalized Well balanced image

This technique is widely applied in various fields, such as medical imaging, remote sensing, and photography. For instance, in medical imaging, histogram equalization helps enhance the visibility of features in X-rays or MRI scans, aiding diagnosis. In remote sensing, it improves satellite images, making geographical features more distinct.

While histogram equalization offers several advantages, it is not without limitations. One challenge is the potential for over-enhancement, where the image may appear artificially high-contrast, leading to the loss of some details. Additionally, it is less effective on color images because altering intensity values individually in each color channel can distort the image's overall color balance. To address this, advanced methods such as adaptive histogram equalization (AHE) and contrast-limited adaptive histogram equalization (CLAHE) have been developed. These methods work on smaller regions of the image, preserving local details while minimizing artifacts.

The simplicity and effectiveness of histogram equalization have made it a standard technique in image processing. However, its application should be carefully evaluated based on the specific requirements of the task. For images where preserving natural appearance and subtle details is critical, alternative contrast enhancement methods may be preferable.

6.4 PROCEDURE TO PERFORM HISTOGRAM EQUALIZATION

Histogram equalization enhances image contrast by redistributing intensity values more uniformly across the available range. The procedure to perform histogram equalization involves the following steps:

STEP 1 **Computing the Histogram**: The first step is to calculate the histogram of the image, which shows the frequency of each pixel intensity value (e.g., 0–255 for an 8-bit grayscale image). This histogram provides an overview of how intensities are distributed in the image.

STEP 2 **Calculate the Probability Density Function (PDF):** In the second step, for each intensity level, its probability is computed by dividing the frequency by the total number of pixels in the image as follows:

$$PDF(i) = \frac{n_i}{N}$$

Where,

n_i is the number of pixels with intensity i

N is the total number of pixels in the image.

STEP 3 **Computing the Cumulative Distribution Function (CDF):** This step determines how intensities should be redistributed. Here, the cumulative probability for each intensity level is calculated as:

$$CDF(i) = \sum_{j=0}^{i} PDF(i)$$

STEP 4 **Normalizing the Cumulative Distribution Function (CDF):** **In the fourth step** the CDF values are scaled to fit the intensity range. For example, 0–255 for an 8-bit image.

$$CDF_{scaled}(i) = round(CDF(i)X\ (L-1))$$

Where L is the total number of possible intensity levels.

STEP 5 **Mapping the Original Intensities:** In this stepthe normalized CDF is used as a mapping function. Each pixel's intensity in the original image is replaced with the corresponding scaled CDF value:

$$I_{equalized}(x,y) = CDF_{scaled}(I(x,y))$$

Here, (x) is the original intensity at pixel location (x, y) and $I_{equalized}(x,y)$ is the new intensity.

STEP 6 **Constructing the Equalized Image:** In the final step the image is reconstructed using the updated intensity values for all pixels.The new image has enhanced contrast, with details in both dark and bright regions being more visible.

Example:

Let us consider a 4X4 grayscale image with pixel intensity values. The pixel intensity values range from 0 to 7, as follows:

$$\begin{bmatrix} 0 & 1 & 1 & 2 \\ 2 & 3 & 4 & 4 \\ 5 & 5 & 6 & 6 \\ 6 & 7 & 7 & 7 \end{bmatrix}$$

STEP 1 **Computing the Histogram**

Intensity (i)	**Frequency (n_i)**
0	1
1	2
2	2
3	1
4	2
5	2
6	3
7	3

STEP 2 **Calculating the Probability Density Function (PDF):** The PDF is computed by dividing each frequency by the total number of pixels (N =16).

Intensity (i)	Frequency (n_i)	PDF(i)= (n_i)/N
0	1	1/16 = 0.0625
1	2	2/16 = 0.1250
2	2	2/16 = 0.1250
3	1	1/16 = 0.0625
4	2	2/16 = 0.1250
5	2	2/16 = 0.1250
6	3	3/16 = 0.1875
7	3	3/16 = 0.1875

STEP 3 Compute the Cumulative Distribution Function (CDF)

Intensity (i)	PDF(i)= (n_i)/N	CDF (Cumulative Sum)
0	0.0625	0.0625
1	0.1250	0.0625+0.1250=0.1875
2	0.1250	0.1875+0.1250=0.3125
3	0.0625	0.3125+0.0625=0.3750
4	0.1250	0.3750+0.1250=0.5000
5	0.1250	0.5000+0.1250=0.6250
6	0.1875	0.6250+0.1875=0.8125
7	0.1875	0.8125+0.1875=1.0000

STEP 4 **Normalize the CDF:** The CDF values are mapped to the intensity range [0, 7] by multiplying each CDF value by the maximum intensity (7) and round to the nearest integer:

Intensity (i)	CDF	Normalized Intensity
0	0.0625	0
1	0.1875	1
2	0.3125	2
3	0.3750	3
4	0.5000	4
5	0.6250	4
6	0.8125	6
7	1.0000	7

STEP 5 **Mapping the Pixel Intensities:**Replacing the original pixel values in the image with their corresponding normalized intensity values:

Original Image

$$\begin{bmatrix} 0 & 1 & 1 & 2 \\ 2 & 3 & 4 & 4 \\ 5 & 5 & 6 & 6 \\ 6 & 7 & 7 & 7 \end{bmatrix}$$

After equalization, new image

$$\begin{bmatrix} 0 & 1 & 1 & 2 \\ 2 & 3 & 4 & 4 \\ 4 & 4 & 6 & 6 \\ 6 & 7 & 7 & 7 \end{bmatrix}$$



CHECK YOUR PROGRESS

i. What is the main purpose of histogram equalization?

a) To compress an image
b) To increase the brightness of an image
c) To enhance the contrast of an image
d) To reduce noise in an image

ii. Histogram equalization is a technique based on which domain?

a) Spatial domain
b) Frequency domain
c) Time domain
d) Compression domain

iii. Histogram equalization works best for images with:

a) Uniform intensity distribution
b) High contrast
c) Low contrast
d) Random noise

iv. Which property of an image does histogram equalization modify?

a) Color balance
b) Pixel intensity distribution
c) Image dimensions
d) Spatial resolution

v. What is a drawback of histogram equalization?

a) It increases noise
b) It is computationally expensive
c) It can over-enhance noise in some images
d) It reduces image resolution

vi. Histogram equalization is especially useful for images:
   a) With a single predominant color
   b) Captured under uniform lighting
   c) With excessive noise
   d) With poor lighting and low contrast

vii. What does the cumulative distribution function (CDF) represent in histogram equalization?
   a) Frequency of each intensity
   b) Total number of pixels in the image
   c) Proportional sum of pixel intensities
   d) Accumulated probability of pixel intensities

viii. How is the histogram of an image calculated?
   a) By counting the frequency of each pixel intensity level
   b) By summing all pixel values
   c) By dividing the total pixels by intensity levels
   d) By converting the image to binary

ix. The cumulative distribution function (CDF) is calculated by:
   a) Taking the derivative of the histogram
   b) Adding the histogram frequencies cumulatively
   c) Dividing each frequency by the total number of pixels
   d) Subtracting the mean intensity from each level

x. The cumulative probability in the CDF is scaled to which range?
   a) [0, 255]
   b) [0, 1]
   c) [1, 256]
   d) [0, max intensity]



6.5 LET US SUM UP

- Histogram equalization is an image processing technique that enhances contrast by redistributing pixel intensity values to achieve a more uniform distribution.
- The histogram represents the frequency of intensity values in an image. A narrow histogram indicates low contrast; which histogram equalization aims to correct.
- By spreading intensity values across the full range (e.g., 0–255 for grayscale images), histogram equalization makes details in both dark and bright areas more visible.
- Key Steps in histogram equalization:
  - Compute the Histogram: Determine the frequency of each intensity level in the image.
  - Calculate the CDF: Derive the cumulative distribution function to redistribute intensities.
  - Normalize CDF: Scale CDF values to fit the intensity range (e.g., 0–255).
  - Map Intensities: Replace original pixel values with scaled CDF values.
- Histogram equalization is widely used in medical imaging, remote sensing, and photography to enhance visibility and distinguish features in poorly contrasted images.
- Advantageshistogram equalization is that it issimple to implement, effective for improving contrast in grayscale images, and useful for enhancing details in images obscured by poor lighting or device limitations.
- Limitations of histogram equalization are:
  - Risk of over-enhancement, leading to unnatural high-contrast images.
  - Ineffectiveness on color images due to potential distortion in color balance.

- Advanced Methods: Adaptive histogram equalization (AHE) and contrast-limited adaptive histogram equalization (CLAHE) improve upon standard methods by working on smaller regions and minimizing artifacts.



6.6 FURTHER READING

- Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods, Publisher: Pearson, 4th Edition (2018).
- Computer Vision: Algorithms and Applications by Richard Szeliski, Publisher: Springer, 1st Edition (2010).
- Digital Image Processing and Analysis by J. R. Parker, Publisher: Society of Photo-Optical Instrumentation Engineers, 1st Edition (1997).
- Image Processing: The Fundamentals by Maria M. P. van Loon and Johan G. S. van der Meer, Publisher: CRC Press,1st Edition (2017).



6.7 ANSWERS TO CHECK YOUR PROGRESS

i. c ii. a iii. c iv. b v. c vi. d vii. d
viii. a ix. b x. a



6.8 MODEL QUESTIONS

Q 1: What is histogram equalization, and why is it used in image processing?

Q 2: How does histogram equalization improve the contrast of an image?

Q 3: Explain the difference between global and local histogram equalization.

Q 4: What types of images benefit the most from histogram equalization?

Q 5: What are the limitations of histogram equalization?

Q 6: How does histogram equalization affect the dynamic range of an image's pixel intensity values?

Q 7: Discuss the role of the cumulative distribution function (CDF) in histogram equalization.

Q 8: Describe real-world applications where histogram equalization is commonly used.

Q 9: What are the key steps involved in performing histogram equalization on a grayscale image?

Q 10: Explain how to calculate the probability distribution of pixel intensities during histogram equalization.

Q 11: How is the cumulative distribution function (CDF) computed, and why is it important?

Q 12: How are new pixel intensity values determined using the CDF during histogram equalization?

Q 13: Illustrate the steps of histogram equalization with a numerical example, starting from a given intensity distribution.

*****

UNIT 7: FILTERING

UNIT STRUCTURE

7.1 Learning Objectives

7.2 Introduction

7.3 Spatial Filtering

7.4 Linear Filtering

7.5 Low Pass Filter

7.6 Limitations of Low Pass Filter

7.7 Weighted Average Filter

7.8 Bartlett Filter

7.9 Gaussian Filter

7.10 Median Filter

7.11 High Pass Filtering

7.12 Let Us Sum Up

7.13 Further Reading

7.14 Answers to Check Your Progress

7.15 Model Questions

7.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- understand the principles of spatial filtering and linear filtering
- evaluate the characteristics and limitations of low-pass filters
- apply and compare various weighted average filters (e.g., Bartlett and Gaussian filters)
- understand and implement the median filter in image processing
- differentiate between high-pass and low-pass filtering in terms of edge enhancement and noise removal

7.2 INTRODUCTION

Spatial filtering is a fundamental concept in image processing, involving the manipulation of pixel values in an image based on their spatial

distribution. It is employed to enhance images, suppress noise, or extract specific features. Linear filtering is a specific type of spatial filtering where the output pixel value is a linear combination of its neighbouring pixel values. The weights used in the combination determine the effect of the filter, such as smoothing or edge detection.

A low-pass filter is a common linear filter used to reduce high-frequency components, such as noise, while retaining low-frequency information, such as smooth transitions. It achieves this by averaging the pixel values within a neighbourhood, effectively softening sharp changes. However, low-pass filters have limitations. They often blur edges and fine details, which can be problematic in applications where preserving these features is crucial. Weighted average filters are an improvement over basic low-pass filters, where the weights are assigned based on the relative importance of neighbouring pixels, providing a more refined smoothing effect.

The Bartlett filter, also known as the triangular filter, is a specific type of weighted filter that applies a linear tapering to the weights, giving more emphasis to the central pixels and gradually reducing the influence of pixels further away. Similarly, the Gaussian filter is another popular weighted average filter that uses a Gaussian function to assign weights. Its smooth, bell-shaped curve ensures that the influence of a pixel diminishes as its distance from the center increases, making it highly effective in reducing noise while maintaining the integrity of edges.

Non-linear filters, such as the median filter, provide an alternative approach to spatial filtering. Instead of averaging, the median filter replaces each pixel value with the median value of its neighbours. This method is particularly effective at removing salt-and-pepper noise without significantly affecting image details.High-pass filtering, in contrast, emphasizes high-frequency components in an image, such as edges and fine details, by suppressing low-frequency components. These filters are essential in applications requiring edge detection or feature enhancement. However, their sensitivity to noise is a drawback, often necessitating pre-processing steps to ensure clean results.

By carefully selecting and combining these filters, it is possible to address a wide range of image processing challenges, balancing noise reduction with feature preservation to achieve desired outcomes.

7.3 SPATIAL FILTERING

Spatial filtering refers to the process of modifying or enhancing an image based on its spatial frequency properties. Its applications include noise reduction, edge detection, and feature enhancement in fields like medical imaging and satellite image analysis. It involves applying a filter kernel (mask) to the image, either to emphasize certain features or to suppress noise. The filtering operation is performed in the spatial domain by convolving the filter kernel $h\,(x, y)$ with the image $f\,(x, y)$. The result is given by:

$$g(x,y) = \sum_{i=-m}^{m} \sum_{j=-n}^{n} h(i,j) f(x+i, y+j)$$

where $g\,(x, y)$ is the filtered image, m and n define the kernel size, and $h\,(i, j)$ is the filter kernel.

For example, smoothing filters like averaging reduce noise by replacing each pixel value with the average of its neighbours, while sharpening filters emphasize edges by highlighting rapid intensity changes.

Example of a Smoothing using a 3x3 averaging filter kernel:

$$h(x,y) = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Example of Sharpening usinga Laplacian filter kernel:

$$h(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

7.4 LINEAR FILTERING

Linear filtering is a spatial filtering technique where the output pixel value is a linear combination of its neighbouring pixel values, determined by

the filter kernel. Linear filtering is applied in image de-noising, feature extraction, and preprocessing for machine learning tasks.The operation can be expressed mathematically as:

$$g(x,y) = \sum_{i=-m}^{m} \sum_{j=-n}^{n} h(i,j) f(x+i, y+j)$$

where $h\,(i,j)$ is the filter kernel.

Linear filters preserve additive properties, meaning that applying the filter to a sum of images is the same as filtering each image separately and summing the results.

For example, the Fig. 7.1 show the original image, box filtered image and Gaussian image.

Box Filter isused for image smoothing, the kernel is a uniform matrix where each element has the same value, such as $\frac{1}{9}$ for a 3x3 filter.

Gaussian Filter is used for blurring while preserving edges, the weights decrease with distance from the center, following a Gaussian distribution:

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

where σ controls the spread and x and y are the coordinates relative to the center of the kernel.



Figure 7.1. Original Image, Box Filtered Image & Gaussian Image

7.5 LOW PASS FILTER

A low-pass filter (LPF) allows low-frequency components of an image to pass through while attenuating high-frequency components, such as edges and noise. Applications of LPF include noise removal in digital images, reducing aliasing in signal processing, and enhancing image quality by suppressing unwanted high-frequency details.This operation smooths the image by reducing sharp intensity variations.

Mathematically, in the spatial domain, an LPF applies convolution using a kernel like:

$$h(x,y) = \frac{1}{M\ X\ N}\begin{bmatrix} 1 & 1 & .. & 1 \\ 1 & 1 & :: & 1 \\ :: & :: & :: & :: \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

where M and N are the dimensions of the filter.

Example of a 3x3 averaging filter:

$$h(x,y) = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

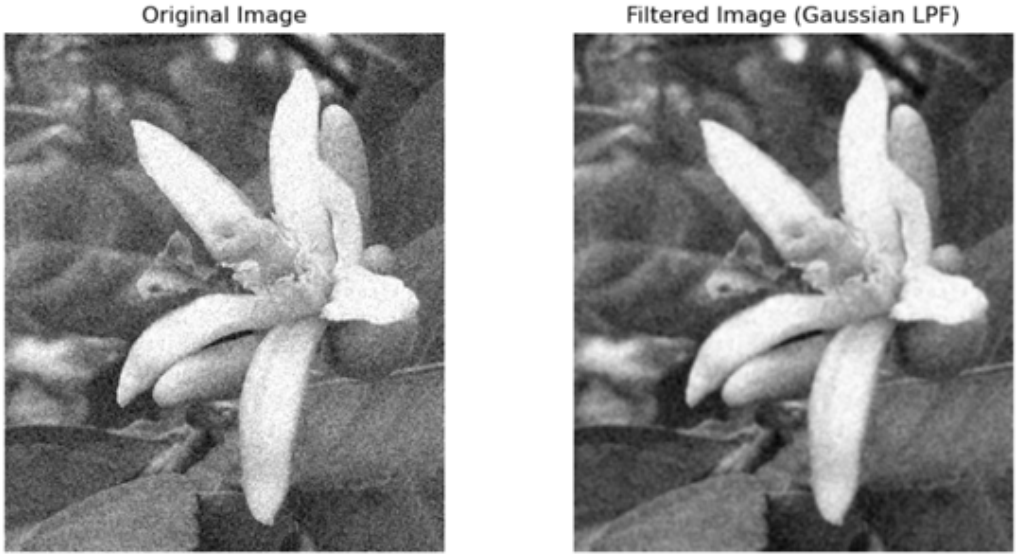


Figure 7.2: Original Image with noise and Filtered Image

In Fig 7.2 a 3x3 kernel filled with equal values (1/9), is applied on a noisy image, ensuring that every pixel in the region is averaged equally. The

function applies the kernel over the image by sliding it across each pixel, performing convolution. The original and filtered images are displayed side by side.

7.6 LIMITATIONS OF LOW PASS FILTER

Low-pass filters, while effective for noise reduction, have significant limitations:

- Loss of Detail: They smooth out high-frequency details, such as edges and fine textures, leading to blurring.
- Edge Distortion: Important image features, like boundaries, are suppressed, making the image appear less sharp.
- Global Nature: Uniform filtering can affect areas with both noise and important features, indiscriminately attenuating high-frequency information.
- Artifacts: When applied in the frequency domain, an improper cutoff frequency may introduce ringing artifacts around sharp transitions.
- Dependency on Kernel Size: Larger kernels provide better noise reduction but at the cost of significant detail loss, while smaller kernels may not sufficiently reduce noise.

7.7 WEIGHTED AVERAGE FILTER

A weighted average filter assigns different weights to pixels in the kernel, emphasizing the center or certain areas while reducing the influence of others. Applications include reducing Gaussian noise and improving smoothing without excessive blurring.The formula is:

$$g(x,y) = \sum_{i=-m}^{m} \sum_{j=-n}^{n} w(i,j) f(x+i, y+j)$$

where (i, j)represents the weights.

For Example, Fig. 7.3 show an original image with Gaussian noise and after applying weighted average filtered imageusing a 3x3 filter emphasizing the center.

$$w(x,y) = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

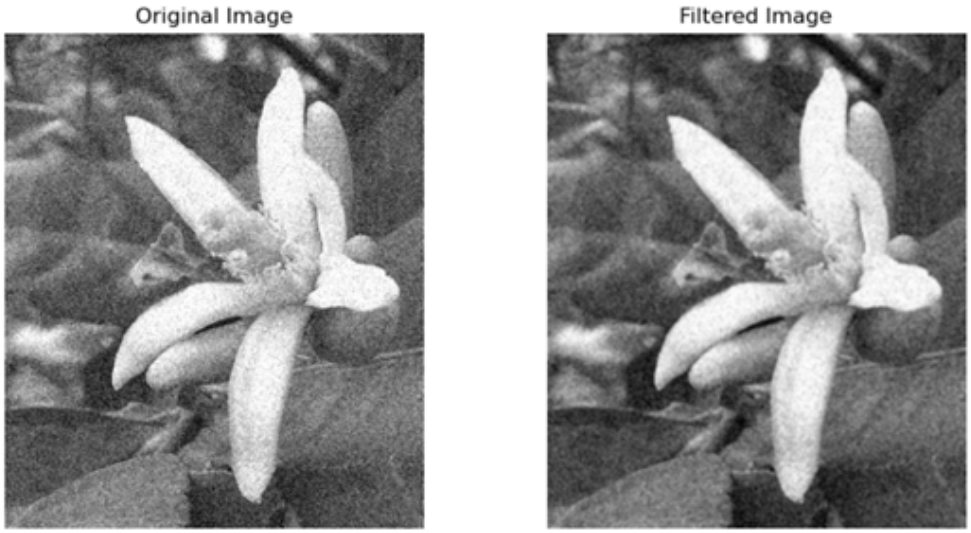


Figure. 7.3: Original Image and Weighted Average Filtered Image

7.8 BARTLETT FILTER

The Bartlett filter, also known as the triangular filter, is a linear smoothing filter often used in image processing to reduce noise while preserving important features like edges. It is named after its triangular shape, which assigns weights to pixels linearly increasing to a peak at the center and then decreasing symmetrically.By offering a weighted average with a triangular distribution, the Bartlett filter balances noise reduction with edge preservation, making it a valuable tool for many image and signal processing tasks. The Fig. 7.3 shows an image with salt and pepper noise and the image after applying Bartlett Filter.

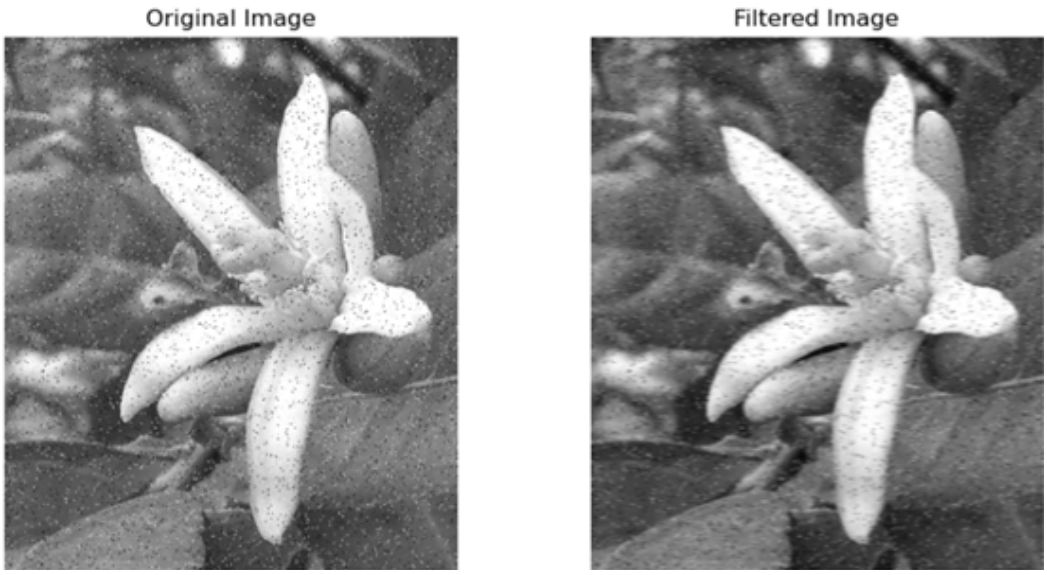


Figure. 7.3: Original Image and Bartlett Filtered Image

For a 1D Bartlett filter with size N, the weights $w(i)$ are given by:

$$w(i) = \begin{cases} 1 - \dfrac{|i|}{\dfrac{N-1}{2}}, & |i| \le \dfrac{N-1}{2} \\ 0, & otherwise \end{cases}$$

where

i represents the index of a filter element, and N is always an odd number to ensure symmetry around the center.

In 2D applications, the Bartlett filter is constructed as a separable filter, meaning the 2D kernel is formed by the outer product of two 1D Bartlett filters:

$$h(x,y)=w(x)\cdot w(y)$$

where $w(x)$ and $w(y)$ are the 1D weights in the horizontal and vertical directions, respectively.

Example of 5x5 Bartlett Filter

The 1D Bartlett weights for N =5 are:

w = [0.2,0.4,0.6,0.4,0.2].

The 2D kernel can then be calculated as:

$$h(x,y) = \begin{bmatrix} 0.04 & 0.08 & 0.12 & 0.08 & 0.04 \\ 0.08 & 0.16 & 0.24 & 0.16 & 0.08 \\ 0.12 & 0.24 & 0.36 & 0.24 & 0.12 \\ 0.08 & 0.16 & 0.24 & 0.16 & 0.08 \\ 0.04 & 0.08 & 0.12 & 0.08 & 0.04 \end{bmatrix}$$

7.9 GAUSSIAN FILTER

A Gaussian filter is a type of linear smoothing filter that reduces noise and detail in an image while preserving edges to some extent. Its kernel weights are derived from the Gaussian function, a bell-shaped curve that assigns higher weights to pixels closer to the center of the filter and smaller weights to those further away. This approach minimizes abrupt transitions in the filtered image, creating a smooth and natural effect.

The Gaussian function in two dimensions is expressed as:

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

where x and y are the coordinates relative to the center of the kernel, σ (sigma) is the standard deviation, which controls the spread or width of the Gaussian function.

The kernel is normalized such that the sum of all weights equals 1, preserving the overall intensity of the image.

For example, a 3×3 Gaussian kernel with σ=1, an approximation is:

$$h(x,y) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

When normalized, the kernel becomes:

$$h(x,y) = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

The Gaussian filter is widely used for firstly in Noise Reduction asit smooths out random variations in pixel values, especially in cases of Gaussian noise. Secondly in pre-processing in Computer Vision, to prepare images for tasks like edge detection and segmentation by reducing detail. And lastly for Blurring Effects to create a soft-focus effect in photography and image editing.

7.10 MEDIAN FILTER

The median filter is a non-linear image processing technique used primarily to reduce noise while preserving edges. Unlike linear filters, which compute a weighted sum of neighbouring pixel values, the median filter replaces each pixel value with the median of the pixel values within a defined neighbourhood or kernel around it.

For each pixel (x) in the image:

- Define a neighbourhood or kernel (e.g., 3×3,5×5 etc.) centered at the pixel.
- Collect all pixel intensity values within the kernel into a list.
- Sort the list of values in ascending order.
- Select the middle value (median) from the sorted list. If the list has an even number of values, the median is the average of the two middle values.
- Replace the center pixel's intensity with the computed median.

Mathematically, for a neighbourhood (x):

$g(x,y)$=median $\{f(i,j)|(i,j)\in N(x,y)\}$,

where

(x,) is the filtered pixel value.

For example, consider a 3×3 neighbourhood:

$$\begin{bmatrix} 12 & 15 & 14 \\ 10 & 11 & 13 \\ 18 & 17 & 16 \end{bmatrix}$$

Sorting the values: {10,11,12,13,14,15,16,17,18}

The median is 14 so the center pixel is replaced by 14.

Applications of Median filter include:

- Noise Reduction: It is effective for removing "salt-and-pepper" noise, where pixels appear as random black or white dots.
- Edge Preservation: Unlike averaging filters, the median filter preserves edges since it does not compute an average of neighboring values, which can blur sharp boundaries.

However median filters are limited by computation intensive for large kernels or high-resolution images and are less effective for Gaussian or other types of noise compared to linear filters.

7.11 HIGH PASS FILTERING

High-pass filtering (HPF) is a technique used in image processing to enhance the high-frequency components of an image, such as edges, fine details, and rapid intensity changes. This filter works by allowing high-frequency information to pass through while attenuating low-frequency components like smooth regions or background details.

In the context of spatial domain filtering, high-pass filtering is typically achieved by subtracting a low-pass filtered version of the image from the original image. The result highlights the areas where the intensity changes rapidly, such as edges.

Mathematically, the high-pass filtering operation can be expressed as:

$g(x,y)=f(x,y)-(f(x,y)*h_{\text{low-pass}}(x,y))$,

Where $(x,)$ is the high-pass filtered image. $(x,)$ is the original image.

$h_{\text{low-pass}}(x, y)$ is the low-pass filter kernel applied to the image.

The symbol " denotes convolution.

Alternatively, a high-pass filter kernel $h_{\text{high-pass}}(x, y)$ can be designed directly, where the kernel is typically the complement of a low-pass filter kernel. For example, in the frequency domain, high-pass filters are designed by taking the inverse of the low-frequency components of the image.

For example, a common high-pass filter kernel used for image sharpening is the Laplacian filter, which emphasizes rapid intensity changes (edges) by applying a kernel such as:

$$h(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

This kernel is centered on a value of 4 (the center pixel), with the surrounding pixels having values of -1. This filter subtracts the average of the neighbouring pixels from the center pixel, thereby enhancing sharp edges.

Applications of High-Pass Filtering:

- Edge Detection: HPFs are commonly used in edge detection algorithms, such as the Sobel or Prewitt operators, which highlight areas of the image with significant intensity changes. This is crucial in object recognition, image segmentation, and medical imaging.
- Image Sharpening: High-pass filters can be used to sharpen images by emphasizing the edges and fine details, making the image appear clearer and more defined. The sharpened image is obtained by adding the high-pass filtered result to the original image, enhancing sharp transitions.
- Enhancement of Fine Features: In applications like remote sensing or medical imaging, HPFs are used to bring out subtle details in images, making it easier to analyze features that are not immediately apparent.

While high-pass filters are effective for emphasizing edges and fine details, they can also amplify noise in regions where no significant intensity transitions exist. Thus, careful tuning and sometimes combined filtering strategies (e.g., combining HPF with a noise reduction technique) are needed to achieve the desired results.



CHECK YOUR PROGRESS

i. What is the main purpose of spatial filtering in image processing?

a) To reduce noise in the image

b) To enhance specific features of the image

c) To resize the image

d) Both a and b

ii. Which of the following is a characteristic of linear filtering?

a) It only uses the image's pixel value.

b) It applies a kernel to the image to modify pixel values.

c) It does not affect the contrast of the image.

d) It only works with grayscale images.

iii. What is the primary function of a low-pass filter in image processing?

a) To sharpen edges
b) To remove high-frequency noise
c) To highlight fine details
d) To increase contrast

iv. What is one limitation of low-pass filters in image processing?

a) They can blur the image and remove details.
b) They do not work with color images.
c) They enhance sharp edges.
d) They are computationally expensive.

v. Which of the following describes a weighted average filter?

a) It replaces each pixel value with its own value.
b) It replaces each pixel value with the average of its neighbors, weighted by distance.
c) It applies equal weights to all pixels in a neighborhood.
d) It is only used in high-pass filtering.

vi. The Bartlett filter is known for which of the following?

a) Its ability to sharpen images
b) Its use in frequency domain filtering
c) Its triangular shape in the frequency domain
d) Its effectiveness in edge detection

vii. What is a Gaussian filter used for in image processing?

a) To enhance edges
b) To smooth images and reduce noise
c) To perform morphological operations
d) To detect corners

viii. Which of the following best describes a median filter?

a) It averages all the pixel values in a neighborhood.
b) It replaces each pixel with the median value of its neighbors.
c) It applies a low-pass filter to remove noise.
d) It enhances edges in the image.

ix. High-pass filters are typically used to:

a) Enhance edges and fine details
b) Remove high-frequency noise
c) Blur the image
d) Create a uniform appearance across the image

x. What is the main advantage of using a median filter over a linear filter?

a) It is faster to compute
b) It works better for Gaussian noise
c) It enhances sharp features in the image
d) It does not blur edges as much as linear filters

xi. What is the behavior of a high-pass filter when applied to an image?

a) It removes edges and fine details
b) It enhances low-frequency regions of the image
c) It highlights edges and fine textures
d) It reduces noise

xii. A weighted average filter can be considered a type of:

a) Low-pass filter
b) High-pass filter
c) Median filter
d) Nonlinear filter



7.12 LET US SUM UP

- Spatial Filteringinvolves modifying an image using a filter or mask that is applied to pixel neighbourhoods, typically to enhance features like edges or reduce noise.
- Linear Filtering method applies a linear transformation to the image by computing a weighted sum of neighbouring pixels. It is often used in edge detection and smoothing processes.
- Low Pass Filter allows low-frequency components (such as smooth gradients and overall structure) to pass through while attenuating higher-frequency noise, useful for noise reduction.
- Low Pass Filters are effective for noise reduction, they can blur the image and cause loss of fine details, particularly high-frequency components such as edges and textures.
- In Weighted Average Filter each pixel is averaged with its neighbours, but with different weights assigned to each pixel, helping to retain more relevant details while smoothing the image.
- Bartlett Filter is a simple low-pass filter that applies a triangular window, which can provide moderate smoothing but is less effective at preserving sharp edges compared to other filters like Gaussian.
- Gaussian Filter is a widely used filter in image processing that applies a Gaussian function to smooth the image. It is particularly effective at reducing noise while maintaining the image structure more clearly than other filters.
- Median Filter: A nonlinear filter used to reduce noise in an image, particularly useful for removing salt-and-pepper noise. It works by replacing each pixel with the median value from its neighbourhood.
- High Pass Filteringtechnique emphasizes high-frequency components in an image, enhancing edges and fine details while reducing smooth areas. It is often used for edge detection.
- Low-pass filters are commonly used for applications such as blurring and noise reduction, but their tendency to blur edges can limit their usefulness in tasks that require fine detail preservation

- The Bartlett filter is simpler and computationally cheaper but less effective than the Gaussian filter for tasks requiring precision in smoothing while preserving sharp transitions.
- The median filter is particularly good at removing outliers and preserving edges, unlike linear filters that might blur important features.
- High-pass filters highlight rapid changes in intensity, helping detect edges, whereas low-pass filters smooth out such changes, often making high-pass filters better suited for feature extraction and edge detection



7.13 FURTHER READING

- Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods, Publisher: Pearson, 4th Edition (2018).
- Computer Vision: Algorithms and Applications by Richard Szeliski, Publisher: Springer, 1st Edition (2010).
- Digital Image Processing and Analysis by J. R. Parker, Publisher: Society of Photo-Optical Instrumentation Engineers, 1st Edition (1997).
- Image Processing: The Fundamentals by Maria M. P. van Loon and Johan G. S. van der Meer, Publisher: CRC Press, 1st Edition (2017).



7.14 ANSWERS TO CHECK YOUR PROGRESS

i. d	ii. b	iii. b	iv. a
v. b	vi. c	vii. b	viii. b
ix. a	x. d	xi. c	xii. a



7.15 MODEL QUESTIONS

Q 1: What is spatial filtering in image processing?

Q 2: How does linear filtering apply to image processing tasks?

Q 3: What are the types of linear filters used in image enhancement?

Q 4: Describe the process of applying a filter in the spatial domain.

Q 5: What is the significance of the convolution operation in linear filtering?

Q 6: What are some common examples of linear filters in image processing?

Q 7: How do boundary conditions affect the outcome of spatial filtering?

Q 8: Explain how convolution is performed in the context of spatial filtering.

Q 9: What role does kernel size play in spatial and linear filtering?

Q 10: What is a low-pass filter in image processing?

Q 11: How does a low-pass filter affect high-frequency components of an image?

Q 12: In what scenarios would a low-pass filter be beneficial for image enhancement?

Q 13: What are the limitations of a low-pass filter in terms of image detail preservation?

Q 14: Why do low-pass filters blur images, and how can this be controlled?

Q 15: How does a weighted average filter differ from a standard averaging filter?

Q 16: Describe the Bartlett filter and its use in image smoothing.

Q 7: What are the advantages of using a Bartlett filter over other types of filters?

Q 18: How is the Bartlett window applied in signal processing, and what are its key features?

Q 19: What are the trade-offs between using a weighted average filter and a simple averaging filter in image enhancement?

Q 20: What is a Gaussian filter, and how does it work in image smoothing?

Q 21: Why is the Gaussian filter preferred over other filters for noise reduction?

Q 22: Explain the difference between Gaussian and other low-pass filters like the Bartlett filter.

*****

UNIT 8: IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

UNIT STRUCTURE

8.1 Learning Objectives

8.2 Introduction

8.3 Low Pass Filtering in Frequency Domain

8.4 High Pass Filter in Frequency Domain

8.5 Butterworth High Pass Filter

8.6 Homomorphic Filter

8.7 Let Us Sum Up

8.8 Further Reading

8.9 Answers to Check Your Progress

8.10 Model Questions

8.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- grasp the concepts of low-pass and high-pass filtering in the frequency domain.
- learn how to analyze signals and images using frequency representations.
- understand its applications in image smoothing and noise reduction.
- study the characteristics of the Butterworth high-pass filter, including its gradual frequency cutoff and smooth transition.
- understand the principle of homomorphic filtering to separate illumination and reflectance components in images.

8.2 INTRODUCTION

In image and signal processing, filtering in the frequency domain is a fundamental approach used to enhance, modify, or extract information from data. The frequency domain representation of signals allows for more efficient manipulation of specific frequency components. Image

enhancement in the frequency domain involves modifying the frequency components of an image to improve its visual quality or to make certain features more prominent. This technique utilizes the Fourier Transform to shift from the spatial domain to the frequency domain, where the image is represented by its frequency components. After applying enhancements, the image is transformed back to the spatial domain using the inverse Fourier Transform.

Let us understand how image enhancement works in the frequency domain. The Fourier Transform decomposes an image into sinusoidal functions of different frequencies. In the context of image processing, this means breaking the image into high and low-frequency components. High frequencies correspond to rapid changes in intensity, such as edges and fine details. Low frequencies correspond to smooth regions, where intensity changes slowly.

The transformation to the frequency domain can be represented by the following formula:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-j2\pi(ux+vy)} dxdy,$$

Where $(x,)$ is the image in the spatial domain. $(u,)$ is the frequency-domain representation. u and v represent frequency components in the x and y directions.

The Inverse Fourier Transform (IFT) is used to convert the frequency-domain image back to the spatial domain:

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, e^{j2\pi(ux+vy)} dudv,$$

Among the key types of frequency domain filters are low-pass filters, high-pass filters, Butterworth high-pass filters, and homomorphic filters, each serving distinct purposes in the processing pipeline.

Low-pass filtering is an essential technique for preserving the lower frequency components of a signal while attenuating its higher frequency components. This is particularly useful for reducing noise, which typically resides in higher frequencies, or for smoothing signals and images. In the frequency domain, this is achieved by suppressing the magnitude of the

high-frequency components while allowing the low-frequency components to remain largely unaffected.

In contrast, high-pass filters focus on retaining the higher frequency components of a signal while attenuating its lower frequency components. This is particularly useful in applications requiring the enhancement of edges and fine details in an image, as these features are represented by high-frequency components. High-pass filters are crucial in edge detection, texture analysis, and sharpening images, providing a means to emphasize subtle changes in intensity or color.

The Butterworth high-pass filter is a specific type of high-pass filter characterized by a smooth and monotonic frequency response. Unlike ideal filters, which have an abrupt cutoff, Butterworth filters transition gradually from passband to stopband, minimizing artifacts such as ringing in the processed signal. This makes them particularly advantageous in applications requiring both precision and smooth transitions, such as medical imaging, audio processing, and computer vision.

Homomorphic filtering represents a sophisticated approach combining spatial and frequency domain techniques to separate multiplicative components of a signal. This is particularly effective for applications like illumination normalization and image enhancement, where signals contain both low-frequency components (illumination) and high-frequency components (reflectance). By transforming the multiplicative model into an additive one using a logarithmic transformation, homomorphic filtering enables independent manipulation of these components.

8.3 LOW PASS FILTERING IN FREQUENCY DOMAIN

Low-pass filtering in the frequency domain is a technique used to suppress high-frequency components of an image while preserving the low-frequency components. High frequencies in an image correspond to rapid changes in intensity, such as noise or sharp edges, while low frequencies represent smoother variations or overall structure. Applications of low pass filtering in frequency domain includes noise reduction by

smoothing out high-frequency noise while retaining the general structure of the image. And blurring in image preprocessing for tasks like object detection or segmentation to simplify input data.

To apply Low-Pass Filtering following steps are followed:

- Fourier Transform: First, the image is converted from the spatial domain to the frequency domain using the 2D Fourier Transform. This step decomposes the image into its frequency components.
- Filtering Operation: A low-pass filter mask is applied to the frequency spectrum. This mask is designed to allow low-frequency components (close to the origin in the frequency domain) to pass through while attenuating or blocking high-frequency components (far from the origin).
- Inverse Fourier Transform: After filtering, the frequency-domain data is transformed back into the spatial domain using the Inverse Fourier Transform. The result is a smoothed version of the original image.

Mathematical a Low-pass filter in the frequency domain is represented using the filtered image $G(u, v)$ in the frequency domain is obtained by multiplying the original image's Fourier Transform $F(u, v)$) with a low-pass filter $H(u, v)$ as:

$G(u,v)=F(u,v)\cdot H(u,v)$,

where $H(u, v)$ is the filter's transfer function.

Ideal Low-Pass Filter:

An ideal low-pass filter passes all frequencies below a specified cutoff and blocks those above it. The transfer function is given by:

$$H(u,v) = \begin{cases} 1, & D(u,v) \le D_0 \\ 0, & D(u,v) > D_0 \end{cases}$$

Where $D(u, v)$ is the distance from the frequency domain origin, calculated as:

$$D(u,v) = \sqrt{(u-u_c)^2 + (v-v_c)^2}$$

Where (u_c, v_c) being the center of the frequency domain.

8.4 HIGH PASS FILTER IN FREQUENCY DOMAIN

High-pass filtering in the frequency domain is a method used to enhance the high-frequency components of an image, such as edges, fine textures, and rapid intensity variations, while suppressing low-frequency components like smooth or uniform areas. This technique is widely used in image sharpening, edge detection tasks and feature enhancement. Image Sharpening improves clarity by emphasizing high-frequency details. Edge detection enhances edges by retaining rapid intensity changes, critical for object recognition and segmentation tasks, while feature enhancement highlights fine structures and textures in medical or satellite images.

In the frequency domain, high-frequency components are located at the outer regions of the spectrum, while low-frequency components are concentrated near the center. A high-pass filter (HPF) selectively retains these high-frequency components, attenuating the rest.

To apply High-Pass Filtering following steps are followed:

- Fourier Transform: Convert the image from the spatial domain to the frequency domain using the 2D Fourier Transform:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})}$$

  where $f(x, y)$ is the image in the spatial domain, and $F(u, v)$ represents its frequency spectrum.

- Apply High-Pass Filter: Multiply the frequency spectrum $F(u, v)$ by a high-pass filter mask $H(u, v)$ designed to suppress low frequencies while allowing high frequencies:

$$G(u,v)=F(u,v)\cdot H(u,v).$$

  For an ideal high-pass filter, the mask is defined as:

$$H(u,v) = \begin{cases} 0, & D(u,v) \le D_0 \\ 1, & D(u,v) > D_0 \end{cases}$$

  where $D(u, v)$ is the distance from the origin of the frequency domain, and D_0 is the cutoff frequency.

- Inverse Fourier Transform: Transform the modified frequency spectrum back to the spatial domain using the inverse 2D Fourier Transform to obtain the filtered image.

A commonly used high-pass filter is the Gaussian high-pass filter, where the mask is defined as:

$$H(u,v) = 1 - e^{-\frac{D(u,v)^2}{2D_0^2}}$$

ensuring a smooth transition that avoids abrupt changes in the frequency spectrum.

8.5 BUTTERWORTH HIGH PASS FILTER

The Butterworth High Pass Filter (BHPF) is a widely used frequency-domain filter designed to pass high-frequency components while attenuating low-frequency components of an image. Unlike an ideal high-pass filter, which creates abrupt transitions, the Butterworth filter offers a smoother cutoff, minimizing artifacts such as ringing. Characteristics of BHPF
is firstly smoother transition, unlike ideal high-pass filters, which sharply separate frequencies, the BHPF provides a gradual transition, reducing abrupt intensity changes in the processed image. Secondly, order dependency, the order n dictates the steepness of the filter's frequency response. A higher n allows for better high-frequency enhancement but may introduce slight distortions.
The transfer function of the Butterworth high-pass filter is defined as:

$$H(u,v) = \frac{1}{1 + \left(\frac{D_0}{D(u,v)}\right)^{2n}},$$

where:

- $(u,)$ is the filter's response at frequency coordinates (u, v).
- $D\,(u, v)$ is the Euclidean distance from the origin of the frequency domain, calculated as:

  $$D(u,v) = \sqrt{(u-u_0)^2 + (v-v_0)^2}$$

  where (u_0,v_0)represents the center of the frequency domain.

- D_0 is the cut-off frequency, determining the transition point between low and high frequencies.
- n is the order of the filter, which controls the sharpness of the transition; higher values of n result in steeper transitions.

8.6 HOMOMORPHIC FILTER

The homomorphic filter is a technique used in the frequency domain to enhance images by separating and processing their illumination and reflectance components. This method is particularly useful for correcting uneven lighting and improving the details in images with significant brightness variations. Homomorphic filtering effectively reduces uneven illumination while enhancing finer details, making it invaluable for applications requiring consistent brightness and improved visibility of features.

An image $(x,)$ can be expressed as the product of its illumination $i\,(x, y)$ and reflectance $r\,(x, y)$:

$f(x,y)=i(x,y)\cdot r(x,y)$,

where:

- $(x,)$: Represents the low-frequency components (illumination variations).
- $(x,)$: Represents the high-frequency components (details and texture).

Since the illumination and reflectance are multiplicative, the homomorphic filter uses a logarithmic transformation to convert this into an additive relationship:

$$\log(f(x,y)) = \log(i(x,y)) + \log\big(r(x,y)\big).$$

To apply Frequency Domain Processing homomorphic filter, following steps are followed:

- **Log Transformation:** Apply a logarithmic transformation to the image to separate the multiplicative components into additive terms.
- **Fourier Transform:** Perform a 2D Fourier Transform to convert the image into the frequency domain.

- **Filtering:** Apply a high-pass filter to suppress the low-frequency components (illumination) and enhance the high-frequency components (reflectance).
- **Inverse Fourier Transform:** Transform the filtered image back to the spatial domain using the inverse Fourier Transform.
- **Exponential Transformation:** Revert the log-transformed image to its original scale using an exponential operation.



CHECK YOUR PROGRESS

i. What is the primary purpose of a low-pass filter in the frequency domain?

a) Attenuate low frequencies
b) Enhance high frequencies
c) Allow low frequencies to pass while attenuating high frequencies
d) Block all frequencies

ii. Which frequency components are preserved in a low-pass filter?

a) High frequencies
b) Low frequencies
c) Zero frequencies
d) All frequencies equally

iii. What is the purpose of a high-pass filter in the frequency domain?

a) Suppress low frequencies
b) Amplify low frequencies
c) Pass low frequencies while attenuating high frequencies
d) Block all frequencies

iv. Which of the following is a characteristic of an ideal high-pass filter?

a) Passes low frequencies perfectly
b) Rejects high frequencies completely
c) Passes high frequencies perfectly
d) Has a linear phase response

v. What is the Fourier transform's role in frequency-domain filtering?
   a) To compute the convolution
   b) To decompose signals into frequency components
   c) To amplify high frequencies
   d) To shift frequencies

vi. What is a common application of high-pass filtering?
   a) Image smoothing
   b) Edge detection
   c) Noise suppression
   d) Brightness enhancement

vii. What is the defining characteristic of a Butterworth filter?
   a) Linear phase response
   b) Maximally flat magnitude response in the passband
   c) Sharp cutoff transition
   d) Exponential decay in the stopband

viii. Which of the following is a common application of Butterworth high-pass filters?
   a) Noise smoothing
   b) Bandwidth compression
   c) Motion blur removal
   d) Image sharpening

ix. What is the main objective of homomorphic filtering?
   a) Sharpen edges in images
   b) Enhance the dynamic range of images
   c) Separate multiplicative and additive components of a signal
d) Suppress high frequencies

x. In homomorphic filtering, the logarithmic transformation is used to convert multiplication into what operation?
   a) Subtraction
   b) Division
   c) Addition
   d) Exponentiation

xi. Which component is typically suppressed during homomorphic filtering?
   a) Low-frequency illumination variations
   b) High-frequency illumination variations
   c) High-frequency reflectance components
   d) Noise components

xii. What is the primary difference between low-pass and high-pass filters?
   a) High-pass filters suppress low frequencies, and low-pass filters suppress high frequencies.
   b) High-pass filters amplify all frequencies, and low-pass filters attenuate all frequencies.
   c) Both suppress high frequencies.
   d) Both suppress low frequencies.

xiii. Which type of filter is more robust to noise: high-pass or low-pass?
   a) Low-pass filter
   b) High-pass filter
   c) Both are equally robust
   d) Neither is robust

xiv. What mathematical tool is essential for frequency domain filtering?

a) Laplace transform
b) Fourier transform
c) Z-transform
d) Wavelet transform



8.7 LET US SUM UP

- Low-pass filtering in the frequency domain is a technique used to retain low-frequency components of an image or signal while attenuating high-frequency components.
- Low-pass filtering is often applied to remove noise or smooth an image.
- Low-pass filtering process involves applying a low-pass filter function to the Fourier transform of the image, then performing an inverse Fourier transform to obtain the filtered result.
- High-pass filtering in the frequency domain emphasizes high-frequency components while suppressing low-frequency components.
- High-pass filtering method is used to highlight edges, fine details, or sudden changes in an image. Similar to low-pass filtering, it involves applying a high-pass filter to the Fourier-transformed data and then reconstructing the signal or image with an inverse Fourier transform.
- The Butterworth high-pass filter is a specific type of high-pass filter characterized by its smooth transition between passband and stopband.
- The Butterworth filter achieves a gradual attenuation, making it less prone to ringing artifacts. It is defined by an order parameter and a cutoff frequency, which determine the sharpness and range of the filtering effect.
- Homomorphic filtering is a technique used in image processing to separate illumination and reflectance components of an image.

- Homomorphic filtering involves logarithmic transformation, frequency domain filtering, and exponential transformation.
- Homomorphic filtering method is particularly effective for enhancing images with non-uniform lighting or correcting uneven illumination, as it helps to normalize brightness while preserving details.



8.8 FURTHER READING

- Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods, Publisher: Pearson, 4th Edition (2018).
- Computer Vision: Algorithms and Applications by Richard Szeliski, Publisher: Springer, 1st Edition (2010).
- Digital Image Processing and Analysis by J. R. Parker, Publisher: Society of Photo-Optical Instrumentation Engineers, 1st Edition (1997).
- Image Processing: The Fundamentals by Maria M. P. van Loon and Johan G. S. van der Meer, Publisher: CRC Press, 1st Edition (2017).



8.9 ANSWERS TO CHECK YOUR PROGRESS

i,c	ii, b	iii, a	iv, c
v, b			
vi, b	vii, b	viii, d	ix, c
x, c			
xi, a	xii, a	xiii, a	xiv, b



8.10 MODEL QUESTIONS

Q 1: What is the principle of low-pass filtering in the frequency domain, and what are its primary applications in image processing?

Q 2: Explain the mathematical operation performed by a low-pass filter on the Fourier transform of an image.

Q 3: What is the effect of applying an ideal low-pass filter to an image?

Q 4: How does high-pass filtering in the frequency domain enhance the edges and fine details in an image?

Q 5: Discuss the differences between ideal, Butterworth, and Gaussian high-pass filters in terms of sharpness and selectivity.

Q 6: What are some typical applications of high-pass filtering in image and signal processing?

Q 7: What is the mathematical form of a Butterworth high-pass filter, and how is it parameterized?

Q 8: Compare the Butterworth high-pass filter with the ideal high-pass filter in terms of performance and practical usability.

Q 9: Why is the Butterworth filter considered smoother than an ideal filter, and what advantages does this provide in practical applications?

Q10: Provide an example of how a Butterworth high-pass filter can be implemented in the frequency domain using an image.

Q11: What is homomorphic filtering, and how is it applied to enhance the dynamic range of an image?

Q12: Explain how a homomorphic filter can be designed as a combination of high-pass and low-pass filters.

Q13: In what scenarios is homomorphic filtering particularly useful in image processing?

*****

UNIT 9: FUNDAMENTALS OF IMAGE

UNIT STRUCTURE

9.1 Learning Objectives

9.2 Introduction to Image Restoration

9.3 Image Degradation

9.3.1 Types of Image degradation

9.4 Classification of Image Restoration Technique

9.4.1 Unconstraint Method

9.4.2 Constraint Method

9.5 Image Restoration Model

9.6 Noise Modelling

9.6.1 Noise Categories Based on Distribution

9.6.2 Noise Categories Based on Correlation

9.6.3 Noise Categories Based on Nature

9.6.4 Noise Categories Based on Source

9.7 Restoration Filter

9.7.1 Inverse Filter

9.7.2 Pseudo Inverse Filter

9.7.3 Wiener Filter

9.7.4 Constrained Least Square Filters

9.8 Iterative Image Restoration Method

9.9 Blind Image Restoration Method

9.10 Geometrical Transformation for Image Restoration

9.11 Let Us Sum Up

9.12 Further Reading

9.13 Answer to Check Your Progress

9.14 Model Questions

9.1 LEARNING OBJECTIVES

This chapter provides an overview of digital image restoration. It introduces the concepts related to degradation model and image restoration

techniques. After studying this chapter, the reader will become familiar with the following:

- Image restoration
- Degradation and its types
- Image restoration model
- Noise models and noise removal filters
- Iterative method
- Blind image restoration and Geometrical for image transformation

9.2 INTRODUCTION

Image restoration refers to the process of extracting clean and beneficial information from noisy and blurred input images. It involves various techniques such as filtering, denoising, and hybrid sparsity learning to restore degraded images to their original state, improving accuracy and preserving details. This is crucial in applications such as astronomical imaging, medical imaging, digital media restoration, and law enforcement, where images may be corrupted by atmospheric noises, camera sensors, and other factors.

9.3 INTRODUCTION TO IMAGE DEGRADATION

No image is perfect. An image gets corrupted by different types of degradation that appear from the stage of image acquisition itself. Images are degraded by noise, blurs, and distortions or artefacts. The process of image degradation takes place in all the stages – image acquisition, image processing, image storage, and transmission. Noise is a disturbance that causes fluctuations in pixel values. Similarly, when an image-capturing system does not capture a point as a point, it results in degradation. Artefacts are extreme intensity and color imperfections that make images meaningless. Thus, image formation itself introduces problems.

Image restoration is the process of retrieving an original image from a degraded image. The idea is to obtain an image as close to the original image as possible. This is possible by removing or minimizing degradations. This is often difficult in case of extreme noise and blurs, and is often called an inverse problem. An inverse problem aims to find the cause from effects.

Often, inverse problems are ill-posed problems, Jacques Hadamard defined a well-posed problem as one that has a unique solution and its behavior changes continuously with the initial condition. An image restoration problem is not a well-posed problem; hence there is no guarantee of a solution, and even if a solution is found, it can be unusable. Therefore, additional assumptions are made about the degraded image and incorporated as a smoothness function. This process is called regularization, which makes recovery of the original image feasible.

9.3.1 Types of Image Degradation

To remove image degradation, it is necessary to understand them. Degradation is of three types and are shown in Fig. 9.1.

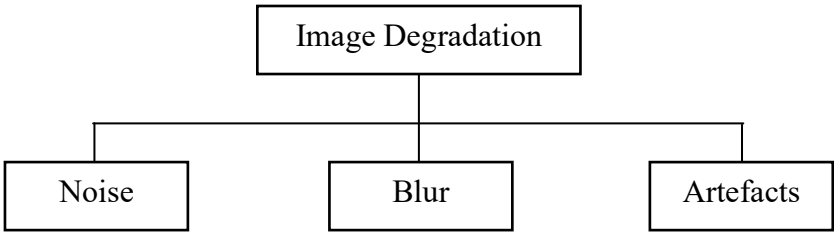


Fig. 9.1: Types of image degradation

1) Noise

Noise is a disturbance that causes fluctuations in pixel values. Hence, pixel values show random variations, and this cannot be avoided. Thus, suitable strategies should be designed to model and manage noise. An original image is shown in Fig. 9.2(a). Gaussian noise is added to it, and the resultant image is shown in Fig. 9.2(b).

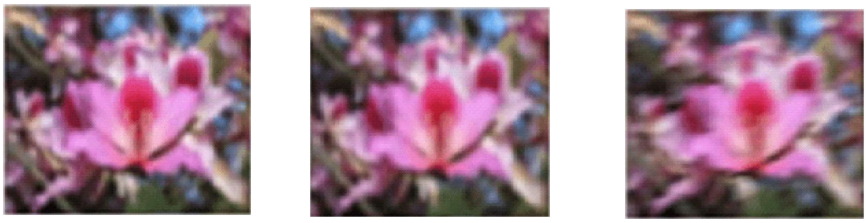


(a) Original image (b) Gaussian noise

Fig. 9.2: (a) Original image (b) Resultant image after adding Gaussian noise

2) Blurs

A blur is a degradation that makes an image less clear, thus making the process of image analysis difficult. Some of the common blurs are Gaussian blur and motion blur. Gaussian blur is the application of a mathematical function to an image in order to blur it. These blurs are as part of image smoothing operation to reduce the noise level. A motion blur occurs because of the movement of the object or camera during the image acquisition process. Gaussian and motion blurs are illustrated in Fig. 9.3(b) and 9.3(c), respectively, for the original image shown in Fig. 9.3(a). Similarly, a lens-type blur is shown in Fig. 9.3(d).



(a) Original image (b) Gaussian blur (c) Motion blur

Fig. 9.3: (a) Original image (b) Gaussian blur (c) Motion blur

3) Artefacts

Distortions or artefacts (the term artefact is mostly used in medical imaging) are extreme intensity or colour fluctuations such as translation, rotation, or change in scale. A distorted image is shown in Fig. 9.4, where it can be observed that the car in the image is slightly elongated. In medical images, an artefact is not a random noise. It is a spurious error or some systematic error that is present in the image due to the faulty image-capturing process or due to the properties of the image object. Artefacts can be mistaken for real organs, which may mislead doctors. The misunderstanding of X-ray films (e.g. scratches), movement of patients during the scanning or importer following of radiological procedures, and dust in the

imaging system can result in systematic errors in image called artefacts.

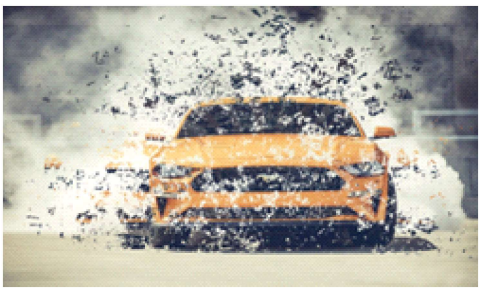


Fig. 9.4: Distorted image



CHECK YOUR PROGRESS 1

1. What is image restoration?
2. Name the types of image degradations?
3. Degradation can be estimated by _____, _____, and _____. (Fill up the blanks)
4. What is motion blur?

9.4 CLASSIFICATION OF IMAGE RESTORATION TECHNIQUE

The different types of image restoration techniques are shown in Fig. 9.5.

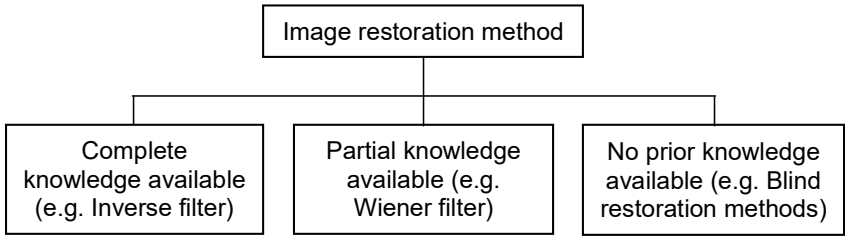


Fig. 9.5: Types of image restoration methods

Based on the knowledge available for the blurring function, the following three scenarios are possible:

1. Complete knowledge of the blurring function is available. This is a simple case in which an original image can be retrieved by applying an inverse filter. This approach is called deconvolution or inverse filtering.

2. Only partial knowledge of the blurring function ***H*** is available. A wiener filter is helpful in this scenario.
3. There is no knowledge of the blurring function. Approaches like blind restoration techniques are used in this scenario. Blind restoration (or blind convolution) is a procedure where an attempt is made to deconvolve the degraded image without any prior knowledge of the cause of degradation.

Let us first review some of the fundamental of image restoration algorithms. Algebraic method is very popular for image restoration, as it uses the concepts of matrices and linear algebra instead of those of integrals. The two types of algebraic methods are as follows:

1. Unconstrained method.
2. Constrained method.

9.4.1 Unconstrained Method

As the name suggests, the unconstrained approach has no constraints. The basic approach in the unconstrained method is that the approximate image $(\hat{f})$ of the original image f is recovered by minimizing the noise. The general image degradation model is given as follows:

$$g = Hf + n$$

Therefore, $n = g - Hf$

In the absence of any knowledge regarding n, it is desirable to seek $\hat{f}$ such that $H\hat{f}$ approximates g in the least square sense. The norm of $\|n\|^2$ is given by $n^T n$.

Therefore,

$\|n\|^2 = \left\|g - H\hat{f}\right\|^2$ where $\left\|g - H\hat{f}\right\|^2 = (g - H\hat{f})^T(g - H\hat{f})$

This can be equivalently stated as a minimization problem and can be written with respect to $\hat{f}$, as an image restoration minimization problem, as $J(\hat{f}) = \left\|g - H\hat{f}\right\|^2$. Minimization involves differentiating and setting $\hat{f}$ to zero. Therefore, minimization of gives the following relation:

$$\frac{\partial J(\hat{f})}{\partial \hat{f}} = 0$$

$$-2H^T\left(g - H\hat{f}\right) = 0$$

$$-2H^T g + 2H^T H\hat{f} = 0$$

This simplifies to the following relation:

$$HH^T\hat{f} = H^T g$$

Further simplification leads to the following equation:

$$\hat{f} = (H^T H) = (H^T H)^{-1} H^T g = H^{-1} g$$

The expression $f = H^{-1}g$ is used by an inverse filter for implementing a simple degradation.

9.4.2 Constrained Method

Constraint restoration method is another image restoration process that models image restoration as a minimization function of the form

$$\left|\underline{O}\hat{f}\right|^2 = |n|^2$$

Unlike the unconstrained restoration method, the focus of the constrained method is to minimize the error between the approximate and the original images. This is equivalent to the following minimization problem:

$$J(\hat{f}) = \left(\left\|\underline{O}\hat{f}\right\|^2 + \alpha\left(\left\|g - H\hat{f}\right\| - \|n\|^2\right)\right)$$

Here, α is called the Lagrange multiplier, $\left\|\underline{O}\hat{f}\right\|^2$ is called the smoothness term or regularization term, and the rest is called the restoration term. This method is called regularization, which is performed to solve ill-posed image restoration problems. The regularization term aims to force g to the closet original image f. The rest of the term enforces the prior knowledge of the image. Using calculus, differentiation of this equation yields the following result:

$$\frac{\partial\left(J\left(\hat{f}\right)\right)}{\partial \hat{f}} = 0$$

$$= 2\underline{O}^T\hat{f} - 2\alpha H^T(g - H\hat{f})$$

Solving this equation for $\hat{f}$, the following relation is obtained:

$$\hat{f} = \left(H^T H + \gamma \underline{O}^T \underline{O}\right)^{-1} H^T g \text{ where } \gamma = \frac{1}{\alpha}$$

This parameter is important and should be turned to satisfy the optimality constraint. In other words, an increase in γ emphasizes the restoration term and decrease emphasizes the smoothness term. It is possible to convert the algebraic solution to the frequency domain by applying Fourier transform.



CHECK YOUR PROGRESS 2

1. What are the different types of image restoration techniques?
2. What are the two popular algebraic method used for image restoration?
3. Write the full form of PSF and OTF?

9.5 IMAGE RESTORATION MODEL

Image restoration starts with an image degradation or restoration model, which is shown in Fig. 9.6.

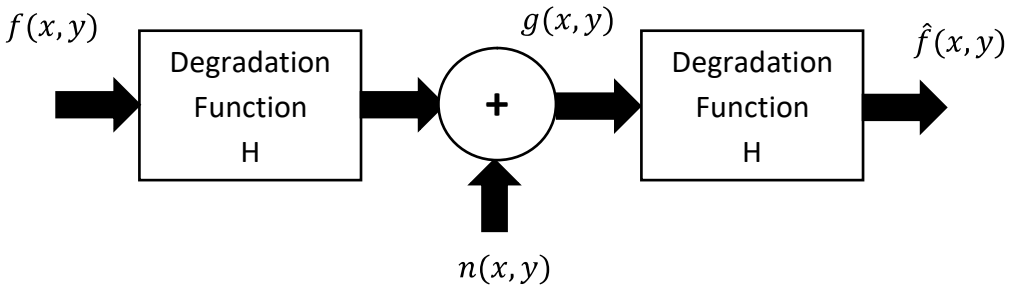


Fig. 9.6: Image restoration model

Let the original image be $f(x, y)$. When the image is acquitted by the imaging system, the process of degradation starts. In a linear space-invariant (LSI) system, degradation is modelled as an operator *H*, which is linear, position-invariant operator and has following properties:

1. It is a linear operator.
2. It obeys the rules of homogeneity.
3. It is a positive invariant.

The degradation function is denoted as $h(x, y)$. Noise is known to affect an image and is denoted as $n(x, y)$. For an LSI system, the degraded image is expresses as follows:

$$g(x, y) = f(x, y) * h(x, y) + n(x, y)$$

Here, the symbol '*' represents the convolution operation. By applying Fourier transform, the equation in the frequency domain becomes as follows:

$$G(u, v) = F(u, v) * H(u, v) + N(u, v)$$

The original image can be retrieved by rearranging this equation to get the following expression:

$$F(u, v) = \frac{G(u, v)}{H(u, v)} - \frac{N(u, v)}{H(u, v)}$$

The original image $f(x, y)$ can be obtained by applying an inverse Fourier transform to $F(u, v)$.

9.6 NOISE MODELLING

First, let us try to categorize and model noise patterns. Some of the frequent noises that are encountered in image processing are categorized based on the criteria of distribution, correlation, nature, and source. This is shown in Fig. 9.7.

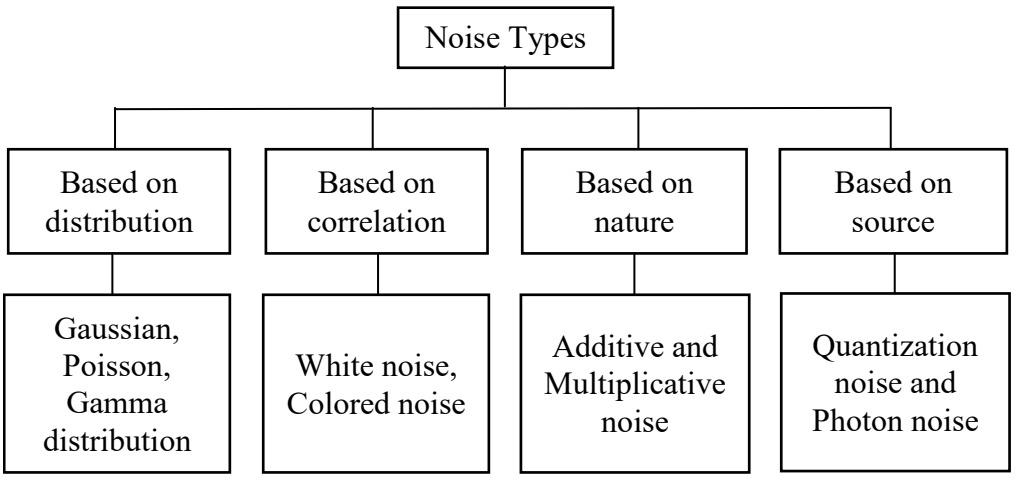


Fig. 9.7: Types of noise

9.6.1 Noise Categories Based on Distribution

Since noise is a fluctuation in pixel values, it is characterized as a random variable. A random variable probability distribution is an equation that links the values of the statistical result with its probability of occurrence. Categorization of noise based on distribution, noise can be classified as follows:

1) Gaussian noise

A random noise that enters a system can be modelled as a Gaussian or normal distribution, which is a bell-shaped curve. Gaussian noise is mathematically denoted as follows:

$$F = f(x, y) \pm N_a$$

where N_a is the Gaussian PDF and $f(x, y)$ is the noiseless image. A sample image and its corresponding Gaussian noise-affected images are shown in Figs. 9.8(a) and 9.8(b), respectively.



(a) Original image (b) Gaussian noise

Fig. 9.8: Gaussian noise (a) original image and (b) image with Gaussian noise (mean=0, variance=0.01)

The Gaussian probability density function (PDF) is given as follows:

$$P(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(z-m)^2}{2\sigma^2}}$$

Here *z* is the grey level of the pixel, and *m* and *í* are the mean and standard deviation of the image, respectively. It has been observed that the error due to Gaussian noise is very small and the average of all errors is 0. Gaussian noise is also known as additive noise and can be modelled as a simple additive process.

2) Impulse or salt-and-pepper noise

This noise is known by various names such as short noise, salt-and-pepper noise, and binary noise. It is caused by a sudden disturbance in the image signal. Mostly this noise is caused by sensor and memory problems due to which pixels are assigned incorrect maximum values. Images corrupted with salt-and-pepper noises of different densities are shown in Fig. 9.9 (a) and (b).



(a) Original image (b) Salt-and-pepper noise

Fig. 9.9: Salt-and-pepper noise (a) image with default density (original image) and (b) image with noise density=0.2

The PDF of impulse noise is given as follows:

$$P(z) = \begin{cases} P_a & for\ z = a \\ P_b & for\ z = b \\ 0 & otherwise \end{cases}$$

Here *z* is the pixel value. Variables *a* and*b* are grey levels. If *b*>*a*, then level *a* will appear as a light spot and vice versa. If and is zero, then the impulsive noise is called unipolar noise. If they are not zero and are approximately equal, the resulting noise is called bipolar noise or salt-and-pepper noise.

3) Poisson noise

Poisson noise manifests as a random structure of texture in images, and is very common in X-ray images. The PDF of Poisson noise is given as follows:

$$P(z) = \frac{(np)^2}{z!} \times e^{-np}$$

Where *n* is the total number of pixels and *p* is the ration of noise pixels to the total number of pixels.

4) Exponential noise

This type of noise occurs in images mostly due to illumination problems. The PDF of exponential noise is given as follows:

$$P(z) = \begin{cases} a \times e^{-az} \; for \; z \geq 0 \\ 0 \qquad otherwise \end{cases}$$

The mean and variance are given as $\frac{1}{a}$ and $\frac{1}{a^2}$, respectively.

5) Gamma noise

Gamma distributed noise also occurs mostly due to illumination problems. The DF of gamma noise is given as follows:

$$P(z) = \begin{cases} \frac{a^b \times z^{b-1}}{(b-1)!} e^{-a^2} & for \; z \geq 0 \\ 0 \qquad otherwise \end{cases}$$

The mean and variance are given as $\frac{b}{a}$ and $\frac{b}{a^2}$ respectively.

6) Rayleigh noise

This type of noise is mostly present in range images. Range images are used in many remote sensing applications where the pixel value indicates the distance between the object and the camera system. The PDF of Rayleigh noise is given as follows:

$$P(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}} \; for \; z \geq 0 \\ 0 \qquad\qquad\qquad otherwise \end{cases}$$

The mean is given as $a + \sqrt{\frac{\pi b}{4}}$ and the variance as $\frac{b(4-\pi)}{4}$.

7) Uniform noise

Uniform noise is another popular noise encountered in images where different values of the noise are equally probable. It occurs due to the quantization process. The PDF of uniform noise is defined as follows:

$$P(z) = \begin{cases} \frac{1}{b-a} \; for \; a \leq f(x,y) \leq b \\ 0 \qquad\qquad\qquad otherwise \end{cases}$$

The mean is $(a+b)/2$ and the variance is $(b-a)^2/12$. Here *a* and *b* are respectively, the minimum and maximum grey-scale values of the image.

9.6.2 Noise Categories Based on Correlation

The statistical dependence among pixels is called a correlation. If pixel is independent of its neighbouring pixels, it is called an uncorrelated pixel; otherwise, it is termed as a correlated pixel. Correlation is a statistical word that implies the degree of association between random variables. The degree of association or the level of dependency can be determined by a covariance matrix. If the elements of the covariance matrix of two random variables, except the diagonal elements, are 0, then the random variables are uncorrelated. Uncorrelated noise is called white noise. Mathematically, for white noise, the noise power spectral density remains constant with frequency.

Coloured noise is more difficult to characterize as its origin is mostly unknown, that is finding the correlated characteristics of coloured noise is difficult because of the involvement of may components the origins of which are uncertain. One such coloured noise is pink noise. Its power spectrum, unlike that of white noise, is not constant but is proportional to the reciprocal of frequency. This is also called of $\frac{1}{f}$ the flicker image.

9.6.3 Noise categories based on nature

Based on the nature of its presence, a noise can be modelled either as an additive or as a multiplicative process.

1) Additive noise

In case of additive noise, an image can be perceived as the image plus noise. This is a linear problem. Thus, the perceived image $g(x,y) = f(x,y) + n(x,y)$.

2) Multiplicative noise

Multiplicative noise can be modelled as a multiplicative process. Speckle noise is one of the most encountered multiplicative noises in image processing. This noise is widely present in medical images, especially ultrasound images. It is a random signal in which the average amplitude increases with the overall signal intensity. It appears as bright specks in the lighter regions of an image. It can be modelled as a pixel value multiplied by the random value. Speckle noise can be modelled as follows:

$$I = f(x, y) + (f(x, y) \times N_g)$$

Where N_g is a random noise having a zero mean Gaussian probability distributive function.

9.6.4 Noise Categories Based on Source

Two of the noise categories based on source frequently encountered in image processing are quantization noise and photon noise. ***Quantization noise*** occurs due to a difference between the actual and the allotted values, and is inherent in the quantization process. ***Photon noise*** occurs due to the statistical nature of the electromagnetic waves. Due to the statistical variation, generation of photons is not constant. This causes variations in photon count, which can be termed as photon noise. Photon noise is present in many medical images, as the pixel value is the count of photons in these images.



CHECK YOUR PROGRESS 3

1. Draw the image restoration model.
2. What is noise? Under what condition, noises can be classified?
3. What are the types of noises based on distribution?
4. What is additive noise?
5. What is photon noise?

9.7 RESTORATION FILTER

Restoration filters are the types of filters that are used for operation of noisy image and estimating the clean and original image. It may consist of processes that are used for blurring or the reverse processes that are used for inverse of blur. Filter used in restoration is different from the filter used in enhancement process. There are three types of restoration filters: inverse filter, pseudo inverse filter, and wiener filter. These are explained below:

9.7.1 Inverse Filter

The process of removing blurs and noise is called deconvolution for inverse filtering. A simple deconvolution starts with an assumption that a blur is characterized by the PSF of the impulsive response of the system. It assumes that most blurs and linear, and the output of the imaging system is the convolution of the impulsive response and the input image. In the presence of noise, the degraded image is given as follows:

$$g(x,y) = f(x,y) * h(x,y) + n(x,y)$$

By applying Fourier transform, this equation yields the following relation:

$$G(u,v) = F(u,v)H(u,v) + N(u,v)$$

Therefore,

$$F(u,v) = \frac{G(u,v)}{H(u,v)} - \frac{N(u,v)}{H(u,v)}$$

If the noise is assumed to be zero, then the deconvolution is equivalent to the following expression:

$$G(u,v) = F(u,v)H(u,v)$$

Therefore,

$$F(u,v) = \frac{1}{H(u,v)} G(u,v)$$

In other words, an inverse filter is the inverse of the degradation function. The inverse filter acts as a high-pass filter, causing blurring and increasing noise. One major problem is that $H(u, v)$ may be zero or closer to zero, and hence, the inverse may result in infinity. Therefore, one possible solution is to design an inverse filter not with the transfer function $\frac{1}{H(u,v)}$, but with the following transfer function:

$$H'(u,v) = \begin{cases} \frac{1}{H(u,v)}, if\ |H(u,v)| \geq \in \\ 1, \qquad if\ |H(u,v)| < \in \end{cases}$$

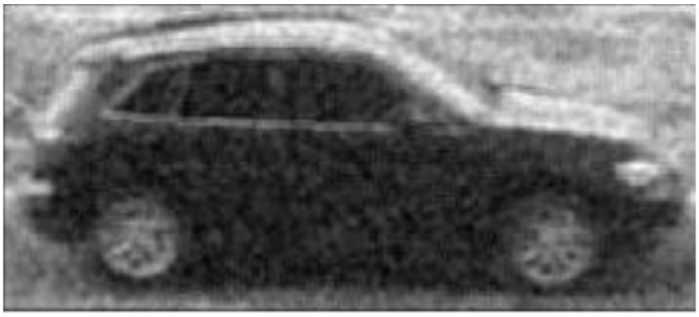
Here, $\in$ is the threshold value. It is used to mitigate the effect of zeros in the degraded function. An original image is shown in Fig. 9.10 (a) and the distorted image is shown in Fig. 9.10 (b). The restored image using an inverse filter is shown in Fig. 9.10 (c).



(a) Original image



(b) Image with distortion



(c) Restored image with inverse filter

Fig. 9.10: Application of inverse filter (a) Original image (b) image with distortion (c) Restored image using inverse filter.

9.7.2 Pseudo Inverse Filter

Pseudo inverse filter is the modified version of the inverse filter and stabilized inverse filter. Pseudo inverse filtering gives more better result than inverse filtering but both inverse and pseudo inverse are sensitive to noise. Pseudo inverse filtering is defined as:

$$H'(u,v) = \begin{cases} \dfrac{1}{H(u,v)}, & if\ H(u,v) \neq 0 \\ 0, & otherwise \end{cases}$$

9.7.3 Wiener Filter

When partial knowledge of the degradation function is available, Weiner filters can be used for image restoration. Let us assume that the image has additive while Gaussian noise, and the aim is to minimize the mean square error between the original and the restored image. Therefore, to design a Wiener filter, an estimation of the original image and the additive noise is required. In other words, this method requires prior knowledge of the power spectra of the noise and original image. It can be observed that the spectral power density is a function of power distribution over different frequencies.

Similar to the unconstrained method, a Wiener filter finds an estimate $\hat{f}(x,y)$ of the original image $f(x,y)$ such that the mean square error is minimized. Thus, a Wiener filter is an optimal filter in terms of the

least mean square of the estimated and the original image.It not only restores the image but also removes noise by image smoothing. The minimized error is given as follows:

$$e^2 = E\{f(x,y) - \hat{f}(x,y)^2\}$$

Where $E(\cdot)$ is the expected value. To know the error, the correlation matrices of f and n are required. Let us assume that R_j and R_n are the correlation matrices of f and n respectively. These correlation matrices are denoted in terms of expectation operation, as follows:

$$R_f = E\{f \cdot f^T\}$$

$$R_n = E\{n \cdot n^T\}$$

The use of the unconstrained form and substitution of $Q^+ Q = R_f^{-n} R_n$ give the following estimation:

$$\hat{f} = (H^T H + \gamma R_f^{-1} R_n)^{-1} H^{-1} g$$

This filter in the frequency domain corresponds to the following expression:

$$F(u,v) = \left[\frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma \left|\frac{S_n(u,v)}{S_f(u,v)}\right|}\right] \cdot G(u,v)$$

Where $S_n(u,v)$ and $S_f(u,v)$ are the spectral power densities of the noise and image, respectively. A cruder version of the Wiener filter often works well in situation where the ratio of $\frac{S_n(u,v)}{S_f(u,v)}$ is assumed to be constant K. This equation is given as follows:

$$F(u,v) \cong \left[\frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma K}\right] \cdot G(u,v)$$

In the absence of any knowledge of noise, K can be assumed to be inverse of the SNR of the image, which is averaged over all frequencies. Thus, K is released to the low-frequency aspect of the filter, and if $K = 0$, the filter becomes an inverse filter. The value of γ is very crucial, and it should be tuned so that $\|n\|^2 = \left\|g - H\hat{f}\right\|^2$. If γ=1, the expression simply reduces to a simple inverse filter. The

expression γ=1 represents a Wiener filter, and when γ assumes different values, this leads to a parametric Wiener filter.

9.7.4 Constraint Least Square Filter

The problem with a Wiener filter is that it is necessary to know the power spectrum of the original image and noise. The constrained approach is used for implementing this technique. This technique also tries to enforce a constraint to represent some degree of smoothness so that the resultant image is smooth and noise free. Let Q be represented as follows:

$$Q = min\left\{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\nabla^2 f(x,y)]^2\right\}$$

It can be observed that Q is selected to approximate a Laplacian or second-order derivative of the image, so the problem becomes minimization of Q.

The approximation of the second order derivative is possibly by convolving the image with a mask

$$p(x,y) = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Let $p\ (u,\ v)$ be the Fourier transform of the matrix. Minimization of the second-order derivative of the image subject to the constraint $\left|g - H\hat{f}\right|^2 = |n|^2$ yields a solution $\left|Q\hat{f}\right| = \hat{f}^T Q^T Q\hat{f}$. This leads to a transfer function in the frequency domain, which is given as follows:

$$\hat{F}(u,v) = \left[\frac{H*(u,v)}{|H(u,v)^2| + \gamma|p(u,v)|^2}\right] G(u,v)$$

Here γ is the parameter used to tune the degree of smoothness. Let the residual difference be $r_s = H - g\hat{f}$. Then, $|r_s|^2 = |n|^2 \pm c$, where $|n|^2 = MN(\sigma_n^2 + m_n^2)$ for an optimal filter. Here, M and N are the dimensions of the image, s_n and m_n are the mean and variance of the image, respectively.

For obtaining the optimal filter, the parameter should be tuned. The procedure for tuning is as follows:

1. Specify an initial value of γ.
2. Computer $\hat{f}$ and $|r|^2$.
3. Check whether $R(u,v) = G(u,v) - H(u,v)\hat{F}(u,v)$; if yes then stop.
   If $\|r\|^2 < \|n\|^2$, then increase the value of γ.
   Else, $\|r\|^2 > \|n\|^2 + \alpha$, then decrease the value of .

9.8 ITERATIVE IMAGE RESTORATION METHOD

Unlike the linear image restoration model, the iterative approach produces successive improvements to the restoration process until a termination condition is reached. Van Cittert introduced this filter in 1990 using the following equation:

$$f_{k+1} = f_k + (g - h * f_k)$$

Here, f is the blur kernel, and when $n = f^0$ is the initial guess and the initial observed degraded image; k is a variable that represents the number of iterations. We will discuss a modified procedure here.

The process starts with a guess, and it repeatedly updates the guess until it arrives at a solution by solving a least square solution iteratively. The process starts with an initial guess based on the observed image g.
Let $\hat{f}_0$ be the initial guess and let $\hat{f}_0(x,y) = \gamma g(x,y)$. The important issue here is the selection of γ. For faster convergences, the value of should be selected based on the following condition:

$$|1 - \gamma H(u,v)| < 1$$

Here γ is a positive constant and its value ranges between 0 and 1. Then, the guess is updated as follows:

$$\hat{f}_{k+1}(x,y) = \hat{f}_k(x,y) + \gamma(g(x,y) - \hat{f}_k(x,y) * h(x,y))$$

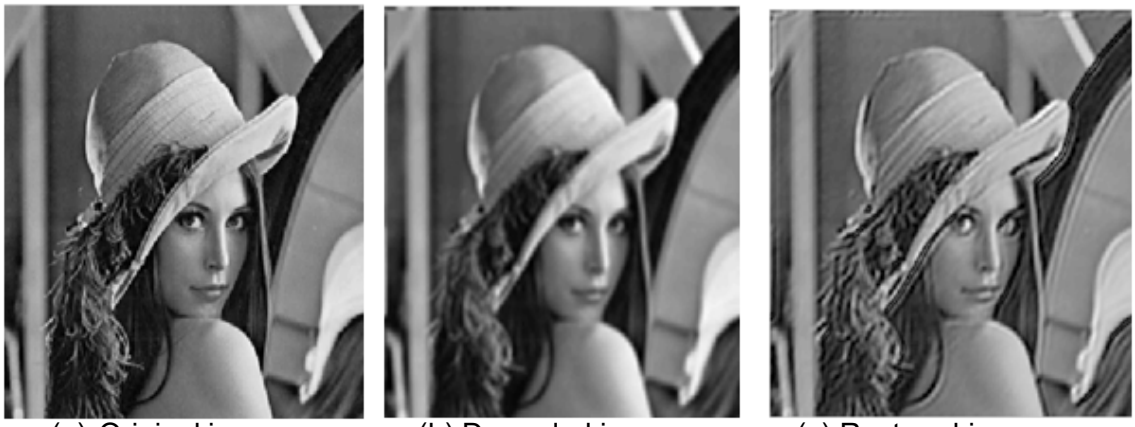
Here the symbol '*' denotes a convolution operation.The idea is that if the approximation $\hat{f}_k$ is correct, then the entire term multiplied by γ becomes zero and the term itself vanishes, leading to convergence. The equation in the frequency domain is given as follows:

$$\hat{F}_{k+1}(u,v) = \hat{F}_k(u,v) + \gamma(G(u,v) - \hat{F}_k(u,v) \times H(u,v))$$

The main advantage of this method is the rapidity of recovering the original image from its degraded image with few assumptions, and it also uses fewer operations. However, this method has some disadvantages, for example, it is sensitive to noise and tends to be unstable as the number of iteration increases.The Lucy-Richardson deconvolution technique is one of the simplest algorithms for iterative image resolution, which is used for restoring astronomical images. The process is illustrated as follows:

$$f_{k+1} = f_k * \left(h * \frac{g}{h * f_h}\right)$$

Here, k is the number of iterations. It can be observed that in every iteration, the method uses the previous degraded image. The Lucy-Richardson method is useful in the presence of noise also. However, the noise increases with an increase in the number of iterations. An original image and the degraded imageare shown in Fig. 9.11 (a) and Fig. 9.11 (b), respectively. The image restored using an iterative filter is shown in Fig. 9.11 (c).



(a) Original image (b) Degraded image (c) Restored image

Fig. 9.11: Application of iterative filter (a) Original image (b) Degraded image (c) Image restored using Lucy-Richardson filter.

9.9 BLIND IMAGE RESTORATION METHOD

All the image restoration techniques that have been discusses so far are based on the premise that either the complete knowledge of the blurring function is available (inverse filter) or partial knowledge is available (Wiener

filter). However, there may be some situations where we may not have any knowledge of the degradation function is known as the blind restoration (or blind deconvolution) technique. Often, restoration of images in the absence of any knowledge of the degradation function is a difficult task. To apply blind restoration, an initial guess is made about the degradation function. Often this is done using homomorphic ideas.

Blind convolution techniques aim to estimate H. One such popular estimation technique breaks the input image f and degraded image g into M smaller blocks, and computes the Fourier transforms. Then, $\log(H)$ is computed by taking the average of the sum of the differences between degraded image blocks and input image blocks. After this estimation of is completed, a Wiener filter can be used to restore the images.

An original image is shown in Fig. 9.12 (a) and its degraded image in Fig. 9.12 (b). The image restored using blind convolution is shown in Fig. 9.12 (c).



(a) Original image (b) Blurred image (c) Result of blind convolution

Fig. 9.12: Blind image restoration (a) Original image (b) Blurred image (c) Result of blind convolution

9.10 GEOMETRIC TRANSFORMATION FOR IMAGE RESTORATION

Geometrical transformations include spatial transforms and image interpolation algorithms. Spatial transforms are useful when pixels can be rearranged. Let us assume that (x, y) represent the original coordinates

and (x', y') the deformed image. The spatial transformation can then be represented as follows:

$x' = T_1(x, y)$and $y' = T_2(x, y)$

The transformations are represented by the following equations:

$T_1(x, y) = c_1x + c_2y + c_3xy + c_4$ and $T_2(x, y) = c_5x + c_6y + c_7xy + c_8$

These equations can be solved for the unknown coefficients, to get the transformation function for the given requirement.



CHECK YOUR PROGRESS 4

1. What is mean filter and state the different types of mean filter.
2. What is inverse filter?
3. Define the pseudo filer in expression.
4. What is wiener filter?
5. What is blind image restoration method?
6. Geometrical transforms include _____, and _____ algorithms. (Fill up the blanks)



9.11 LET US SUM UP

- Mathematical morphology is a framework that uses set theory and logic to analyze and process geometric structures in images.
- Binary images consist of only two pixel values—0 (black) and 1 (white)—and are often used for masking and shape analysis.
- Morphological operations work on image sets using structuring elements to modify shapes, boundaries, and structures in binary images.
- The basic morphological operations include:
- Erosion: Shrinks objects by removing boundary pixels.
- Dilation: Expands objects by adding boundary pixels.
- Opening: Erosion followed by dilation; removes small noise.

- Closing: Dilation followed by erosion; fills gaps and connects objects.
- These operations rely on set operations (union, intersection, difference) and logical operations (AND, OR, NOT, XOR).
- Structuring elements define the neighborhood for morphological operations and can be rectangular, elliptical, or linear in shape.
- Morphological image processing is widely used in noise removal, object segmentation, edge detection, and boundary enhancement.
- Properties like idempotence, duality, associativity, and translation invariance govern the behavior of these operations.
- Applications include medical image analysis, OCR, industrial quality control, and satellite image interpretation.



9.12 FURTHER READING

- "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods, Pearson, 4th Edition (2018).
- "Mathematical Morphology and Its Applications to Image Processing" by Jean Serra, Kluwer Academic Publishers, 1st Edition (1992).
- "Image Processing, Analysis, and Machine Vision" by Milan Sonka, Vaclav Hlavac, Roger Boyle, Cengage Learning, 4th Edition (2014).
- "Morphological Image Analysis: Principles and Applications" by Pierre Soille, Springer, 1st Edition (2003).
- "Fundamentals of Digital Image Processing" by Anil K. Jain, Prentice-Hall, 1st Edition (1989).



9.13 ANSWER TO CHECK YOUR PROGRESS

Check Progress 1

1. Image restoration is the process of recovering the original image from a degraded image.
2. Degradations can be noise, blurs, and artefacts.

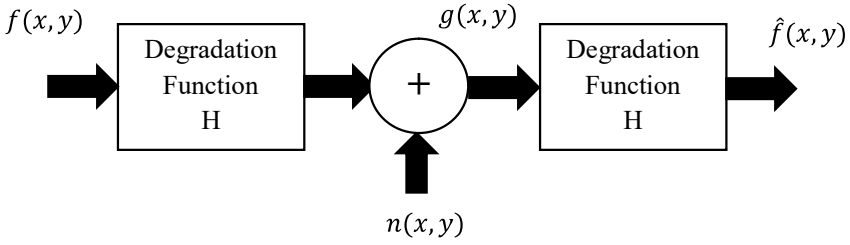
3. Degradations can be estimated by observation, experiment, and modelling.
4. A motion blur occurs because of the movement of the object or camera during the image acquisition process.

Check Progress 2

1. There are three main types of image restoration techniques available which are as follows:
   a) Complete knowledge of the blurring function is available. This is a simple case in which an original image can be retrieved by applying an inverse filter.
   b) Only partial knowledge of the blurring function is available. A wiener filter is helpful for this scenario.
   c) No prior knowledge of the blurring function is available. Blind restoration technique is used in this scenario.
2. The two types of algebraic methods used for image restoration are unconstrained and constrained methods.
3. PSF stands for point spread function and OTF stands for optical transfer function.

Check Progress 3

1. Image restoration model is given below:



2. Noise is a disturbance. Noises can be classified based on their distributions, correlation, nature, and source.
3. Noise can be classified based on distribution are: Gaussian noise, Impulse or Salt-and-pepper noise, Poisson noise, Exponential noise, Gamma noise, Rayleigh noise, and Uniform noise.

4. The combination of a picture with noise is known as additive noise. This is a linear problem. Thus, the perceived image
$g(x,y) = f(x,y) + n(x,y)$
5. Photon noise develops because electromagnetic waves are statistical. The statistical variance causes the photon production to fluctuate. This creates changes in photon count, known as photon noise.

Check Progress 4

1. Mean filters compute the average of the pixels that fall within the selected window and replace the central pixel. The different types of mean filters are arithmetic mean filters, geometric mean filters, harmonic mean filters, contra-harmonic filters, and Yp-mean filters.
2. An inverse filter removes blurs and noise using the deconvolution process.
3. Pseudo inverse filter is the modified version of the inverse filter and stabilized inverse filter. Pseudo filter is defined as:

$$H'(u,v) = \begin{cases} \frac{1}{H(u,v)}, & if\ H(u,v) \neq 0 \\ 0, & otherwise \end{cases}$$

4. A wiener filter is an optimal filter. It finds an estimate $\hat{f}(x,y)$ of the original image $f(x,y)$ using the least mean square.
5. In situations where there is little or no prior knowledge of blurs, blind image restoration methods are useful. Here, restoration is carried out by estimating the blurring functions either directly or indirectly.
6. Geometrical transforms include spatial and image interpolation algorithms.



9.14 MODEL QUESTIONS

Q1: What are the factors that can cause image degradations?

Q2: Classify degradations and explain the classification briefly.

Q3: What is the role of the point spread function?

Q4: What is blur? What are the types of blurs?

Q5: What is the significance of MTF and MTF plot?

Q6: List out some of the kernels or masks of the PSF.

Q7: What is artefact?

Q8: Explain in detail an image restoration model.

Q9: How are degradation functions estimated? List out the techniques.

Q10: What is noise? How can noises be classified?

Q11: Elucidate the types of noise based on its probability distribution.

Q12: What ae white noise and pink noise?

Q13: What re the sources of image noise? Can noise be controlled? Justify.

Q14: What is speckle noise? What kind of noise is this?

Q15: Describe in detail the types of mean filters.

Q16: What is an inverse filter?

Q17: What is a Wiener filter? Write in detail the construction and implementation of a Wiener filter. List out the advantages and disadvantages of a Wiener filter.

Q18: Explain the concept of algebraic image restoration methods.

Q19: What is meant by the blind restoration method?

Q20: Explain the role of geometrical transform in image restoration.

Q21: Consider the following list of elements:

A = {12, 23, 45, 45, 67, 89, 89, 90, 100}

Compare the arithmetic mean, geometric mean and harmonic mean of list A.

Q22: Consider the following list of elements:

A = {22, 43, 65, 75, 87, 80, 81, 92, 100}

Compute the median, maximum, minimum, and midpoint of list A.

Q23: Consider the following list of elements:

A = {15, 15, 25, 25, 55, 79, 80, 80, 110}

Apply alpha-trimming filter with a clipping factor $T = 2$, find the replacing pixel value?

Q24: What is the result of applying an order-statistic filter on the following image?

$$\begin{pmatrix} 1 & 3 & 5 \\ 4 & 4 & 3 \\ 5 & 2 & 2 \end{pmatrix}$$

*****

UNIT 10: IMAGE NOISE

UNIT STRUCTURE

1.1 Learning Objectives

10.2 Introduction

10.2.1 Sources of Image Noise

10.3 Classification of Image Noise

10.3.1 Additive Noise

10.3.2 Multiplicative Noise

10.3.3 Impulse Noise

10.4 Filters for Noise Reduction

10.4.1 Median Filter

10.4.2 Mean Filter

10.4.3 Trimmed Average Filter

10.5 Let Us Sum Up

10.6 Further Reading

10.7 Answers to Check Your Progress

10.8 Model Questions

10.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- understand the concept of image noise and its impact on the quality and interpretability of images
- identify the different sources of image noise
- classify the types of image noise and understand their characteristics and effects on image processing
- explore and compare various filtering techniques such as median filters, mean filters, and trimmed average filters for noise reduction in images
- evaluate the effectiveness of different filters in reducing specific types of noise

10.2 INTRODUCTION

Images are often affected by noise , resulting in unwanted random variations in the pixel values. Image noise may degrade the quality of the image significantly. Effective interpretation or processing of the image may be hampered by this interference, which can mask important features. Noises in may be induced from various factors such as environmental factors, sensor related issues and many other. Noise is an inevitable part of image acquisition and processing, but understanding its origins and types allows for the development of effective reduction techniques.

Mathematically, image noise can be represented as:

$$I_{noisy}(x,y) = I_{original}(x,y) + N(x,y) \qquad (10.1)$$

where:

I_{noisy} is the intensity of the noisy image at pixel $f(x, y)$.

$I_{origina}$ is the intensity of the original, noise-free image at pixel .

$N(x, y)$ represents the noise introduced at pixel (x, y).

As shown in equation 10.1, a noisy image I_{noisy} is the sum of the original image, $I_{origina}(x, y)$, and a noise function represented by N. The noise function may vary according to the type of noise (e.g., Gaussian noise, salt-and-pepper noise, etc.).

10.2.1 Sources of Image Noise

Image noise can arise from anywhere during the acquisition, transmission, or processing of images. It is described in detail by the following sources of image noise. Table 10.1 summaries some common sources of image noise.

- ***Sensor Noise*** arises due to the imperfections and inherent limitations in imaging devices that carry sensors, such as CCD (Charge-Coupled Device) or CMOS (Complementary Metal-Oxide-Semiconductor). This kind of noise is caused by various physical and electronic factors. For example, thermal noise is caused by the random fluctuations of heat energy in the components of the sensor, mainly observed in low-light

situations or with long exposure photography. Shot noise arises due to the quantum behavior of light; its statistical nature leads to a random fluctuation in the pixel brightness because of how light is detected. Additionally, read noise is induced in the process of transferring electrical signals from the sensor to digital data, at times including an error in the conversion process. Together, these noises frequently contribute to grainy patterns or even random speckles when pictures are shot under low-light conditions or with other adverse conditions.

- ***Environmental factors*** significantly affect image quality. Poor lighting can cause underexposure or overexposure, amplifying noise in darker areas. Rapid movements create streaking or ghosting effects. Atmospheric turbulence causes distortions in light, especially in long-distance or outdoor shots. The image quality is also degraded by dust or smudges on the lens or sensor. These factors usually cause localized distortions or uneven brightness.
- ***Transmission noise*** occurs in the process of sending or storing an image, particularly in digital communication or data compression. This may be caused by interference in communication channels, where electromagnetic disturbances during wireless transmission corrupt the image data. Lossy compression methods such as JPEG reduce file size by discarding fine details, leading to blocky artefacts or loss of sharpness. Bit errors during transmission can also modify pixel values, causing anomalous patterns. The effect of transmission noise is often found in the form of pixel corruption, block artefacts, or image blurriness, especially in low-bandwidth.
- ***Quantization noise*** arises in the process of signal conversion from analog to the digital format. It usually samples and discretises continuous values of intensity into discrete values. In this process rounding errors or inaccuracies may arise at certain levels. Low resolutions of sampling analog signals mean

losing subtle details. Quantization noise often manifests as banding or contouring artifacts, especially in regions of smooth intensity variation, such as skies or gradients.

- The electronic circuits in cameras or imaging devices that magnify weak signals to make them useful are the source of ***amplifier noise***. When amplifying gain, weak signals are amplified to increase brightness, but the background noise contained in the signal is also amplified. Moreover, electronic crosstalk, which is the way different electronic components in the circuit interact with each other, may introduce unwanted signals that constitute part of the noise.

Table 10.1: Some Common sources of image noise

Source	Description
Sensor Noise	Noise caused by imperfections in imaging sensors, such as thermal energy fluctuations or electronic interference during signal conversion.
Environmental Factors	Issues like poor lighting, motion blur, or atmospheric turbulence that distort captured images.
Transmission Noise	Noise introduced during image transmission, often due to interference in communication channels or compression algorithms.
Quantization Noise	Errors arising when converting an analog signal to a digital format, leading to inaccuracies in pixel intensity values.
Amplifier Noise	Noise generated by the electronic circuits used in cameras or imaging devices to amplify the captured signal.

10.3 CLASSIFICATION OF IMAGE NOISE

There are various types of image noise, each with unique properties that influence how it may affect image quality. Examples of typical noise types include additive, multiplicative, and impulsive noise. Additive noise just adds random variations to pixel values in an image without regard to any information within the original image. While multiplicative noise scales

the pixel values with their intensity and has a greater effect on brighter pixels, impulse noise consists of sudden, random spiking that often appears as individual, black or white single pixel points. In the sections that follow, some common kinds of image noise are covered.

10.3.1 Additive Noise

Additive noise is simply added to the image, and it does not depend on the image content. As shown in equation 10.2, the noisy image I_{noisy} can be expressed as the sum of the original image $I_{original}(x, y)$ and the noise $N_{additive}(x, y)$, in case of additive noise.

$$I_{noisy}(x, y) = I_{original}(x, y) + N_{additive}(x, y) \quad (10.2)$$

where,

$I_{original}(x, y)$ *is the intensity of the original image at pixel* (x, y).

$N_{additive}(x, y)$ *is the additive noise at pixel* (x, y).

I_{noisy} *is the noisy image at pixel* (x, y).

10.3.2 Multiplicative Noise

Multiplicative noise scales the pixel values based on their intensity. The noisy image I_{noisy} is obtained by multiplying the original image $I_{original}(x, y)$ with the noise $N_{multiplicative}(x, y)$. Equation 10.3 presents the mathematical expression for multiplicative noise.

$$I_{noisy}(x, y) = I_{original}(x, y) \cdot N_{multiplicative}(x, y) \quad (10.3)$$

where:

$I_{original}(x, y)$ *is the intensity of the original image at pixel* (x, y).

$N_{multiplicative}(x, y)$ *is the multiplicative noise at pixel* (x, y).

I_{noisy} *is the noisy image at pixel* (x, y).

10.3.3 Impulse Noise

Impulse noise is characterised by sudden random spikes in the image. The noisy image I_{noisy} is the original image $I_{original}$ with impulse noise $N_{multiplicative}(x, y)$ added at certain locations.Equation 10.4 presents the mathematical expression for impulse noise.

$$I_{noisy}(x,y) = \begin{cases} I_{original}(x,y) + N_{impulse}(x,y), & if\ impulse\ noise\ occurs\ at(x,y) \\ I_{original}(x,y), & otherwise \end{cases} \quad (10.4)$$

where:

$I_{original}(x, y)$ *is the intensity of the original image at pixel* (x, y).

$N_{impulse}(x, y)$ *is the impulse noise at pixel* (x, y), *typically* $\pm A$ *(where A is a constant).*

Impulse noise often appears as salt-and-pepper noise with values:, $N_{impulse}(x,y) \in \{0, A\}$, *where A represents the intensity of the noise.*

Table 10.2: Different Classes of Noise

Type	Characteristics
Additive Noise	Independent of the original image content, this noise adds random variations directly to pixel values. Gaussian noise and uniform noise are common examples.
Multiplicative Noise	Scales pixel values based on their intensity, often observed as speckle noise in radar and medical images.
Impulse Noise	Appears as isolated black or white dots in an image, often caused by malfunctioning sensors or transmission errors.



CHECK YOUR PROGRESS

1. Which type of noise is caused by random spikes, often appearing as isolated black or white dots in an image?
   a) Additive Noise
   b) Impulse Noise
   c) Multiplicative Noise
   d) Quantization Noise

2. Which filtering technique is most commonly used for reducing salt-and-pepper noise?
   a) Mean Filter
   b) Median Filter

c) Gaussian Filter
d) Trimmed Average Filter

3. What does transmission noise often result in during digital communication?
   a) Loss of dynamic range
   b) Localized distortions
   c) Block artifacts and corrupted pixels
   d) Amplification of background noise

4. State True or False
   a. Additive noise depends on the intensity of the original image.
   b. Quantization noise occurs during the process of converting analog signals into digital format.
   c. Amplifier noise is primarily caused by atmospheric turbulence.

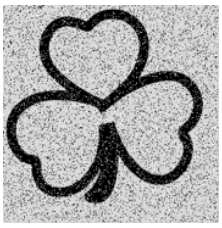
5. Impulse noise is often referred to as _____ noise due to its appearance as random black and white dots.
6. Multiplicative noise scales the pixel values based on their _____.
7. _____ noise arises due to imperfections and limitations in imaging sensors like CCD or CMOS.

10.4 FILTERS FOR NOISE REDUCTION

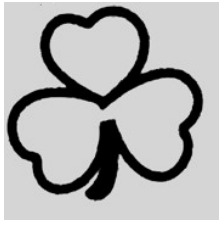
Filters are the mathematical operations that are performed over the image for modifying certain aspects of the image. Filters may be used for various purposes- for sharpening the details for noise removal etc. Filters operate on the pixel values of the image and apply particular mathematical operations based on the pixels surrounding it. Filtering are used for decreasing the effect of noise by altering pixel values so that minimal changes take place but the important information is retained. There are different types of filters to minimize noise, and each one is suitable for a particular type of noise.

10.4.1 MEDIAN FILTER

A median filter replaces each pixel value with the median value of its surrounding pixels. In general, a window of size 3X3 or 5X5 is considered. Within this window, the pixels values of the image are sorted and then the sender pixel is replaced by the median value. Median filters a very effecting reducing impulse noise such as salt and pepper noise. Figure 10.1 shows the effect of application of median filter over an image. The image in Figure 10.1 a) is effected with salt and pepper noise, which is reduced significantly by applying a median filter, giving the resultant output in figure 10.1 b).



a) Image with salt and pepper noise



b) Resultant image after applying median filter

Figure 10.1 : Application of median filter over a noisy image

2	5	1	2	3
6	0	7	1	25
5	4	4	3	2
26	2	5	7	9
8	5	7	3	0

a) Original pixel values

2	5	1	2	3
6	0	7	1	25
5	4	4	3	2
26	2	5	7	9
8	5	7	3	0

b) Application of 3X3 median filter over the image

2	5	1	2	3
6	4	7	1	25
5	4	4	3	2
26	2	5	7	9
8	5	7	3	0

c) Changes in the pixel values after application of the median filter

2	5	1	2	3
6	4	3	3	25
5	5	4	5	2
26	5	4	4	9
8	5	7	3	0

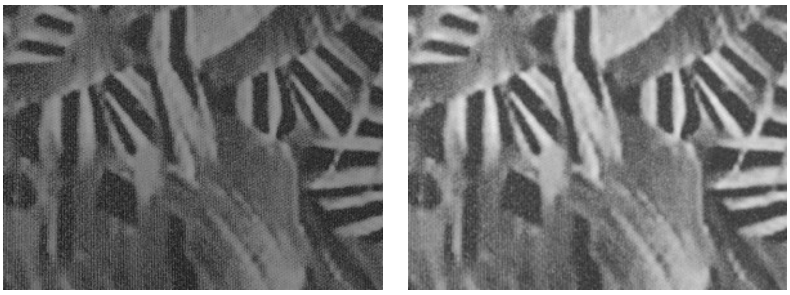
d) final pixel values after application of the median filter

Figure 10.2: Illustration of application of median filter

Figure 10.2 depicts the application of median filter over a image. A 3X3 median filter has been considered in this example. The filter is place over the over the pixels. In figure 10.2 b), we can see that the pixels values in this window are- 2, 5, 1, 6, 0, 7, 5, 4, 4. The sorted pixel values are- 0, 1, 2, 4, 4, 5, 5, 6, 7 with median 4. Thus the centre pixel which earlier had value 0 is replaced by the median 4. This process is repeated as the window slides across the whole image. Figure 10.2 d) shows the final pixel values of the image after application of the median filter over the whole image.

10.4.2 Mean Filter

The mean filter also operates using a window (normally 3X3 or 5X5). However, in mean filtering technique, the centre pixel is replaced by the average of all the pixel values in that window. Mean filters are widely used for reducing Gaussian noise. Mean filter, though easy to implement, has certain limitation. It tends to blur the image , particularly at the edges. It cannot differentiate between noise and important details, thus may not be able to preserve the fine details. Also, as extreme values are also averaged out, it may cause further distortion in the image instead of improving it. None the less, mean filter is still one of the most generally used filtering techniques of noise reduction. Figure 10.3 shows the application of mean filter over an image. Figure 10.4 depicts the steps involved in applying mean filter over an image. Here a 3x3 window is consider at each stage. As shown in figure 10.4 b) , the average values in this window is 3.78, which can be rounded of to 4. Thus the central pixel is updated to 4 as shown in figure 10.4 c). The the process is repeated across the whole image and the final updated vales of the pixels can be seen in the figure 10.4 d).



a) Original Image b) Filtered image

Figure 10.3 : Application of mean filter over a noisy image

2	5	1	2	3
6	0	7	1	25
5	4	4	3	2
26	2	5	7	9
8	5	7	3	0

a) Pixel values of the original image

2	5	1	2	3
6	0	7	1	25
5	4	4	3	2
26	2	5	7	9
8	5	7	3	0

b) Application of mean filter

2	5	1	2	3
6	4	7	1	25
5	4	4	3	2
26	2	5	7	9
8	5	7	3	0

c) Changes in the pixel values after applying mean filter

2	5	1	2	3
6	4	3	5	25
5	7	4	7	2
26	7	4	4	9
8	5	7	3	0

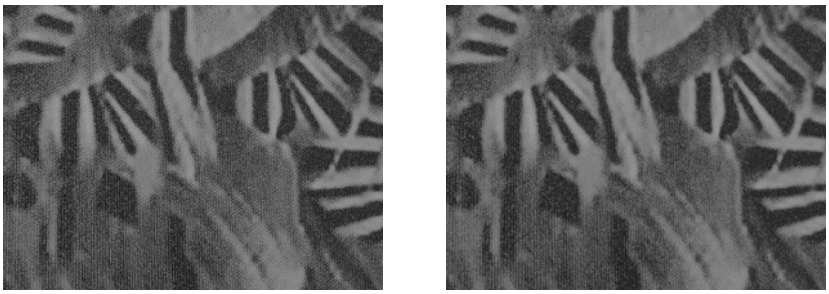
d) Final image after application of mean filter over the whole image

Figure 10.4: Illustration of application of mean filter

10.4.3 Trimmed Average Filter

The working principle of trimmed average filter is same as mean filtering. However, before calculating the average, this techn ique removes the outliers. This is done by considering a trimmed parameter, lets say p, and then removing the p highest and lowest values within the window. The centre pixel is replaced by the average

of the remaining pixels. Trimmed average filtering is less sensitive to extreme noise compared to simple mean filtering, making it better at maintaining a balance between smoothness and image detail. Figure 10.5 shows the application of trimmed average filter over an image.



a) Original Image b) Filtered image

Figure 10. 5 : Application of trimmed average filter over a noisy image



CHECK YOUR PROGRESS

Fill in the Blanks

8. A _____ filter replaces each pixel value with the median value of its surrounding pixels and is effective for reducing salt-and-pepper noise.
9. The _____ filter works by averaging the pixel values within a defined neighborhood, making it effective for reducing Gaussian noise but prone to blurring edges.
10. The _____ average filter removes extreme values in a neighborhood before calculating the average, balancing noise reduction and detail preservation.
11. Filters in image processing modify pixel values based on _____ pixels to reduce noise and enhance features.

State true/false

12. The median filter is particularly useful for Gaussian noise as it averages pixel values within a window.

13. The mean filter tends to blur edges because it does not differentiate between noise and important features.
14. The trimmed average filter removes outliers before calculating the average, making it less sensitive to extreme noise.
15. Filters only reduce noise and cannot enhance image details.



10.5 LET SUM UP

- Image noise refers to unwanted random variations in pixel values that degrade image quality and make it difficult to analyze or interpret effectively.
- Noise can originate from various sources such as sensor imperfections (thermal noise, shot noise, read noise), environmental factors (lighting issues, motion blur), transmission issues (bit errors, lossy compression), quantization errors, and amplifier noise.
- Types of image noise include additive noise, multiplicative noise, impulse noise etc.
- Additive noise adds random variations to the image, independent of the image content.
- Multiplicative noisescales pixel values based on their intensity.
- Impulse noise causes sudden spikes in pixel values, often appearing as salt-and-pepper noise.
- Noise reduction filters include median filter, mean filter, trimmed average filter etc.
- Median filter replaces each pixel with the median of surrounding pixels, effective for impulse noise.
- Mean filter averages surrounding pixel values to reduce random noise but may blur edges.
- Trimmed average filter removes extreme values before averaging, preserving image details better than the mean filter.
- Each filtering technique has its advantages and limitations, especially regarding the preservation of edges and image details.



10.6 FURTHER READING

1) Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing* (3rd ed.). Pearson.

2) Pratt, W. K. (2007). Digital Image Processing: PIKS Inside (4th ed.). Wiley.

3) Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.



10.7 ANSWERS TO CHECK YOUR PROGRESS

1. b) Impulse Noise
2. b) Median Filter
3. c) Block artifacts and corrupted pixels
4. a) False. b) True. c) Flase
5. salt-and-pepper
6. intensity
7. Sensor
8. median
9. mean
10. trimmed
11. surrounding
12. False
13. True
14. True
15. False



10.8 MODEL QUESTIONS

Q1: Explain the concept of image noise and its significance in image processing.

Q2: Describe the different sources of image noise and how each source contributes to the degradation of image quality.

Q3: Classify the types of image noise and elaborate on the characteristics of additive noise, multiplicative noise, and impulse noise.

Q4: Discuss how sensor noise affects image quality and explain the role of thermal noise, shot noise, and read noise in image degradation.

Q5: How do environmental factors such as lighting, motion blur, and dust affect the quality of images? Provide examples of each.

Q6: What is transmission noise, and how does it occur during image transmission or storage?

Q7: Explain the concept of quantization noise and its effect on digital image quality.

Q8: What is amplifier noise, and how does it impact image clarity, particularly in low-light conditions?

Q9: Describe the working principle of the median filter and its effectiveness in removing impulse noise from images. Give one example.

Q10: Compare the mean filter and median filter in terms of their advantages and limitations for noise reduction in images.

Q11: How does a trimmed average filter work, and why is it considered more effective than the mean filter in preserving image details? Discuss with an example.

Q12: Evaluate the importance of noise reduction filters in maintaining image integrity and enhancing the quality of processed images.

*****

UNIT 11: IMAGE SEGMENTATION

UNIT STRUCTURE

1.1 Learning Objectives

11.2 Introduction

11.3 Classification of Image Segmentation

11.4 Local and Global Segmentation

11.5 Image Segmentation Techniques

- 11.5.1 Edge-based Segmentation
- 11.5.2 Region Detection
- 11.5.3 Boundary Detection

11.6 Clustering Technique Image Segmentation Based on Thresholding Limitations

11.7 Edge Linking

11.8 Watershed Transformation

11.9 Let Us Sum Up

11.10 Further Reading

11.11 Answer to Check Your Progress

11.12 Model Questions

11.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- define image segmentation techniques
- apply edge-based segmentation, line-based segmentation
- apply region-based segmentation, and boundary detection

11.2 INTRODUCTION

Image segmentation involves dividing an image into meaningful structures. These structures depict the objects that constitute an image. Segmentation is the first step in image analysis and pattern recognition. Different features in images are identified and extracted by segmentation algorithms. Accuracy of extracted features decide the accuracy of automatic

recognition algorithms. Thus, a suitable and rough segmentation algorithm should be chosen very carefully. Selection of a suitable algorithm is highly application dependent. Image segmentation is one of the most difficult tasks in image processing. Generally, many image processing tasks are aimed at finding a group of pixels in an image that are similar or connected to each other in some way. We are showing the step image analysis, object representation, visualization, understanding, classification in Fig.11.1.

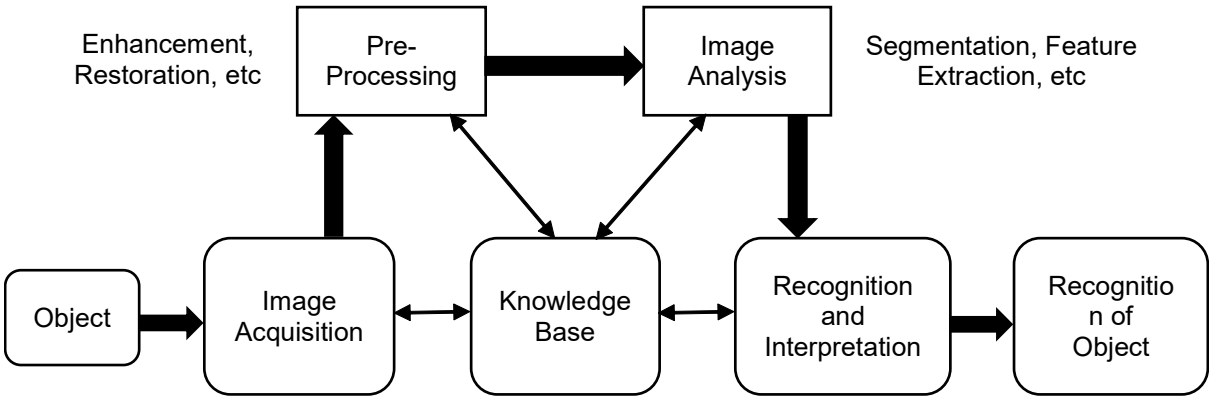


Fig.11.1: Image processing steps

An image scene is processed for autonomous machine perception and is mapped to knowledge. Here, the system tries to imitate human recognition and ability to make decision according to the information contained in the image, showing in Fig.11.2.

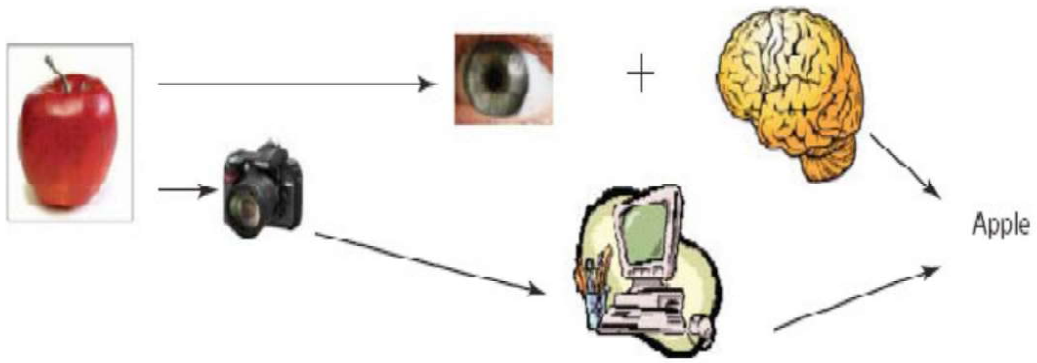


Fig.11.2: Image to knowledge mapping

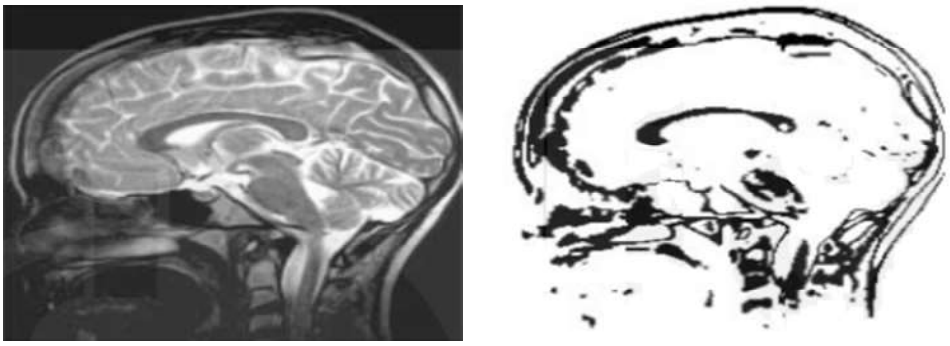
Image segmentation is the fundamental step in image analysis, understanding, interpretation and recognition tasks. It is the process of

decomposing a scene into different components. Segmentation partitions an image into multiple homogeneous regions with respect to some characteristics. In practice, it groups the pixels having similar attributes into one group. The result of image segmentation is a set of regions that collectively cover the entire image or a set of contours extracted from the image. Each pixel in a particular region is similar to the other pixel with respect to some characteristics such as colour, edge, texture, etc. Segmentation is an intermediate stage between low- level and high-level image processing tasks. Low-level tasks manipulate pixel values for irregularity correction or for image enhancement, whereas high-level tasks manipulate and analysis a group of pixels that convey some information.

Segmentation is the most important step in automated recognition system which has numerous applications and some of them are discussed below:

1. **Medical imaging**

   Segmentation is used to locate tumours, measure tissue volumes, computer aided surgery, diagnosis, treatment planning, study of anatomical structures, etc. Fig.11.3 shows the segmented portion of brain.



(a) Original image (b) Segmented image

Fig.11.3: Example of image segmentation in medical imaging

2. **Satellite imaging**

   Segmentation is used to locate objects like roads, forest, etc. in satellite images for monitoring of ecological resources like seasonal dynamics of vegetation, deforestation, mapping of underground

minerals etc. Segmentation methods based on texture, colour, wavelength, etc. are generally used here.

3. **Movement detection**

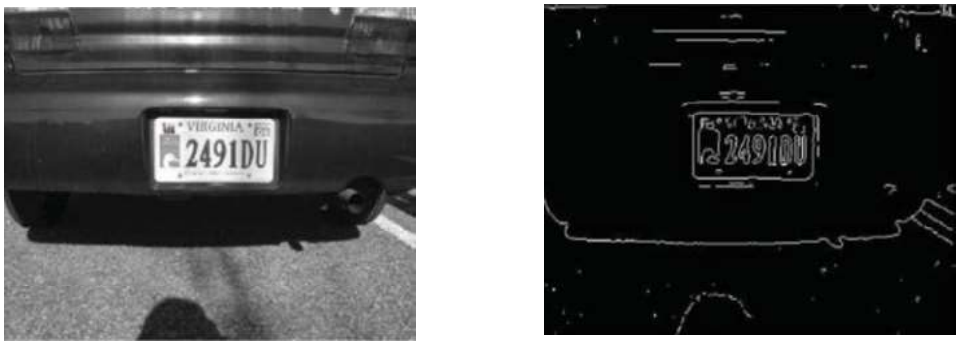
   Monitoring crowded urban environment is a goal of modern vision systems. Knowledge of the size of the crowd and tracking its motion can be used to monitor traffic intersection. Intelligent walk signal system can be designed based on the number of people waiting to cross the road. Knowledge of the size of the crowd is helpful in general safety, crowd control and planning urban environment.

4. **Security and surveillance**

   Security of the national assets such as bridges, dams, tunnels etc. is critical in today's world. Automated smart system to detect 'suspicious' movements or activities, to detect left baggage or vehicle is crucial for safety. Automated face detection systems try to match a criminal's face in a crowded place.

5. **License plate recognition (LPR)**

   Automated license plate reading is a very useful and practical approach as it helps in monitoring existing and illegally acquired license plates. LPR can be used in private parking management, traffic monitoring, automatic traffic ticket issuing, automatic toll payment, surveillance and security enforcement. Fig.11.4 shows the segmented license plate.



(a) Original license plate (b) Segmented license plate

Fig.11.4: Example of image segmentation in LPR

6. **Industrial inspection and automation**

   Here the objective is to find mission components (Integrated Circuits (IC)), in a Printed Circuit Board (PCB) or a missing pill in a packet of medicine or to check the quality of the baked biscuits etc. These automated systems help in reducing manpower in the plants and in reducing error committed by humans due to repetitive task.

7. **Robot navigation**

   Robots are used in a variety of industrial, medical and research applications. Pre-assigned tasks can be repeatedly and efficiently done by robots in hazardous environment. Image analysis and understanding help robots navigate and execute the given task.



CHECK YOUR PROGRESS A

1. What is image segmentation?

.....

.....

2. State two applications of image segmentation techniques.

.....

.....

11.3 CLASSIFICATION OF IMAGE SEGMENTATION

Image segmentation partitions an image into set of regions. In many applications, regions represent meaningful areas in an image. In other applications, regions might be set of border pixels grouped into structures such as line segments, edges, etc. The level of partitioning of an image depends on the problem to be solved. Segmentation should stop when the objects of interests have been isolated. Segmentation has two objectives:

- To decompose an image into regions for further analysis.
- To perform a change of representation of an image for faster analysis.

There are a number of segmentation techniques available. Segmentation is application dependent. A single segmentation technique may not be

suitable for different applications. Hence, these techniques have to be combined with the domain knowledge in order to effectively solve the problem. Generally, all segmentation algorithms are based on one of the two properties of gray level values of pixels.

a) Discontinuity

In this approach, images are portioned based on the difference or discontinuity of the gray level values. This means segmentation contour starts from the point where pixels appear to have very different gray values. Edge based segmentation methods fall in this category.

b) Similarity

Images are partitioned based on the similarity of the gray level values of the pixel according to a pre-defined criterion. This means as long as pixels have gray values close to each other, they are considered to be in the same segmentation block. Thresholding, region-based clustering, and matching-based segmentation techniques fall in this category.

We categorize segmentation techniques as follows:

1) Edge based segmentation

Edge based method is a commonly used technique to detect boundaries and discontinuities in an image. With this technique, detected edges are assumed to represent object boundaries and used to identify objects. The assumption here is that every part of the object is sufficiently uniform such that they can be separated on the basis of discontinuity alone.

2) Region based segmentation

Edge based techniques find the object boundaries and then locate the objects, whereas region-based techniques look for uniformity within a sub-region based on a suitable property like intensity, colour, texture etc. Region based segmentation starts in the middle of an object and then ‘grows’ outwards till it meets the object boundary.



CHECK YOUR PROGRESS B

1. Segmentation algorithms are generally based on which two properties of intensity?

.....

.....

11.4 LOCAL AND GLOBAL SEGMENTATION

In local segmentation, small windows are generated on a whole image and segmenting sub image. Global segmentation constitutes segmenting whole image. Global segmentation is used on large number of pixels. Local segmentation is used to deal with lesser number of pixel when compared to global segmentation. A global thresholding technique is one which makes use of a single threshold value for the whole image, whereas local thresholding technique makes use of unique threshold values for the partitioned sub images obtained from the whole image.

11.5 IMAGE SEGMENTATION TECHNIQUES

In this section, we shall discuss the various techniques of image segmentation.

11.5.1 Edge-based Segmentation

As object can be fully represented by its edges, segmentation of an image into separate objects can be achieved by location edges of these objects. Edge detection is a fundamental tool in image processing and computer vision. The aim is to identify points in an image at which the brightness changes sharply. Changes in image brightness can be due to discontinuities in depth, discontinuities in surface orientation, changes in material property or variation in scene illumination. Result of an edge detection algorithm is a set of connected curves that may indicate object boundaries. This significantly reduces the amount of data to be processed, making the overall object representation algorithm simpler and quicker.

If edge detection step is successful, subsequent task of interpreting the information is also successful. A typical approach to segmentation using edges is:

a) Compute an edge image, containing all edges of an original image.
b) Process the edge image so that only closed object boundaries remain.
c) Transform the result to an ordinary segmented image by filling in the object boundaries.

The first step of edge detection is discussed in the following sections. Difficulty lies in the second step which often requires removal of edges that are caused by noise or other artefacts, bridging the gaps at locations where no edge was detected and decision to connect the edges that make up a single object.

An edge may be closely defined as a line of pixels showing an 'observable' difference. For example, consider two sub-images shown in Fig.11.5. In the sub image Fig.11.5 (b), there is a clear difference between the gray levels in the second and third columns which can be easily picked by the human eye where as in the sub-image of Fig.11.5 (a) no such difference can be seen.

51	52	53	59
54	52	53	62
50	52	53	68
55	52	53	55

11.5 (a)

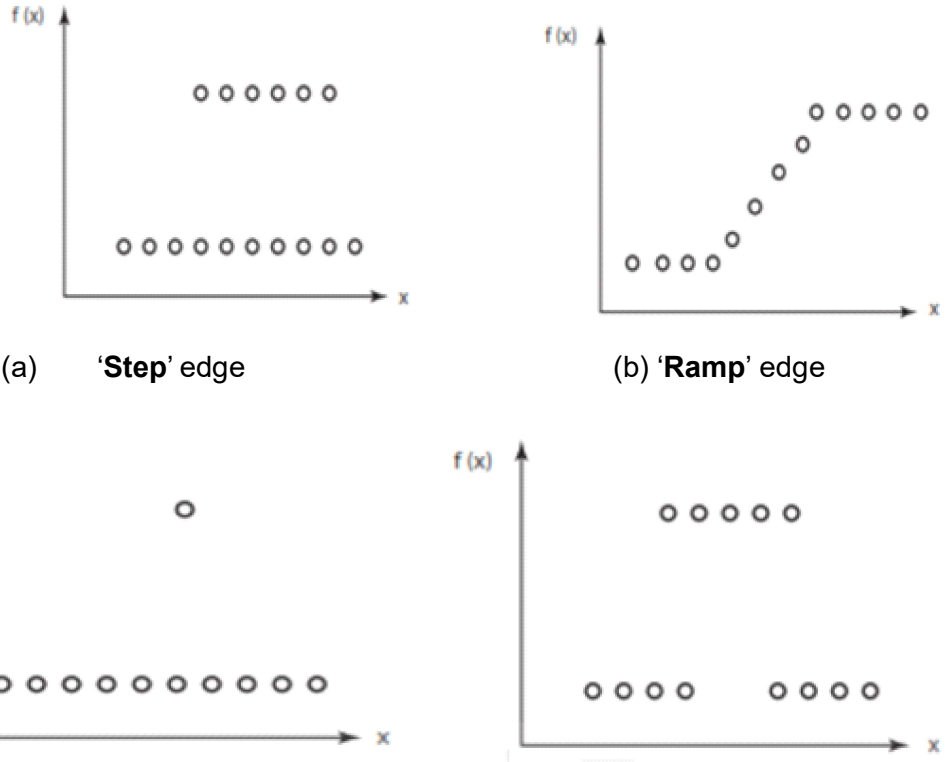
50	53	150	160
51	53	150	180
58	55	154	170
54	56	156	155

11.5 (b)

Fig.11.5: Two sub-images

Different edge models have been defined based on their intensity profiles. Fig.11.6 (a) shows a '**step**' edge which involves transition between two intensity values over a distance of one pixel. This is an ideal edge where no additional processing is needed for identification. A '**ramp**' edge is shown in Fig.11.6 (b), where the transition between two intensity levels takes place

over several pixels. In practice, all the edges get blurred and noisy because of focusing limitations and inherent noise present in electronic components. Fig.11.6 (c) shows a '**point**' which is defined as only one or two isolated pixels having different gray level values as compared to its neighbours. Fig.11.6 (d) shows a '**roof**' edge that is defined as multiple pixels having same or similar gray level values which are different from their neighbours.



(a) '**Step**' edge (b) '**Ramp**' edge

(c) '**IsolatedPoint**' edge (d) '**Roof**' edge

Fig.11.6: Gray Levels Across Edges

The first order derivative at a point x of a one-dimensional function $f(x)$ is given by

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x) \qquad (1)$$

The second order derivative of is given by

$$\frac{\partial^2 f}{\partial^2 x} = f''(x) = f(x+1) + f(x-1) - 2f(x) \qquad (2)$$

Let the values of the ramp edge in Fig.11.6 (b) from left to right be

f(x)=	20	20	20	20	20	50	100	180	180	180	180	180
1st derivative f′(x)= [using Eq. (1)]	-	0	0	0	0	+30	+50	+80	0	0	0	-
2nd derivative f′′(x)= [using Eq. (2)]	-	0	0	0	0	30	20	-80	0	0	0	0

Generally, for the implementation of first or second derivative, masks are generated. These masks are convolved with the image to get the result.
For 3 x 3 mask shown in Fig.11.7, the output is calculated by,

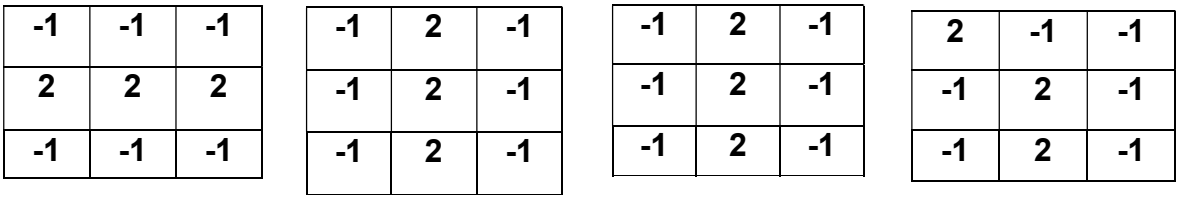
$$g(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x+i, y+j)\, w(i,j) \qquad (3)$$

Where is the image with which mask w is being multiplied.

W (-1,1)	W (0, -1)	W (1, -1)
W (-1,0)	W (0,0)	W (1,0)
W (-1,1)	W (0,1)	W (1,1)

Fig.11.7: 3 x 3 mask

Now, before we discuss the edge detection approaches, let us discuss line detection. A line can be a small number of pixels of a different colour or gray level on an otherwise unchanging background. For the sake of simplicity, it is assumed that the line is only single pixel thick. Fig. 11.8 shows line detection masks in various directions. The first mask in Fig.11.8 (a) responds strongly to lines (one pixel thick) oriented horizontally and the second mask in Fig.11.8 (b) responds to vertical lines. Third and fourth in Fig.11.8 (c) and Fig.11.8 (d) masks respond to lines at and respectively.



(a) Horizontal (w_1) **(b)** Vertical (w_2) **(c)** (w_3) **(d)** (w_4)

Fig.11.8: Line Detection Mask

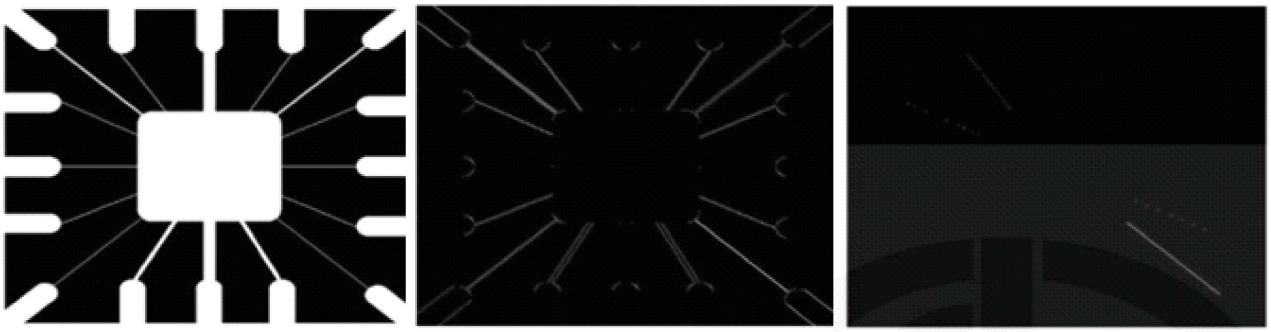
Let g_1, g_2, g_3, and g_4 be the response of each of these masks from left to right respectively. If then that point is more likely to be associated with horizontal line. Consider the electronic circuit shown in Fig.11.9 (a), the results of applying masks of Fig.11.8 (a) to Fig.11.8 (d) on this circuit are shown in Fig.11.9 (a) to Fig.11.9 (g).

Edge detection technique:

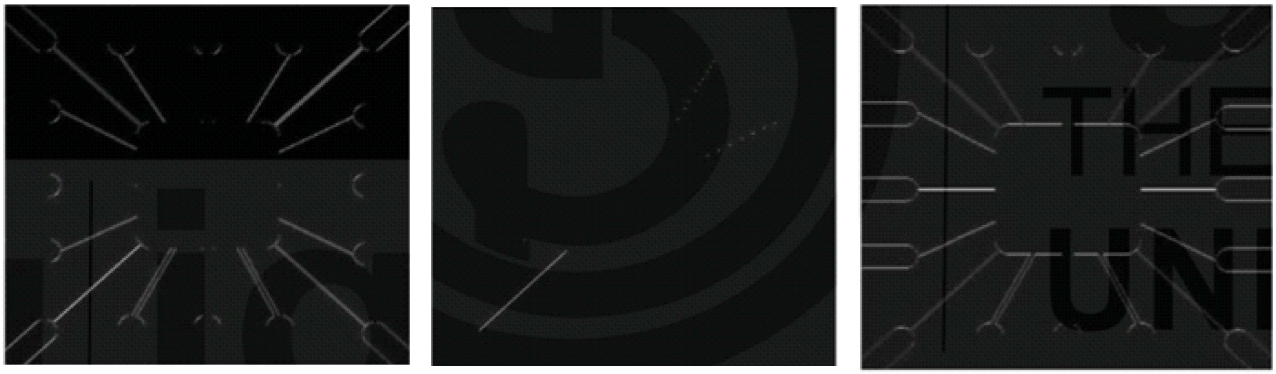
Edge detection is the most common approach used in segmentation. A typical edge may be the border between a block of red colour and a block of yellow colour. Edge can also be the boundary between two regions with relatively distinct gray level properties. Computation of a local derivative operator can enhance edges. Fig.11.10 shows the response of first and second derivative to light strip on dark background (a) and dark strip on light background (b). Edge is the smooth change in gray levels. **First derivative** is positive at the leading edge of transition and negative at the trailing edge of transition. It is zero in areas of constant gray level. The response of second derivative is different from first derivative. It is positive for dark side, negative for light side and zero in constant area. **'Magnitude' of first derivative** is used to detect the presence of an edge. **'Sign'** of second derivative is used to determine whether edge pixel lies on dark side or on light side of edge. **'Zero crossing'** is at the midpoint of a transition in gray level.

Edge detection is a non-trivial task. Edge-detection is not as simple as it looks. In practice, edges are corrupted by noise and blurring. This can be illustrated by the following examples of edge detection on a one-dimensional array. In the Fig.11.11 (a), we can intuitively say that there is an edge between 4^{th} and 5^{th} pixel. But, if the intensity difference between 4^{th} and 5^{th} pixel is

smaller because of noise and if intensity difference between the 5th and 6th pixels is higher as in Fig.11.11 (b), it would not be easy to identify the location of the edge precisely. As the edges are corrupted by noise, we observe multiple edges instead of a single edge.



(a) The original image (b) Result from -45 mask (c) -45 line



(d) Result of +45 mask (e) +45 line (f) Result of horizontal mask



(g) horizontal line

Fig.11.9: Circuit

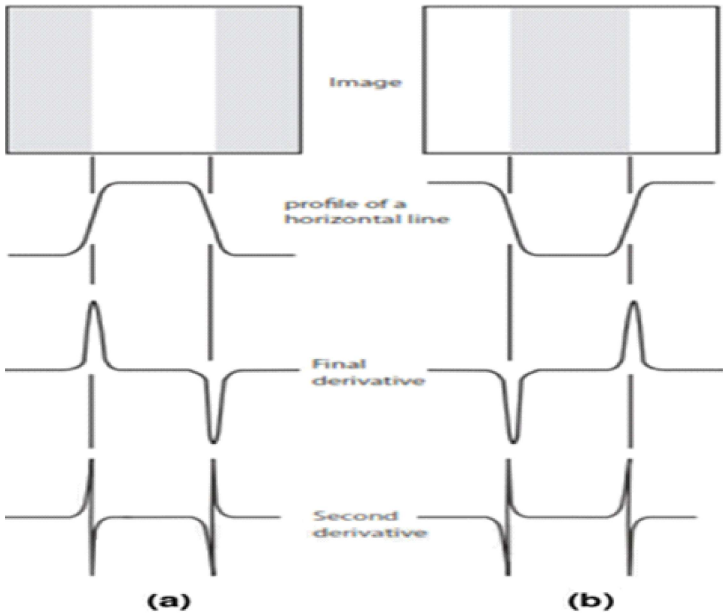


Fig.11.10: First and second derivative to (a) light strip on dark background (b) dark strip on light background

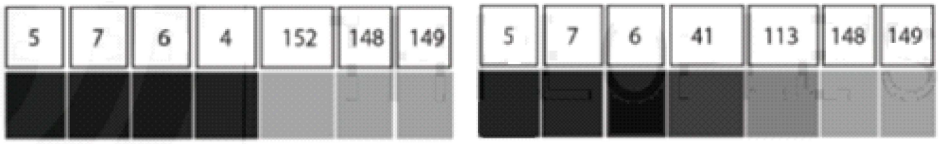


Fig.11.11 (a) **Fig.11.11 (b)**

Fig.11.11: Examples of edge detection on a one-dimensional array

11.5.1.1 Gradient operator

There are many edge detection methods. Broadly, we can classify them into two categories*:* **First order Gradient based search method**, and **Laplacian based zero crossing method**. The first order derivative-based search methods detect edges by computing the gradient magnitude and then searching for local directional maxima of the gradient magnitude. The zero-crossing based methods search for the zero crossing in a second order derivative computed from the image to find edges. Gradient is a first order derivative and is defined for an image $f(x, y)$ as

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \qquad (4)$$

It points in the direction of maximum rate of change of f at a point (x, y). Magnitude of the gradient is $mag\ (\nabla f) = \left[G_x^2 + G_y^2\right]^{1/2}$.

This can be approximated as and direction of the gradient is given by

$(\nabla f) = |G_x| + |G_y|$, where is measured with respect to x-axis.

Now, we present again discussion on various gradient operators: **Prewitt operator** and **Sobel operator**.

1) Prewitt operator

It uses a 3 x 3 size mask that approximate first derivative. The x direction mask and y direction mask are shown in Fig.11.12. The approach used is

$G_x = (Z_7 + Z_8 + Z_9) - (Z_1 + Z_2 + Z_3)$

and $G_y = (Z_7 + Z_8 + Z_9) - (Z_1 + Z_2 + Z_3)$

The convolution results in greatest magnitude indicating gradient direction. These are also called 'Compass operators' because of its ability to determine gradient direction.

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Fig.11.12: Prewitt operator

2) Sobel operator

Sobel operator is similar to Prewitt's, but it uses a weight of 2 in the centre coefficient to give it little prominence. Soble x and y direction mask is given in Fig.11.13.

$G_x = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)$

and $G_y = (Z_3 + 2Z_6 + Z_0) - (Z_1 + 2Z_4 + Z_7)$

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	0
-1	1	1

Fig.11.13: Sobel operator

Sobel operator has the advantage of providing both a derivative and a smoothing effect. This smoothing effect has noise suppression characteristics. Diagonal edges can be detected by Prewitt and Sobel masks by rotating the earlier masks by counter clockwise. Fig.11.14 shows the two masks.

-1	-1	0
-1	0	0
0	1	1

0	1	1
-1	0	1
-1	-1	1

0	1	2
-1	0	1
-2	-1	0

-2	-1	0
-1	0	1
0	1	2

(a) Prewitt (**b)** Sobel

Fig.11.14: Prewitt and Sobel Masks for Diagonal Edge Detection

11.5.1.2 Laplacian operator

Laplacian, a second order derivative is defined for a 2D function $f(x, y)$, as

$$\nabla^2 f = \frac{\partial f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Two Laplacian masks are shown in Fig. 11.15. The Laplacian equation for Fig.11.15 (a) is

$$\nabla^2 f = 4Z_5 - (Z_2 + Z_4 + Z_6 + 8_8)$$

Similarly, the Laplacian equation for Fig.11.15 (b) is

$$\nabla^2 f = 8Z_5 - (Z_1 + Z_2 + Z_3 + Z_4 + Z_6 + Z_7 + Z_8)$$

The centre coefficient of these operators are positive and other coefficients are negative. Sum of all the coefficients is zero.

0	1	0
1	-4	1
0	1	0

(a)

1	1	1
1	-8	1
1	1	1

(b)

Fig.11.15: 3 x 3 Laplacian

Laplacian is hardly used in practice for edge detection because it is very sensitive to noise, and produces double edges. Edge direction is not detected by Laplacian. It is used to find the location of edge using zero crossing detection. When the image contains noise, the best approach is to use Laplacian of 2D Gaussian function for edge detection application. 2D Gaussian filter is given by

$$H(x,v) = e^{-1x^2+y^2/2\pi^2} \text{ or } H(r) = e^{-r^2/2\sigma^2},$$

Where, $r^2 = x^2 + y^2$ and σ is the standard deviation.

First derivation of Gaussian filter is $H'(r) = \frac{1}{\sigma^2} r e^{-r^2/2\sigma^2}$.

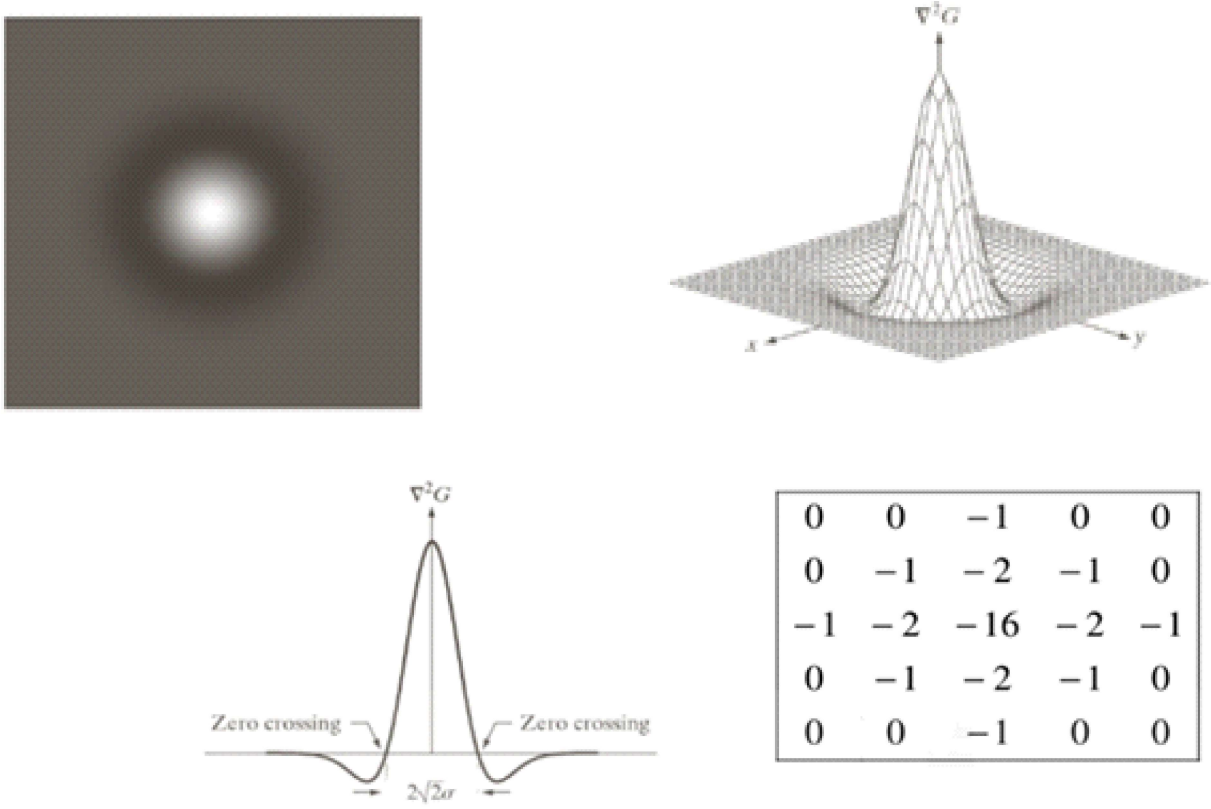
Second derivative of Gaussian filter is

$$H'(r) = \frac{1}{\sigma^2}\left(\frac{r^2}{\sigma^2} - 1\right)e^{-r^2/2\sigma^2} .$$

After returning to original coordinates x, y and introducing a normalizing coefficient C, a convolution mask of LOG (Laplacian of Gaussian) operator is given by

$$H''(x,y) = C\left(\frac{x^2 + y^2 - \sigma^2}{\sigma^4}\right)e^{-(x^2+y^2)/2\sigma^2}$$

A 5×5 LOG mask is given in Fig.11.16. Due to its shape, LOG is also known as 'Mexicanhat'. Computing second derivative in this way is robust and efficient. Zero crossings are obtained at $r = \pm\sigma$.



0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	−16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Fig.11.16: LOG as an image, LOG 3D plot, LOG

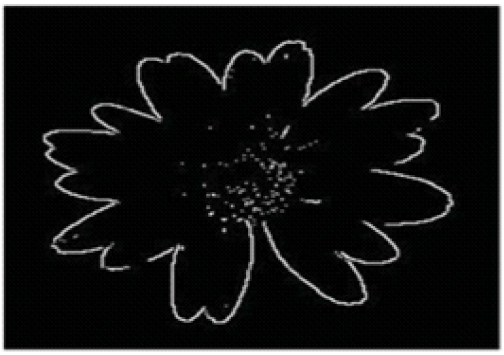
Fig.11.17 (a) is the input image, Fig. 11.17 (b) is the output of Prewitt filter, Fig.11.17 (c) is the output of Robert filter, Fig.11.17 (d), Fig.11.17 (e), and Fig.11.17 (f) are outputs of Laplacian, Canny and Sobel filters respectively. As it is clear from the figures, each filter extracts different edges in the image. Laplacian and Canny filters extract lot if inner details while Sobel and Robert filters extract only the boundary. Prewitt filter extracts the entire boundary of the flower without any gaps in the boundary.



(a) Original image



(b) Output of Prewitt filter



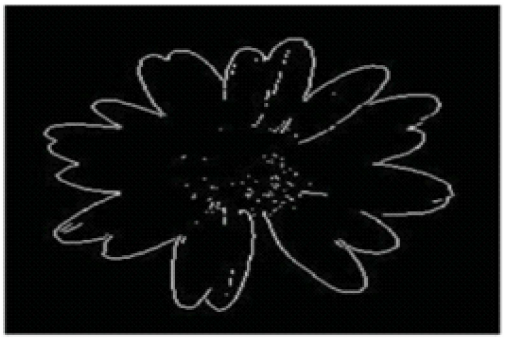
(c) Output of Robert filter



(d) Output of Laplacian filter



(e) Output of Canny filter



(f) Output of Sobel filter

Fig.11.17: (a) Original image (b) Output of Prewitt filter (c) Output of Robert filter (d) Output of Laplacian filter (e) Output of Canny filter (f) Output of Sobel filter



CHECK YOUR PROGRESS C

1. What is an edge?
2. List the properties of the second derivative around an image?

3. Define Gradient operator?
4. Why is Laplacian generally not used in its original form for edge detection?
5. Give a 5 x 5 LOG mask.

11.5.2 Region detection

Object boundary was detected to locate an object. In this section, object region will be used to locate the object. In theory, locating an object by locating either its boundary or region should result in the same object as boundary and region are just different representations of the same object. But in practice, edge-based segmentation approach may give totally different result than a region-based approach. Imperfect images, imperfect methods can be the reason which causes that the result of object detection using boundary-based approach to be different from region-based approach. Region based methods rely on the assumption that neighbouring pixels within one region have similar values. The common approach is to compare a pixel with its neighbours. A pixel belongs to the same cluster if the pixel and its neighbours satisfy the same similarity criterion. Region based segmentation methods employ two basic steps:

a) Merging and
b) Splitting.

Image segmentation using **merging** has the following steps:

Step 1: Obtain the initial segmentation of the image.

Step 2: Merge two adjacent segments to form a single segment if they are similar in same way.

Step 3: Repeat step 2 until no segment to be merged remains.

The initial segmentation can be all individual pixels. The basic idea is to combine two pixels (regions) if they are similar. The similarity criteria can be based on gray level similarity, texture of the segment, etc.

Image segmentation using **Splitting** has the following steps:

Step 1: Obtain an initial segmentation of an image.

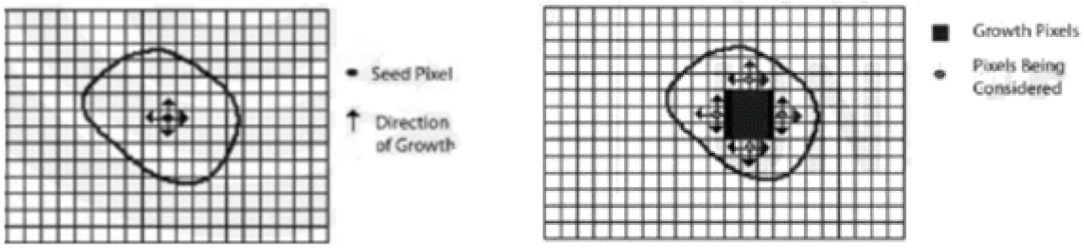
Step 2: Split each segment that is in homogeneous in some way.

Step 3: Repeat step 2 until all segments are homogeneous.

Here, initial segmentation may be the entire image (no segmentation). The criterion for inhomogeneity of a segment may be the variance of gray levels or the difference in its textures etc. Both splitting and merging methods seem to be top-down and bottom-up approach of the same method. But there is a basic difference. Merging two segments is straight forward, but in splitting, we need to know the subsegment boundary.

Region growing:

Region growing is a process of merging adjacent pixel segments into one segment. It is one of the simplest and very popular method of segmentation which is used in many applications. It needs a set of starting pixels called '**seed**' points. The process consists of picking a seed from the set and examining all 4 or 8 connected neighbours of this seed and merging similar neighbours to the seed as shown in Fig.11.18 (a). The seed point is modified based on all merged neighbours in Fig.11.18 (b). The algorithm continuous until the seed set is empty.



(a) Start of growing region (b) Growing process after a few iterations

Fig.11.18: Region

Region growing algorithm for a single seed point and same gray level value as similarity measure to merge the pixels is as follows:

The algorithm uses a data structure entity stack to keep track of seed points. Two operations push and pop are used here.

Push: puts the pixel coordinates on the top of the stack.

Pop: takes a pixel from the top of the stack.

Input to the algorithm is an image 'f'. Initial seed has coordinates of (x, y) and gray level value g at = (x, y) $f(x, y) = g$. The goal is to grow a region with all pixels having gray level value of g and assigning them gray level $k = 1, k \neq 1, k \neq g$. Let (x, y) be the coordinates of initial seed, and let be the coordinates of pixel under investigation.

The algorithm

```
Push (x, y)
        Do till stack is not empty
Pop (a, b)                      / (take input from top of the stack)
        If f (a, b) = g         / if input = desired value of g
                Set f (a, b) = 1        / segmented pixel is assigned
                                          as 1
                Push (a, b + 1          / Test all four neighbours of
                                          (a, b)
                Push (a, b – 1)         / Test all four neighbours of
                                          (a, b) by pushing it on the top
                                          of the stack
                Push (a +1, b)
                Push (a – 1, b)
        End
```

This is a recursive algorithm. The final region is extracted by selecting all pixels having gray level value as 1(k). The algorithm can be modified by changing the similarity measure to incorporate a range of values for merging. The statement if $f(a, b) = g$ can be changed to $g_1 < f(a, b) < g_2$.

Thus, if the gray level value of pixel (a, b) is between g_1 and g_2 then, it is segmented. The algorithm can be further modified to incorporate multiple seed points. In the above algorithm, only four neighbours are considered. It can be modified for eight neighbourhood. Instead of using four push operation, eight push instruction can be used with coordinates of all eight neighbours.

There are several issues about region growing which are listed below:

1) Suitable selection of seed points is important. Seed point selection is user dependent. Histogram and image properties can be taken into account by user to select the seed point.
2) Selection criterion for similarity measure is important. It generally depends on the original image and object to be segmented. Band of intensity values, colour, texture, shape are some of the similarity measures generally used.
3) Formulation of a stopping rule is also very important in region growing. Growing of a region should stop when no more pixel satisfies the similarity criteria. Algorithm can be made more powerful, if size and shape of the grown region are also considered.

Advantage: The main advantage of region growing algorithm is that it correctly separates the regions based on the property defined by the user. The algorithm is very simple to implement. Only input needed are the seed points and selection criterion. Multiple criteria can also be applied. The algorithm works well in noisy environment also.

Disadvantage: The major disadvantage of region growing is that the seed points are user dependent. Selection of wrong seed points can lead to wrong segmentation results. The algorithm is highly iterative and requires high computational time and power.

Example 1:

In the image segment given in Fig.11.19 (a) seed points are given at (3, 2) and (3, 4). Similarity criterion is gray level difference. Find segmented image, if a) T=3 and b) T=8.

	1	2	3	4	5
1	0	0	5	6	7
2	1	[1]	5	8	7
3	0	1	6	[7]	7
4	2	0	7	6	6
5	0	1	5	6	5

Fig.11.19 (a): Input image segment for example

Solution:

For $T = 3$, region growing starts with pixel $(3, 2)$. All the pixels having gray level difference <3 are assigned as a and denoted as region R_1. Another region growing starts at $(3, 4)$. All pixels with gray level value <3 are assigned as b and denoted as region R_2. The output is shown in Fig.11.19 (b).

For $T = 8$, all pixels have gray level difference less than $3 \Rightarrow$ only one region is formed, with all pixels being assigned as 'a'. The output is shown in Fig.11.19 (c).

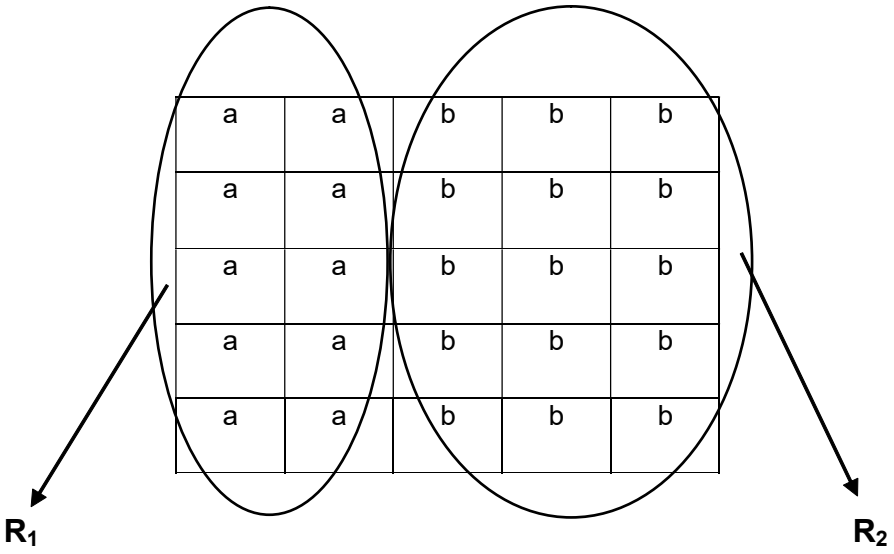


Fig.11.19 (b): Output for $T = 3$

Output for $T = 3$

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

Fig.11.19 (c): Output for $T = 8$

Example 2: Use region growing to segment the object in the image given in Fig.11.20. The seed is the centre pixel of the image. Region is grown in the following directions.

a) In horizontal and vertical directions only (4 neighbourhood)

b) In horizontal and vertical and diagonal directions (8 neighbourhood).

Similarly, criterion is the difference between two-pixel values is less than or equal to 5.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	66	60
10	59	10	60	70	63	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

Fig.11.20: Input image

Solutions:

1. Region growing starts with seed point pixel with gray value 60 in the centre. It moves horizontally up and down, vertically up and down to check how much given pixel value differs from 60.

If the difference is less than equal to 5, then it is assigned as 'a' and merges with the region, else it is assigned as 'b'. Fig.11.21 (a) shows the output.

2. If diagonal elements are also included then the region grows more as shown in Fig.11.21 (b).

b	b	b	b	b	b	b
b	b	b	b	b	b	b
b	b	a	a	a	b	b
b	a	b	a	b	b	b
b	b	a	a	b	b	b
b	b	b	b	b	b	b
b	b	b	b	b	b	b

Fig.11.21 (a):
Output for 4-neighbourhood

b	b	b	b	b	b	b
b	b	b	b	b	b	b
a	b	a	a	a	b	a
b	a	b	a	b	a	a
b	a	a	a	b	b	a
b	b	b	b	b	b	b
b	b	b	b	b	b	b

Fig.11.21 (b):
Output for 8-neighbourhood

Now, let us discuss Split and Merge method. As mentioned before, there is a fundamental problem in splitting procedure. It needs suitable sub segments to be established before performing splitting. The problem of how to split has to be resolved. We can decide whether a particular segment needs to be split or not by checking if

- Gray level difference (variance) exceeds a threshold or
- The variance of texture measure exceeds a threshold or
- Histogram entropy or any other histogram measure exceeds threshold, or
- Edge pixels exist in the segment.

We start by splitting the entire image into four quadrants. The object can be in any or some or of all the four quadrants, as the object can be anywhere in the image. Then, further subdivision of the quadrants is done. Merging operation is added to the segmentation process, recursive splitting and merging of image segments is done as shown in Fig.11.22. Original image is shown in Fig.11.22 (a) and Fig.11.222 (b) shows, entire image split into four segments. Fig.11.22 (c) shows

a further split of four segments. In Fig.11.22 (d), further segmentation is done if gray level variance in the sub block is non-zero. Merging of the sub blocks having same gray levels is shown in Fig.11.22 (e). Continuing this process, we end up in two segments one being the object and other being the background.

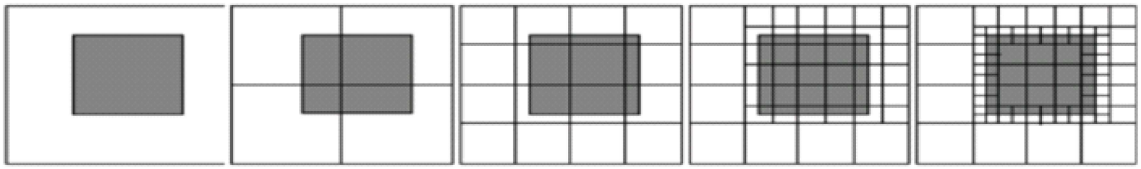


Fig.11.22 (a) **Fig.11.22 (b)** **Fig.11.22 (c)** **Fig.11.22 (d)** **Fig.11.22 (e)**

Fig.11.22: Example of split and merge segmentation

Split and merge algorithm uses '**Quad tree**' for representing segments. Fig.11.23 shows that there is a one-to-one relation between splitting an image recursively into quadrant sand the corresponding quad tree representation. However, there is a limitation in this representation, as we cannot model merging of two segments at different level of the pyramid. Let R represent the entire image. A homogeneity criterion it is selected. If the region R is not homogeneous $(H(R) = false)$, then split R into four quadrants R_1, R_2, R_3, and R_4. Any four regions with the same parent can be merged into a single homogeneous region if they are homogeneous. The steps of split and merge algorithm are as follows:

Step 1: Define an initial segmentation into regions, a homogeneity criterion and a pyramid data structure.

Step 2: If any region R in the pyramid data structure is not homogeneous (H(R) = False), split it into four child regions.

Step 3: When no further splitting is possible, merge two adjacent regions Ri which are homogeneous $(H(R_i \cup R_j) = True)$.

Step 4: Stop when no further merging is possible.

<table>
<tr><td rowspan="2">R_1</td><td colspan="2">R_{21}</td><td>R_{22}</td></tr>
<tr><td colspan="2">R_{23}</td><td>R_{24}</td></tr>
<tr><td rowspan="3">R_2</td><td>R_{411}</td><td>R_{412}</td><td rowspan="2">R_{42}</td></tr>
<tr><td>R_{413}</td><td>R_{414}</td></tr>
<tr><td colspan="2">R_{44}</td><td>R_{43}</td></tr>
</table>

Fig.11.23 (a):
Naming convention of quadrants

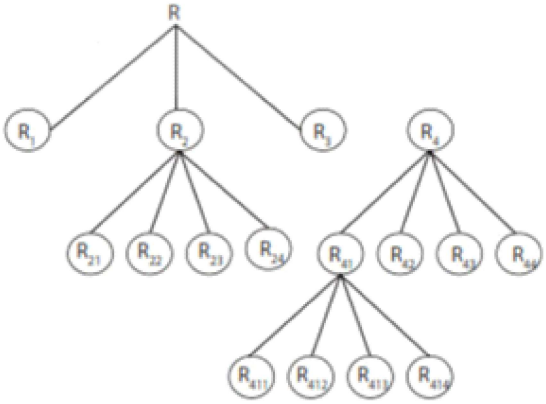


Fig.11.23 (b):
Quadtree representation

Fig.11.23: Example of recursive splitting of an image by a quad

Several modifications to this basic algorithm are possible. For example, in Step 2, merging of homogeneous regions is allowed which results in a simpler and faster algorithm.

Advantage of this algorithm: The image could be split progressively according to our required resolution because the number of splitting levels is decided by the user.

Disadvantage of this algorithm: It may produce blocky segments as splitting is done in rectangular quadrants. This problem can be solved by splitting at higher level but this will increase the computational time.

Example 3: Segment the image in the Fig.11.24 (a) by split and merge algorithm. Homogeneity criteria is the gray levels of the pixels.

Solution: The image is divided into four quadrants. Fig.11.24 (b) shows the image and its quad tree. Quadrants 2 and 3 are homogeneous. Thus, no further splitting is done. Quadrant 1 and 4 are non-homogeneous, hence they are divided further into 4 quadrants. Fig.11.24 (c) shows the splitting and corresponding quad tree. Now only one quadrant, 12 is still non-homogeneous. Hence, it is further subdivided. Fig.11.24 (d) shows the segmented image and its final quad tree structure. Now all regions are homogeneous and hence no further splitting is possible.

Now merging operation takes place between adjacent quadrant. Quadrants 43 and 44 are merged into a single region. Quadrants 123 and 124 are merged, quadrant 11 and 14 are merged. Finally merge all quadrants which are homogeneous. Fig.11.24 (e) is the final segmented image after merging is complete.

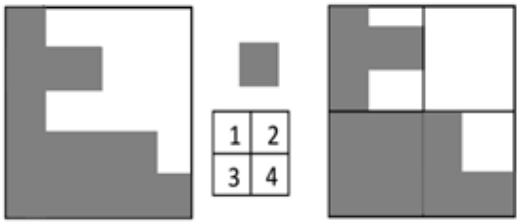


Fig.11.24 (a): Split and merge algorithm

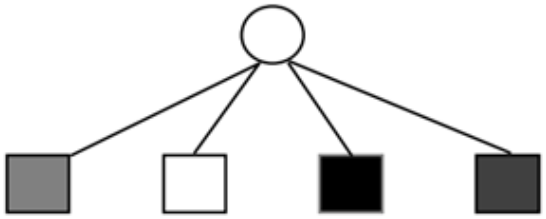


Fig.11.24 (b): Image and its quad tree

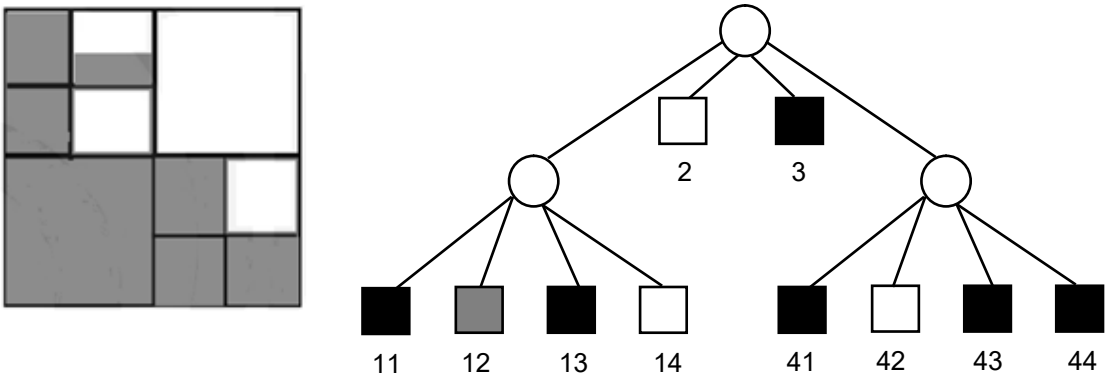


Fig.11.24 (c): Splitting and corresponding quad

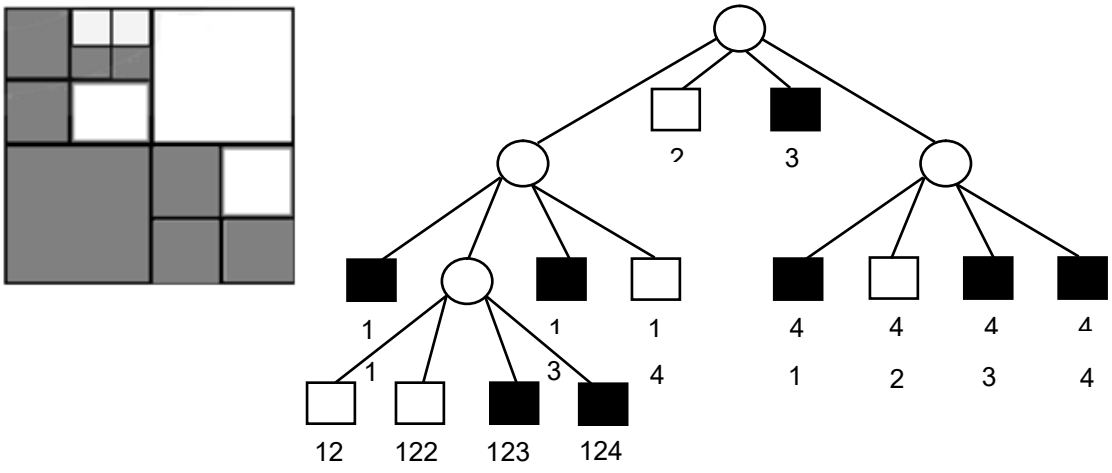


Fig.11.24 (d): Segmented image and its final quad tree

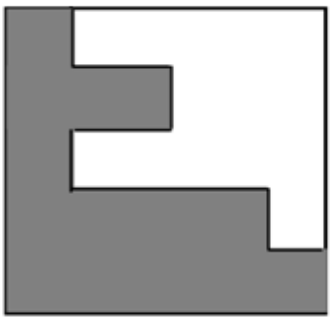


Fig.11.24 (e): Split and merge algorithm



CHECK YOUR PROGRESS D

1. Distinguish between image segmentation based on thresholding with image segmentation based on region-growing techniques.
2. Consider the image segment
3. What are the advantages or disadvantages if we use more than on seed in a region-growing technique?

11.5.3 Boundary detection

After performing image segmentation, a region may be represented in terms of:

- External characteristics (boundaries).
- Internal characteristics (texture).

A region may be described by its boundary in terms of features such as its length, the orientation of the straight line joining its extreme points and the number of concavities in the boundary. Many algorithms require the points in the boundary of a region in an ordered clockwise (or anticlockwise) direction. For boundary detection or following or tracking, the following assumptions are made:

1) The image is binary where 1=foreground and 0=background
2) The image is padded with a border of 0's so an object cannot merge with the border.
3) W limit the discussion to single regions. The extension is straight forward.

Moore Boundary Tracking Algorithm:

Given a binary region R or its boundary.

Step 1: Let the starting point b_0, be the uppermost, leftmost point in the image labelled 1.

Step 2: Denote by c_0 the west neighbour of b_0. C_0 is always a background point.

Step 3: Examine the 8-neighbours of b_0, starting at c_0 and proceeding in a clockwise direction.

Step 4: Let b_1 denote the first neighbour encountered whose value is 1.

Step 5: Let c_1 denote the background point immediately preceding b_1 in the sequence.

Step 6: Store the locations of b_0 and b_1 for use in Step 10.

Step 7: Let $b = b_1$ and $c = c_1$.

Step 8: Let the 8-neighbours of b starting at c and proceeding clockwise be denoted as n_1, n_2, …, n_8. Find the first n_k which is foreground (i.e., a "1").

Step 9: Let $b = n_k$ and $c = n_{k-1}$.

Step 10: Repeat steps 8 and 9 until $b = b_0$, that is, we have reached the first point and the next boundary point found is b_1.

The sequence of b points found when the algorithm stops is the set of ordered boundary points.

		1	1	1	1	
	1			1		
		1		1		
	1			1		
	1	1	1	1		

Fig.11.25 (a): Image

	c_0	b_0	1	1	1	
	1			1		
		1		1		
	1			1		
	1	1	1	1		

Fig.11.25 (b): Step 1 and Step 2

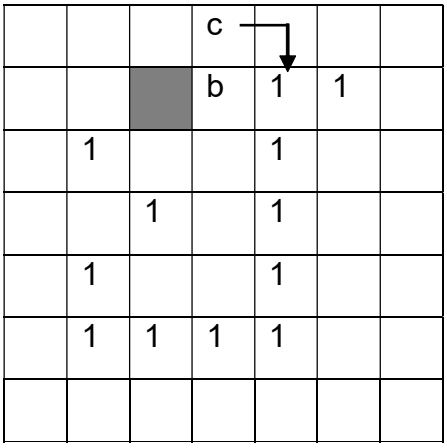


Fig.11.25 (c): Step and Step 9

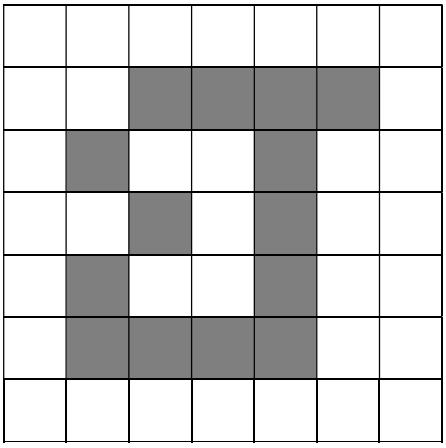


Fig.11.25 (d): Step 10

Fig.11.25: Illustration in Boundary following algorithm.

There is a need for stopping rule as stated in Step 10. We would only include the spur at the right if we stop when we reach the initial point without checking the next point. Starting from topmost leftmost point in Fig.11.26 (a) results in the Fig.11.26 (b). In the Fig.11.26 (c), the algorithm has returned to the starting point again. Rest of the boundary could not be traced.

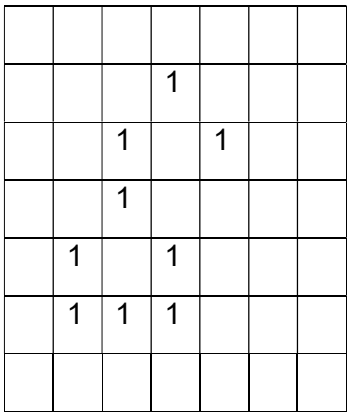


Fig.11.26 (a)

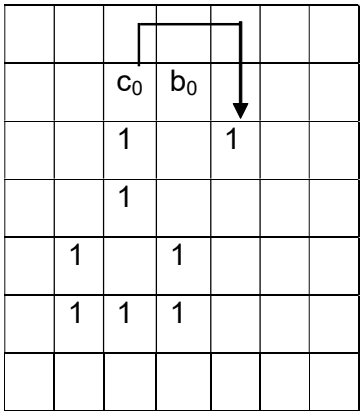


Fig.11.26 (b)

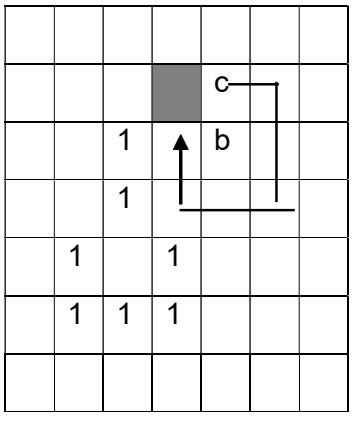


Fig.11.26 (c)

Fig.11.26: Example of erroneous result of Boundary Detection Algorithm

Chain Codes:

Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. Freeman code [1961] represent a boundary by the sequence of straight-line segments of specified length and direction. The direction

is coded by a numbering scheme (4 or 8 connectivity) as shown in Fig.11.27.

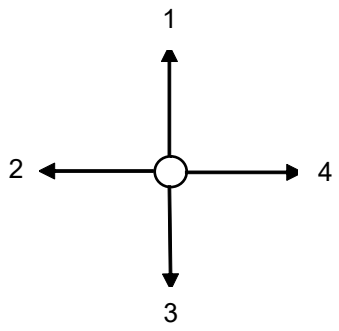


Fig.11.27 (a): 4-direction chain code

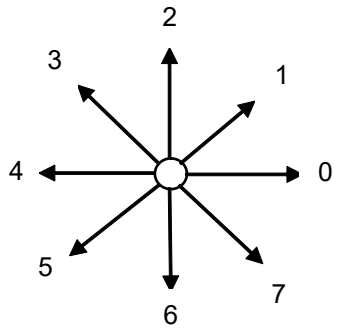


Fig.11.27 (b): 8-direction chain code

Fig.11.27: Direction numbers

Images are acquired and processed in a grid format with equal spacing in the x direction and y direction. Directions, so a chain code can be generated by the following a boundary in clockwise direction and assigning a direction to the segment connecting every pair of pixels. To avoid noise degradation and long chains a resampling of the image grid is commonly used to describe the boundary at a coarser level as shown in Fig.11.28 (a), Fig.11.28 (b) shows resampled points and Fig.11.28 (c) shows 8-direction chain code.

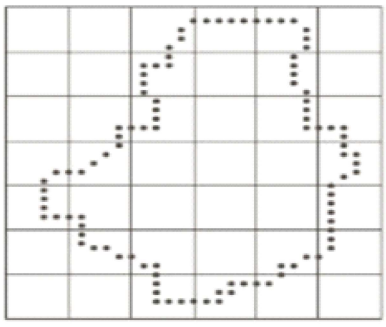


Fig.11.28 (a): Digital boundary

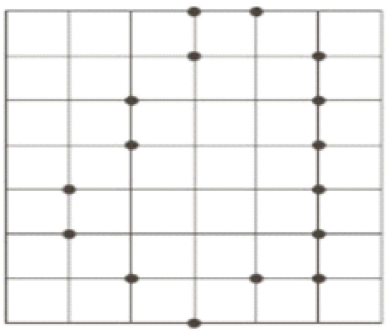


Fig.11.28 (b): Result of with resampling grid resampling

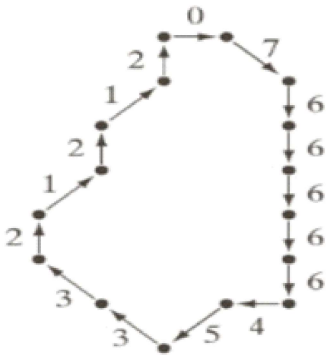


Fig.11.28 (c): 8-direction chain coded

Fig.11.28

If we start from topmost leftmost corner, chain code for Fig.11.28 is

0 7 6 6 6 6 6 4 5 3 3 2 1 2 1 2

The chain code depends on the starting point. To normalize it, we treat the code as a circular sequence of direction number sand redefine the starting point so that the resulting sequence forms an integer of minimum magnitude. To account for rotation, we use the first differences of the chain code instead of the code itself.

The first difference is obtained by counting the number of direction changes in counter clockwise direction that separate two adjacent elements of the code.

First difference is **6 7 7 0 0 0 0 6 1 1 0 7 7 1 7 1**

First value is calculated by considering the code as a circular sequence of integers, hence coding the sign changes from 2 to 0 counter clockwise and so on.

Example 4: Find chain code and first difference of the following boundary shapes.

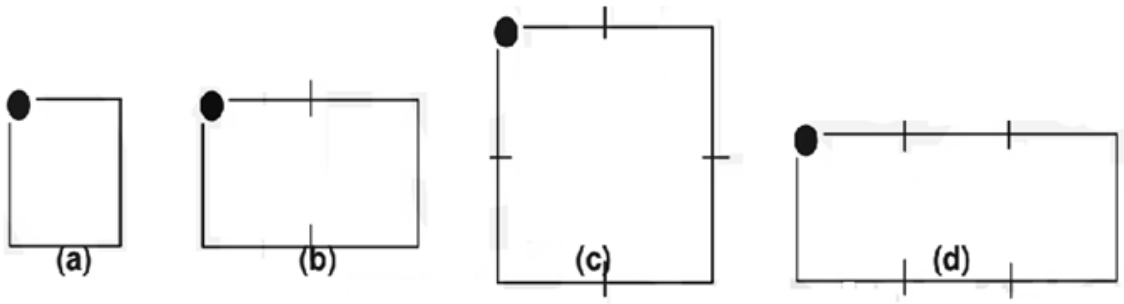
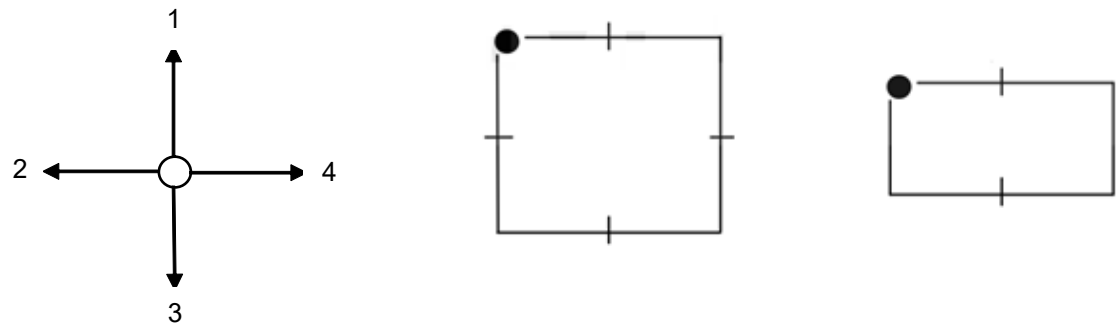
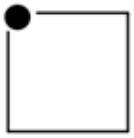


Fig.11.29

Solution:



Chain code:	0 0 3 3 2 2 1 1	0 0 3 2 2 1
Difference:	3 0 3 0 3 0 3 0	3 0 3 3 0 3





Chain code: 0 0 3 2 2 1 0 0 0 3 2 2 2 1

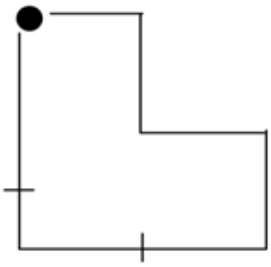
Difference: 3 0 3 3 0 3 3 0 0 3 3 0 0 3

4-connected codes are used in this example. First differences are computed by treating the chain as a circular sequence.



CHECK YOUR PROGRESS:

1. Find chain code and first difference of the following boundary shape.



11.6 CLUSTERING TECHNIQUE ON IMAGE SEGMENTATION

Clustering is a method to perform image segmentation of pixel-wise segmentation. Clustering-based segmentation groups pixels into clusters based on their similarity to the feature. It is used in image compression and content-based image retrieval. It is also used when objects in an image have distinct features that feature vectors can capture. In this type of segmentation, we try to cluster the pixels that are together. There are two approaches for performing the segmentation by hierarchical clustering.

1) Clustering by Merging
2) Clustering by Divisive

1) Clustering by Merging or Agglomerative Clustering:

This method uses a bottom-up approach, which involves assigning the pixel nearest to the cluster. The algorithm for performing the agglomerative clustering as follows:

- Take each point as a separate cluster.
- For a given number of epochs or until clustering is satisfactory.
- Merge two clusters with the smallest inter-cluster distance.
- Repeat the above step.

2) Clustering by Division or Divisive Clustering:

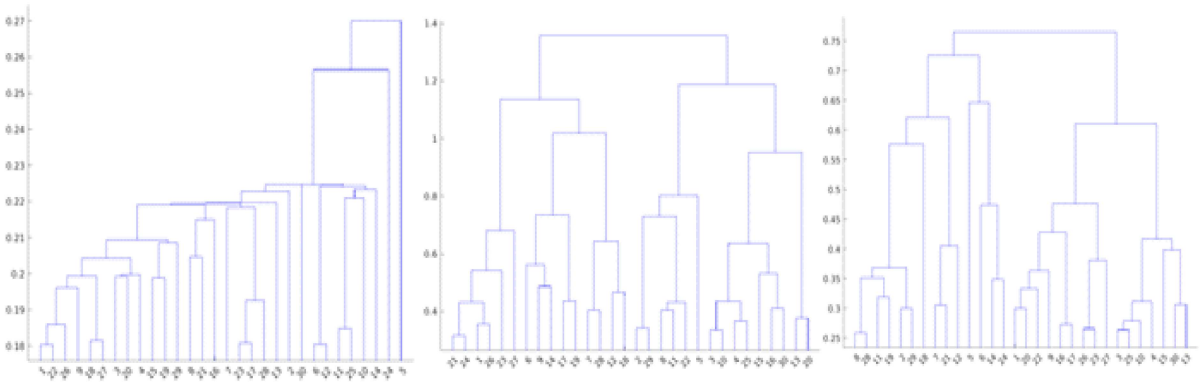
This method uses a top-down approach, which involves assigning the pixel nearest to the cluster. The algorithm for performing the agglomerative clustering as follows:

- Construct a single cluster containing all points.
- For a given number of epochs or until clustering is satisfactory.
- Split the cluster into two clusters with the largest inter-cluster distance.
- Repeat the above steps.

The agglomerative clustering is represented by Dendrogram. It can be performed in three methods:

- By selecting the closest pair for merging.
- By selecting the farthest pair for merging.
- By selecting the pair which is at an average distance (neither closest nor farthest).

The following dendrogram represents various types of clustering:



(a) Nearest clustering (b) Average clustering (c) Farthest clustering

Fig.11.30: Different types of agglomerative clustering

Here, we will be discussing how to perform the K-Means Clustering.

K-Means Clustering:

K-Means clustering is a very popular clustering algorithm which applied when we have a dataset with labels unknown. The goal is to find certain groups based on some kind of similarity in the data with the number of groups represented by K. This algorithm is generally used in areas like market segmentation, customer segmentation, etc. But it can also be used to segment different objects in the images on the basis of the pixel values.

The algorithm for image segmentation works as follows:

1. First, we need to select the value of in K-means clustering.
2. Select a feature vector for every pixel (Colour values such as RGB value, texture, etc.).
3. Define a similarity measure B/W feature vectors such as Euclidean distance to measure the similarity B/W any two points or pixel.
4. Apply K-Means algorithm to the cluster centers.
5. Apply connected component's algorithm.
6. Any component smaller than the threshold may be combined with a similar component until no further combinations are possible.

Following are the steps for applying the K-means clustering algorithm:

1. Select points and assign them one cluster center each.
2. Until the cluster center won't change, perform the following steps:
   a. Allocate each point to the nearest cluster center and ensure that each cluster center has one point.
   b. Replace the cluster center with the mean of the point assigned to it.
3. End

How to determine the value of K?

The objective function of the K-Means algorithm is WCSS

$$C = \sum i = 1M \sum k = 1N|xik - Ci|2C = \sum i = 1M \sum k = 1N|xki - Ci|2$$

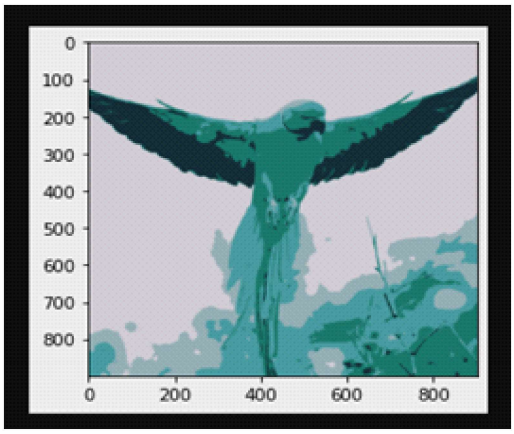
A graph plotted between WCSS and K popularly known as the Elbow method. The point where the graph makes a sharp bend is noted and the

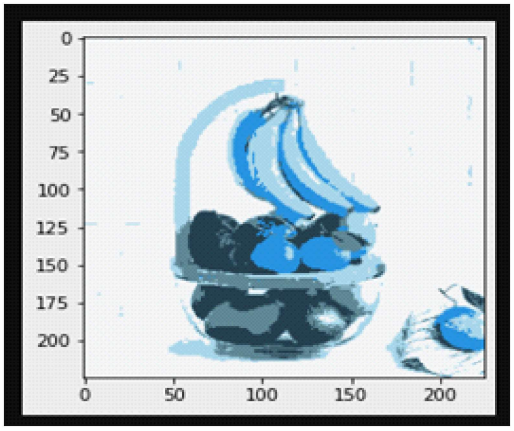
corresponding value of is taken. Determination of is done using the Elbow method, a rather brute-force approach.

Example: Python Implementation K-Means image segmentation

```
import numpy as np
import cv2
import matplotlib. pyplot as plt
from sklearn. cluster import KMeans
%matplotlib inline
image_1 = cv2.imread("/content/image_1.jpg",
cv2.IMREAD_UNCHANGED)
image_2 = cv2.imread("/content/image_2.jpg",
cv2.IMREAD_UNCHANGED)
vector_1 = image_1. reshape ((-1,3))
vector_2 = image_2. reshape ((-1,3))
kmeans_1 = KMeans (n_clusters=5, random_state = 0, n_init=5).
fit(vector_1)
c = np. uint8 (kmeans_1. cluster_centers_)
seg_data= c [kmeans_1. labels. Flatten ()]
seg_image = seg_data. reshape ((image_1. shape))
plt. Imshow (seg_image)
plt. pause (1)
kmeans_2 = KMeans (n_clusters=5, random_state = 0, n_init=5).
fit(vector_2)
c = np. uint8(kmeans_2. cluster_centers_)
seg_data= c [kmeans_2. labels. Flatten ()]
seg_image = seg_data. reshape ((image_2. shape))
plt. imshow (seg_image)
plt. pause (1)
```

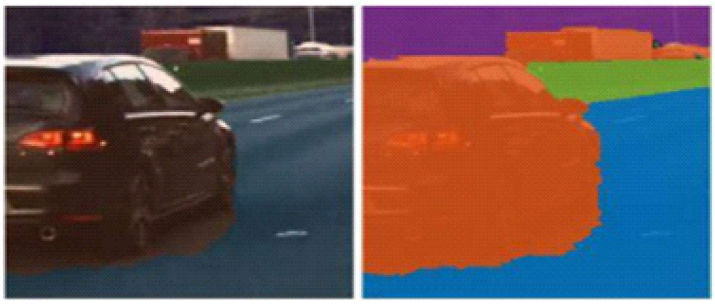
Output:



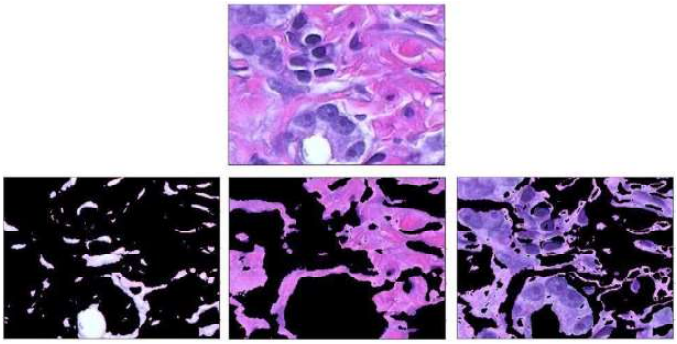


Application of Image Segmentation using K-Means clustering

1. ***Autonomous vehicles:*** During the design of the perception system of self-driving vehicles semantic segmentation is used to separate objects from the surrounding in the vicinity of the car.



2. ***Imaging in Medical Field:*** Clustering is used to separate and identify types of tissues in cancerous cell images.



3. **3-D Scene Segmentation:** In Robotic vision technology, systems are required to identify surrounding and localize objects. The Scene generated is 3D in nature. Clustering Algorithms are used for this purpose.

11.7 IMAGE SEGMENTATION BASED ON THRESHOLDING

Image segmentation based on thresholding is a fundamental technique used to partition an image into different regions based on pixel intensity values. It is commonly employed in various image processing tasks such as object detection, image analysis, and medical imaging. In thresholding, pixel values in an image are compared to a threshold value. Depending on whether the pixel's intensity is above or below this threshold, it is assigned to one of two categories (typically foreground or background). This process creates a binary image, where each pixel is either part of an object or the background.

Steps for thresholding-based image segmentation:

1. **Grayscale conversion:** If the image is in colour, it is first converted to grayscale, as thresholding generally works on single-channel images (intensity values).
2. **Choosing a threshold:** The threshold value T is chosen, which could be:
   - A global threshold:A single threshold value applied to the entire image.
   - Adaptive thresholding:The threshold value changes based on local regions of the image.
3. **Segmentation Process:** After selecting the threshold , the pixel values are compared to it:
   - If the pixel intensity at a location is greater than the threshold , it is assigned to one class (e.g., object).
   - If the pixel intensity is less than or equal to the threshold , it is assigned to another class (e.g., background)
4. **Output of the segmented image:** The result of the thresholding operation is a binary image, often represented with values 1 for the foreground and 0 for the background.

Types of thresholding

There are several methods of thresholding which are as follows:

1. Global thresholding:

- The same threshold value is applied to the entire image.
- Simple method but may not work well if lighting conditions vary across the image.

For a grayscale image, the thresholding operation can be expressed as:

$$I_{output}(x,y) = \begin{cases} 1 \; if \; I(x,y) \geq T \\ 0 \; if \; I(x,y) < T \end{cases}$$

Where:

$I(x, y)$ is the intensity of the pixel at position (x, y).

T is the threshold value

$I_{output}(x, y)$ is the binary image (1 for background, 0 for background)

2. Adaptive thresholding:

- The threshold value T is computed locally for different regions of the image. This method is helpful when there is uneven lighting or shadows in the image.
- In this case, the threshold value varies based on local regions:

$$I_{output}(x,y) = \begin{cases} 1 \; if \; I(x,y) \geq Local\ Threshold(x,y) \\ 0 \; if \; I(x,y) < Local\ Threshold(x,y) \end{cases}$$

Where $Local\ Threshold\ (x, y)$ is determined based on the local pixel intensities in the neighbourhood of (x, y).

3. Otsu's thresholding:

- An automated method that finds the optimal threshold by maximizing the variance between the foreground and background pixel values. This technique is often used when the image has bimodal (two distinct intensity peaks) histograms.

4. Multiple thresholding:

- More than one threshold value is used to segment the image into multiple regions (e.g. background, object, and other objects).

Practical Applications:

- **Object Detection**: Thresholding can help in identifying objects by isolating them from the background.
- **Medical Imaging**: Segmentation of different tissues or organs based on intensity values.

- **Image Analysis**: Identifying regions of interest, such as bright or dark spots in an image.

Example of Global Thresholding (in Python):

Here's a simple Python code using OpenCV for thresholding an image:

```
import cv2
import numpy as np
# Load the image in grayscale
image = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
# Apply a global threshold
threshold_value = 127 # You can change this value
_, segmented_image = cv2.threshold(image, threshold_value, 255, cv2.THRESH_BINARY)
# Show the original and segmented images
cv2.imshow('Original Image', image)
cv2.imshow('Segmented Image', segmented_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Considerations:

- **Threshold Selection**: The performance of thresholding depends on the proper choice of threshold value. If the threshold is too high or too low, important features may be missed, or the image may be over-segmented.
- **Noise**: Thresholding is sensitive to noise. Pre-processing techniques like smoothing may be needed to improve results.

11.8 LIMITATIONS OF IMAGE SEGMENTATION

Image segmentation techniques have a lot to offer users. But they also have certain challenges. Some of the major limitations are listed as follows:

a) Data quality and diversity

The quality and diversity of the training data significantly influence the performance of image segmentation models. Limited or biased training data may lead to poor generalization and suboptimal segmentation

results. Therefore, gathering a diverse range of high-quality images that represent various scenarios, viewpoints, lighting conditions, and object classes relevant to your application is advised. Also, you must use augmentation techniques. These include rotation, flipping, scaling, cropping, and colour jittering. They make the training dataset more diverse and improve model strength. Lastly, cleaning and preprocessing the training data to remove noise, artifacts, and inconsistencies is essential. Techniques like normalization, histogram equalization, and image registration can enhance data quality.

b) Algorithm selection and tuning

As there is a wide range of algorithms available, choosing the right segmentation algorithm and optimizing its parameters for a specific task can be challenging. To overcome this challenge, you must compare and evaluate different segmentation algorithms (e.g., U-Net, Mask R-CNN, FCN) based on their performance metrics, computational efficiency, and suitability for the application domain. Additionally, techniques like grid search, random search, or Bayesian optimization can be used to conduct systematic experiments and architecture configurations of the selected algorithms. You can also combine multiple segmentation models or algorithms using ensemble techniques to leverage their complementary strengths and improve overall performance.

c) Computational requirements

Image segmentation often requires significant computational resources, especially when processing large datasets or deploying models on resource-constrained devices. To optimize segmentation algorithms for efficiency, you can use parallel processing, GPU acceleration, and model compression techniques such as pruning, quantization, and knowledge distillation. Additionally, you can also utilize cloud computing services to offload computational tasks, scale resources dynamically, and use specialized hardware accelerators for faster inference.

d) Integration with existing systems

Adding image segmentation to existing software is complex. This is true especially when dealing with many data formats. These include protocols and environments. To get over these issues, you can design segmentation systems as modular components with well-defined interfaces and APIs to facilitate seamless integration with other software systems and platforms. Additionally, you can provide comprehensive documentation, code samples, and developer support to assist integration efforts and address potential challenges.

e) Accuracy and precision

Achieving accurate and precise segmentation results is essential for many applications, but it can be challenging due to factors such as complex object shapes, occlusions, and variations in image quality. It can evaluate segmentation models using appropriate performance metrics such as Intersection over Union (IoU), Dice coefficient, pixel accuracy, and class-wise F1 score to assess accuracy and precision. You can further continuously refine segmentation models through iterative training, validation, and fine-tuning processes to improve accuracy and reduce errors.

11.9 EDGE LINKING

Edge Linking: In this step, we try to refine the edge detection by linking the adjacent edges and combine to form the whole object. The edge linking can be performed using any of the two methods below:

- *Local Processing*: In this method, we used gradient and direction to link the neighbourhood edges. If two edges have a similar direction vector then they can be linked.
- *Global processing*: This method can be done using HOG transformation

Pros:

- This approach is similar to how the humans brain approaches the segmentation task.

- Works well in images with good contrast between object and background.

Limitations:

- Does not work well on images with smooth transitions and low contrast.
- Sensitive to noise.
- Robust edge linking is not trivial and easy to perform.

11.10 WATERSHED TRANSFORMATION

In the watershed-based image segmentation, the image is splitted into different areas. Normally watershed-based segmentation will lead to an over-segmentation of the image, especially for noisy image such as medical CT images. To avoid this over segmentation, either the input image must be pre-processed or the different regions of the image must be merged based on similarity.

It is easy to understand the watershed technique by consider the image as a plain surface where high pixel values correspond to peaks and low pixel values correspond to valleys. Same as actual watersheds in irrigation, if a drop of water falls on any particular point of the contour, automatically it will flow to lower ground till it reaches a local minimum which is catchment basin and the collected points in a particular basin is called as a member of that watershed. Expressions (1) and (2) represent the watershed transform.

$$B = G * \nabla I \qquad (1)$$

Where ∇I is defined such that

$$\nabla I(i,j) = [I(i,j+1 - I(i,j)]^2 + [I(i,j+1 - I(i+1,j)]^2 \qquad (2)$$

The next step is to find the local minima of B. An element m (i, j) is said to be a local minimum if $m(i,j) < p(i_0,j_0) \forall (i_0,j_0) \in N(i,j)$ where, $N(i,j)$ represents spatial neighbourhood of the element at row i and column j in eight connectivity. The local minima m_i represents the catchment basins of the image and are each assigned with a unique label greater than zero. The final step is to follow each element in B towards its lowest valued neighbour until it merges into one of the catchment basins $m_{i.}$ Equations

(3) and (4) give the watershed and an edge map equation respectively. Once an element merges with catchment basin $m_{i,}$ it assumes the label of that basin. The above function can be represented as WS. So, the watershed W is defined as in the equation (3).

$$W = WS(G * \nabla I) \quad (3)$$

The resulting edge map E is defined in the expression (4).

$$E(i,j) = \begin{cases} 1 & \nabla W(i,j) > 0 \\ 0 & therwise \end{cases} \quad (4)$$

Where $W(i,j)$ is the label of watershed in which element (i,j) is a member after all of the watersheds have been labelled.

11.10.1 Limitations of watershed transform-based image segmentation

Noise is a great challenge in the segmentation process of an image with watershed-based technique and over segmentation is another important drawback with watershed- based image segmentation. These drawbacks can be avoided by using region margin method. A combination of wavelet and watershed-based image segmentation method can also be used to address the above problems.

11.10.2 Steps of the watershed algorithm-based image segmentation

Step 1: Read the original image (I0).

Step 2: Take L level wavelet transform of image (IL).

Step 3: Smooth the image and remove noise particles using wavelet denoising.

Step 4: Apply watershed segmentation.

Step 5: Resulting merged region image is projected onto the IL-1 layer by an Inverse wavelet transform.

Step 6: Go to step 3 and repeat the above procedure until L equal 0 (initial stage).

Acquisition the source image: Get the MRI image, which is the source image for the segmentation process obtained from a clinic.

Contrast maximization: Maximization of contrast of an image is adopted to identify and minimize number of valley point. Wavelet transform is preferred for contrast enhancement.

To maximize the contrast, first compute $\alpha_{n,s}$ using the Equation (5) and (6).

Let n be the decomposition level,

$$E_{n,s} = \frac{\sum_{i,j} |x_{n,s}(x,j)|}{N} \quad (5)$$

$$E_{n,s} = \frac{\sum_{i,j} |x_{n,s}(x,j)|}{N} \quad (6)$$

Where s is the sub band which indicates $LH, HL, \text{and } HH$. $E_{n,s}$ is the average absolute value of a specific sub bands s in level n. N is the number of coefficients of each sub band, $x_{n,s}(i,j)$ is enhanced sub band from each sub band $x_{n,s}(i,j)$ and can be computed by equation (7).

$$x_{n,s}(i,j) = \alpha_{n,s} x_{n,s}(i,j) \quad (7)$$

Enhanced image is computed for each sub band and reconstructed the image using inverse wavelet transform. Resulting image is the contrast-enhanced image.

Conversion of objects of interest: The watershed transform detects intensity valleys in the image. The enhanced image is converted into the objects of interest with high intensity valleys.

Detection of intensity valleys: All the intensity valleys are below a particular threshold. The location rather than the size of the regions is important. The min function modifies the image to contain only those valleys found by the extended min function. The min function will also change a valleys pixel value to zero (deepest possible valley for until 8 images). All regions containing an imposed minimum will be detected by the watershed transform.

Watershed segmentation: Watershed segmentation of the imposed minima image is accomplished with the watershed function. The watershed function returns a label matrix containing non-negative numbers that correspond to watershed regions. Pixels that do not fall into any watershed region are given a pixel value of zero.

However, the image is degraded by noise and so cause over-segmentation. Therefore, over-segmentation images are further merged in some regions. The region to be merged is based on homogeneity and similarity criteria based on the wavlet coefficients. Mean, second and third order central moments values of the wavelet coefficients of each of the regions are calculated.

Mergin over-segmentation regions: The watershed transform is sensitive to the change of the data therefore the noise objective brim and then changes inside objectives can lead to over-segmentation. Over segmentation is caused by the watershed transform, and the image background is divided into several parts and merged using the following conditions:

a) The regions that will be combined are neighbours.
b) The characteristics of the region, which will be combined, should be similar.
c) The big region after combining is useful.

The regions which are neighbours and similar pixel can be combined. After combination, the background and the target image are separated. Usually, similitude can be defind according to gray, texture, and so on.

If B_i is set of boundary points of catchment basin $I(CB)$, N_i is the number of B_i, then the definition of $B\ mean\ (i)$ is given by equation (8).

$$B\ mean(i) = f(pj)/Ni \qquad (8)$$

During projection from I^{th} layer to $(I-1)^{th}$ layer, a parent-child spatial relationship between the image elements of two successive

layers is defined. This relationship is evaluated by means of a similarity measure. The children of a layer may belong to different patterns in the upper layer. The similarity between a child image element and its possible parents describes the similarities.



11.11 LET US SUM UP

In this unit, we have discussed the following:

1. Image segmentation techniques
2. Edge based segmentation
3. Line based segmentation
4. Various region-based segmentation techniques, and
5. Boundary detection algorithm.



11.12 FURTHER READING

- "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods, Pearson, 4th Edition (2018).
- "Mathematical Morphology and Its Applications to Image Processing" by Jean Serra, Kluwer Academic Publishers, 1st Edition (1992).
- "Image Processing, Analysis, and Machine Vision" by Milan Sonka, Vaclav Hlavac, Roger Boyle, Cengage Learning, 4th Edition (2014).
- "Morphological Image Analysis: Principles and Applications" by Pierre Soille, Springer, 1st Edition (2003).
- "Fundamentals of Digital Image Processing" by Anil K. Jain, Prentice-Hall, 1st Edition (1989).



11.13 ANSWERS TO CHECK YOUR PROGRESS

CHECK YOUR PROGRESS A

1. Image segmentation partitions an image into set of regions. In many applications, regions represent meaningful areas in an image. In other

application, regions might be set of border pixels into structures such as line segments, edges, etc. The level of portioning of an image depends on the problem to be solved. Segmentation should stop when the objects of interests have been isolated. Segmentation has two objectives:

a. To decompose an image into regions for further analysis.
b. To perform a change of representation of an image for faster analysis.

2. Segmentation is the most important step in automated recognition system which has numerous applications and some of them are discussed below:

a. Medical imaging.
b. Satellite.
c. Movement detection.
d. Security and surveillance.
e. License plate recognition (LPR).
f. Industrial inspection and automation.
g. Robot navigation.

CHECK YOUR PROGRESS B

1. Generally, all segmentation algorithms are based on one of the two properties of gray level values of pixels.

a. Discontinuity: In this approach, images are portioned based on the difference or discontinuity of the gray level values. Edge based segmentation methods fall in this category.

b. Similarity: Images are partitioned based on the similarity of the gray level values of the pixels according to a pre-defined criterion. Thresholding, region-based clustering and matching-based segmentation techniques fall in this category.

CHECK YOUR PROGRESS C

1. A typical edge may be the border between a block of red colour and a block of yellow. Edge can also be the boundary between two regions with relatively distinct gray-level properties.

2. 'Sign' of second derivative is used to determine whether edge pixel lies on dark side or on light side of edge. 'Zero crossing' is at the midpoint of a transition in gray level.
3. Gradient Operator: Gradient is a first order derivative and is defined for an image $f(x, y)$ as

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}$$

If points in the direction of maximum rate of change of f at a point (x, y).

Magnitude of the gradient is $mag(\nabla) = \left[G_x^2 + G_y^2\right]^{1/2}$.

This can be approximated as $(\nabla f) = |G_x| + |G_y| mag$

and the direction of the gradient is given by $\alpha(x, y) = \tan^{-1}\left[\frac{G_x}{G_y}\right]$,

where α is measured with respect to x-axis.

Laplacian is hardly used in practice for edge detection because it is very sensitive to noise, and produced double edges. Edge direction is not detected by Laplacian. It is used to find the location of edge using zero crossing detection.

4. **5 x 5 LOG mask**

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	-16	-2	-1
0	-1	-2	-1	0
0	0	1	0	0

CHECK PROGRESS D

1. Image segmentation based on thresholding applies a single fixed criterion to all pixels in the image simultaneously. Hence, it is grid. On the other hand, image segmentation based on the region-based approach is more flexible. Hence it is possible to adjust the acceptance

criteria in the course of region-growing process so that they can depend on the shape of the growing regions if desired.

2. Step 1: Computation of the histogram of the input image.

   The histogram of the image gives the frequency of occurrence of the gray level. The histogram threshold is fixed as 32. Now the input image is divided into two regions as follows:

   Region 1: Gray level < 32

   Region 2: Gray level > 32

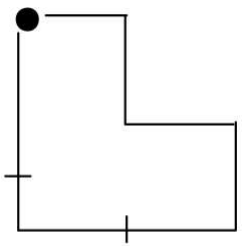
   The input image after this decision is given as

2	2	2	2	2	1	1	1
2	2	2	2	2	1	1	1
1	1	2	2	2	2	2	2
1	2	2	2	2	1	2	2
2	2	2	2	2	2	1	1
2	2	2	2	2	2	1	1
1	2	1	2	2	2	2	2
1	1	2	2	2	2	2	2

3. The advantage of using more than one seed is that better segmentation of the image can be expected, since more seeds lead to more homogeneous regions. The drawback of using more than one seed is that the probability of splitting a homogeneous region in two or more segments increases.

CHECK PROGRESS E

1.



Chain code : 0 3 0 3 2 2 1 1

Difference : 3 3 1 3 3 0 3 0



11.14 MODEL QUESTIONS

Q1: What is the formal definition of segmentation?

Q2: What are the different ways in which segmentation algorithms can be classified?

Q3: List the edge operators.

Q4: What is meant by first and second order derivatives of an image?

Q5: What is he problem of second-order derivatives?

Q6: What is meant by edge linking?

Q7: What is meant by crack edge?

Q8: What is the objective of edge relaxation?

Q9: What is a Hough transform?

Q10: Why and how are edges combined?

Q11: What is meant by a contour?

Q12: How can segmentation algorithm be evaluated?

Q13: How are isolated points and lines recognized?

Q14: Explain in detail the stages of edge detection algorithms. How are they present in edge operators?

Q15: Explain in detail the edge linking algorithms.

Q16: Explain in detail the active contour models.

Q17: Explain watershed segmentation.

Q18: Explain how the edge segmentation algorithms are evaluated?

Q19: Consider one dimensional image $f(x) = [10\ 10\ 10\ 10\ 40\ 40\ 40\ 40\ 20\ 20]$. What are the first and second derivatives? Locate the position of edge.

Q20: Consider an image $F = \begin{pmatrix} 1 & 2 & 5 \\ 5 & 5 & 5 \\ 5 & 3 & 2 \end{pmatrix}$. Show the output of any edge detection algorithm.

Q21: A black object in a white background is known to occupy 20%, 40%, and 80% of the histogram. What is the threshold value? If the colours

of the object and the background are reversed, what is the threshold value?

Q22: Consider the following image:

8	4	5
8	4	1
7	5	1

Show the crack image.

Q23: For the given image, apply graph theoretic algorithm and show the flow of the edges.

9	7	9
2	4	5
7	5	1

Q24: Explain in detail the graph theoretic algorithm, with respect to the following image:

4	5	6
11	9	38
3	11	13

Q25: Consider the following image:

1	1	9	8	7
0	1	8	8	8
0	0	7	9	8
0	1	8	8	9
1	2	8	8	9

What is the result if the threshold is 3? If the threshold condition is changed as (pixel-seed) < 0.1 (maximum – minimum) of 8-neighbourhood, what is the resultant image?

Q26: For the given image,

$$F = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 9 & 9 & 9 & 9 & 2 & 2 & 2 & 2 \\ 9 & 9 & 9 & 9 & 2 & 2 & 2 & 2 \\ 9 & 9 & 9 & 9 & 2 & 2 & 2 & 2 \\ 9 & 9 & 9 & 9 & 2 & 2 & 2 & 2 \\ 9 & 9 & 9 & 9 & 2 & 2 & 2 & 2 \\ 9 & 9 & 9 & 9 & 2 & 2 & 2 & 2 \\ 9 & 9 & 9 & 9 & 2 & 2 & 2 & 2 \end{pmatrix}$$

Show the result of the split-and-merge algorithm.

28. The contingency table of a segmentation algorithm is given as follows:

Expert human	**Present**	**Absent**
ROI present	8	1
Absent	1	1

What is the efficiency of the segmentation algorithm?

*****

UNIT 12: IMAGE COMPRESSION TECHNIQUES

UNIT STRUCTURE

12.1 Learning Objective

12.2 Introduction

12.3 Requirement for Image Compression

12.4 Redundancy in Images

12.5 Image Compression Scheme

12.6 Huffman Coding

12.7 Arithmetic Coding

12.8 Transformed Based Coding

12.9 Vector Quantisation

12.10 Let Sum Up

12.11 Further Reading

12.12 Answers to Check Your Progress

12.13 Model Questions

12.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- understand the concept and importance of image compression.
- explore the types of redundancies in images.
- learn about various image compression schemes and their applications.
- understand the principles and algorithms behind popular image compression techniques: Huffman coding, Arithmetic coding, Transform-based coding, and Vector quantization.

12.2 INTRODUCTION

In the digital age, high-resolution imaging devices produce large image files, leading to increased demands for storage space and bandwidth during transmission. Image compression addresses these challenges by reducing file sizes while maintaining acceptable visual quality. This unit

explores the fundamental concepts and techniques of image compression, including the identification and elimination of redundancies in image data. You will learn about both lossless compression methods, such as Huffman coding and arithmetic coding, which preserve all original data, and lossy compression techniques, like transform-based coding and vector quantization, which sacrifice some data for greater efficiency. By understanding these methods, you will gain insights into how compression optimizes storage, enhances transmission speed, and improves compatibility across devices, making it indispensable in modern digital applications.

12.3 REQUIREMENT FOR IMAGE COMPRESSION

The size of image files has grown dramatically since the introduction of high-resolution imaging equipment. However, with growing resolution storage space and bandwidth requirement also increases. Thus, image compression has become important in many aspects. The following are some points highlighting the advantages of image compression.

- **Storage Efficiency**: The amount of storage space needed for the images as their size gets reduced due to compression.
- **Enhanced Transmission Speed:** Lowering Image size also lowers the bandwidth demand and transmission takes a shorter time.
- **Faster Loading:** Long files take time to load. By compressing images, file sizes can be reduced significantly shortening the loading time for the page.
- **Mobile Optimisation:** Mobile users do not require very high resolution images since mobile devices operate with small screen size and lower resolutions. Therefore, using compression reduce the images and make them more mobile friendly while improving the experience and speeds of loading time.
- **Compatibility:** Makes image sharing and distribution across different platforms more practical.
- **Cost Reduction:** Reduces storage and transmission costs.

12.4 REDUNDANCY IN IMAGES

The data representing an image often may have many redundancies which may be categorised into the following types-

- Spatial Redundancy: Occurs when neighbouring pixels have similar values. For example, large areas of uniform colour in an image.
- Spectral Redundancy: Occurs due to correlation between different colour channels in an image (e.g., RGB channels).
- Temporal Redundancy: Present in image sequences or video frames where changes between consecutive frames are minimal.
- Perceptual Redundancy: Certain details in an image may not be perceptible to the human eye and can be safely eliminated.

12.5 IMAGE COMPRESSION SCHEMES

There are various schemes available for image compression. Most of the compression schemes generally consists of the following steps:

- **Transformation:** Transfers image data to a domain where redundancies are more detectable.
- **Quantization:** Reduces the precision of transformed coefficients to eliminate less significant information.
- **Encoding:** the quantised image is then encoded using different coding schemes to reduce the size.

Compression schemes can broadly be classified into the following two categories -

Lossless Compression: It a compression scheme where when the encoded image is decoded, we get back the original image (such as PNG or GIF).

Lossy Compression: Some loss is incurred during this type of compression techniques (Such as JPEG or MP3). We cannot reconstruct the exact original image.



CHECK YOUR PROGRESS

State True /False

1. High-resolution imaging devices lead to increased storage and bandwidth requirements for image files.
2. Lossless compression allows the exact original image to be reconstructed after decoding.
3. Temporal redundancy occurs when neighbouring pixels in an image have similar values.
4. Quantization in image compression reduces the precision of transformed coefficients to eliminate less significant information.

Fill in the Blanks:

5. _____ redundancy occurs when changes between consecutive frames in a video sequence are minimal.
6. _____ compression schemes ensure no data is lost during the compression process.
7. Quantization is a step in image compression that reduces the _____ of transformed coefficients.

12.6 HUFFMAN CODING

It is a simple and efficient lossless data compression scheme, where codes are assigned to the input characters. The codes are of variable-length, which depends on the frequency of the characters. More frequently appearing characters receive shorter codes, while infrequently appearing characters receive longer codes.

Steps:

- Calculate the frequency of each symbol in the image data.
- Build a binary tree, combining symbols with the lowest frequencies.
- Assign binary codes to each symbol based on their position in the tree.

Example 12.1:

The process can be illustrated with the following example. Let us consider the string “AAABBCCDDDDD”. First we need to construct a frequency table, where we store the frequency count of each symbol as shown in table 12.1, and then sort the table in ascending order. We then repeatedly combine the symbols with smaller frequencies.

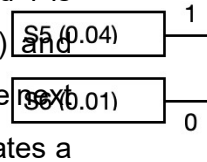
Table 12.2: Sorted frequency table

Symbol	Frequency	Code
B	2	110
C	2	111
A	3	10
D	5	0

Table 12.1: Frequency table

Symbol	Frequency
A	3
B	2
C	2
D	5

We first combine B (frequency 2) and C (frequency 2) (lowest frequencies) to create a new node BC with frequency 4(2+2). While combining two nodes, we make the one with the lower frequency as the left child and the other one as the right child. Label 0 is then assigned to the left edge and 1 is assigned to the right edge. Next, we combine A(frequency 3) and BC(frequency 4) and create a node A(BC) with frequency 7 (3+4). The next combination is -A(BC)(frequency 7) and D (frequency 5) which creates a node D(A(BC)) (frequency 12). The final tree is depicted in figure 12.1.



The next step is to assign codes to the symbols. We traverse the tree from the root node till we encounter the symbol in one of the leaves.

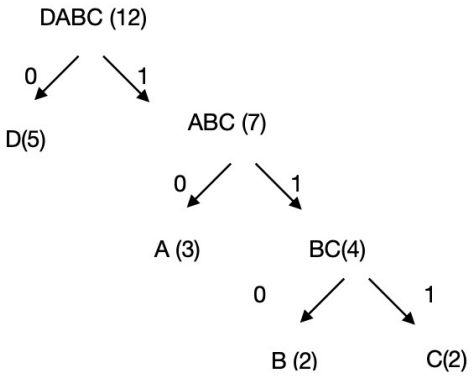


Figure 12.1: Huffman Tree for "AAABBCCDDDDD"

As we traverse, we note down the labels of each edge which finally is assigned as the Huffman code for the symbol. The final codes for the symbols are shown in table 12.2.

Example 12.2: The process can also be applied to images, where we consider the intensity levels of the pixels. Consider intensity levels of pixels in an image and their corresponding probability distributions. By applying Huffman coding algorithm, we can construct a tree as shown in figure 12.2. The final Huffman codes are shown in table 12.3. As we can see from the table we need 8 bits to represent the given intensity levels of each pixel. However, with Huffman coding, we can reprint the intensities with code lengths that varies from 1to 5.

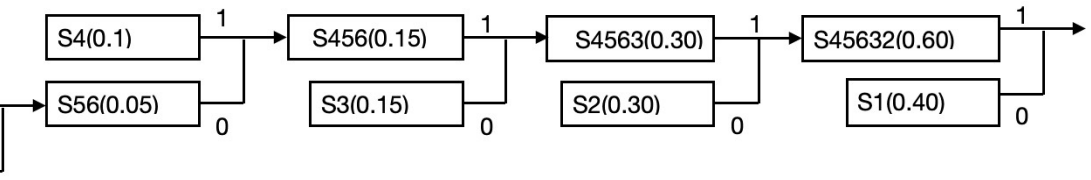


Figure 12.2: Huffman tree for the symbols in table 12.3

Table 12.3: Huffman coding for image pixels

Pixels (with intensity value gray scale)	Probability (arranged in decreasing order)	Binary Code	Huffman Code	Length of Huffman Code
S1 - 50	0.4	00110010	0	1
S2 - 100	0.3	01100100	10	2
S3 - 200	0.15	11001000	110	3
S4 - 150	0.1	10010110	1111	4
S5 - 255	0.04	11111111	11101	5
S6 - 175	0.01	10101111	11100	5

12.7 ARITHMETIC CODING

Arithmetic coding is a form of entropy encoding in lossless data compression. Arithmetic coding differs from Huffman coding because it encodes the whole message into a single number: a fraction between 0 and 1. The steps to perform in the arithmetic coding scheme are as follows:

Steps:

- **Input Data and Probabilities:** Prepare the input data into a format with data point and corresponding probability of appearance.
- **Divide the Range [0, 1]:** Divide the range of probability [0,1] into subintervals. Assign a subinterval to a symbols proportional the probability of owcuurance.
- **Encode the Message:** Start with the first symbol. As a symbol is processed, extract the range of probability distribution of that symbol and then divide that range again into subintervals as was done in the previous step. Repeat the step until all the symbols are processed.
- **Generate the Final Code:** Once all the symbols are processed select any value from the final interval. This value is used to represent the whole image.

- **Decode:** In the decoding process, the same process is repeated in the reverse order to contract the original image.

Example 12.3: Encode the message "ABBA"

Given probabilities:

- Symbol A: 0.4
- Symbol B: 0.3
- Symbol C: 0.2
- Symbol D: 0.1

Step 1: Divide the Range [0, 1]

Step 2: Encode Each Symbol

Start with the full range [0, 1]. For each symbol, narrow the range:

- **First Symbol: A**

  Range for A: [0.0, 0.4).

Table 12.4: Probability Table

Symbol	Probability	Range
A	0.4	[0.0, 0.4)
B	0.3	[0.4, 0.7)
C	0.2	[0.7, 0.9)
D	0.1	[0.9, 1.0)

Narrowed range:

- **Second Symbol: B**

  Range for B in the current range [0.0, 0.4):

Table 12.5 Arithmetic Coding for “ABBA”

A	B	B	A
0	0	0.16	0.208
0.1	0.04	0.172	0.2116
0.2	0.08	0.184	0.2152
0.3	0.12	0.196	0.2188
0.4	0.16	0.208	0.2224
0.5	0.2	0.22	0.226
0.6	0.24	0.232	0.2296
0.7	0.28	0.244	0.2332
0.8	0.32	0.256	0.2368
0.9	0.36	0.268	0.2404
1	0.4	0.28	0.244

Subdivide [0.0, 0.4) into proportions based on probabilities.

New range for B: [0.16, 0.28).

- **Third Symbol: B**

  Range for B in the current range [0.16, 0.28):

  Subdivide [0.16, 0.28) into proportions.

  New range for B: [0.208, 0.244).

- **Fourth Symbol: A**

  Range for A in the current range [0.208, 0.244):

  Subdivide [0.208, 0.244) into proportions.

  New range for A: [0.208, 0.2224).

Step 3: Final Encoded Value

Choose any value in the final range [0.208, 0.2224) as the encoded value. For example, **0.213**.

12.8 TRANSFORM- BASED CODING

Transform coding converts spatial domain image data into the frequency domain using mathematical transforms, such as the Discrete Cosine

Transform (DCT) or Discrete Wavelet Transform (DWT). High-frequency components (less perceptible details) are often discarded to achieve compression. Effective for lossy compression with good visual quality. Forms the basis of popular formats like JPEG.

Steps:

- Divide the image into smaller blocks.
- Apply a transform (e.g., DCT) to each block.
- Quantize and encode the transformed coefficients.

12.9 VECTOR QUANTIZATION (VQ)

It a lossy compression technique, where the input data is considered as a collection of large number of input vectors. These vectors are then mapped with a finite set of representative vectors and in the compressed image, the index of these representative vectors are mentioned in place of the original vectors. The representative set of vectors is called as **"Codebook".** The following are some of the important concepts involved in vector quantization.

- **Input Vectors**: The input high dimensional data like fracture vectors, pixels intensities etc.
- **Codebook**: The set of representative vectors. Each vector from the input vector set is finally represented with one of the vectors from codebook.
- **Quantization**: In the compressed image, each input vector of the original image is replaced with the index of the closest representative vector from the codebook.
- **Reconstruction**: In this process, the image is reconstructed with the help of the indices stored in the compressed image and the code book. Each index is replaced by the corresponding vector from the code book. As, the codebook has only representative vectors not the original vectors, it is not possible to get back the original image and thus the process is lossy.

The block diagram of the vector quantisation processis given in figure 12.3. An important subprocess of this procedure is the construction of the

Codebook. For this K-means clustering algorithm may be use, where the set of input vectors are divided into k clusters. The centroid vector of each closer can be used a the representative vector for that cluster in the codebook. Thus in the codebook, there will be only k vectors. Each input vector is then replaced with the index of the nearest codebook vector using a distance metric, such as the **Euclidean distance.** During reconstruction, the codebook vectors corresponding to the indices are used to approximate the original data.

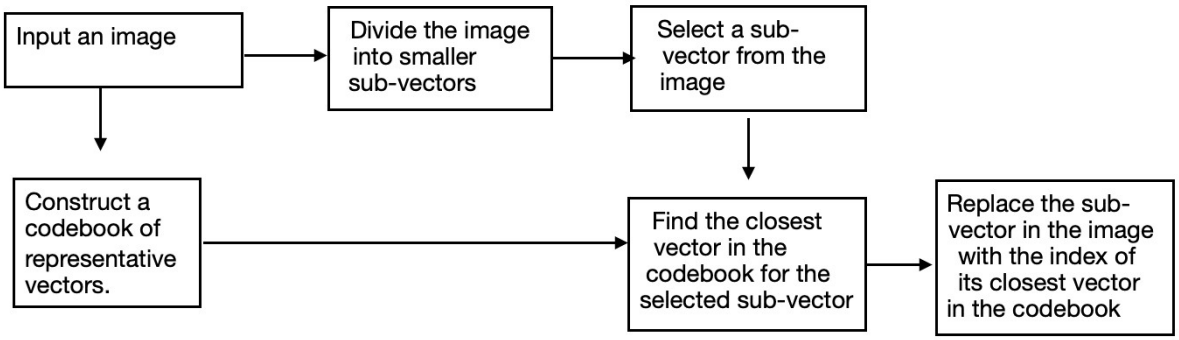


Figure 12.2: Huffman tree for the symbols in table 12.3



CHECK YOUR PROGRESS

State True /False

8. Huffman coding assigns variable-length codes based on the frequency of symbols, with more frequent symbols receiving longer codes.
9. Arithmetic coding encodes the entire message into a single fractional number.
10. Transform-based coding converts spatial domain image data into the frequency domain using methods like DCT or DWT.

Fill in the Blanks:

11. Huffman coding is a _____ compression technique where codes are assigned based on the frequency of symbols.
12. In arithmetic coding, the final encoded value is a _____ in the range [0, 1].

13. In vector quantization, the _____ contains the representative vectors used to approximate the original image.
14. The _____ algorithm can be used to construct the codebook in vector quantization.



12.10 LET SUM UP

- Image compression reduces the storage space and bandwidth requirements for high-resolution images. It also improves transmission speed, page loading time, and compatibility across devices.
- An image data can have redundancies, which can be categorised into spatial, spectral, temporal, or perceptual redundancy. Removal of these redundancies optimize storage and bandwidth requirements.
- Compression schemes are classified into lossless (e.g., PNG) and lossy (e.g., JPEG) based on data reconstruction.
- Lossless compression retains the original information, thus it is possible to get back the original image.
- Lossy compression techniques results in some information loss as it discard the non-essential data.
- Huffman coding assigns shorter binary codes to frequent symbols for efficient lossless compression.
- Arithmetic coding encodes an entire message into a single fractional value between 0 and 1.
- Transform-based coding (e.g., DCT) shifts image data to the frequency domain, discarding high-frequency components for lossy compression.
- Vector quantization maps input data to representative vectors in a codebook for lossy compression.



12.11 FURTHER READING

1) Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing* (3rd ed.). Pearson.

2) Pratt, W. K. (2007). Digital Image Processing: PIKS Inside (4th ed.). Wiley.

3) Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.



12.12 ANSWER TO CHECK YOUR PROGRESS

1. True
2. True
3. False
4. True
5. Temporal
6. Lossless
7. Precision
8. False
9. True
10. True
11. Lossless
12. Fraction
13. Codebook
14. K-means Clustering



12.13 MODEL QUESTIONS

Q1: Why has image compression become important in the era of high-resolution imaging devices?

Q2: List the key needs for image compression.

Q3: Explain how image compression enhances transmission speed, mobile optimization and increase storage efficiency.

Q4: Define the following types of redundancies in images- Spatial redundancy, Spectral redundancy, Temporal redundancy, Perceptual redundancy. Give example of each.

Q5: What are the main steps involved in most image compression schemes?

Q6: Discuss the Huffman encoding scheme. Assign Huffman codes to the symbols in the string "BBBCCCDDDAAABCADDDACDBADDDDCCC".

Q7: Using the probabilities A: 0.4, B: 0.3, C: 0.2, D: 0.1, calculate the range for encoding the message "BACCD".

Q8: What is transform-based coding, and how does it contribute to image compression? Name two common transforms used in transform-based coding. What happens to high-frequency components during transform-based compression?

Q9: What is a codebook in vector quantization, and how is it constructed?

Q10: With a block diagram Explain the steps involved in vector quantization. Why is vector quantization considered a lossy compression technique?

*****

UNIT 13: BINARY IMAGE PROCESSING

UNIT STRUCTURE

13.1 Learning Objectives

13.2 Introduction

13.3 Binarization

13.4 Mathematical Morphology

13.5 Structuring Element

13.6 Morphological Image Processing

- 13.6.1 Basic Set Theory
- 13.6.2 Logical Operations

13.7 Standard Binary Morphological Operations

- 13.7.1 Dilation
- 13.7.2 Erosion
- 13.7.3 Opening
- 13.7.4 Closing
- 13.7.5 Properties of Morphological Operations

13.8 Application of Morphological Operations

13.9 Let Us Sum Up

13.10 Further Reading

13.11 Answers to Check Your Progress

13.12 Model Questions

13.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- define mathematical morphology and its basis in set theory.
- identify the role of structuring elements in morphological operations.
- recognize the difference between binary and grayscale morphological operations.
- understand and apply basic morphological operations such as erosion, dilation, opening, and closing.

- comprehend the effects of each operation on an image, including their impact on object boundaries and shape.
- apply morphological operations for noise removal, object isolation, and enhancement in binary and grayscale images.
- use opening and closing for removing small noise elements and closing small holes in images.
- explore applications of morphological processing in noise removal, object detection, and segmentation.

13.2 INTRODUCTION

In this unit we will discuss about binary image processing, its significance, applications, and various techniques used to process binary images. The content begins with the concept of binarization, the basic process of converting grayscale or color images into binary formats, followed by efficient and simple image analysis and then goes into detail in mathematical equations in, a powerful framework based on set theory to analyze the shape, structure, and spatial relationships of two images. An important element in shape processing is a predefined structuring element, dimension or kernel that interacts with the image in shape transformation Understanding how classification objects work will give you an insight into the different approaches they take so two images are processed and analyzed. The content will then introduce basic set theory and logical operations, which form the mathematical backbone of morphological image processing.We will explore the basic binary morphology operations, including erosion, dilation, opening and closing, and their properties and practical applications. These operations are the building blocks of complex image analysis tasks such as edge detection, noise filtering, and object segmentation.

13.3 BINARIZATION

Binarization is a process that transforms a grayscale or color image into a binary image. Binary images are also known as *black and white (BW)*

images as the pixels are either black or white. The black pixels are represented as 0 while white pixels are represented as 1. Usually, the black pixels in a binary image represent background and white pixels represent foreground. Thresholding is the most common technique used to convert a grayscale image into binary image. Thresholdingis a technique of image processing in which intensity value of a pixel is set as 1 if its value is greater or equal to a given threshold, otherwise it is set as 0. Thresholding can be categorized into global thresholding and local thresholding. In global thresholding like Otsu method, one threshold is used for the entire image while in local thresholding thresholds are chosen locally for a region or part of an image. The thresholding operation is defined by equation 13.1.

$$BW(i,j) = \begin{cases} 1 & if\ I(i,j) \geq thresh \\ 0 & otherwise \end{cases}$$ ——————-Eq. 13.1

where *BW* is the binary image and *I* is the grayscale image, *(i,j)* represents a pixel position.

Binary images consume less memory and computational power as they have only 1-bitplane. Thus, they are easy to handle and process. Although, high-end machines are available now at an affordable rate, the use of binary images are still relevant. They are useful in analysing the shapes, structures and geometry of the objects present within the image. The objects and structures present in color or grayscale images are often represented using masks. These masks are nothing but binary images, where the objects of interests are appeared as white and rest as black.



CHECK YOUR PROGRESS

Q1. Which of the following techniques is commonly used for image binarization?

a) Gaussian Blur
b) Otsu's Method
c) Sobel Edge Detection
d) Hough Transform

Q2. In image binarization, what does the threshold value determine?

a) The intensity range of colors in the image

b) The separation of foreground from the background

c) The direction of edge gradients

d) The connectivity of pixels

Q3. Adaptive thresholding is particularly useful for:

a) High-resolution images

b) Noisy images

c) Images with varying illumination

d) Edge detection

MATHEMATICAL MORPHOLOGY

Mathematical morphology is a theory used torepresent and process structures and geometry present in images based on set theory, topology, and lattice algebra. In mathematical morphology the sets are used to represents the objects present in images. The pixels belonging to objects represented as one set (foreground) and rest belong to another set (background).It provides a set of operations known as morphological operations to process shapes, sizes, and structures of objects present in images. The basic operations are erosion and dilation.

Mathematical morphological operations are applicable to various image processing tasks including image enhancement, segmentation, edge detection, and feature extraction. They are widely used in medical image analysis, pattern recognition, document processing.

13.4 STRUCTURING ELEMENT

A *structuring element* is the most integral part of mathematical morphology which interacts with an image to perform morphological operations. It is a predefined small matrix which has binary values (either 0 or 1). It defines the size and shape of the neighborhood area used to perform the morphological operations. It determines the impact of any morphological operations on an image. The shape of a structuring element can be round, rectangular, oval, line etc; size can vary depending on the requirements. Figure 13.1 presents 3 examples of structuring elements.

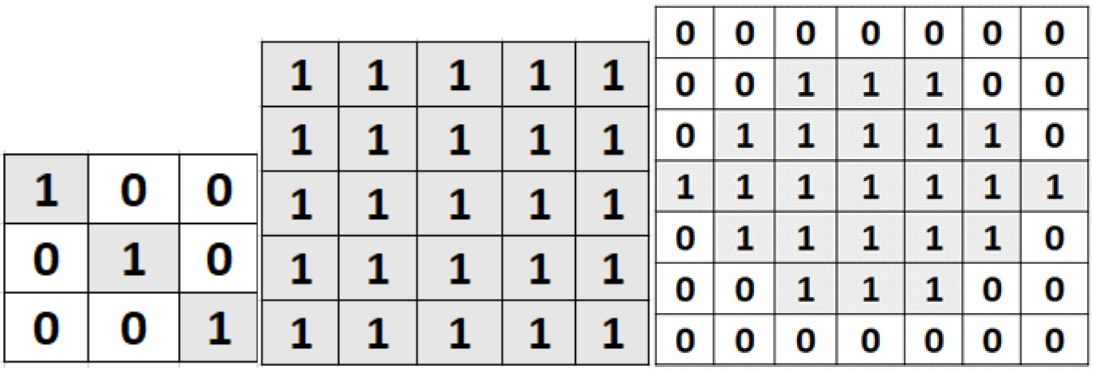


Figure 1.1: Structuring elements with various shape and size. (a) Line structuring element of size 3x3, (b) Rectangular structuring element of size 5x5 and (c) Eliptical structuring element of size 7x7.

During a morphological operation, the structuring element is swamped over the image from left-top corner to right-bottom corner. At every step the central point of the structuring element is placed over a pixel, P of the image currently being processing and the intensity of each pixel in the neighbourhood of P is compared with the value of the corresponding point of the structuring element. When the intensities of all neighbourhood pixels of P match the values of the structuring element, it is called a *fit*. If none matches it is called a *miss* and when there is a partial match it is termed a *hit*. Figure 13. 2 shows a sample structuring element and sample binary image in matrix form. The results of scanning the sample image by the structuring element as given be figure 13.2 is demonstrated in figure 13.4.

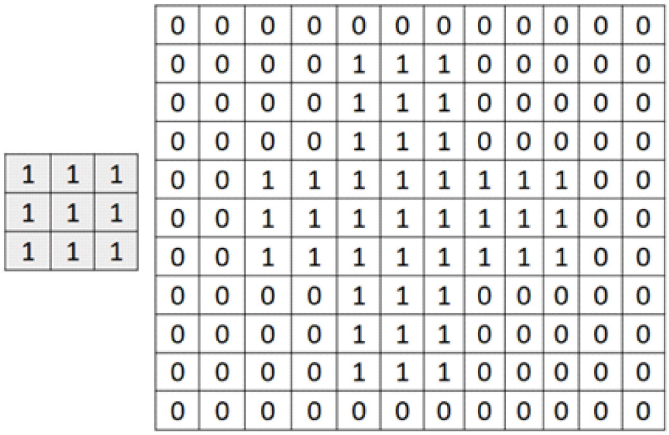


Figure 13.2 (a) 3x3 rectangular structuring element, (b) a sample black and white image

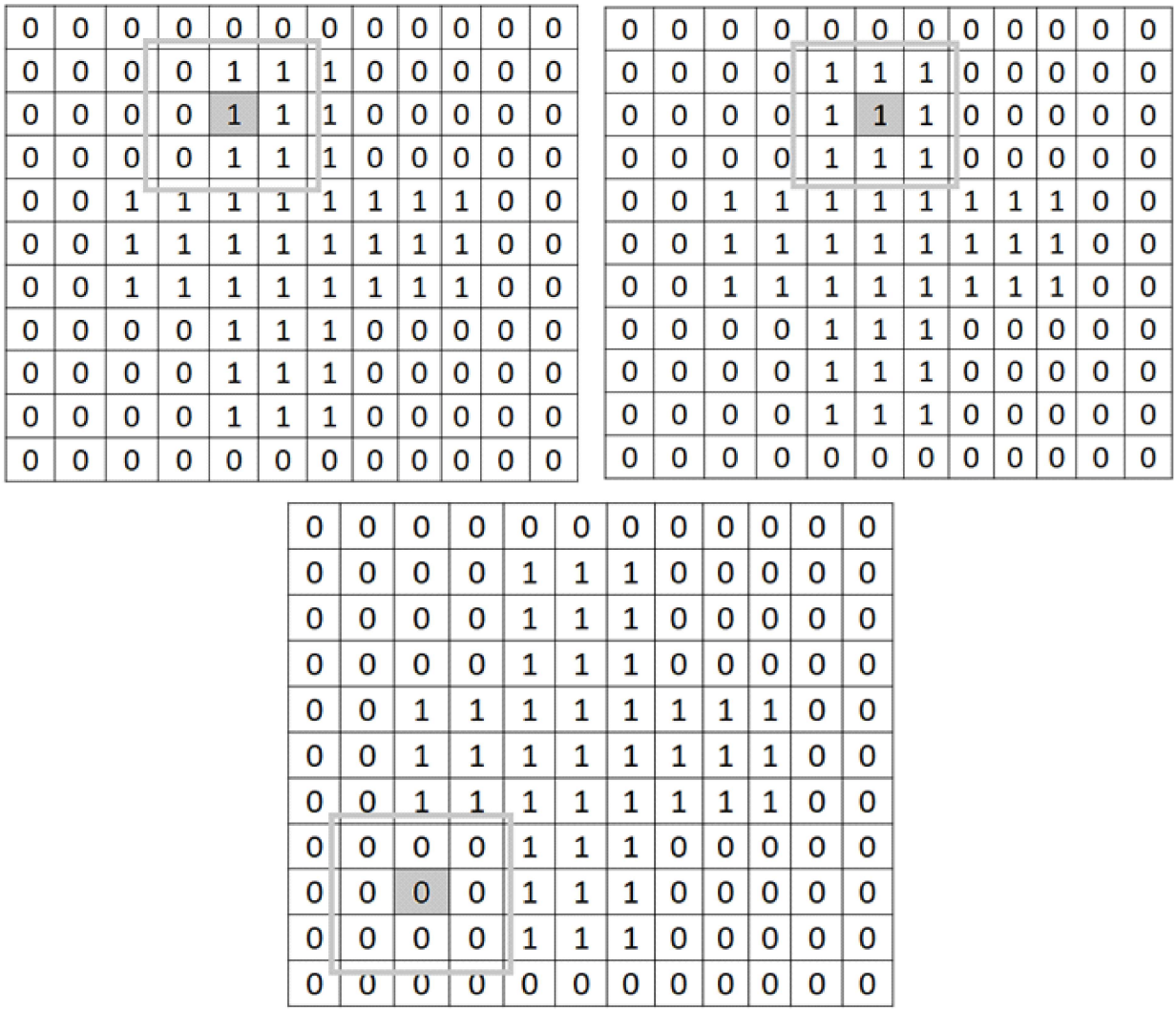


Figure 13.3 Result of scanning the binary image in 13.2 (b) by structuring element in 13.1 (a). (1) Hit, (b) Fit, and (c) Miss



CHECK YOUR PROGRESS

Q4. Mathematical morphology is based on which mathematical theory?

a) Linear Algebra

b) Set Theory

c) Differential Equations

d) Fourier Analysis

Q5. The structuring element in morphological operations is:

a) Always a square matrix

b) A geometric shape used for interaction with the image

c) A gradient mask

d) A histogram of pixel intensities

Q6. What does the “hit” in hit-or-miss transform refer to?

a) The structuring element partially overlapping with the image

b) The structuring element perfectly matching the foreground of the image

c) The detection of edges in the image

d) The removal of noise from the image

Q7. What does the “miss” in hit-or-miss transform refer to?

a) The structuring element matching the background of the image

b) A failed detection of an object

c) Noise left after the operation

d) Partial overlap of the structuring element with the image

Q8. The term “fit” in morphological operations means:

a) The structuring element aligns perfectly within the region of interest

b) The structuring element partially covers the region of interest

c) The structuring element enhances object boundaries

d) The structuring element detects intensity gradients

13.5 MORPHOLOGICAL IMAGE PROCESSING

Morphological image processing allows manipulation of objects and structures found within images. Basic goal is to modify the shapes, boundaries, and connectivity of the geometric structures with the help of a structuring element. Fundamental morphological image processing techniques perform tasks like, shrinkage, expansion, segmentation, noise removal, boundary detection etc. The basis of all morphological operations is basic set theory. Most of the morphological operations applies set

operations along with logical operations. The context of basic set theory and logical operation in relation to morphological image processing explained in the following subsections.

13.5.1 Basic Set Theory

Basic set theory provides the mathematical foundation for most of the image processing techniques including morphological image processing. A set is a collection of distinct elements that represents objects, pixels, or features in an image. In binary images, objects are treated as sets of pixels for the coordinates of their pixels represent an element in the set. Set theory defines various operations that can be used directly in image processing. For instance, **union** and **intersection** are used to form or find the common elements in the sets, just like merging or overlapping regions in an image, as well as to find the pixels that belong only to one of the two sets. **Difference** operations find pixels affecting only one set and not the other; this produces a result from subtracting objects or isolating boundaries. **Complementation** can identify all elements not in each set; this is related to inversion of an image.

Basic morphological operations, erosion and dilation, can be defined using set theory; erosion represents Minkowski subtraction of a structuring element from the image set, while dilation corresponds to Minkowski addition. These set-theoretic principles form the basis for understanding the application of image morphological techniques. Set theory provides such a logical and algebraic foundation that one could describe a part of a requirement to implement various kinds of image processing tasks in this structure. That makes it possible to minimize effort in analysing and manipulating the data's spatial and structural relationships.

13.5.2 Logical Operations

Logical image processing naturally represents the data to be manipulated through pixel-wise operations using Boolean logic

and thus can be easily used to access and extend the power of typical image processing. Logical operations are elementary in image processing, used to manipulate and analyse binary images by performing pixel-wise operations based on Boolean logic. Common logical operations include AND, OR, NOT, and XOR; these relatively simple formulations operate directly on binary pixel values (0 or 1) from one or more images. These operations prove to be an extensive methodology for numerous tasks involved in processing images. The AND operation uses a pixel to be set as 1, only if corresponding pixels of each input image are ones. This can be used to specify intersection areas or masking portions of the image. The OR operation sets a pixel to 1 if any corresponding pixels are 1 in the input images, combining areas. NOT inverts the pixel binary values of the image and switches black to white pixels. XOR finds differences between input images and produces them as 1. It is a powerful tool in preprocessing, segmentation, and image analysis-simple for computation purposes. Its use could involve combining results of different algorithms, isolating objects of interest, or deleting unwanted regions. Thus, logical operations are some of the major tools that make possible various functionalities in image-processing workflows.

13.6 STANDARD BINARY MORPHOLOGICAL OPERATIONS

In this section we will discuss about four basic morphological operations: erosion, dilation, opening and closing and their properties. As mentioned earlier, these operations are performed by placing a structuring element over each pixel of an image.

13.6.1 Erosion

Erosion helps in shrinking object size by removing pixels from object boundaries. Erosion records the pixels whose

neighbourhood fit with the structuring element. Mathematically erosion can be expressed as:

$$I \ominus S = z \in E | S_z \subseteq I \}$$

where S_z is the translation of S_z by a vector *z*, and *E* stands for Euclidean space. The following condition determines the pixel value of the eroded picture *E* at (x, y) when the structural element *S* is positioned at the neighbourhood $N(x, y)$ of image *I* centred at pixel position (x, y):

$$E(x,y) = \begin{cases} 1 & if\ S\ fits\ with\ N(x,y) \\ 0 & otherwise \end{cases}$$



Figure 13.5 (a) Original Image, (b) Result of Erosion applied on image (a) and Result of Dilation applied on image

13.6.2 Dilation

Dilation expands objects by adding pixels to the boundaries. It records the pixels whose neighbourhood hit with the structuring element. Mathematically dilation can be expressed as:

$$I \oplus S = \{z \in E | S_z \cap I / = \Phi\}$$

where S_z is the translation of S_z by a vector *z*, and *E* stands for Euclidean space. The following condition determines the pixel value of the dilated picture *D* at (x, y) when the structural element *S* is positioned at the neighbourhood $N(x, y)$ of image *I* centred at pixel position (x, y):

$$D(x,y) = \begin{cases} 1 & if\ S\ hits\ with\ N(x,y) \\ 0 & otherwise \end{cases}$$

Figure 13.5 shows the results of erosion and dialation.

13.6.3. OPENING

The opening is the result of erosion operation followed by dilation. It is useful in removing noise and separating objects that are close. It can be expressed as:

$$I \quad S = (I \ominus S) \oplus S$$

Figure 13.6 demonstrate the result of opening.



Figure 13.6 (a) Original Image, (b) Result of Opening applied on image (a)

13.6.4 Closing

The closing is does the opposite of opening. In, the resultant image is obtained by performing dilation after erosion. It can be used to fill small gaps between boundary pixels and connecting disjoint components. Mathematically it can be expressed as:

$$I \bullet S = (I \oplus S) \ominus S$$

Figure 13.7Shows the result of closingoperation on a binary image.

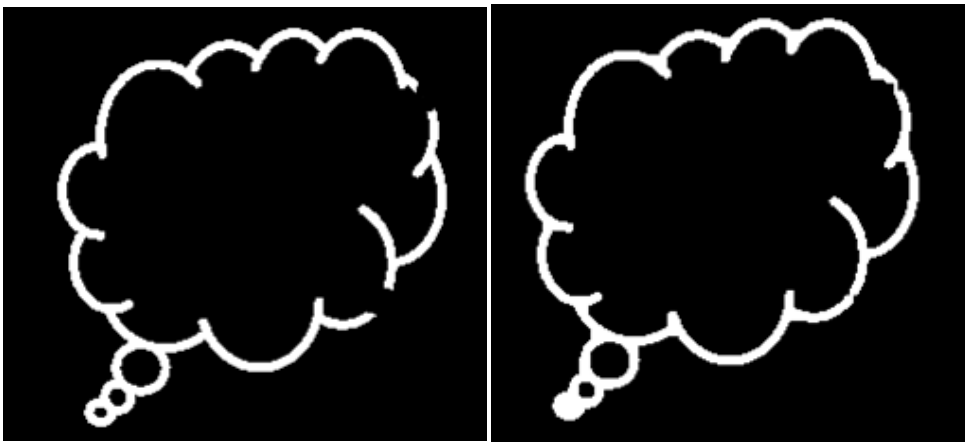


Figure 13.7 (a) Original Image, (b) Result of Closing applied on image (a)

13.7 PROPERTIES OF MORPHOLOGICAL OPERATIONS

The key properties of erosion, dilation, opening and closing are listed below.

i. They are translation invariant.

ii. The dilation is commutative.

iii. The dilation is associative.

iv. Erosion and dilation satisfy the duality.

v. Opening and closing satisfy the duality.

vi. The dilation is distributive over set union.

vii. The erosion is distributive over set intersection.

viii. Opening and closing are idempotent.

ix. Opening is anti-extensive.

13.8 APPLICATION OF MORPHOLOGICAL OPERATIONS

Morphological Operations were widely used for image processing tasks like noise removal, image segmentation, background correction. Few application scenarios of morphological operations are briefly discussed below:

A. Removal of Noise

One of the more popular applications would be in the removal of noise from binary or grayscale images. Erosion can help remove small specks of noise, while dilation can help fill tiny holes within objects. In an efficient way, for example, morphological operations can be used to remove noise from the scanned documents in Optical Character Recognition (OCR) systems so that text recognition can be performed more cleanly and accurately. Figure 13.8 shows the result of noise removal using morphological operators.



Figure 13.8 (a) Original Image, (b) After Noise removal

B. Detection and Segmentation of Objects

Morphological techniques like opening and closing are fundamental in the detection and isolation of objects enclosed in an image. In image segmentation, these operations improve the results of thresholding techniques by eliminating pruned minor details such as a small noise or even connecting broken parts of objects. One possible example can be found in satellite imaging, wherein morphological operations aid in feature detection and segmentation of roads, buildings, and vegetation.

C. Preprocessing Medical Imaging

The principle of morphological processing is important in the field of medical image analysis, wherein it enhances important features while suppressing unwanted information. For instance: Erosion and dilation assist in the segmentation of structures, with cells, tumors, and blood vessels being the most sought-after in segments. Opening can remove small artifacts in X-rays or MRIs. Closing helps to fill in gaps in regions of interest, increasing the visibility of continuous structures such as veins or bones.

D. Shape and Boundary Analysis

Morphological operations can be efficiently applied for shape and boundary analysis. The boundary of objects can be highlighted with basic operations like dilation and erosion. This is extremely important in the case of industrial quality control, where the shape of manufactured components is scrutinized for defects.An example of boundary detectionis shown in figure 13.9.



Figure 13.9 (a) Original Image, (b) Detected Boundary



CHECK YOUR PROGRESS

Q9. Erosion in morphological operations is primarily used to:

a) Remove noise

b) Shrink objects in an image

c) Enhance object boundaries

d) Dilate small objects

Q10. What happens during the erosion operation?

a) The structuring element adds pixels to object boundaries

b) The structuring element removes pixels from object boundaries

c) The image is converted to grayscale

d) Noise is amplified

Q11. Dilation is used to:

a) Fill gaps in objects

b) Reduce object size

c) Remove edges from objects

d) Apply a Gaussian blur

Q12. What is the effect of applying dilation on a binary image?

a) Expands the boundaries of foreground objects

b) Shrinks the boundaries of background objects

c) Converts binary to grayscale

d) Enhances noise in the image

Q13. The opening operation is a combination of:

a) Dilation followed by erosion

b) Erosion followed by dilation

c) Two successive erosions

d) Dilation followed by thresholding

Q14. Which operation is most effective for removing small objects or noise from an image?

a) Closing

b) Opening

c) Erosion

d) Dilation

Q15. Which of the following is an application of morphological operations?

a) Edge detection in binary images

b) Noise removal in grayscale images

c) Feature extraction in medical imaging

d) All of the above



13.9 LET US SUM UP

- Mathematical morphology provides a framework to analyse and process geometric structures in images using set theory, topology, and lattice algebra.
- Mathematical morphology is applied to binary or grayscale images. Binary images are 1-bit images in which 0 represents black and 1represent white.
- Images are treated as sets, wherein objects are represented by one set of pixels and background is represented by another set.
- Erosion, dilation, opening, and closing are the fundamental morphological operations.
- These operations are based on basic set operations like union, intersection, difference and logical AND, OR operations.

- Erosion shrinks objects and dilation expands. Closing helps in separating supposed to be disconnected objects whereas opening helps in reducing gap in the boundary.
- Morphological operations use a predefined binary matrix called as structuring element. A structuring element can be circular, rectangular, line, elliptical etc.
- Morphological image processing is used in image enhancement, segmentation, edge



13.10 FURTHER READING

- "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods, Pearson, 4th Edition (2018).
- "Mathematical Morphology and Its Applications to Image Processing" by Jean Serra, Kluwer Academic Publishers, 1st Edition (1992).
- "Image Processing, Analysis, and Machine Vision" by Milan Sonka, Vaclav Hlavac, Roger Boyle, Cengage Learning, 4th Edition (2014).
- "Morphological Image Analysis: Principles and Applications" by Pierre Soille, Springer, 1st Edition (2003).
- "Fundamentals of Digital Image Processing" by Anil K. Jain, Prentice-Hall, 1st Edition (1989).



13.11 ANSWERS TO CHECK YOUR PROGRESS

Q 1: b) Otsu's Method

Q 2: b) The separation of foreground from the background

Q 3: c) Images with varying illumination

Q 4: b) Set Theory

Q 5: b) A geometric shape used for interaction with the image

Q 6: b) The structuring element perfectly matching the foreground of the image

Q 7: a) The structuring element matching the background of the image

Q 8: a) The structuring element aligns perfectly within the region of interest

Q 9: b) Shrink objects in an image

Q10: b) The structuring element removes pixels from object boundaries

Q11: a) Fill gaps in objects

Q12: a) Expands the boundaries of foreground objects

Q13: b) Erosion followed by dilation

Q14: b) Opening

Q15: d) All of the above



13.12 MODEL QUESTIONS

Q1: Explain the importance of morphological image processing in computer vision and image analysis.

Q2: What is image binarization? Discuss its significance and methods used for thresholding, including Otsu's method and adaptive thresholding.

Q3: Define mathematical morphology. How does it differ from other image processing techniques?

Q4: What is a structuring element in mathematical morphology? Explain its role and types used in morphological operations.

Q5: Discuss the concept of morphological operations and how they are applied to binary images for noise removal and shape analysis.

Q6: Explain how basic set theory concepts, such as union, intersection, and complement, are used in morphological operations.

Q7: Describe the role of logical operations like AND, OR, and NOT in the context of binary image processing.

Q8: Provide a detailed explanation of standard binary morphological operations and their impact on image structure.

Q9: What is dilation in morphological image processing? Explain its working and applications with suitable examples.

Q10: Define erosion and explain how it is used to refine object boundaries or shrink objects in binary images.

Q11: What is the opening operation in mathematical morphology? Discuss its advantages in removing small objects or noise.

Q12: Define the closing operation and explain its application in filling small gaps or holes in binary images.

Q13: Discuss the key properties of morphological operations, including idempotence, duality, and invariance under translation.

Q14: Enumerate and explain the applications of morphological operations in fields like medical imaging, object detection, and character recognition.

Q15: How do morphological operations contribute to feature extraction and preprocessing in machine learning and deep learning pipelines?

*****

UNIT 14: COLOR IMAGE PROCESSING

UNIT STRUCTURE

14.1 Learning Outcomes

14.2 Introduction

14.3 Color Formation

14.4 Human Perception of Color

14.5 Color Model

- 14.5.1 RGB Model
- 14.5.2 CMY Color Model
- 14.5.3 HIS Color Model
- 14.5.4 YIQ Color Model

14.6 Histogram Equalization of ColorImage

14.7 Color Image Filtering

14.8 Let Us Sum Up

14.9 Further Reading

14.10 Answers to Check Your Progress

14.11 Model Questions

14.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- understand color Models and Representations
- understand how color images are captured, stored, and represented digitally in various file formats.
- perform transformations between different color spaces and understand the purpose of such conversions.
- analyzecolor histograms for each channel to assess intensity distribution and apply histogram equalization.
- apply image enhancement techniques (e.g., contrast adjustment, saturation control, filtering) specific to color images.

14.2 INTRODUCTION

Color image processing is focuses on analysing images based oncolor information. Grayscale images represent intensity in terms of one channel, while color images capture pixel intensity in terms of three channels to represent colors as combinations of primary components, such as red, green, and blue (RGB), or other color spaces like HIS, CMY, and YIQ. Color information carries very crucial information required for image analysis for fields like computer vision, human computer interactions.

Color image processing involves various techniques, including color transformation, enhancement, segmentation, and feature extraction, tailored to handle the multidimensional nature of color images. These techniques enable the analysis of images for tasks like object detection, scene understanding, and pattern recognition, where color plays a pivotal role. Moreover, converting between color spaces can simplify complex tasks, such as identifying objects under varying lighting conditions or enhancing image details.

14.3 COLOR FORMATION

The human visual system and the way light interacts with an item combine to create the fascinating phenomena of color. Light is made up electromagnetic waves. Electromagnetic waves have a range of wavelengths that correlate to various hues. When light strikes an item, certain wavelengths are absorbed by the surface while others are reflected. The wavelength reflected determines our perception of an object's color. For an instance, an object appears red because it reflects red light while absorbing the other wavelengths.

Forming colors can also take place through additive or subtractive mixing. Additive color mixing is the combination of the primary colors- red, green, and blue (or RGB)- to see some colors, as in a digital screen or display. When mixed, all three colors produce a white light in equal intensity. In contrast, a subtractive color mix involves the pigments cyan, magenta, and yellow (or CMY). This subtractive process applies mostly in printing,

where certain wavelengths are absorbed and reflect the rest, producing a new color.

Environmental factors play other roles in the formation of colors, which include the intensity of the light source and its composition. For instance, seeing the color of an object can vary depending on the type of artificial light it is illuminated with, such as daylight, fluorescent, or incandescent. Humans see color because of the cones in their eyes: these cones respond to certain wavelengths, and thus the physical phenomenon of color gets translated into vivid, meaningful visual experiences.

14.4 HUMAN PERCEPTION OF COLOR

Color is perceived by humans through a complex interplay of light, the eye and the brain. Light enters through the eye through the cornea and lens; they focus the incoming light on the retina, where two very special types of photoreceptor cells are found, called cones and rods. Rods are sensitive to light and dark, allowing one to see in dim light, while cones are important for color perception. Cones are sensitive to three specific wavelengths, which correspond with receptors for red, green, and blue light.

The signals from cones then go through the different neural pathways in order to perceive many colors at the time. The process involves a trichromatic theory that states that there are three kinds of cones: red, green, and blue which work for different stimuli to produce all the visible colors. The second model is called the opponent process theory, which explains the visual system's responses in terms of opposing color pairs, for example, red-green and blue-yellow.

According to conditioning, color perception is also affected by other adaptive influences such as the unique effects of variations in lighting and nearby colors and individual differences like color blindness. The extremely stable color perceived for all visible objects, under the various conditions of illumination, is the color constancy. Factors beyond physiological mechanisms, such as culture, emotion, and psychology, too, shape the way humans interpret and ascribe meaning to colors.

14.5 COLOR MODEL

Color models are mathematical representations which describe and manipulate colors in a standardized way. These models define how colors are represented as combinations of primary components, making it easier to process, store, and reproduce colors digitally or physically. Different models serve specific applications, such as image processing, printing, and broadcasting. Below are some commonly used color models:

14.5.1 RGB Model

The RGB color model has three primary components: red, green, and blue. Each color component has an intensity value, typically ranging from 0 to 255 in an 8-bit system. The number of pixels used to represent a pixel of an RGB image is called as a pixel depth. As each color component is if 8-bit, the pixel depth of an RGB image is of 24-bit. Thus, this combination of these three components allows for the representation of over 16 million (2^{24}=16,777,216)unique colors. In this model, pure black is represented with (0,0,0), pure red with (255, 0, 0), pure green with (0, 255, 0), pure blue with (0, 0, 255) and pure white is represented with (255, 255, 255). Any intermediate value represents a color based on the contribution of these three primary colors. It is an additive model in which light added to these color components to create hue. It directly correlates to the way human vision perceives light and is particularly suited for devices like screens that emit light.Figure 14.1 demonstrate the RGB color cube and RGB image generation using three color components.

The RGB model is useful mainly for hardware. Electronic displays like televisions, computer screens, smartphones, digital cameras, and projectors use RGB models.

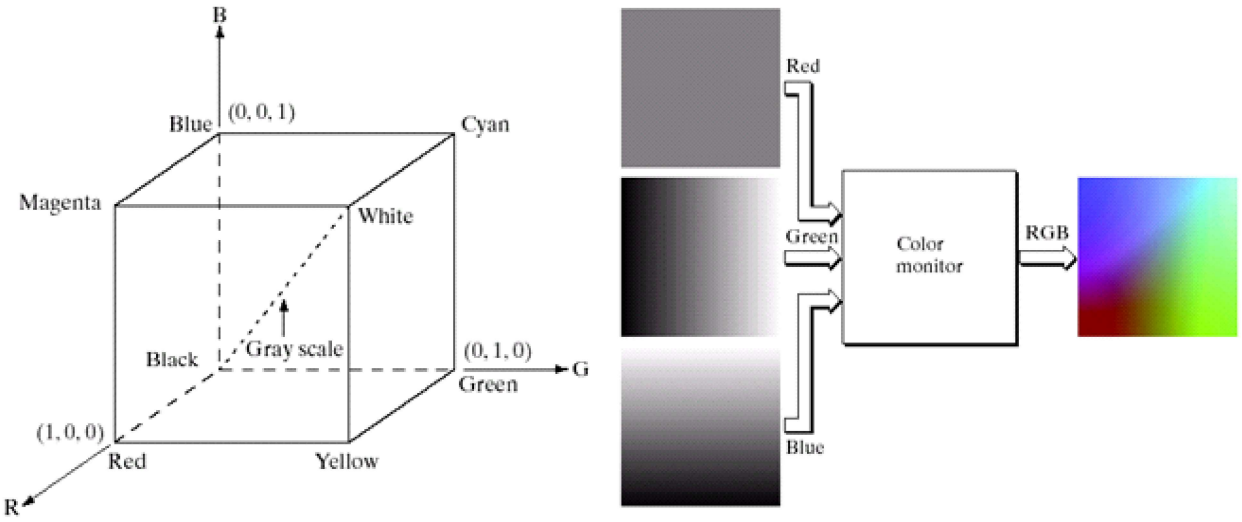


Figure 14.1. (a)The RGB Color cube, (b) Generating RGB image using three color planes (image source [1])

The main advantage of RGB model is ability to produce a huge range of colors making it suitable for applications that require detailed and accurate color production. Also, it is highly compatible with hardware ensuring faster processing and rendering of graphics. However, RGB model is not intuitive for human interpretation as human eyes perceive color in terms of hue, saturation, and brightness rather than red, green, blue. Another limitation comes with inconsistencies of across devices. Different devices come with different calibration for color production leading to difference in color output by different devices for the same amount of red, green, blue values.

In summary, RGB model is essential for all light-based systems, offering excellent precision and compatibility with digital devices. However, for certain tasks like color analysis or manipulation, transformations to more perceptually uniform color models may be necessary.

14.5.2 CMY Color Model

CMY, A subtractive color space which is popularly used for color reproducing in printed and other physical media. Unlike the

CMY model which subtracts the light to reproduce colors, the RGB model adds lights to create colors. In this model for example, cyan is made by reflecting green and blue while absorbing red light, magenta is made by reflected red and blue while absorbing green light, and yellow is made by reflected red and green while absorbing blue light. Forming black by mixing the three combines inks of red, green and blue can depict how every visible colorcan be produced by taking out the red light. Alas the mix creates a dark brownish color which could mean that the other two inks are filled instead of red because the cuff will make Beagle (or purple). To adjust this shortcoming the CMYK model includes black 'K' letter which has been said to stand for Key color black because it is used as the key plate in printing. This detail cut down the volume of ink used as well as increased the contrast of dark shades and details. The transformation from RGB to CMY can be expressed as equation 14.1.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \text{————equation (14.1)}$$

The CMY/CMYK model's operational method relies on the premise that each color comprises its own set of absorbed light wavelengths by ink layers. As an illustration, cyan ink soaks up red light and reflects green and blue, indicating that the color is of a cyan tint. Different colors are produced by adding pixels of various inks and, through that process, subtracting light in some form. For example, blue pigments are formed after cyan and red green rays are subtracted when cyan and magenta are combined.

The CMY/CMYK pattern is important in a variety of applications, especially in printing, where it is the standard for accurate reproduction of characters on paper and other media Widely used in packaging and painting for the home, office, industrial printing systems, physical advertising, packaging and brochures

The strength of the model is its optimization for printing, where it better renders physical characters from digital characters, and reduces ink consumption and maintains high fidelity If you include a black (K) component a, it also increases black tone accuracy, improves contrast and texture for texture. However, the CMY/CMYK model has its limitations. This is not suitable for screens and other light displays, as it relies on subtractive color mixing rather than the additive approach used in RGB. Additionally, converting a digital RGB format to CMYK for printing can cause some color discrepancies due to differences in the color environment.

In summary, the CMY/CMYK model plays an important role in the printing industry, enabling efficient ink handling and accurate color reproduction on physical media even though it is not designed for digital presentation but supports RGB models is brought about by the distinction between digital structures and their physical manifestations.

14.5.3 HIS Color Model

The HSI color model has been formulated to suit the intuitive nature of humans when perceiving and describing colors. Three dimensional spaces of hue, saturation and intensity represent colors whereas chromatic elements are segregated from brightness: achromatic ones. This separation makes the model more comprehensible and functional in respect of many spheres related to image processing.

Hue refers to the type of color such as red, green, or blue. It is often described as an angle along a color wheel which extends from 0° to 360°. Consequently, the red hue corresponds to 0 degrees while the green and blue hues correspond to 120 and 240 degrees respectively. Saturation is the strength of the color, with a value of zero representing a tone of gray while a value higher than zero represents a more vivid color. Intensity, however, is defined as the average brightness of a color which may have been created from a

mixture of all colors, which means all the RGB values in this case as well. The HSI model has a wide range of applications such as in image segmentation, it is used to colorize certain regions of an image to identify several objects in an image. Also, some of the colors are enhanced to improve the images for diagnostic purposes in medical imaging. Furthermore, the model has numerous application in image enhancement where the lightness and contrast of an image can be adjusted separate to the hue and saturation of the image. Figure 14.2 shows the conceptual relationship between RGB and HSI model.

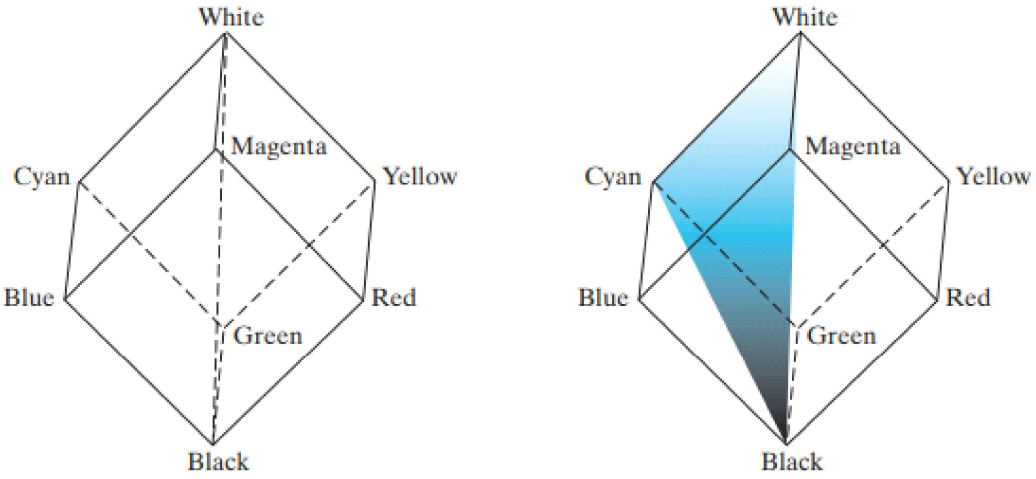


Figure 14.2 Conceptual relationship between RGB and HSI model (image source [1])

The HSI model has one of its best features in the ability to explain color concepts in an intuitive manner. If brightness is separated from the color components, it is possible to change brightness without changing the hue and saturation. This property can be very important in analysis work, like the one done in object recognition, where brightness as well ascolor information can also be processed independently.

The HSI model, however, has its own drawbacks. The conversion from RGB to HSI needs non-linear equations, which are computation intensive. Also, the model explains color non-uniformity,

wherein the visual differences of the color do not vary proportionately with the differences in the HSI values, and this can result in errors in use for some applications.

In summary, the HSI color model offers a simple and efficient means to represent and alter colors. It is a great advantage in almost all cases as it is possible to differentiate the luminance from the colors. Its application in image processing and in medical imaging is going to be most valuable. Its application is however weighed down by its complexity and non-uniformity of its application.

14.5.4 YIQ Color Model

The YIQ model is a color space historically used in television broadcasting, especially under NTSC (National Television Systems Commission) standards. It divides the image into three distinct categories: brightness (Y), in-phase (I), and quadrature (Q). The Y axis represents the bright or gray area of the image, ensuring compatibility with a black-and-white television. The I phase captures color information along the orange-to-teal axis, while the Q phase transmits color information along the purple-to-green axis. Together, the I and Q components produce chrominance, which carries details of the color, while the Y component brightens the image. The transformation from RGB to YIQ can be performed using equation 14.2.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \text{———equation (14.2)}$$

This color model had the necessary features for high-definition use. Analog was a key feature of NTSC systems, enabling the transmission of video signals with color and brightness information. Moreover, the separation of brightness and chrominance improved the bandwidth used. This design took advantage of the fact that human vision is more sensitive to light than color details, allowing for better compression without noticeable loss of quality

The YIQ model has several strengths. Its efficiently distributed bandwidth ensures that more data is loaded into light, which is critical for human perception, and reduces the amount of data needed for chrominance Furthermore, the model was designed to match old black-and-white televisions and white supporting color TV will move backwards towards A smooth transition is ensured in the early days

However, the YIQ model has limitations that have limited its usefulness in modern applications. It has been largely replaced by more advanced color zones such as YUV and YCbCr which are better suited for digital systems.In addition, it is mainly limited to the use of analog video systems, making it mostly irrelevant in modern digital image processing and broadcasting.

In summary, the YIQ model was a leading color space that played an essential role in the television industry. It balanced luminance with chrominance for optimum usage of bandwidth and showed reverse compatibility. However, due to its obsoletion by newer technologies with enhanced performance in digital systems, it has been supplanted.

14.6 HISTOGRAM EQUALIZATION OF COLOR IMAGE

Histogram equalization is a popular image processing technique used to enhance the contrast of an image by redistributing its intensity levels. For color images, this technique should be used carefully to produce naturally balanced colors while enhancing the visual effect. Since histogram equalization applied directly to any color channel (RGB) can distort the color balance of an image, the process often involves converting the image into a color space that separates intensity (or brightness) from chromatic components such as HSV (Hue, Saturation) inside , Value)) or YCbCr. The histogram equalization is then applied only to the intensity component (e.g., the V channel of HSV or the Y channel of YCbCr), excluding the chromatic components (hue and saturation).

After matching, the intensified component is recombined with the original chromatic information, and the image is converted back to the original RGB color space This keeps the colors visually the same, while difference is all significantly increased. Histogram equalization is particularly useful in applications where detail is required, such as medical imaging, satellite imagery, and photography It can help reveal hidden information in areas that are less obvious or overexposed, and making it a valuable tool for both system design and hands-on image analysis. Figure 14.3 shows the result of histogram equalization when applied to a low contrast image.

14.7 COLOR IMAGE FILTERING

Color image filtering is an important technique in image processing. This technique is applied to improve, alter, or even extract features from color images. In grayscale images, filtering is performed on one intensity channel; however, color images contain multiple channels, such as red, green, and blue. Every channel is associated with a particular color component, and filtering can be applied directly to these channels or to derived components in other color spaces, such as HSV or YCbCr. Filtering may be used for smoothing textures, enhancing edges, or even to make features stand out in an image. One of the most common applications of color image filtering is noise reduction, where Gaussian, median, or bilateral filters are used to smooth out irregularities while preserving the most important details.

Gaussian filters apply a weighted averaging process to reduce high-frequency noise, whereas median filters are effective at removing salt-and-pepper noise by replacing each pixel with the median of its neighbors. Bilateral filters go a step further by smoothing an image while retaining the details of edges, hence highly useful in color images in which edges are boundaries to objects. Another important application of color image filtering is edge detection. This can be done by employing Sobel, Prewitt, or Laplacian filters to detect intensity or color transitions and emphasize the boundaries between objects within the image. In general, when working with color

images, these filters are applied to the intensity component of a color space such as HSV because edge information is naturally represented in brightness rather than in chromatic channels. Color-based segmentation or enhancement can also be applied using filtering. For instance, filters can be devised to extract specific color bands or to emphasize

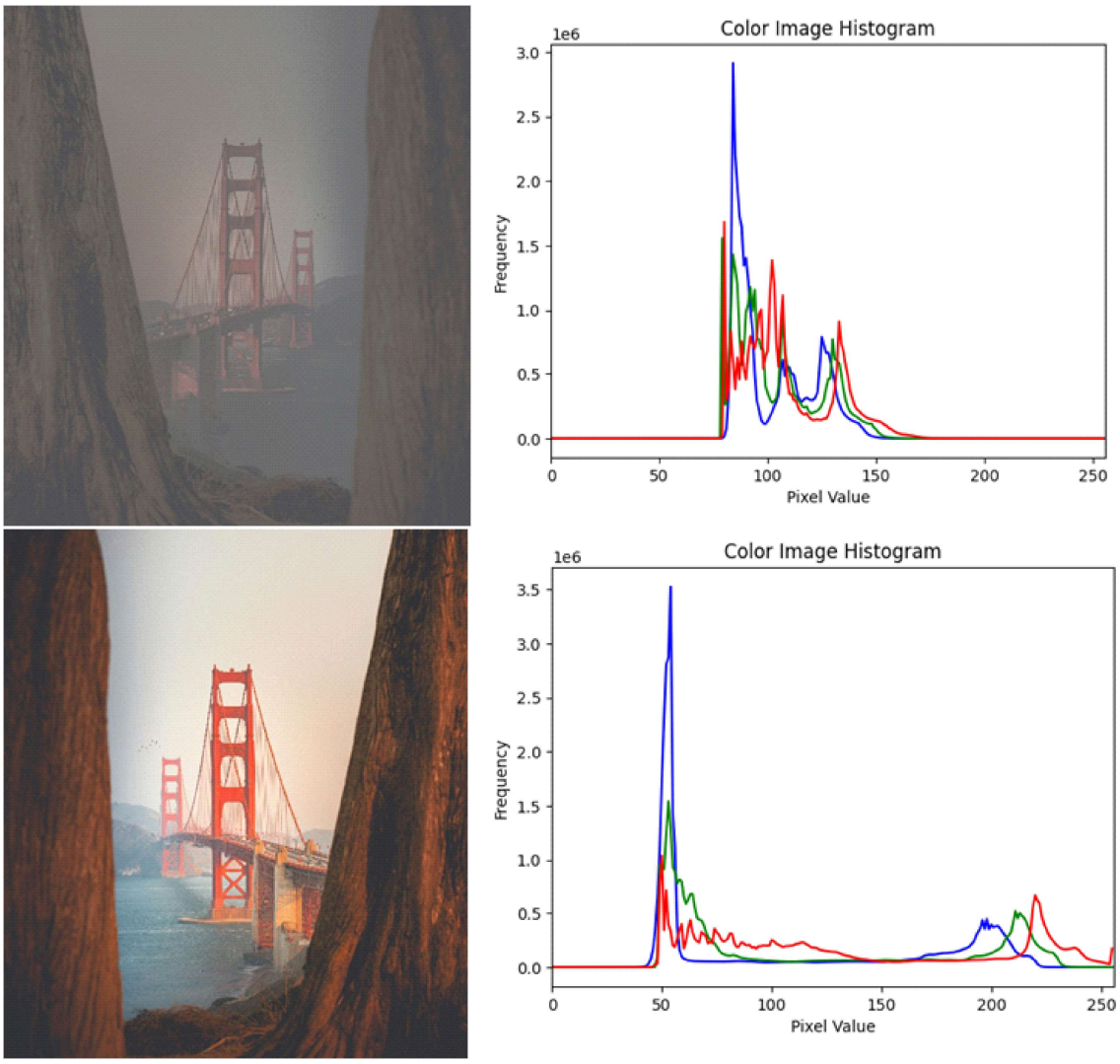


Figure 14.3. (a) A low contrast image (b) Histogram of the image (a), (c) Image (a) after histogram equalization and (d) Histogram of the image (c)

certain colors in an image, and this is helpful in object recognition, medical imaging, and remote sensing applications where specific features may not easily be seen by emphasizing the same. Figure 14.4 shows the results of image smoothing and sharpening filters.

In summary, color image filtering is a versatile tool that is important in tasks such as noise reduction, edge detection, feature enhancement, and segmentation. The choice of filtering method and its application to specific color channels or intensity components depends on the requirements of the task. Properly applied, filtering can significantly improve the quality and usability of color images in many domains.



Figure 14.4 (a) Original Lena Image, (b) image (a) after applying image smooting using a 5x5 average filter and (c) image (a) after image sharpening using the Laplacian (image source [1])



CHECK YOUR PROGRESS

Q1. Which of the following tasks is NOT typically a part of color image processing?

a) Color transformation
b) Image filtering
c) Edge detection
d) Audio signal processing

Q2. How does the human eye perceive color?

a) Through rods sensitive to all wavelengths of light
b) Through cones sensitive to red, green, and blue wavelengths
c) Through a combination of rods and cones sensitive to ultraviolet light
d) Through the optic nerve directly detecting RGB signals

Q3. What type of model is the RGB color model?

a) Subtractive

b) Additive

c) Multiplicative

d) Perceptual

Q4. Why is the CMYK model preferred in printing over the RGB model?

a) It is an additive model better suited for displays.

b) It separates brightness and color components.

c) It includes a black component for better contrast and efficiency.

d) It uses only primary colors for a broader color gamut.

Q5. In the HSI model, what does the “Saturation” component represent?

a) The brightness of the color.

b) The type of color, such as red or green.

c) The purity or vividness of the color.

d) The intensity of red, green, and blue combined.

Q6. What is the primary advantage of the YIQ color model in broadcasting?

a) High compatibility with digital displays.

b) Efficient bandwidth utilization by separating brightness and color details.

c) Superior color accuracy for printing purposes.

d) Backward compatibility with modern RGB systems.

Q7. Which channel is typically used for histogram equalization in the YIQ model?

a) Y (Luminance)

b) I (In-phase)

c) Q (Quadrature)

d) All three channels simultaneously

Q8. What is the main purpose of histogram equalization in color image processing?

a) To reduce noise in color images.

b) To enhance contrast by redistributing pixel intensities.

c) To convert between color models.

d) To sharpen edges in a color image.

Q9. What is the key difference between additive and subtractive color models?

a) Additive models are used for printing, while subtractive models are for displays.

b) Additive models mix light to form colors, while subtractive models mix pigments to absorb light.

c) Additive models use CMY components, while subtractive models use RGB components.

d) Subtractive models are hardware-based, while additive models are perceptual.

Q10. In color image filtering, why might one process the luminance component separately?

a) To reduce computational complexity.

b) To ensure color accuracy is maintained.

c) Because luminance is more critical to human perception than chrominance.

d) To apply histogram equalization directly on chrominance components.



14.8 LET US SUM UP

- Human vision perceives color through cones in the retina, sensitive to red, green, and blue wavelengths.
- Colors are interpreted based on hue, saturation, and brightness.
- RGB is an additive color model combining red, green, and blue light to form colors.

- Widely used in digital displays like monitors, TVs, and cameras.
- Subtractive color model like CMY used in printing, where inks absorb specific wavelengths to create colors.
- CMYK adds a black component for better contrast and efficiency.
- Optimized for physical media but requires conversion from RGB for digital designs.
- Cylindrical color model HSI is based on how humans perceive color and useful in image editing and segmentation for its intuitive representation.
- YIQ model is optimized for human sensitivity to brightness but less relevant in modern digital systems.
- Histogram Equalization (HE) of color images enhances contrast in images by redistributing pixel intensities.
- Color image filtering has a range of applications including noise reduction, edge enhancement, and feature extraction.



14.9 FURTHER READING

- "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods, Pearson, 4th Edition, 2018.
- "Color Image Processing: Methods and Applications" by Rastislav Lukac and Konstantinos N. Plataniotis, CRC Press, 1st Edition, 2006.
- "Image Processing, Analysis, and Machine Vision" by Milan Sonka, Vaclav Hlavac, and Roger Boyle, Cengage, 4th Edition, 2014.
- "Fundamentals of Multimedia" by Ze-Nian Li, Mark S. Drew, and Jiangchuan Liu, Springer, 2nd Edition, 2014.



14.10 ANSWERS TO CHECK YOUR PROGRESS

Q1. d) Audio signal processing

Q2. b) Through cones sensitive to red, green, and blue wavelengths

Q3. b) Additive

Q4. c) It includes a black component for better contrast and efficiency.

Q5. c) The purity or vividness of the color.

Q6. b) Efficient bandwidth utilization by separating brightness and color details.

Q7. a) Y (Luminance)

Q8. b) To enhance contrast by redistributing pixel intensities.

Q9. b) Additive models mix light to form colors, while subtractive models mix pigments to absorb light.

Q10. c) Because luminance is more critical to human perception than chrominance.



14.11 MODEL QUESTIONS

Q 1: Explain the significance of color image processing and its key applications in various domains.

Q 2: Describe how humans perceive color and the role of cones in the human visual system.

Q 3: What are the primary differences between additive and subtractive color models? Provide examples of each.

Q 4: Discuss the RGB color model in detail, including its working principle, applications, and limitations.

Q 5: Explain the CMY/CMYK color model. Why is the black (K) component important in printing?

Q 6: Compare the RGB and CMYK color models. Highlight their advantages and disadvantages in practical applications.

Q 7: What is the HSI color model? Describe its components and explain how it aligns with human perception of color.

Q 8: Discuss the YIQ color model, its components, and why it is suitable for analog television broadcasting.

Q 9: Illustrate how histogram equalization is applied to color images and explain its importance in enhancing image contrast.

Q10: Describe the process and benefits of filtering color images. How is it different from filtering grayscale images?

Q11: Explain the transformations required to convert an image from the RGB color space to the HSI color space.

Q12: Discuss the strengths and limitations of the YIQ model compared to modern color spaces like YUV or YCbCr.

Q13: What challenges are involved in applying histogram equalization to color images compared to grayscale images?

Q14: Describe the role of luminance and chrominance in color models like YIQ and explain why they are separated.

Q15: How does color image filtering enhance image quality? Provide examples of techniques like smoothing and sharpening.

*****

UNIT 15: WAVELET-BASED IMAGE PROCESSING

UNIT STRUCTURE

15.1 Learning Objectives

15.2 Introduction

15.3 Wavelet

15.4 Discrete Wavelet Transform

15.5 Haar Wavelet

15.6 Filter Bank

15.7 Embedded Image Coding

15.8 Let Us Sum Up

15.9 Further Reading

15.9 Answers to Check Your Pogress

15.10 Model questions

15.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- explain the concept of wavelet
- explain the theoretical background of continuous wavelet transform
- describe the significance of discrete wavelet transform
- describe the usefulness of Haar Wavelet Transform
- explain the significance of filter bank
- explain the significance of embedded image encoding

15.2 INTRODUCTION

The Fourier Transform has long been a fundamental mathematical tool in signal and image processing, enabling the analysis and optimization of spectral characteristics in the frequency domain. However, its global nature can limit its effectiveness when handling localized or transient features in images. Wavelet transforms provide a multi-resolution approach that captures both frequency and spatial information, addressing this limitation. This capability has contributed to the widespread adoption of wavelet-based

techniques across a variety of image processing applications. Wavelet transforms decompose an image into a series of scaled and shifted versions of a base function referred to as a wavelet. In contrast to the infinite sinusoidal waves of the Fourier Transform, wavelets are localized waveforms that offer a powerful mechanism for analyzing image features at different scales and locations. This multi-resolution approach allows for the efficient representation of both smooth regions and sharp transitions, rendering wavelets particularly well-suited for tasks such as image compression, denoising, and feature extraction. The adaptability of wavelets to local image characteristics has driven the development of numerous wavelet families, each tailored to specific applications and image properties.

15.3 WAVELET

Wavelets are waveforms that are localized and confined within specific, well-defined time and frequency ranges, in contrast to the sinusoidal waves employed in Fourier analysis, which are confined solely to the frequency domain but extend infinitely in time. This unique localization property of wavelets, in both time and frequency, makes them well-suited and advantageous for the analysis of signals and images containing transient features or discontinuities. The ability to localize in both the time and frequency domains allows wavelets to effectively capture and characterize the temporal and spectral properties of the input data, which is crucial for applications such as feature extraction, signal denoising, and image compression.

A wavelet is a function $\psi(t)$ (often called the “mother wavelet”) that satisfies the admissibility condition:

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(f)|^2}{|f|} df < \infty$$

where $\hat{\psi}(f)$ is the Fourier transform of $\psi(t)$. This condition essentially ensures that the wavelet has zero mean and sufficient decay in both time and frequency domains.

From the mother wavelet, a family of wavelets is generated through scaling and shifting:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a})$$

where:

- a is the scaling parameter, which controls the width or frequency of the wavelet. Larger values of a correspond to lower frequencies (wider wavelets) and vice-versa.
- b is the translation parameter, which shifts the wavelet in time.

Continuous Wavelet Transform

The *Continuous Wavelet Transform (CWT)* of $x(t)$ a signal is then defined as the inner product of the signal with the scaled and shifted wavelet:

$$CWT(a,b) = \int_{-\infty}^{\infty} f(t)\frac{1}{\sqrt{|a|}}\psi^*(\frac{t-b}{a})dt$$

where:

- $\psi(t)$ is the mother wavelet, a localized waveform.
- a is the scaling parameter, controlling the wavelet's width (frequency).
- b is the translation parameter, shifting the wavelet in time.
- $\psi^*(t)$ denotes the complex conjugate of $\psi(t)$.

The CWT provides a time-frequency representation of the signal, showing how the frequency content changes over time.

15.4 DISCRETE WAVELET TRANSFORM

The Discrete Wavelet Transform is a powerful mathematical tool used for analyzing signals and images at different scales and resolutions. It's derived from the Continuous Wavelet Transform but differs in how it samples the scale and translation parameters. While the CWT uses continuous values for these parameters, the DWT restricts them to discrete values, typically powers of two. This discretization makes the DWT computationally efficient and suitable for digital implementation.

The DWT discretizes the scaling and translation parameters:

$$DWT(j,k) = \sum_{n=0}^{N-1} f(n)2^{-j/2}\psi(2^{-j}n - k)$$

where:

- j is the scale parameter (integer).

- k is the translation parameter (integer).
- N is the length of the discrete signal $f(n)$.

The CWT provides a highly redundant representation of the signal, meaning it contains more information than necessary for perfect reconstruction. This redundancy can be beneficial for certain applications but comes at the expense of increased computational complexity. In contrast, the DWT, by utilizing discrete scales and translations, achieves a non-redundant or critically sampled representation. This reduction in redundancy significantly decreases the computational burden, rendering the DWT more practical for many applications. In essence, the DWT offers a more practical and efficient approach to wavelet analysis compared to the CWT, while still retaining the ability to analyze signals at multiple resolutions.

The Discrete Wavelet Transform is extensively utilized across diverse applications due to its capacity to analyze signals at multiple resolutions, effectively capturing both frequency and spatial information. In the realm of image and video compression, such as JPEG2000, DWT plays a crucial role in reducing data redundancy by decomposing images into approximation and detail coefficients. Moreover, it finds application in denoising and signal processing, allowing the separation of noise from meaningful data while preserving critical features. Within the domain of medical imaging, DWT enhances images and compresses large datasets, facilitating efficient storage and transmission. Additionally, DWT is employed in biometric systems for feature extraction and in finance for time-series data analysis, establishing its versatility across numerous domains.

Implementation of Discrete Fourier Transform using Python's PyWavelets package:

```
import matplotlib.pyplot as plt

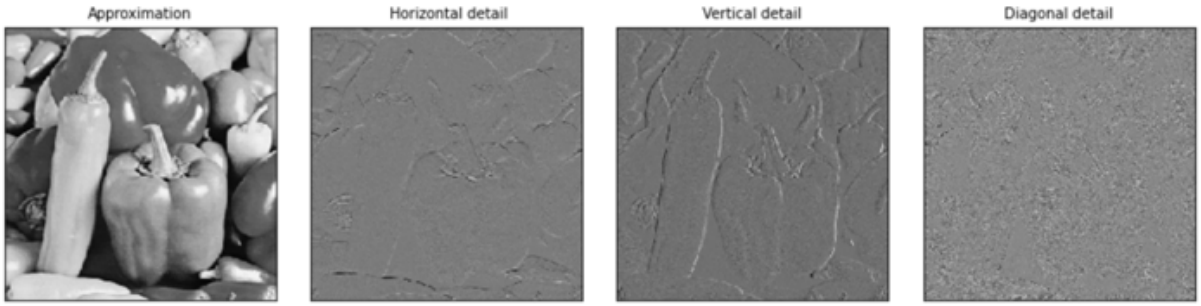
import pywt
import pywt.data

img = cv2.imread('peppers_gray.tif',0)

titles = ['Approximation', ' Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = pywt.dwt2(img, 'bior1.3')
LL, (LH, HL, HH) = coeffs2
fig = plt.figure(figsize=(12, 3))
for i, a in enumerate([LL, LH, HL, HH]):
    ax = fig.add_subplot(1, 4, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)
    ax.set_title(titles[i], fontsize=10)
    ax.set_xticks([])
    ax.set_yticks([])

fig.tight_layout()
plt.show()
```

Output:



15.5 HAAR WAVELET

The Haar wavelet is one of the simplest and most widely used wavelets in the domains of signal processing and numerical analysis. It serves as a foundational concept for understanding the fundamental principles of wavelet transforms and proves particularly advantageous for applications that require fast, computationally efficient transformations.

The Haar mother wavelet, $\psi(t)$ is defined as:

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 0.5, \\ -1 & \text{if } 0.5 \leq t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

The corresponding scaling function, $\phi(t)$ is:

$$\phi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

The key features of Haar Wavelet are as follows:

- **Simplicity**: Its simple mathematical form, consisting of just constant values (+1, -1, and 0), makes it easy to understand, implement, and compute. This simplicity contributes to its computational efficiency, making it suitable for real-time applications.
- **Compact Support**: The Haar wavelet is only non-zero over a finite interval. This localized nature allows it to effectively capture sharp transitions and discontinuities in signals and images.
- **Orthogonality**: The Haar wavelet and its scaled and shifted versions are orthogonal to each other. This property is crucial for efficient signal decomposition and reconstruction, as it ensures that the wavelet coefficients represent independent pieces of information.
- **Fast Computation**: Due to its simple structure and orthogonality, the Haar wavelet transform can be computed very quickly using fast algorithms. This makes it attractive for applications where computational speed is critical.
- **Edge Detection**: The Haar wavelet's sensitivity to abrupt changes makes it well-suited for detecting edges and discontinuities in images. This is exploited in various image processing tasks, such as feature extraction and object recognition.

Haar Wavelet Transform

The Haar wavelet transform decomposes a signal into approximation (low-frequency components) and detail (high-frequency components) coefficients. It is performed iteratively, allowing multilevel decomposition.

15.6 FILTER BANK

Wavelet-based image processing relies on filter banks to perform the Discrete Wavelet Transform. These filter banks decompose an image into distinct frequency sub-bands, effectively separating the image's features at various scales. The image undergoes a series of high-pass and low-pass filtering operations. The high-pass filter extracts the high-frequency details (edges, textures), while the low-pass filter captures the low-frequency information (smooth areas). Following the filtering step, the output is down-sampled by a factor of 2, reducing the data size while maintaining essential information. A multi-resolution representation of the image is created by iteratively applying this process to the low-pass output from the previous step. To reconstruct the original image, the process is reversed using up-sampling and inverse filters.

Implementation of Filter Bank with Haar Wavelet using Python's PyWavelets package:

```
import pywt
import numpy as np
import cv2
import matplotlib.pyplot as plt

def apply_dwt_filter_bank(image, wavelet='haar', level=1):
    coeffs = pywt.wavedec2(image, wavelet, level=level)
    return coeffs

def reconstruct_from_dwt(coeffs, wavelet='haar'):
    return pywt.waverec2(coeffs, wavelet)

image_path = 'peppers_gray.tif'
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

# Apply DWT filter bank
wavelet_type = 'haar'
level = 2 # Decomposition level
coeffs = apply_dwt_filter_bank(image, wavelet=wavelet_type, level=level)

# Display approximation and detail coefficients
cA = coeffs[0]
details = coeffs[1:]  # List of (cH, cV, cD) at each level
```

Plotting the images based in the different coefficients:

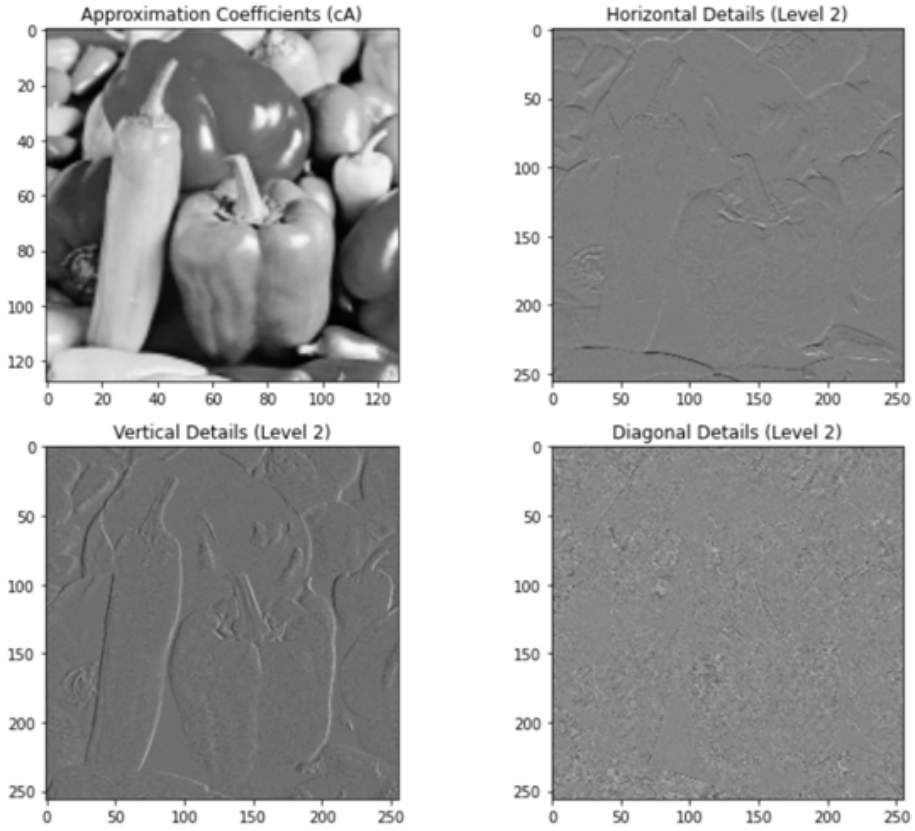
```
plt.figure(figsize=(10, 8))
plt.subplot(2, 2, 1)
plt.title("Approximation Coefficients (cA)")
plt.imshow(cA, cmap='gray')

for i, (cH, cV, cD) in enumerate(details):
    plt.subplot(2, 2, 2)
    plt.title(f"Horizontal Details (Level {i+1})")
    plt.imshow(cH, cmap='gray')

    plt.subplot(2, 2, 3)
    plt.title(f"Vertical Details (Level {i+1})")
    plt.imshow(cV, cmap='gray')

    plt.subplot(2, 2, 4)
    plt.title(f"Diagonal Details (Level {i+1})")
    plt.imshow(cD, cmap='gray')

plt.tight_layout()
plt.show()
```



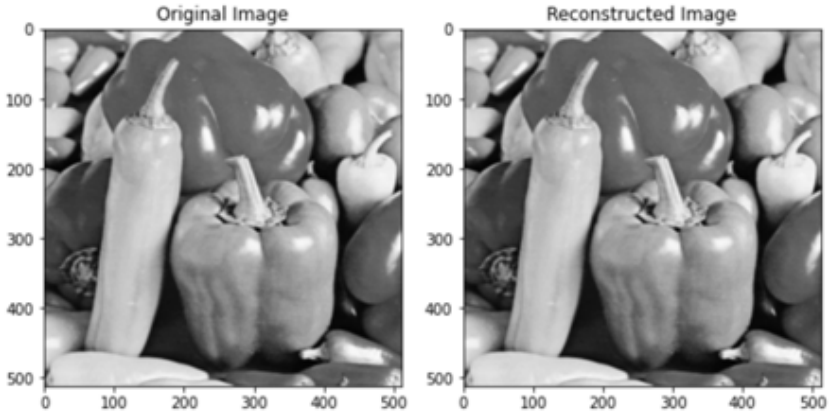
Plotting the original and the reconstructed image:

```
# Reconstruct the image
reconstructed_image = reconstruct_from_dwt(coeffs, wavelet=wavelet_type)

# Show original and reconstructed images
plt.figure(figsize=(8, 4))
plt.subplot(1, 2, 1)
plt.title("Original Image")
plt.imshow(image, cmap='gray')

plt.subplot(1, 2, 2)
plt.title("Reconstructed Image")
plt.imshow(reconstructed_image, cmap='gray')

plt.tight_layout()
plt.show()
```



15.7 EMBEDDED IMAGE CODING

Embedded image coding, also known as *embedded zerotree wavelet coding*, is a powerful compression technique used in wavelet-based image processing. It generates a bitstream that is progressively refinable, meaning that the image quality improves as more bits are received.

Embedded image coding's key advantage is its capacity for progressive image transmission. This feature proves valuable in bandwidth-constrained scenarios or when quick previews are required before downloading the full-resolution image. Additionally, embedded coders provide granular control over the bit rate, facilitating compression to target file sizes.

The embedded bitstream's ability to be truncated at any point further enables the generation of lower-resolution or lower-quality image versions as needed.



CHECK YOUR PROGRESS

1. Which of the following best describes a key advantage of wavelet transforms over Fourier transforms?
   A. Wavelets are easier to implement
   B. Wavelets are global in nature
   C. Wavelets capture both frequency and spatial information
   D. Fourier transforms are better for localized features

2. The Discrete Wavelet Transform (DWT) differs from the Continuous Wavelet Transform (CWT) in that DWT:
   A. Uses continuous scaling parameters
   B. Is computationally more complex
   C. Provides a redundant representation
   D. Uses discrete scales and translations

3. Which wavelet is the simplest and often used for introductory learning and fast computation?
   A. Daubechies wavelet
   B. Haar wavelet
   C. Morlet wavelet
   D. Biorthogonal wavelet

4. In wavelet decomposition, high-pass filters are used to extract:
   A. Low-frequency components (smooth areas)
   B. Noise components
   C. High-frequency components (edges and textures)
   D. Color information

5. What is the primary purpose of embedded image coding in wavelet-based image processing?
   A. To increase image resolution
   B. To create infinite compression
   C. To enable progressive image transmission
   D. To reduce color distortions

6. Which of the following properties is NOT associated with the Haar wavelet?
   A. Compact support
   B. Orthogonality
   C. Continuous smooth curve
   D. Fast computation



15.8 LET US SUM UP

- Wavelet transforms provide a multi-resolution approach to image processing by capturing both spatial and frequency information, making them highly effective for localized and transient feature analysis.
- Unlike the Fourier Transform, which uses global sine and cosine functions, wavelets are localized functions capable of analyzing signals at different scales and positions.
- A wavelet is a scaled and shifted version of a mother wavelet function, satisfying the admissibility condition, which ensures it has a zero mean and is suitable for signal decomposition.
- The Continuous Wavelet Transform (CWT) offers a redundant, high-resolution analysis of signals in time-frequency space, but it is computationally intensive.
- The Discrete Wavelet Transform (DWT) uses discrete values of scale and translation, resulting in a compact and efficient representation suitable for digital implementation.

- DWT is widely used in image compression (e.g., JPEG2000), denoising, biometric feature extraction, medical imaging, and financial data analysis due to its efficiency and multi-resolution capability.
- The Haar Wavelet is the simplest wavelet, offering fast computation, compact support, and orthogonality, making it suitable for edge detection, signal decomposition, and real-time applications.
- Filter banks are essential components in DWT, separating an image into high-frequency and low-frequency components using high-pass and low-pass filters, followed by down-sampling.
- The reconstruction process in wavelet-based systems uses inverse filters and up-sampling to recover the original image from its wavelet coefficients.
- Embedded Image Coding produces a progressive bitstream that allows scalable and efficient image compression, ideal for applications with bandwidth limitations or quality scalability requirements.
- Wavelet transforms are especially powerful for handling images with sharp edges, discontinuities, and localized patterns, outperforming traditional transforms in many practical scenarios.



15.9 FURTHER READING

- "Digital image processing and pattern recognition", Pakhiramalay K, PHI
- "Digital image processing", Sundaramurthy Sridhar, Oxford
- "Fundamentals of Digital Image Processing", S Annadurai, R Shanmugalakshmi, Pearson



15.10 ANSWERS TO CHECK YOUR POGRESS

1. C – Wavelets capture both frequency and spatial information.
2. D – DWT uses discrete scales and translations.
3. B – The Haar wavelet is the simplest and widely used.

4. C – High-pass filters extract high-frequency components like edges and textures.
5. C – Embedded image coding allows progressive transmission where quality improves as more data is received.
6. C – Haar wavelet is not a continuous smooth curve; it is piecewise constant.



15.11 MODEL QUESTIONS

Q 1: Explain the concept of wavelet transforms. How do they overcome the limitations of Fourier transforms in image processing?

Q 2: Describe the structure and mathematical formulation of the Continuous Wavelet Transform (CWT). What is its significance in signal and image analysis?

Q 3: What is the Discrete Wavelet Transform (DWT)? Discuss its advantages and applications in digital image processing.

Q 4: Write a detailed note on the Haar Wavelet. Discuss its mathematical properties and its role in efficient image analysis.

Q 5: What are filter banks in wavelet-based image processing? Explain how they function in the decomposition and reconstruction of images.

Q 6: Discuss the concept and advantages of embedded image coding. How does it facilitate progressive image transmission and scalable compression?

Q 7: Compare and contrast the Haar Wavelet with other wavelet families in terms of computational efficiency, support, and application.

Q 8: Illustrate with examples how wavelet transforms are applied in real-world scenarios such as medical imaging, biometric systems, and video compression.

*****