Introduction to Image processing: overview, Nature of IP, IP and its related fields, Digital Image representation, types of images.

Digital Image Processing Operations: Basic relationships and distance metrics, Classification of Image processing Operations.

Overview

What is Image Processing?

Image processing is a technique used to manipulate visual information, typically in the form of digital images, to enhance their quality or to extract useful information from them. It involves the following steps:

1. **Image Acquisition:**
   - Capturing images using devices like cameras or scanners.
2. **Digitization:**
   - Converting the captured image into a digital form through sampling (discretizing the spatial domain) and quantization (discretizing the amplitude).
3. **Image Enhancement:**
   - Improving the visual appearance of an image for better human interpretation, such as adjusting brightness and contrast or removing noise.
4. **Image Analysis:**
   - Extracting meaningful information from the image for automatic interpretation by machines, such as identifying objects, patterns, or features within the image.
5. **Image Compression:**
   - Reducing the size of the image file for efficient storage and transmission without significant loss of quality.
6. **Image Restoration:**
   - Recovering an image that has been degraded by using various techniques to reverse known distortions.

Objectives of Image Processing:

- Enhance the quality of pictorial information for better human interpretation.
- Enable automatic machine interpretation of images.

Nature of IP

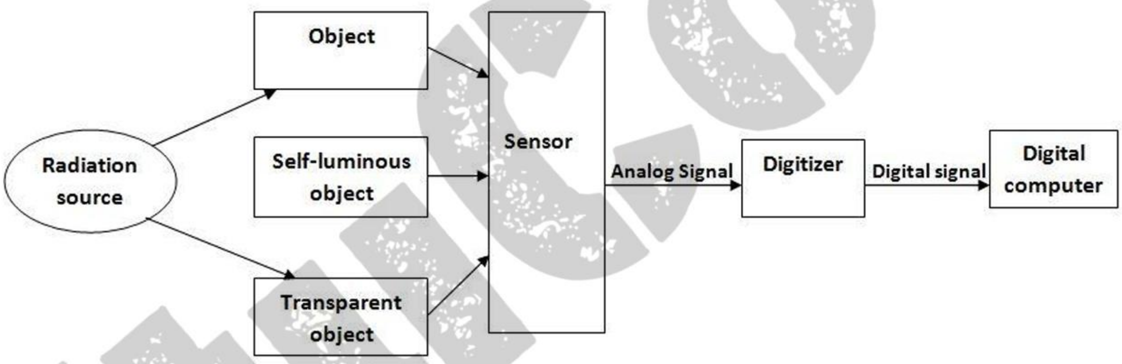


Figure 4.1: Image processing environment

Ways of Acquiring an Image:

1. **Reflective Mode Imaging:**
   - **Description:** In reflective mode imaging, a light source illuminates the object, and the reflected light is captured by a sensor.
   - **Examples:**
     - **Video Cameras:** Capture moving images by recording reflected light in real-time.
     - **Digital Cameras:** Capture still images by recording reflected light.
     - **Scanners:** Capture high-resolution images of documents by moving a light source across the document and recording the reflected light.
   - **Process:**
     - The sensor converts light energy into electrical signals, which are then processed to form a digital image.
2. **Emissive Type Imaging:**
   - **Description:** Emissive type imaging captures images from objects that emit their own light (self-luminous objects) without the need for an external light source.
   - **Examples:**
     - **Thermal Imaging:** Captures images based on the heat emitted by objects. Useful in low-light conditions.
     - **Magnetic Resonance Imaging (MRI):** Captures detailed internal body images using magnetic fields and radio waves.
     - **Positron Emission Tomography (PET):** Captures images of metabolic processes in the body using radioactive tracers.
   - **Process:**

- Sensors detect the radiation emitted directly by the object to form an image.

3. **Transmissive Imaging:**
   - **Description:** In transmissive imaging, a light source passes through the object, and the transmitted light is captured by a sensor.
   - **Examples:**
     - **X-ray Imaging:** Captures images of the internal structure of objects, particularly useful in medical diagnostics.
     - **Microscopic Imaging:** Captures detailed images of small objects using transmitted light.
     - **Ultrasound Imaging:** Captures images based on the transmission and reflection of sound waves.
   - **Process:**
     - Radiation passes through the object, and the attenuated radiation is sensed to form an image.

Types of Image Processing:

1. **Optical Image Processing:**
   - **Description:** Optical image processing involves manipulating images using optical techniques such as lenses and coherent light beams.
   - **Key Aspects:**
     - **Radiation Source:** Light sources like the sun, lamps, and other luminous objects.
     - **Object:** The target for imaging, which can be 2D, 3D, or multidimensional.
   - **Human Vision:** An example of optical image processing, where the eye perceives 2D projections of 3D scenes.
   - **Objective:** Capture and manipulate the 2D distribution of light to gather information about the object.
2. **Analog Image Processing:**
   - **Description:** Analog image processing deals with continuous signals using analog circuits.
   - **Examples:**
     - **Film-Based Imaging Systems:** Use photographic film to capture images.
     - **Analog TV Signals:** Transmit visual information using continuous electrical signals.
   - **Process:**
     - Continuous function f(x,y) representing the image is processed using analog methods.
3. **Digital Image Processing:**

- **Description:** Involves converting analog images to digital form for processing using computers.

- **Process:**
  - **Sampling:** Converts a continuous image into a discrete set of pixels. This step is reversible.
  - **Quantization:** Converts sampled values into discrete integers. This step is not reversible.
- **Applications:**
  - **Image Enhancement:** Improving image quality.
  - **Object Counting:** Identifying and counting objects within an image.
  - **Image Analysis:** Extracting meaningful information from images.

Advantages of Digital Images:

1. **Easy Post-Processing:**
   - Software tools allow for easy correction and enhancement of images.
2. **Easy Storage:**
   - Digital images can be efficiently stored on digital media like hard drives, SSDs, and cloud storage.
3. **Easy Transmission:**
   - Images can be easily shared over the internet and other digital networks.
4. **Environmentally Friendly:**
   - Digital imaging avoids the use of chemicals required for film processing.
5. **Ease of Operation:**
   - Digital cameras and devices are user-friendly and often automated.

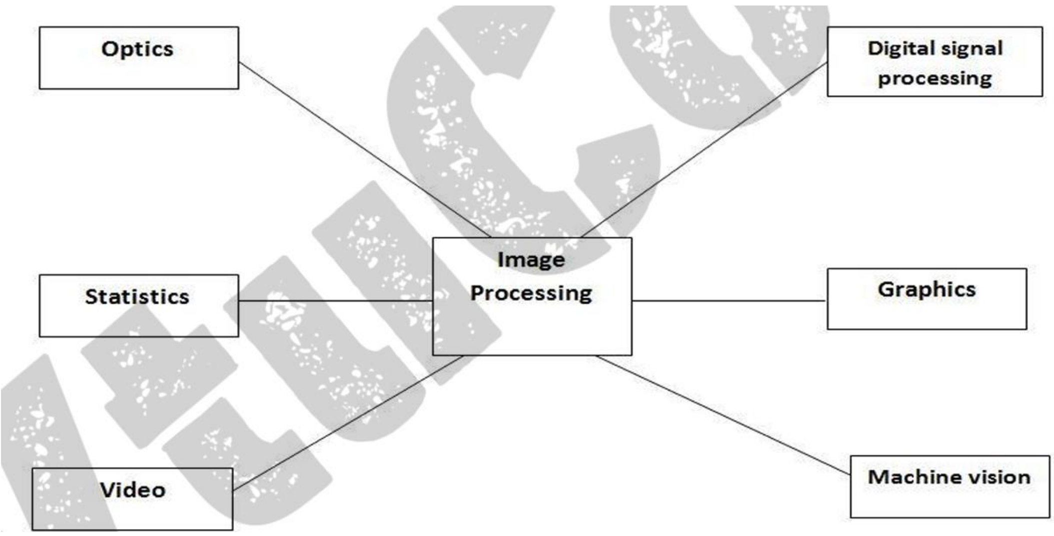
Disadvantages of Digital Images:

1. **Initial Cost:**
   - High-quality digital imaging equipment can be expensive.
2. **Sensor Issues:**
   - Digital sensors can consume a lot of power and may have reliability issues.
3. **Security Concerns:**
   - Digital images can be susceptible to unauthorized access and tampering during storage and transmission.

Final Image Form:

- **Display Image:**
  - For human viewing, digital images must be converted back to optical form using digital-to-analog conversion. This allows the images to be displayed on screens or printed on paper.

IP and its related fields



1) Image Processing and Computer Graphics

- **Computer Graphics:**
  - **Focus:** Deals primarily with vector data, which are mathematical representations of images using lines and curves.
  - **Use Case:** Often used to create images from numerical data, producing outputs like animations, simulations, and abstract drawings.
- **Image Processing:**
  - **Focus:** Deals with raster data or bitmaps, which are stored in a 2D matrix form representing real images.
  - **Use Case:** Enhances and analyzes images to improve their quality or extract information.
- **Relationship:**
  - Image processing is a logical extension of computer graphics. While computer graphics generate images from data, image processing refines and interprets these images.

2) Image Processing and Signal Processing

- **Signal Processing:**
  - **Focus:** Involves the analysis and manipulation of one-dimensional signals (e.g., audio signals).

- **Image Processing:**
    - **Focus:** Deals with visual information that is often in two or more dimensions (e.g., images and videos).
- **Relationship:**
    - Image processing is an extension of signal processing. Techniques used in signal processing can be adapted and extended to handle multi-dimensional visual data.

3) Image Processing and Machine Vision

- **Machine Vision:**
    - **Goal:** Interprets images to extract physical, geometric, or topological properties for automation and inspection.
    - **Applications:** Industrial automation, quality control, and automated inspection systems.
- **Computer Vision:**
    - **Goal:** Mimics human vision to understand and interpret scenes.
    - **Applications:** Scene understanding, object recognition, and autonomous systems.
- **Relationship:**
    - Image processing provides foundational techniques that are used in machine vision and computer vision for further analysis and interpretation of images.

4) Image Processing and Video Processing

- **Video Processing:**
    - **Focus:** Involves the processing of a sequence of images (video) indexed by time.
    - **Applications:** Enhancements, compression, and analysis of video content.
- **Relationship:**
    - Video processing is an extension of image processing. Many image processing algorithms can be adapted to process video data by applying the same techniques to each frame of the video.

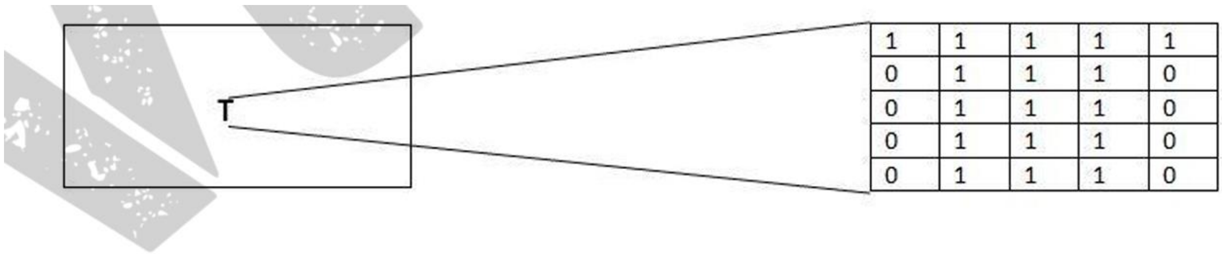
5) Image Processing and Optics

- **Optical Image Processing:**
    - **Focus:** Involves the manipulation of images using optical components like lenses and light.
    - **Applications:** Enhancing image quality through optimal lighting and lens configurations.
- **Relationship:**
    - Optical principles are fundamental to image acquisition and initial processing stages, making optics an essential part of image processing.

6) Image Processing and Statistics

- **Statistics:**
  - **Focus:** Involves the collection, analysis, interpretation, and presentation of data.
- **Image Analysis:**
  - **Focus:** Extracts and analyzes object information from images using statistical methods.
  - **Applications:** Counting objects, measuring areas, and statistical inference for deeper image understanding.
- **Relationship:**
  - Statistical techniques are crucial for image analysis, helping to extract meaningful information and make inferences from image data.

Digital Image representation



1	1	1	1	1
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0

(a) (b)

Figure 4.3: Digital Image Representation (a) Small binary digital image (b) Equivalent image contents in matrix form

1) Defining an Image

- **2D Image:** An image can be seen as a 2D signal that changes over the spatial coordinates x and y. It can be written as f(x,y).
- **3D Image:** Medical images like MRI and CT scans are 3D images represented as f(x,y,z) where x, y, and z are spatial coordinates.

2) Digital Image and Matrix Representation

- **Example Image:**
  - Fig. 4.3(a) shows a small binary digital image.
  - Fig. 4.3(b) shows its matrix form, where each value represents the pixel's intensity.
  - The image has 5 rows and 5 columns.

- **Mathematical Function:** An image f(x,y) is divided into X rows and Y columns, with coordinate ranges {x=0,1,...,X−1}

3) Pixels and Intensity Values

- **Pixels:** The smallest unit of a digital image, located at the intersection of rows and columns in a matrix.
- **Intensity Values:** The value of the function f(x,y)f(x, y)f(x,y) at each point, representing the image's grey value or intensity.
- **Resolution:**
  - **Vertical Resolution:** Number of rows.
  - **Horizontal Resolution:** Number of columns.
  - **Image Size:** Expressed in terms of pixel dimensions (e.g., 256x256, 512x512).

4) Resolution and Bit Depth

- **Resolution:** In digital photography, resolution is the level of detail contained in an image. More specifically, it refers to the number of pixels that exist within that image. The higher the resolution, and the richer the pixel count, the more detail and definition you will see.
- **Spatial Resolution:** Depends on the number of pixels and the bit depth (number of bits used to represent pixel intensity).
- **Bit Depth:** The number of bits required to encode pixel values, determining the range of possible intensity values.
  - **Binary Images:** Bit depth of 1 (values are 0 or 1).
  - **Greyscale Images:** Bit depth of 8 (values range from 0 to 255).
  - **Color Images:** Includes both color value and intensity, with bit depth determining the color gamut or palette.
- **Calculation of Total Bits: Total bits=Number of rows×Number of columns×Bit depth**

5) Artifacts from Resolution Changes

- **False Contouring:** Occurs when bit depth is reduced while keeping the number of pixels constant.
- **Checkerboard Effect:** Happens when the number of pixels is reduced while keeping the bit depth constant.

6) 3D Images

- **Voxel:** In 3D images, the term 'voxel' (volume element) is used instead of 'pixel'. A 3D image is represented as f(x,y,z)f(x, y, z)f(x,y,z), incorporating depth information.

types of images

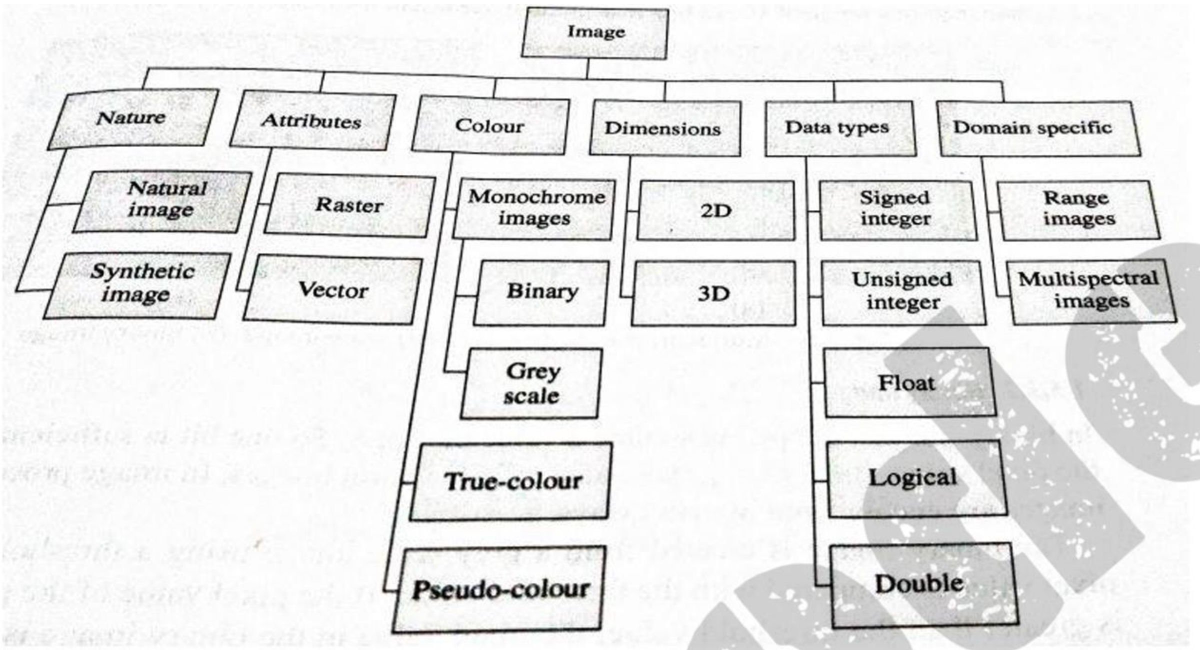


Figure 4.4: Classification of Images

Images can be classified based on several criteria as illustrated in Figure 4.4. Below is a detailed breakdown of these classifications:

Based on Nature

1. **Natural Images:**
   - **Description:** Images of natural objects obtained using devices such as cameras or scanners.
   - **Examples:** Photos of landscapes, portraits, real-world scenes.
2. **Synthetic Images:**
   - **Description:** Images generated using computer programs.
   - **Examples:** Computer graphics, digitally created artwork, simulations.

Based on Attributes

1. **Raster Images:**
   - **Description:** Pixel-based images where the quality depends on the number of pixels.
   - **Characteristics:** Operations such as enlarging or blowing up can result in quality reduction.
2. **Vector Graphics:**

- **Description:** Use basic geometric shapes such as lines and circles to describe an image.
- **Characteristics:** Not resolution-dependent, so quality is maintained regardless of scaling.

Based on Colour

1. **Monochrome Images:**
   - **Description:** Images without a color component.
   - **Types:**
     - **Binary Images:** Pixels can only be black or white (0 or 1).
     - **Grayscale Images:** Contain shades of gray, typically represented by 8 bits per pixel.
2. **True Colour Images:**
   - **Description:** Represent the full range of available colors, stored with full precision.
   - **Characteristics:** Each color component (red, green, blue) is represented using 8 bits.
3. **Pseudo-Colour Images:**
   - **Description:** False color images where color is added artificially based on data interpretation.
   - **Examples:** Used in medical imaging (e.g., Doppler images) and remote sensing.

Based on Dimensions

1. **2D Images:**
   - **Description:** Standard digital images represented as a rectangular array of pixels.
2. **3D Images:**
   - **Description:** Images with an additional dimension, often representing depth or volume.
   - **Examples:** Volume images, CT scans, MRIs, microscopy images, range images.

Based on Data Types

1. **Binary Images:**
   - **Description:** Use 1 bit per pixel (black or white).
2. **Grayscale Images:**
   - **Description:** Stored as 8-bit or 16-bit images, representing different shades of gray.
3. **Colour Images:**
   - **Description:** Typically use 24 or 32 bits to represent color and intensity values.
4. **Other Data Types:**
   - **Signed Integer:** Used to handle negative numbers.
   - **Unsigned Integer:** For non-negative numbers.
   - **Float:** Storing data in scientific notation for precision.
   - **Logical:** Boolean data type.

- **Double:** Double precision floating-point for high accuracy.

Domain Specific Images

1. **Range Images:**
    - **Description:** Pixel values denote the distance between the object and the camera, also known as depth images.
2. **Multispectral Images:**
    - **Description:** Taken at different bands of the electromagnetic spectrum, including visible, infrared, and ultraviolet regions.
    - **Application:** Predominantly used in remote sensing.

Basic relationships and distance metrics

Image Representation

1. **Pixels and Coordinate Systems**:
    - **Logical Pixels**:
        - Logical pixels represent points in a continuous 2D function.
        - They are "logical" because they specify a location but do not occupy physical space.
        - Typically represented in the Cartesian first coordinate system (x and y coordinates are both positive).
        - Example: In a 3x3 grid, logical pixels are placed in the first quadrant with coordinates ranging from (0,0) to (2,2).
    - **Physical Pixels**:
        - Physical pixels occupy actual space when displayed on an output device.
        - These pixels are represented in the Cartesian fourth coordinate system (both x and y coordinates can be negative or zero).
        - Example: A digital image in a 3x3 grid is often represented in the fourth quadrant of the Cartesian coordinate system.
2. **Digital Image Processing**:
    - **Discrete Image Representation**:
        - In digital image processing, the discrete form of the image is commonly used.
        - Discrete images are represented in the fourth quadrant of the Cartesian coordinate system.
        - Programming environments like MATLAB often start with an index of (1,1), which is the first element of the matrix.

Image Topology

1. **Neighbourhoods**:
   - **4-Neighbourhood (N4)**:
     - A pixel's 4-neighbourhood includes the four direct neighbors (horizontal and vertical).
     - Example: For a pixel P(x,y), the 4-neighbourhood includes (x+1,y), (x-1,y), (x,y+1), and (x,y-1).
     - Graphically:

```
0 X 0
X P X
0 X 0
```

   - **Diagonal Neighbours (ND)**:
     - A pixel's diagonal neighbors include the four diagonal pixels.
     - Example: For a pixel P(x,y), the diagonal neighbors are (x-1,y-1), (x+1,y+1), (x-1,y+1), and (x+1,y-1).
     - Graphically:

```
X 0 X
0 P 0
X 0 X
```

   - **8-Neighbourhood (N8)**:
     - Combines the 4-neighbourhood and diagonal neighbors, including all eight surrounding pixels.
     - Example: For a pixel P(x,y), the 8-neighbourhood includes all neighbors in the 3x3 grid centered on P.
     - Graphically:

```
X X X
X P X
X X X
```

2. **Connectivity**:

Let's consider a binary image where pixels can either be 0 (background) or 1 (foreground). We will explore 4-connectivity, 8-connectivity, and mixed connectivity (m-connectivity) using this example.

Binary Image Example

Consider the following 5x5 binary image:

```
1 0 0 1 1
1 1 0 0 0
0 1 1 0 1
0 0 1 0 1
1 1 0 0 0
```

4-Connectivity

In 4-connectivity, a pixel is connected to its immediate horizontal and vertical neighbors.

Example 1: Let's find the 4-connected component that includes pixel at (0,0).

- Start from (0,0): value = 1
- Check 4-neighbors:
  - (0,1): value = 0 (not part of the component)
  - (1,0): value = 1 (part of the component)
- Now, from (1,0):
  - (0,0): already visited
  - (2,0): value = 0 (not part of the component)
  - (1,1): value = 1 (part of the component)
- Now, from (1,1):
  - (1,0): already visited
  - (0,1): value = 0 (not part of the component)
  - (2,1): value = 1 (part of the component)
- Now, from (2,1):
  - (1,1): already visited
  - (3,1): value = 0 (not part of the component)
  - (2,0): value = 0 (not part of the component)
  - (2,2): value = 1 (part of the component)
- Now, from (2,2):
  - (1,2): value = 0 (not part of the component)
  - (2,1): already visited
  - (3,2): value = 1 (part of the component)
- Now, from (3,2):
  - (2,2): already visited
  - (4,2): value = 0 (not part of the component)

- (3,1): value = 0 (not part of the component)
- (3,3): value = 0 (not part of the component)

The 4-connected component including (0,0) is:

```
1 0 0 0 0
1 1 0 0 0
0 1 1 0 0
0 0 1 0 0
0 0 0 0 0
```

8-Connectivity

In 8-connectivity, a pixel is connected to its immediate horizontal, vertical, and diagonal neighbors.

Example 2: Let's find the 8-connected component that includes pixel at (0,3).

- Start from (0,3): value = 1
- Check 8-neighbors:
  - (0,2): value = 0 (not part of the component)
  - (0,4): value = 1 (part of the component)
  - (1,3): value = 0 (not part of the component)
  - (1,2): value = 0 (not part of the component)
  - (1,4): value = 0 (not part of the component)
- Now, from (0,4):
  - (0,3): already visited
  - (1,3): value = 0 (not part of the component)
  - (1,4): value = 0 (not part of the component)

The 8-connected component including (0,3) is:

```
0 0 0 1 1
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

Mixed Connectivity (m-connectivity)

In mixed connectivity, pixels are connected using both 4-connectivity and diagonal neighbors, but ensuring no multiple paths or loops.

Example 3: Let's find the m-connected component that includes pixel at (4,0).

- Start from (4,0): value = 1
- Check 4-neighbors and diagonal neighbors:
    - (3,0): value = 0 (not part of the component)
    - (4,1): value = 1 (part of the component)
- Now, from (4,1):
    - (4,0): already visited
    - (4,2): value = 0 (not part of the component)
    - (3,1): value = 0 (not part of the component)
    - (3,0): value = 0 (not part of the component)

The m-connected component including (4,0) is:

```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
1 1 0 0 0
```

3. **Relations**:
    - **Binary Relations**:
        - Specifies a relationship between two pixels a and b (denoted as aRb).
        - Examples include:
            - **Reflexive**: For any element a in set A, aRa holds.
            - **Symmetric**: If aRb, then bRa also holds.
            - **Transitive**: If aRb and bRc, then aRc also holds.
        - If all three properties are satisfied, the relationship is called an equivalence relation.

Distance Measures

1. **Euclidean Distance**:
    - Measures the straight-line distance between two pixels p and q.
    - Formula: D(p,q)=(x−s)2+(y−t)2D(p,q) = \sqrt{(x-s)^2 + (y-t)^2}D(p,q)=(x−s)2+(y−t)2
    - Advantage: Simple and intuitive.

- Disadvantage: Computationally expensive due to the square root operation.

2. **City Block Distance (D4)**:
   - Measures the distance by summing the absolute differences between the pixel coordinates.
   - Formula: D4(p,q)=|x−s|+|y−t|D4(p,q) = |x-s| + |y-t|D4(p,q)=|x−s|+|y−t|
   - Advantage: Easier to compute compared to Euclidean distance.
3. **Chessboard Distance (D8)**:
   - Measures the maximum absolute difference between the pixel coordinates.
   - Formula: D8(p,q)=max⁡(|x−s|,|y−t|)D8(p,q) = \max(|x-s|, |y-t|)D8(p,q)=max(|x−s|,|y−t|)
   - Advantage: Simple calculation, useful in grid-based environments.

Important Image Characteristics

1. **Connected Set**:
   - A set of pixels that are connected based on a specific connectivity rule (e.g., 4-connectivity or 8-connectivity).
   - Important for identifying objects and regions within an image.
2. **Digital Path**:
   - A sequence of pixels that form a path from one pixel to another.
   - Example: A path from pixel p to pixel q is a sequence p1, p2, ..., pn where p1=p and pn=q.
   - Path length is the number of pixels in the path. If the path starts and ends at the same pixel, it is called a closed path.
3. **Region**:
   - A connected component in an image.
   - Example: A region R is a set of connected pixels sharing the same properties (e.g., intensity).
4. **Adjacency**:
   - Two regions R1 and R2 are adjacent if their union forms a connected component.
   - Example: Adjacent regions share at least one common boundary pixel.
5. **Boundary (Contour)**:
   - The border of a region, separating it from the background or other regions.
   - Example: The boundary of a region includes pixels that have at least one neighbor outside the region.
6. **Edges**:
   - Represent abrupt intensity changes in an image.
   - Can be connected or disjoint.
   - Example: Edges outline objects and are crucial for defining object shapes.

Classification of Image processing Operations

Based on the Domain of Processing

a. Spatial Domain

Operations are performed directly on the image pixels. This includes:

- **Point Operations**: Modify individual pixel values. Examples include contrast adjustment, thresholding, and histogram equalization.
- **Neighborhood Operations**: Modify a pixel based on its neighborhood. Examples include filtering (e.g., blur, sharpen), edge detection, and noise reduction.

b. Frequency Domain

Operations are performed on the image's frequency components, typically after a transformation like the Fourier Transform.

- **Filtering**: Frequency domain filters can be used to remove noise (e.g., low-pass filtering) or enhance edges (e.g., high-pass filtering).
- **Compression**: Techniques like JPEG use frequency domain methods for image compression.

2. Based on Data Type

a. Binary Image Processing

Operations are performed on binary images where each pixel is either 0 (black) or 1 (white).

- **Morphological Operations**: Such as erosion, dilation, opening, and closing, used for shape analysis and noise removal.

b. Grayscale Image Processing

Operations are performed on images with intensity levels ranging from 0 to 255 (for 8-bit images).

- **Histogram Equalization**: Enhances contrast by redistributing the intensity values.
- **Filtering**: Operations like median filtering for noise reduction.

c. Color Image Processing

Operations are performed on images with multiple channels (e.g., RGB).

- **Color Space Transformations**: Converting between color spaces (e.g., RGB to HSV).
- **Color Correction**: Adjusting colors for better visual appearance or to match certain standards.

3. Based on Purpose

a. Image Enhancement

Improving the visual appearance of an image or making it more suitable for a specific task.

- **Contrast Adjustment**: Stretching or compressing the intensity range.
- **Sharpening**: Enhancing edges to make features more distinct.

b. Image Restoration

Recovering an image that has been degraded by noise, blur, or other distortions.

- **Noise Reduction**: Using filters like Gaussian, median, or wavelet-based methods to reduce noise.
- **Deconvolution**: Correcting blur by reversing the effects of the blurring process.

c. Image Compression

Reducing the amount of data required to represent an image.

- **Lossy Compression**: Techniques like JPEG that discard some image data to achieve high compression ratios.
- **Lossless Compression**: Techniques like PNG that preserve all image data.

d. Image Segmentation

Partitioning an image into meaningful regions for analysis.

- **Thresholding**: Simple segmentation based on intensity values.
- **Clustering**: Advanced techniques like k-means or watershed for more complex segmentation.

e. Image Analysis

Extracting meaningful information from images.

- **Feature Extraction**: Identifying key features like edges, corners, or blobs.
- **Pattern Recognition**: Classifying image content using techniques like machine learning or deep learning.

4. Based on Complexity

a. Low-Level Image Processing

Basic operations that are directly applied to pixel values.

- **Noise Filtering**: Removing noise using filters.
- **Histogram Equalization**: Adjusting contrast based on histogram information.

b. Mid-Level Image Processing

Operations that involve feature extraction and object recognition.

- **Edge Detection**: Identifying boundaries of objects.
- **Segmentation**: Dividing the image into meaningful parts.

c. High-Level Image Processing

Involves understanding the content of the image and making decisions.

- **Object Recognition**: Identifying objects within an image.
- **Scene Understanding**: Interpreting a scene's overall context and relationships between objects.

Examples for Each Classification

1. **Spatial Domain - Neighborhood Operations**
   - **Filtering**: Applying a 3x3 median filter to remove salt-and-pepper noise.
2. **Frequency Domain - Filtering**
   - **Low-pass Filtering**: Applying a Gaussian filter in the frequency domain to blur an image.
3. **Binary Image Processing - Morphological Operations**
   - **Erosion**: Reducing noise by removing small white regions.
4. **Grayscale Image Processing - Filtering**
   - **Median Filtering**: Reducing noise while preserving edges.
5. **Color Image Processing - Color Space Transformation**
   - **RGB to HSV**: Converting an image from RGB to HSV color space for easier color manipulation.
6. **Image Enhancement - Contrast Adjustment**

- **Histogram Equalization**: Enhancing the contrast of an image by spreading out the most frequent intensity values.

7. **Image Restoration - Noise Reduction**
   - **Gaussian Filter**: Applying a Gaussian filter to smooth an image and reduce noise.
8. **Image Compression - Lossy Compression**
   - **JPEG Compression**: Reducing file size by compressing the image with a slight loss in quality.
9. **Image Segmentation - Thresholding**
   - **Otsu's Method**: Automatically finding the threshold value for segmenting a grayscale image into binary.
10. **Image Analysis - Feature Extraction**
    - **Canny Edge Detection**: Identifying edges in an image using the Canny algorithm.
11. **Low-Level Processing - Noise Filtering**
    - **Applying a Median Filter**: To reduce noise in a grayscale image.
12. **Mid-Level Processing - Edge Detection**
    - **Using the Sobel Operator**: To detect edges in an image.