International Islamic University Chittagong

IMAGE PROCESSING & PATTERN RECOGNITION

Final Lecture topics

Resources'

https://drive.google.com/drive/folders/1AneGLB4gtoIdsPxrxYbB7xPlxlKBWi-_?usp=sharing

C191267

TASNIM

48th batch

Contents

Syllabus

Lec 19-24	**4. Image Compression:** Fundamental concepts of Image Compression and Data Compression, Data Redundancy, Image Compression models, Error free and Lossy compression, Image Compression Standards.
Lec 25-28	**5. Morphological processing:** Morphological Image Processing, Dilation, Erosion, Opening, Closing, Hit and Miss etc.
Lec 29-32	**6. Segmentation:** Image Segmentation, Different types of Segmentation, Edge linking and boundary detection, Thresholding, Region oriented segmentation,
Lec 33	**Class Test 2**
Lec 34-37	**7. Image Representation:** Object representation and description algorithms, Run Code, Chain Code, Signature, Skeleton, Boundary detection, Feature Extraction few case studies.
Lec 38-41	**8. Pattern Recognition:** Fundamental concepts of Pattern Recognition, Pattern, Pattern Classes, Types of Pattern Recognition, Decision Theoretic methods, Structural method, Statistical method, Neural Network, few case studies like speech recognition, fingerprint recognition, character recognition etc.
Lec 42-45	**Class Test 3 & Review Class**

Zainal Abedin Sir Lecture topics

Segment 1: Image Compression Chapter 8

Loosy / lossless Compression, Redundancy,
coding redundancy, Variable length redundancy, relation with entropy and redundancy, Huffman Coding/decoding , Arithmatic coding , LZW coding.

Exercise: 8.9, 8.18

Segment 2: Morphological processing Chapter9

Dilation, Erosion, opening , closing, Hit, mix related exercise, application.

Exercise: 9.1, 9.2, 9.18 ,9.26

Segment 3: Segmentation **Chapter** 10

Edge detection:
Edge model, edge linking algorithm., (1st, 2nd derivatives, all method related to both derivatives, using Gradient detection, canny edge detection)

Region segmentation , region growing , region splitting

Thresholding , Otsu method
optimal thresholding value find by Otsu method
Class variance,

Segment 4: Image representation

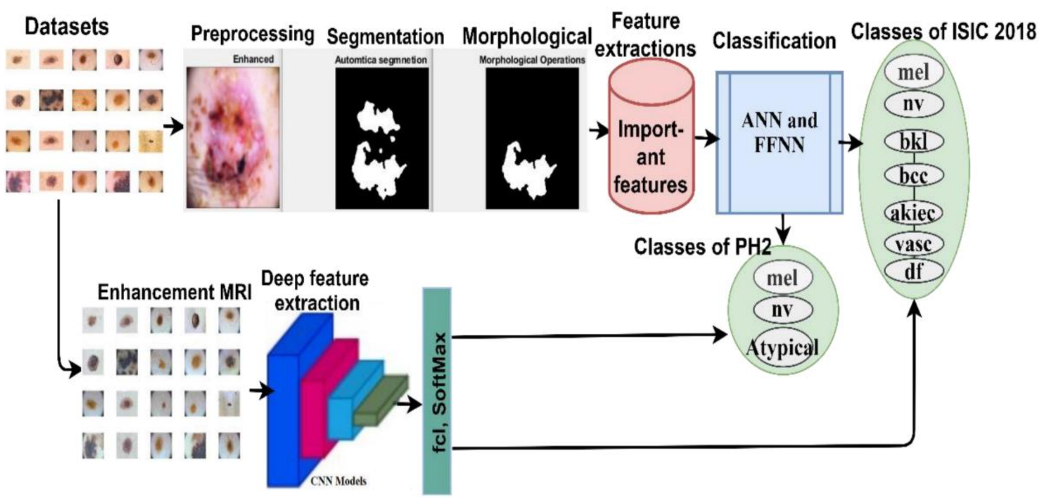
later discuss

Segment 5: Pattern recognition Chapter 12:

Manual feature extraction,
12.3 How Minimum distance classifiers works?
Decision Boundary (Example 12.1) Perceptron, linear
or nonlinear?
if an image is given with filter size then how Convolution layer extract feature, max pooling layer value,

Invariant feature descriptor: how works, type like the scale-invariant feature transform (SIFT), rotational invariant CNN architecture:

0. Defining a dataset
1. How many CNN would need for that dataset?
2. How many pooling layers need
3. Dimension
4. Feature extraction
5. FCN
6. Which classifier will be used?



What is the difference between global thesholding and local thresholding ? Calculate the optimal global thresholding value using Otsu method. for the follow

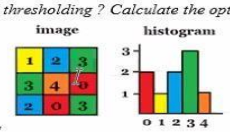


Fig. Image and corresponding histogram

Using region growing concept segment the following image using the center as a seed and the region is grown when the difference between two pixels is equal or less than 5. Use 8-connectivity.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	[illegible]	[illegible]	[illegible]	56	60
10	59	[illegible]	60	[illegible]	10	62
10	60	[illegible]	[illegible]	[illegible]	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

Previous Solve by Tarin (C191270)

Pattern recognition and Image Processing

Final Previous Solution

C191270_Tarin

Spring 2022

1a) "data is a combination of Information and redundant data"- do you agree? justify your answer with proper example

Ans: Same as autumn 2021

1b) Encode the following 4x 4,8 bit image using LZW coding

200 200 129 129

200 200 129 129

2
0
0
2
0
0
1
2
9
1
2
9
2
0
0
2
0
0
1
2
9
1
2
9

A
n
s
:

SP-22

1(b)

200	200	129	129
200	200	129	129
200	200	129	129
200	200	129	129

Current Recognized Sequence	Pixel being processed	Encoded output	Dictionary Location	Dictionary Entry
	200			
200	200	200	256	200-200
200	129	200	257	200-129
129	129	129	258	129-129
129	200	129	259	129-200
200	200			
200-200	129	256	260	200-200-129
129	129			
129-129	200	258	261	129-129-200
200	200			
200-200	129			
200-200-129	129	260	262	200-200-129-129
129	200			
129-200	200	259	263	129-200-200
200	129			
200-129	129	257	264	200-129-129
129		129		

Encoded output:

200 200 129 129
256 258 260 259
257 129

1c) Describe in details how colour based segmentation can be used in an attempt to segment "same coloured" object in real colour images . what problems arise? what are some ways of dealing with them?

Ans:

Color-based segmentation is a technique used to segment objects of the same color in real color images. It involves identifying regions in an image that have similar color characteristics and grouping them together as the samecolored object. Here is a detailed description of how color-based segmentation can be applied and the problems that may arise, along with potential solutions:

1. Color Space Selection: The first step is to select an appropriate color space for representing the image. Commonly used color spaces include RGB, HSV, and Lab. Each color space has its own advantages in terms of separating color information and can affect the segmentation results.

2. Color Thresholding: Once the color space is chosen, color thresholding is performed by defining a range of acceptable color values for the desired object. Pixels falling within this range are considered part of the samecolored object. However, several challenges can arise:

- Illumination Variations: Changes in lighting conditions can cause variations in color appearance. To address this, techniques such as histogram equalization, color normalization, or color constancy algorithms can be applied to reduce the impact of illumination variations.

- Similar Color Interference: Objects with different colors may appear similar due to factors such as lighting conditions, texture patterns, or background clutter. Additional features such as texture, shape, or spatial relationships can be incorporated to differentiate between objects of similar colors.

3. Noise and Artifacts: Image noise and artifacts can negatively impact color-based segmentation. Preprocessing steps like noise reduction techniques (e.g., median filtering) or smoothing filters can help reduce the impact of noise and improve the segmentation results.

4. Parameter Selection: The selection of color thresholds and segmentation parameters is crucial for achieving accurate results. These parameters may vary depending on the specific image and object characteristics. Techniques such as adaptive thresholding or statistical approaches can be used to dynamically determine suitable thresholds.

5. Post-processing: After initial segmentation, post-processing steps can be applied to refine the results. These steps may include morphological operations (e.g., erosion, dilation) to remove small noise or gaps, region merging/splitting techniques for better connectivity, or contour smoothing to improve the object boundaries.

6. Evaluation and Validation: It is important to evaluate and validate the segmentation results using appropriate metrics and ground truth data. This helps assess the accuracy and effectiveness of the color-based segmentation method and make necessary adjustments if required.

While color-based segmentation is effective for segmenting same-colored objects, it has limitations. Some challenges include variations in lighting, similar colors of different objects, and noise interference. To address these challenges, techniques like illumination normalization, incorporating additional features, noise reduction, parameter tuning, and post-processing steps can be employed. Additionally, combining color-based segmentation with other segmentation methods or machine learning approaches can further enhance the segmentation accuracy and robustness in real-color images.

1d) What is bit plane decomposition? What is the benefit of bit plane decomposition with necessary example?

Ans:

Bit plane decomposition is a technique used in image processing to decompose an image into its individual bit planes. In this technique, the binary representation of each pixel in the image is separated into different bit planes, with each bit plane containing the binary values of a specific bit position (e.g., the most significant bit plane, second most significant bit plane, etc.).

The benefit of bit plane decomposition is that it provides a way to analyze and manipulate the image at different levels of detail, based on the significance of the bits. Here's an example to illustrate the benefit:

Let's consider a grayscale image where each pixel value ranges from 0 to 255, represented by 8 bits (1 byte). By performing bit plane decomposition, we can extract the individual bit planes for each pixel value.

For instance, consider a pixel value of 150, which in binary representation is 10010110. Through bit plane decomposition, we obtain the following bit planes:

Bit Plane 7: 1 0 0 1 0 1 1 0 (Most Significant Bit - MSB)
Bit Plane 6: 0 0 0 0 0 0 0 0
Bit Plane 5: 0 0 0 0 0 0 0 0
Bit Plane 4: 0 0 0 0 0 0 0 0
Bit Plane 3: 1 0 0 1 0 1 1 0
Bit Plane 2: 0 0 0 0 0 0 0 0
Bit Plane 1: 1 0 0 0 0 0 0 0
Bit Plane 0: 0 0 0 0 0 0 0 0 (Least Significant Bit - LSB)

Each bit plane represents a different level of information present in the image. The MSB (bit plane 7) contains the most significant information, while the LSB (bit plane 0) contains the least significant information.

The benefit of bit plane decomposition becomes apparent when we observe the impact of manipulating or discarding certain bit planes. For example:

1. Image Compression: By discarding the bit planes with lower significance (e.g., bit planes 0 to 3), we can achieve lossy compression of the image, reducing the storage requirements without significant degradation in visual quality.

2. Image Enhancement: By manipulating or applying image processing operations on specific bit planes, we can enhance specific details or features in the image. For example, amplifying or attenuating the higher bit planes (e.g., bit planes 5 to 7) can enhance fine details or emphasize edges in the image.

3. Image Watermarking: Bit plane decomposition can be utilized in watermarking techniques. By embedding watermarks in specific bit planes, the watermark can be selectively extracted or detected without affecting the overall image appearance.

Overall, bit plane decomposition allows for analysis, manipulation, and selective processing of images at different levels of detail, based on the significance of the bits. It provides flexibility and control in various image processing applications, such as compression, enhancement, and watermarking.

2a) binary image X and structuring element B are given below

i) https://youtu.be/77IvijY3H8g

ii) **What is pruning? why it is important in many image processing applications ?**

Ans:

Pruning in image processing is the process of removing unnecessary or irrelevant data or components from an image or computational model. It is important because it improves efficiency, reduces complexity, enhances image quality, aids in feature selection, and promotes model interpretability.

[2] Why pruning algorithm is necessary in many image processing application?

Ans: The pruning algorithm is a technique used in digital image processing based on mathematical morphology. It is used as a complement to the skeleton and thining algorithm to remove unwanted parastic components. In this case, parastic components refer to branches of a line which are not key to the overall shape of the line and should be removed.

OR,

2a) i) In the pattern recognition and classification there are four basic steps involved which are sensing an image, segmenting the image, extracting the features from the segmented objects , and classification to recognize the specific object . Based on this procedure explain how you can segment and classify a specific object from an 8 bit color image Ans:

To segment and classify a specific object from an 8-bit color image, we can follow the basic steps of pattern recognition and classification: sensing, segmentation, feature extraction, and classification. Here's a detailed explanation of the procedure:

1. Sensing the Image:

- Capture the 8-bit color image using a sensing device, such as a camera or scanner.
- Ensure proper lighting conditions and capture the object of interest with sufficient image resolution.

2. Segmentation:

- Apply segmentation techniques to separate the object from the background:
- Color-based Segmentation: Set a threshold on the color channels to distinguish the object from the background based on color intensity. For example, you can use color thresholding techniques to identify regions that fall within a specified color range.

- Region-based Segmentation: Utilize algorithms such as region growing, graph cuts, or watershed segmentation to group pixels with similar color characteristics that belong to the object.
- Edge-based Segmentation: Employ edge detection algorithms like Canny or Sobel to identify object boundaries by detecting significant changes in color intensity or gradients.

3. Feature Extraction:

- Extract relevant features from the segmented object. These features should capture discriminative information for classification.
- In the case of an 8-bit color image, features can be derived from color attributes such as:
- Color Histogram: Calculate the distribution of color intensities or color channels to represent the color distribution within the object.
- Color Moments: Compute statistical moments, such as mean, standard deviation, or higher-order moments, to describe the color properties of the object.
- Color Texture Descriptors: Analyze texture patterns in the color channels, such as Local Binary Patterns (LBP) or Haralick features, to capture textural information.

4. Classification:

- Utilize a classification algorithm to recognize the specific object based on the extracted features.
- Various classification techniques can be employed, including:
- Supervised Learning: Train a classifier, such as Support Vector Machines (SVM), Random Forests, or Neural Networks, using labeled training data with known object classes. Then, apply the trained classifier to classify new objects.
- Unsupervised Learning: Utilize clustering algorithms like k-means or Gaussian Mixture Models (GMM) to group similar objects together based on extracted features.
- Deep Learning: Apply deep neural network architectures, such as Convolutional Neural Networks (CNNs), for end-to-end feature learning and object classification.

The classification step will provide the recognition or classification result, indicating the specific object present in the 8-bit color image.

It's important to note that the choice of segmentation techniques, feature extraction methods, and classification algorithms can vary depending on the specific characteristics of the objects and the application domain. Experimentation, evaluation, and fine-tuning of each step may be necessary to achieve accurate segmentation and classification results for a given 8-bit color image.

i
i
)
2b)
Answer:
Process of Hit or Miss transformation

3a) An Analytic produce a decision boundary with the equation 5.1 X1 - 0.3 X2 - 8.43 the first and second decision functions are produced as 6.6 X1 + 1.2 X2 - 28.33 and 5.6 X1 + 3.8 X2 - 10.0 respectively. determine the two mean vector (m1 , m2) from the above information Ans:

3c) describe canny edge detection algorithm Ans:

The Canny edge detection algorithm is a popular and widely used method for detecting edges in digital images. It was developed by John F. Canny in 1986 and is known for its ability to accurately detect edges while minimizing noise and producing thin, well-defined edges.

The Canny edge detection algorithm consists of several steps:

1. **Image smoothing**: The algorithm starts by applying a Gaussian blur to the input image. This step helps to reduce noise and remove small details that are not relevant to the edge detection process.

2. **Gradient calculation**: The next step involves calculating the gradients of the smoothed image. The image is convolved with two separate filters (Sobel operators) in the horizontal and vertical directions. These filters highlight regions of the image with strong intensity changes, which are likely to correspond to edges.

3. **Gradient magnitude and direction**: The gradient magnitudes and directions are computed based on the horizontal and vertical gradient components. The magnitude represents the strength of the gradient at each pixel, while the direction indicates the orientation of the gradient.

4. **Non-maximum suppression**: In this step, the algorithm aims to thin out the detected edges to obtain a single-pixel-wide edge. It involves scanning the gradient magnitude image and suppressing pixels that are not local maximums along the direction of the gradient. This ensures that only the most significant edges are preserved.

5. **Double thresholding**: A double thresholding technique is employed to classify the remaining edge pixels into strong, weak, or non-edges. This is done by defining high and low thresholds. If a pixel's gradient magnitude is above the high threshold, it is marked as a strong edge. If it is below the low threshold, it is considered a non-edge. Pixels with values between the low and high thresholds are labeled as weak edges.

6. **Edge tracking by hysteresis**: The final step involves connecting weak edges to strong edges to form continuous edges. This is achieved by analyzing the connectivity of weak edges. If a weak edge pixel is adjacent to a strong edge pixel, it is promoted to a strong edge. This process continues until no more weak edges can be connected.

The output of the Canny edge detection algorithm is a binary image where the detected edges are represented by white pixels and the non-edge regions are black.

The Canny edge detection algorithm is widely used in various computer vision and image processing applications, including object detection, image segmentation, and feature extraction.

4a) shown below A is an original binary image which has some defects. how would you detect the defects of the original image using morphological operators with different structural elements explain each structural element with proper example the detected defect sample is also given below Ans:

4b) i. evaluate the chain Code of figure 1 (use 8 direction) ii. chain code has some limitations . make the chain could in a rotational invariant

iii.find the area of figure 1

Or,

4b) i) describe your method of computing skeleton of the images. what kind of structural elements will you be using what happens when you change the size of the structure in element Ans:

1st part :

To compute the skeleton of an image, I'll describe a commonly used method known as the "thinning" algorithm. Thinning is a morphological operation that iteratively reduces the foreground regions of an image to their skeleton, a thin representation of the original shapes.

Here's an outline of the steps involved in computing the skeleton using the thinning algorithm:

1. **Thresholding**: Convert the input image to a binary image by applying an appropriate thresholding technique. This step separates the foreground objects from the background.

2. **Boundary extraction**: Extract the boundaries of the foreground objects in the binary image. This can be done using contour extraction techniques like the Moore-Neighbor Tracing algorithm or the Freeman Chain Code.

3. **Thinning iteration**: Repeat the following steps until no further changes occur in the image:
   - a. Initialize a blank output image to store the thinned skeleton.
   - b. Iterate over the boundary pixels of the foreground objects.
   - c. Identify each boundary pixel's 8-neighborhood (or 4-neighborhood, depending on the desired connectivity). - d. Determine if the current boundary pixel is a candidate for thinning by evaluating specific conditions based on its neighborhood. These conditions typically involve connectivity, neighborhood patterns, and preservation of topology.
   - e. If the current boundary pixel satisfies the thinning conditions, mark it for deletion in the output image.
   - f. Once all boundary pixels have been processed, delete the marked pixels from the original image.
   - g. Copy the updated image to the output image.
   - h. Repeat the iteration with the updated image.

4. **Post-processing**: After the thinning algorithm has completed, you can perform additional post-processing steps to refine the skeleton if needed. This may include removing noise, closing gaps, or improving connectivity.

The thinning algorithm continues iterating until no further changes occur, indicating that the skeleton has been extracted as much as possible.

It's important to note that the effectiveness of the thinning algorithm can depend on factors such as image quality, noise levels, and the specific characteristics of the objects in the image. Experimentation and adjustments to the thinning conditions and post-processing steps may be necessary to achieve the desired skeleton representation.

2^{nd} part:

To detect defects in an original image using morphological operators, we can use different structural elements to highlight specific characteristics or shapes of the defects. Morphological operators, such as erosion and dilation, are used to modify the shape and structure of objects in an image.

Here are a few examples of structural elements and how they can be used to detect defects:

1. **Square Structural Element**:

A square structural element is a square-shaped neighborhood that is used in morphological operations. It can be defined by its size, such as a 3x3 square or a 5x5 square. This structural element is useful for detecting defects that have a square or block-like shape. When applied with dilation, it can expand the defect areas, making them more prominent. Similarly, applying erosion with the square structural element can help to remove small noise or smooth out irregularities.

2. **Cross Structural Element**:

A cross structural element is a cross-shaped neighborhood with a central pixel and four arms extending horizontally and vertically. This structural element is particularly effective in detecting line-like defects. When used with dilation, it can thicken and enhance the line-like defects, making them easier to identify. Erosion with the cross structural element can remove small line-like details and thin out the defects.

3. **Disk Structural Element**:

A disk structural element is a circular-shaped neighborhood defined by its radius. It is suitable for detecting defects that have a circular or rounded shape. By applying dilation with the disk structural element, circular defects can be expanded and emphasized. Erosion with the disk structural element can help to smooth out the defects and remove small irregularities within them.

4. **Custom Structural Element**:

Depending on the specific characteristics of the defects you are trying to detect, you can design and use custom structural elements. For example, if you are looking for defects with a specific pattern or shape, you can create a structural element that matches that pattern. Custom structural elements offer flexibility in targeting specific defect types or shapes.

It's important to note that the selection of the appropriate structural element depends on the specific characteristics of the defects and the desired outcome. Experimenting with different structural elements and combinations of morphological operations (such as dilation, erosion, opening, and closing) can help in accurately detecting and enhancing defects in the original image.

iii) define local binary pattern example

Local Binary Pattern (LBP) is a texture descriptor that captures the local patterns and variations within an image. It assigns a binary code to each pixel in an image based on the comparison of its intensity value with its neighboring pixels.

To define the Local Binary Pattern, let's consider a 3x3 neighborhood around a central pixel in a grayscale image.

We can assign a binary code to the central pixel based on the intensity comparison with its neighbors.

1. **Central Pixel**: The central pixel is denoted by P(x_c, y_c), where (x_c, y_c) represents its coordinates.
2. **Thresholding**: Compare the intensity value of the central pixel P(x_c, y_c) with its 8 neighboring pixels in a circular pattern. For simplicity, let's assume we move clockwise starting from the top-left neighbor. If the intensity of a neighbor pixel is greater than or equal to the intensity of the central pixel, assign it a value of 1. Otherwise, assign it a value of 0.
3. **Binary Code**: Concatenate the binary values of the 8 neighboring pixels in a clockwise manner to form an 8bit binary code. This binary code represents the local pattern around the central pixel.
4. **Decimal Conversion**: Convert the binary code to decimal. This decimal value represents the LBP value of the central pixel.

For example, let's consider a grayscale image with the intensity values of the 3x3 neighborhood around the central pixel P(x_c, y_c) as follows:

1 4 3

2 6 5

7 9 8

Let's assume the intensity of the central pixel P(x_c, y_c) is 6. We can compare the intensity values of the neighboring pixels with the central pixel and assign binary values:

1 0 0

0 1 1

1 1 1

The binary code formed by concatenating the binary values in a clockwise manner is "00111101".

Converting this binary code to decimal, we get the LBP value for the central pixel P(x_c, y_c) as 61.

This process is repeated for each pixel in the image, resulting in a new image where each pixel represents the LBP value of the corresponding pixel in the original image.

Local Binary Pattern is a powerful texture descriptor that captures the local structure and texture information in an image, making it useful for various computer vision tasks such as object recognition, face recognition, and texture analysis.

5a) In the pattern recognition we have to extract the features from the object and that features are needed to recognized through recognizing algorithm for recognizing as specific object is it possible to recognize the object features through neural network that features are extracted from the object throw local binary pattern justify your answer with 4 x 4 8 bit colour image Ans:

Yes, it is possible to recognize object features through a neural network using features extracted from the object using the Local Binary Pattern (LBP) method. LBP is a widely used texture descriptor that captures local patterns and variations in an image.

To justify the answer, let's consider a 4x4 8-bit color image as an example. The LBP algorithm can be applied to this image to extract local patterns as follows:

1. **Image Conversion**: Convert the 8-bit color image to grayscale. This can be done by taking the average of the RGB values for each pixel or by using other color-to-grayscale conversion methods.

2. **LBP Calculation**: Divide the grayscale image into a grid of cells, such as 4x4. For each pixel within a cell, compare its intensity value with the surrounding neighbor pixels (usually 8 neighbors in a circular pattern). Set a binary code for each neighbor pixel based on whether its intensity is greater or smaller than the central pixel. This results in an 8-bit binary pattern for each pixel within a cell.

3. **Histogram Calculation**: Count the occurrences of each unique binary pattern within each cell and create a histogram. The histogram represents the distribution of local patterns within the image.

4. **Feature Vector**: Concatenate the histograms from all the cells to form a feature vector. In the case of a 4x4 grid, we would have a feature vector of size (16 x Number of unique binary patterns).

Once the feature vector is extracted using the LBP method, it can be used as input to a neural network for object recognition. The neural network can be trained on a dataset of labeled objects, where the extracted LBP features are paired with their corresponding object classes. The network learns to recognize patterns and correlations between the extracted features and the object classes, enabling it to predict the class of new, unseen objects based on their LBP features.

By training a neural network with LBP features extracted from objects, it can learn to recognize object classes based on the extracted local patterns and variations. The network's ability to generalize and make predictions on unseen data depends on the quality and representativeness of the training dataset, the network architecture, and the training process.

5b) template matching is a popular way to identify patterns from images i) explain the formulation of correlation based algorithm with example ii)describe pattern recognition model

OR

5b) describe minimum distance classified built the decision boundary for two pattern class w1 and w2 where to mean vectors are one 5.6 1.2 6.2 and 2 equal 1.40.3 3.7 explain how decision boundary works on classification stage

Spring 2019

1a) A 24 bit image with a 1024 x 840 pixels is compressed by a 16 bit lossy truncation algorithm. calculate the compression ratio and data redundancy of the image.

Ans:

To calculate the compression ratio, we need to first determine the size of the original uncompressed image and compare it to the size of the compressed image.

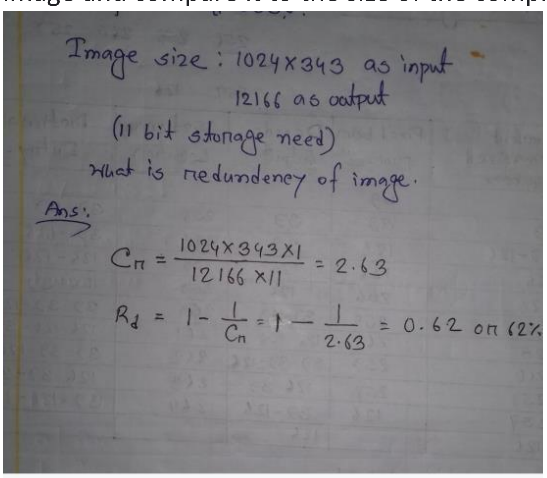


Image size : 1024×343 as input
12166 as output
(11 bit storage need)
what is redundency of image.

Ans:

$$C_r = \frac{1024\times343\times1}{12166\times11} = 2.63$$

$$R_d = 1-\frac{1}{C_r} = 1-\frac{1}{2.63} = 0.62 \text{ or } 62\%$$

Size of the original image:
Each pixel in the original image is represented by 24 bits, or 3 bytes. Therefore, the total size of the image can be calculated as follows:

1024 pixels × 840 pixels × 3 bytes/pixel = 2,764,160 bytes

Size of the compressed image:
Each pixel in the compressed image is represented by 16 bits, or 2 bytes. Therefore, the total size of the compressed image can be calculated as follows:

1024 pixels × 840 pixels × 2 bytes/pixel = 1,771,520 bytes

Compression ratio:
The compression ratio is the ratio of the size of the original image to the size of the compressed image:

Compression ratio = Size of original image / Size of compressed image
Compression ratio = 2,764,160 bytes / 1,771,520 bytes
Compression ratio ≈ 1.56

Data redundancy:

The data redundancy is the difference between the size of the original image and the size of the compressed image, expressed as a percentage of the size of the original image:

Data redundancy = (Size of original image - Size of compressed image) / Size of original image × 100% Data redundancy = (2,764,160 bytes - 1,771,520 bytes) / 2,764,160 bytes × 100% Data redundancy ≈ 35.9%

Therefore, the compression ratio is approximately 1.56 and the data redundancy is approximately 35.9%.

1b) "data and information are not synonymous terms" justify the statement with example . Estimate the Huffman coding for the word 'CSEDIP '

Ans:

Data and information are not synonymous terms because while data refers to raw and unorganized facts, information refers to processed and organized data that has meaning and relevance to the user. Data is a collection of raw and unprocessed facts, figures, and symbols that have no meaning on their own. In contrast, information is data that has been processed, organized, and given context so that it can be understood and used by people.

For example, consider a database of customer orders. The database contains data such as the customer's name, order number, date of purchase, and order details. While this data is valuable to the business, it is not information until it is organized, analyzed, and presented in a meaningful way. Once the data is processed and organized, it becomes information that can be used to make informed decisions about the business.

Huffman coding is a type of variable-length data compression algorithm that assigns shorter codes to more frequently occurring characters or symbols. To estimate the Huffman coding for the word 'CSEDIP', we would first determine the frequency of each character in the word. Counting the frequency of each character, we get:

- C: 1
- S: 1
- E: 1
- D: 1
- I: 1
- P: 1

Since each character appears only once, any Huffman code assigned to each character would be the same length, such as 3 bits. Therefore, the Huffman coding for the word 'CSEDIP' would be:

- C: 000
- S: 001
- E: 010
- D: 011
- I: 100
- P: 101

In this case, the Huffman coding provides a fixed length code of 3 bits for each character, which does not provide any compression benefit. In practice, Huffman coding is most effective when there is a significant difference in the frequency of characters in the data being compressed, allowing shorter codes to be assigned to more frequently occurring characters, resulting in a more efficient compression.

1c) write the advantage of LZW coding over arithmetic coding .

Ans:

LZW (Lempel-Ziv-Welch) coding and arithmetic coding are two popular data compression algorithms. While both techniques are effective at compressing data, they have different advantages and disadvantages.

One of the advantages of LZW coding over arithmetic coding is its simplicity. LZW coding is a relatively simple and easy-to-implement algorithm that requires minimal computational resources. In contrast, arithmetic coding is a more complex algorithm that requires a lot of computational resources and is difficult to implement efficiently.

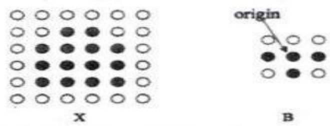
Another advantage of LZW coding is that it is a lossless compression technique, meaning that it can compress data without losing any information. In contrast, arithmetic coding is a lossy compression technique, meaning that it may discard some information during compression, which can lead to a loss of quality in the compressed data.

Additionally, LZW coding is particularly effective at compressing data with repeating patterns, as it can recognize and encode these patterns efficiently. This makes it well-suited for compressing text data and other types of data that have repeating patterns. In contrast, arithmetic coding is better suited for compressing data with a more random distribution of symbols, as it can assign shorter codes to more frequently occurring symbols.

Overall, while arithmetic coding has its advantages, such as providing better compression ratios for certain types of data, LZW coding's simplicity, lossless compression, and effectiveness at compressing data with repeating patterns make it a popular choice for many applications.

2 a) Explain the concept of Morphological Opening and closing. Explain with an example that $A \ominus B \neq B \ominus A$ where $\ominus$ denotes the morphological erosion operator. 4

b) Binary image, X and structuring element, B, are given as follows: 3



Calculate $Y1 = X \ominus B$, where $\ominus$ denotes the morphological erosion operator and $Y2 = X \oplus B$ where $\oplus$ denotes the morphological dilation operator.

c) Write the short note about: i) Hit-or-Miss Transformation, ii) Skeleton, and ii) Boundary filling. 3

2a) Answer:
https://www.geeksforgeeks.org/difference-between-opening-and-closing-in-digital-image-processing/ **2b)Answer:**

2c)

The Hit-or-Miss transformation is a fundamental operation in digital image processing, specifically in the field of morphological image processing. It is used for pattern matching, shape detection, and image analysis tasks in binary images.

The goal of the Hit-or-Miss transformation is to identify pixels or regions in an image that match a specific pattern. This pattern consists of foreground (white) pixels, background (black) pixels, and a third category of pixels that may or may not be present.

The operation is performed using two structuring elements: J, representing the foreground, and K, representing the background. These elements define the pattern to be matched. J contains the pixels that must be present (foreground) in the image, while K specifies the pixels that must be absent (background) in the image.

The Hit-or-Miss transformation

9.5.1 Boundary Extraction

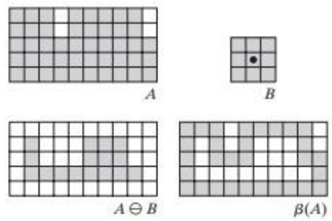
The boundary of a set A, denoted by $\beta(A)$, can be obtained by first eroding A by B and then performing the set difference between A and its erosion. That is,

$$\beta(A) = A - (A \ominus B) \tag{9.5-1}$$

where B is a suitable structuring element.

Figure 9.13 illustrates the mechanics of boundary extraction. It shows a simple binary object, a structuring element B, and the result of using Eq. (9.5-1). Although the structuring element in Fig. 9.13(b) is among the most frequently used, it is by no means unique. For example, using a 5×5 structuring element of 1s would result in a boundary between 2 and 3 pixels thick.

From this point on, we do not show border padding explicitly.



a b
c d

FIGURE 9.13 (a) Set A. (b) Structuring element B. (c) A eroded by B. (d) Boundary, given by the set difference between A and its erosion.

Skeleton:

Skeletonization is a morphological operation used in digital image processing to extract the thin, elongated representation of an object while preserving its connectivity and topology. It reduces the binary image to its skeleton or medial axis, which represents the centerlines of the objects.

Skeletonization is commonly employed in various applications, such as shape analysis, pattern recognition, and feature extraction. The process involves iteratively thinning the binary image until only the skeleton remains.

There are different algorithms for skeletonization, but the basic idea is to repeatedly apply erosion operations to the image until the desired skeleton is obtained. The erosion operation removes the boundary pixels of objects while preserving their connectivity.

During each iteration of the skeletonization process, the image is eroded using a specific structuring element, usually a 3x3 or 5x5 neighborhood. The structuring element is applied to each pixel of the image, and if the pattern matches a specific condition, the pixel is removed. The process continues until no further changes occur, indicating the convergence to the skeleton.

The specific conditions for pixel removal depend on the chosen algorithm. Generally, pixels are removed if they do not affect the connectivity or topology of the objects in the image. This iterative erosion process thins the objects, preserving their essential shape characteristics.

The resulting skeleton represents the medial axis or centerlines of the objects in the image. It provides a simplified representation that retains important information about the shape and structure of the objects while reducing the complexity of the image.

Skeleton morphological operations have proven to be useful in various applications, such as character recognition, shape matching, object tracking, and analysis of vascular structures in medical imaging.

3a)Explain Image Compression model

An image compression system is composed of two distinct functional components: an encoder and a decoder. The encoder performs compression, and the decoder performs the complementary operation of decompression. Input image is fed into the encoder, which creates a compressed representation of the input.This representation is stored for later use, or transmitted for storage and use at a remote location. When the compressed representation is presented to its complementary decoder, a reconstructed output image is generated. In still-image applications, the encoded input and decoder output are and respectively

8.1 ■ Fundamentals 537

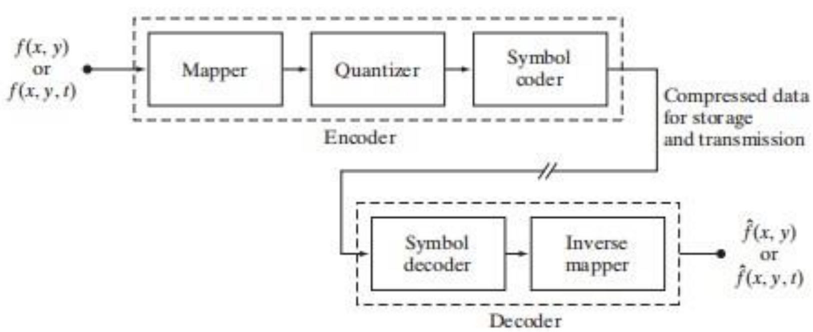


FIGURE 8.5 Functional block diagram of a general image compression system.

The encoding or compression process

In the first stage of the encoding process, a mapper transforms into a (usually nonvisual) format designed to reduce spatial and temporal redundancy

The quantizer in Fig. 8.5 reduces the accuracy of the mapper's output in accordance with a pre-established fidelity criterion. The goal is to keep irrelevant
information out of the compressed representation

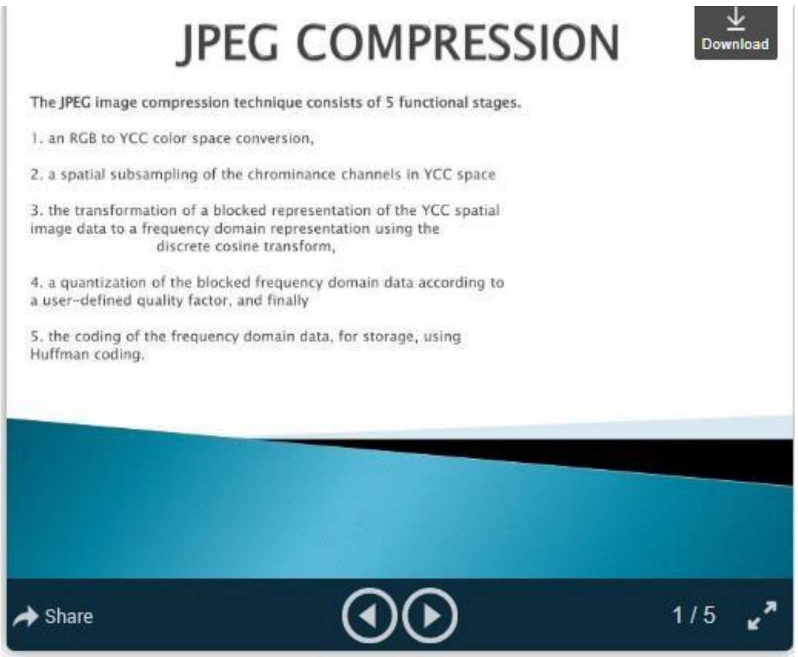
The decoding or decompression process

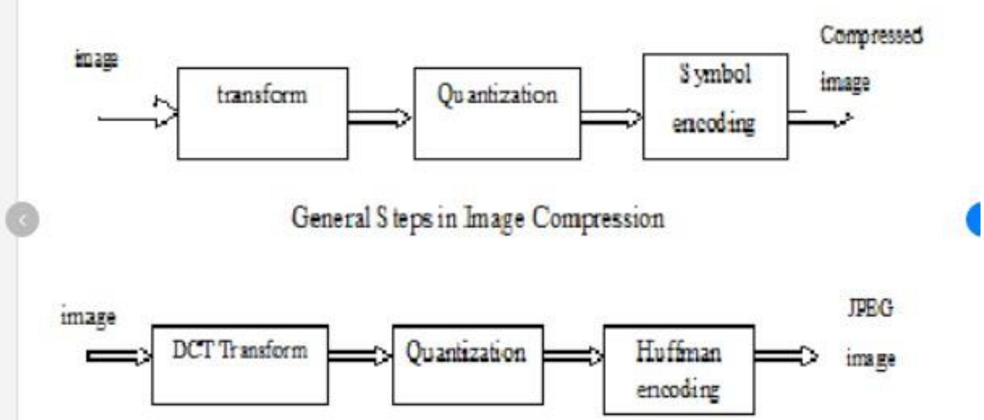
The decoder of Fig. 8.5 contains only two components: a symbol decoder and an inverse mapper. They perform, in reverse order, the inverse operations of the encoder's symbol encoder and mapper. Because quantization results in irreversible information loss, an inverse

quantizer block is not included in the general decoder
mode

In the third and final stage of the encoding process, the symbol coder of Fig. 8.5 generates a fixed- or variablelength code to represent the quantizer output and maps the output in accordance with the code.

3b)describe the steps of JPEG Image Compression





Process Flow of JPEG image compression

3c)Describe bit plan coding? what are the differences between lossy and lossless Image Compression.

The technique, called bit-plane coding, is based on the concept of decomposing a multilevel (monochrome or color) image into a series of binary images (see Section 3.2.4) and compressing each binary image via one of several wellknown binary compression methods

Table 1. Comparison between Lossy and Lossless compression

FACTORS	DATA COMPRESSION	
	LOSSLESS COMPRESSION	LOSSY COMPRESSION
Definition	Lossless compression is a class of data compression algorithms that allow the original data to be perfectly reconstructed from the compressed data[7].	Lossy compression is the class of data encoding methods that uses inexact approximations to represent the content. These techniques are used to reduce the data size for storage, handling, and transmitting content[8].
Algorithm	RLW, LZW, Arithmetic encoding, Huffman coding, Shannon-Fano coding	Transform coding, DCT,DWT, Fractal compression, Rectangle Segmentation and Sparse Matrix Storage (RSSMS).
Uses	Text or programs, images and sound	Images, audio and video.
Images	RAW, BMP, and PNG are all Lossless formats.	JPEG and GUI are lossy image formats.
Audio	WAV, FLAC, and ALAC are all Lossless formats.	MP3, MP4, and OGG are lossy audio formats.
Video	Few lossless video formats are in common consumer use, they would result in video files taking up a huge amount of space.	Common formats like H-264, MKV, and WMV are all lossy. H-264 can provides smaller files with higher qualities than previous generation of video codec because it has a "smaller" algorithm that's better at choosing the data to throw out.
Advantages	It maintains quality. Conversion in any other format possible without loss of audio information.	It can make a multimedia file much smaller than its original size.It can reduce file sizes much more than lossless compression.
Disadvantages	It doesn't reduce the file size as much as lossy compression. Lossless encoding technique cannot achieve high levels of compression.	Conversion to another format only with loss of audio information.It cannot be used in all types of files because it works by removing data. Text and data cannot be compressed because they do not have redundant information.

3d)What is pruning ? why is it so important?

4a)What is segmentation? Describe the necessity of segmentation Answer:

Image segmentation is a commonly used technique in digital image processing and analysis to partition an image into multiple parts or regions, often based on the characteristics of the pixels in the image. Image segmentation could involve separating foreground from background, or clustering regions of pixels based on similarities in color or shape. For example, a common application of image segmentation in medical imaging is to detect and label pixels in an image or voxels of a 3D volume that represent a tumor in a patient's brain or other organs.

Segmentation is a crucial step in image processing that involves dividing an image into meaningful and coherent regions or objects. It plays a vital role in various computer vision tasks, such as object recognition, image analysis, and scene understanding. The primary necessity of segmentation arises from the fact that raw images are essentially a collection of pixels, and without segmentation, it would be challenging to extract meaningful information from them.

Here are some key reasons why segmentation is necessary in image processing:

1. Object Recognition: Segmentation helps in identifying and isolating objects within an image. By separating the foreground objects from the background, it becomes easier to recognize and classify different objects present in an image. Object recognition is crucial for various applications, including autonomous driving, surveillance systems, and robotics.

2. Image Analysis: Segmentation provides a higher-level representation of the image content, allowing for more indepth analysis. By dividing an image into distinct regions, different properties and characteristics of those regions can be studied individually. This enables tasks such as texture analysis, shape detection, and motion tracking, which are essential in medical imaging, satellite imagery analysis, and video processing.

3. Image Understanding: Segmentation helps in understanding the content and structure of an image. By partitioning the image into semantically meaningful regions, it becomes easier to comprehend the relationships between objects, their spatial arrangement, and the overall scene layout. This facilitates higher-level reasoning and decision-making processes in applications like scene understanding, object tracking, and augmented reality.

4. Image Editing and Manipulation: Segmentation is crucial for various image editing tasks, such as background removal, object manipulation, and image retouching. By segmenting the image, specific regions of interest can be isolated, allowing for selective modifications or enhancements. This is widely used in graphic design, photo editing software, and video post-production.

5. Image Compression: Segmentation can be used to exploit the redundancy present in an image and improve compression efficiency. By separating the image into meaningful regions, compression algorithms can allocate more bits to preserve important details while reducing the bits allocated to less critical regions. This leads to higher compression ratios without significant loss of visual quality.

In summary, segmentation is necessary in image processing as it provides a fundamental step for extracting meaningful information from images. It enables object recognition, image analysis, image understanding, image editing, and compression, ultimately facilitating a wide range of computer vision applications across various domains.

4b)Depict the edge linking using the local processing algorithm

Local processing

One of the simplest approaches for linking edge points is to analyze the characteristics of pixels in a small neighborhood about every point (x, y) that has been declared an edge point by one of the techniques discussed in the previous section. All points that are similar according to predefined criteria are linked, forming an edge of pixels that share common properties according to the specified criteria.

The two principal properties used for establishing similarity of edge pixels in this kind of analysis are (1) the strength (magnitude) and (2) the direction of the gradient vector. The first property is based on Eq. (10.2-10). Let S_{xy} denote the set of coordinates of a neighborhood centered at point (x, y) in an image. An edge pixel with coordinates (s, t) in S_{xy} is similar in *magnitude* to the pixel at (x, y) if

$$|M(s,t) - M(x,y)| \leq E \tag{10.2-36}$$

where E is a positive threshold.

The direction angle of the gradient vector is given by Eq. (10.2-11). An edge pixel with coordinates (s, t) in S_{xy} has an *angle* similar to the pixel at (x, y) if

$$|\alpha(s,t) - \alpha(x,y)| \leq A \tag{10.2-37}$$

where A is a positive angle threshold. As noted in Section 10.2.5, the direction of the edge at (x, y) is *perpendicular* to the direction of the gradient vector at that point.

A pixel with coordinates (s, t) in S_{xy} is linked to the pixel at (x, y) if both magnitude and direction criteria are satisfied. This process is repeated at every location in the image. A record must be kept of linked points as the center of the neighborhood is moved from pixel to pixel. A simple bookkeeping procedure is to assign a different intensity value to each set of linked edge pixels.

The preceding formulation is computationally expensive because all neighbors of every point have to be examined. A simplification particularly well suited for real time applications consists of the following steps:

1. Compute the gradient magnitude and angle arrays, $M(x, y)$ and $\alpha(x, y)$, of the input image, $f(x, y)$.
2. Form a binary image, g, whose value at any pair of coordinates (x, y) is given by:

$$g(x,y) = \begin{cases} 1 & \text{if } M(x,y) > T_M \text{ AND } \alpha(x,y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

where T_M is a threshold, A is a specified angle direction, and $\pm T_A$ defines a "band" of acceptable directions about A.

3. Scan the rows of g and fill (set to 1) all gaps (sets of 0s) in each row that do not exceed a specified length, K. Note that, by definition, a gap is bounded at both ends by one or more 1s. The rows are processed individually, with no memory between them.
4. To detect gaps in any other direction, θ, rotate g by this angle and apply the horizontal scanning procedure in Step 3. Rotate the result back by $-\theta$.

https://benchpartner.com/q/explain-about-the-edge-linking-procedures

4c)Define adaptive thresholding. how can thresholding be used in segmentation

Adaptive Thresholding is used in scenes with uneven illumination where same threshold value not usable throughout complete image.
In such case, look at small regions in the image and obtain thresholds for individual sub-images. Final segmentation is the union of the regions of sub-images.
1. Thresholding Segmentation

The simplest method for segmentation in image processing is the threshold method. It divides the pixels in an image by comparing the pixel's intensity with a specified value (threshold). It is useful when the required object has a higher intensity than the background (unnecessary parts).
You can consider the threshold value (T) to be a constant but it would only work if the image has very little noise (unnecessary information and data). You can keep the threshold value constant or dynamic according to your requirements.
The thresholding method converts a grey-scale image into a binary image by dividing it into two segments (required and not required sections).
According to the different threshold values, we can classify thresholding segmentation in the following categories:
Simple Thresholding
In this method, you replace the image's pixels with either white or black. Now, if the intensity of a pixel at a particular position is less than the threshold value, you'd replace it with black. On the other hand, if it's higher than the threshold, you'd replace it with white. This is simple thresholding and is particularly suitable for beginners in image segmentation. Adaptive Thresholding
Having one constant threshold value might not be a suitable approach to take with every image. Different images have different backgrounds and conditions which affect their properties.

Thus, instead of using one constant threshold value for performing segmentation on the entire image, you can keep the threshold value variable. In this technique, you'll keep different threshold values for different sections of an image.

This method works well with images that have varying lighting conditions. You'll need to use an algorithm that segments the image into smaller sections and calculates the threshold value for each of them.

From ChatGPT:

Thresholding is a fundamental technique used in image segmentation to separate objects or regions of interest from the background. It involves selecting a threshold value and classifying each pixel in the image as either foreground or background based on its intensity or other image characteristics.

Here's how thresholding can be used in segmentation:

1. Intensity-based Segmentation: In many cases, objects or regions of interest in an image exhibit different intensity characteristics compared to the background. Thresholding can be employed to separate these regions based on their intensity values. The basic idea is to select a threshold value that effectively distinguishes the foreground objects from the background based on intensity alone.

2. Binary Image Generation: Thresholding converts a grayscale or color image into a binary image, where pixels above the threshold are classified as foreground (object) pixels, and pixels below or equal to the threshold are classified as background pixels. This binary image can be used as a segmentation mask or a preliminary step for further analysis or processing.

3. Adaptive Thresholding: As mentioned earlier, adaptive thresholding adjusts the threshold value locally, taking into account the local characteristics of the image. This can be particularly useful when dealing with images that have non-uniform lighting conditions or variations in contrast. Adaptive thresholding allows for better adaptation to local image properties, leading to more accurate segmentation results.

4. Multiple Thresholds: In some cases, a single threshold may not be sufficient to separate different regions of interest in an image. Multiple thresholds can be utilized to segment an image into multiple classes or categories. This technique, known as multi-level thresholding or multiclass thresholding, enables the identification and separation of distinct objects or regions with different intensity ranges.

5. Feature-based Thresholding: Thresholding can also be applied based on features other than intensity, such as texture, color, gradient magnitude, or other image attributes. By selecting appropriate features and threshold values, it becomes possible to segment the image based on specific characteristics of interest, allowing for more advanced segmentation techniques.

6. Post-processing: Thresholding is often followed by post-processing operations to refine and improve the segmentation results. These operations can include morphological operations (such as erosion and dilation) to remove noise or fill gaps in the segmented regions, or contour extraction to obtain more precise boundaries of the segmented objects.

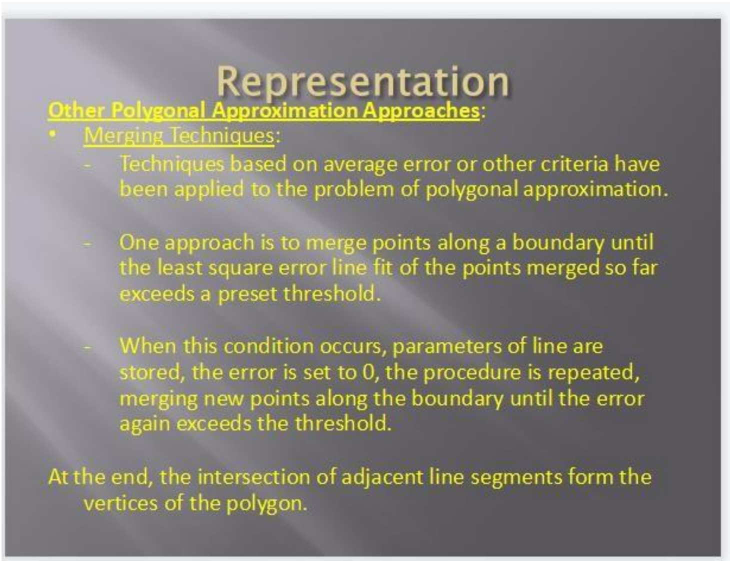
Thresholding is a versatile and widely used technique in image segmentation due to its simplicity and effectiveness. However, it is important to choose appropriate thresholding methods, adjust threshold values carefully, and consider the specific characteristics of the image and the desired segmentation goals for optimal results.

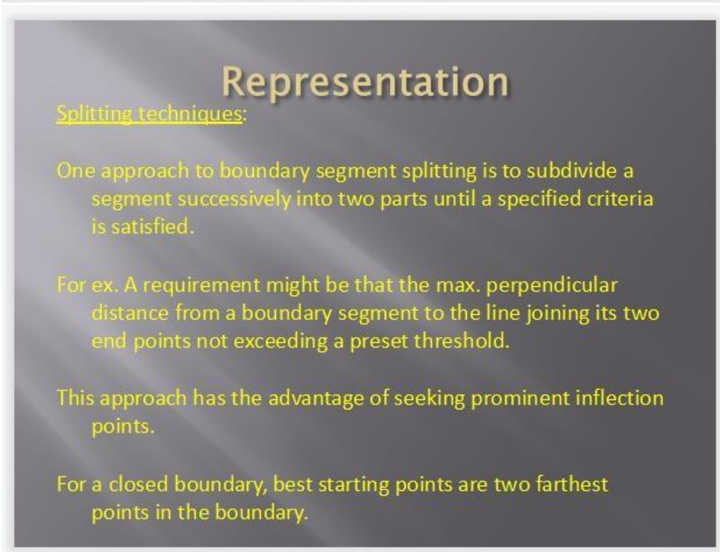
4d)Describe the segmentation process of printed characters

5a)define feature extraction how is the region features a choice
5b)explain the use of motion in segmentation
5c)calculate the area perimeter and shape factor using chaincode
5d)define LBP local binary pattern give examples

6a) define pattern recognition.write down the applications of pattern recognition
6b) describe the minimum distance classifier IRDA decision functions for two pattern class w1 and w2 where to main vectors m1 = [5.1, 1.4,6.5] And m2 = [1.6, 0.4,2.1].
And find the pattern class for X1 = [4.6,1.2,5.2] and X2 = [3.5, 0.3, 1.9] using decision boundary.
6c) describe syntactic pattern recognition
6d) describe a pattern recognition system to recognize 'A' using template meaning matching

7a) describe region splitting and merging algorithm





7b) define the artificial neural network network with 10 inputs and two outputs

7c) describe the pattern recognition system

7d) what are the limitations of the chain code? how can we overcome the shortcomings?

Ans:

The limitations of chain codes in image processing include:

Sensitivity to Noise: Chain codes are highly sensitive to noise and small variations in the boundary of an object. Even minor noise or irregularities can result in significantly different chain code representations, leading to inaccurate boundary descriptions.

Lack of Scale Invariance: Chain codes are not inherently scale-invariant. If an object's size changes, the chain code representation will also change accordingly. This lack of scale invariance makes it challenging to compare or match objects at different scales using chain codes alone.

Rotation Sensitivity: Chain codes are sensitive to object rotations. When an object is rotated, the chain code representation will be different from the original, making it difficult to directly use chain codes for rotation-invariant object recognition or matching.

Autumn 2021

1a) "data is a combination of Information and redundant data" do you agree? justify your answer with proper example OR , write the steps of JPEG algorithm Ans:
Yes, I agree that data is a combination of information and redundant data. Let me justify my answer with an example:

Data refers to raw facts, observations, or measurements that have been collected but do not provide any context or meaning on their own. When this data is processed and organized in a meaningful way, it becomes information. Information is data that has been interpreted, analyzed, and presented in a way that is useful and valuable for decision-making or understanding.

However, within a dataset, there can be instances of redundant data. Redundant data refers to information that is duplicated or unnecessary for the intended purpose of the dataset. It does not add any additional value or new insights but occupies storage space and increases processing time.

For example, let's consider a customer database for an e-commerce company. Each customer entry contains fields such as name, address, email, and order history. The name and address fields provide essential information about the customers. However, if there are multiple entries for the same customer with identical information, it creates redundancy. Storing and processing redundant data in this case would be inefficient and could lead to inaccuracies or inconsistencies if updates or changes are not uniformly applied across all redundant instances.

Therefore, while data can contain both valuable information and redundant data, it is important to identify and eliminate or minimize redundant elements to ensure efficient storage, processing, and analysis of the dataset.

Or:

[7] Steps of JPEG algorithm:

- Compression
  - Phase One: Divide the image.
    divide into 8x8 pixel blocks.
  - Phase Two: Conversion to the frequency domain
    - A direct cosine transformation applied
    - convert spatial domain to frequency domain.
  - Phase Three: Quantization
    - f. info is quantized to remove unnecessary information.
  - Phase Four: Entropy Coding
    - Finally, standard compression techniques compress the final bit stream.

1b) considered the text “Dept of CSE”

A. what is the probability of occurrence of each character

B. using this probabilities construct the Huffman coding table for transmitting this text

C. write the encoded sequence

D. from encoded sequence do decoding

1c) A source contains four symbols(i,i,u,c) with corresponding probability of 0.3 0.2 0.4 and 0.1 respectively construct arithmetic coding to encode and decode a specific word Or,
Encode the following image using LZW coding
126 75 36 75
126 75 36 75
126 75 36 75
126 75 36 75

2a) the shape of structuring element has significant impact of every morphological operation . explain the statement with necessary examples Ans:

The shape of the structuring element has a significant impact on every morphological operation because it determines the neighborhood considered for the operation. Morphological operations, such as erosion and dilation, alter the shape and structure of objects in an image based on the configuration of the structuring element.

Let's consider two common shapes of structuring elements: a square and a cross.

1. Square Structuring Element:
A square structuring element considers a square neighborhood around each pixel during morphological operations. It treats all pixels within the square as neighbors and performs the operation accordingly.

For example, let's say we have a binary image with a square-shaped object and we want to perform erosion to shrink the object. If we use a small square structuring element, such as a 3x3 square, the erosion operation will remove one layer of pixels from all sides of the object, resulting in a smaller square object.

2. Cross Structuring Element:
A cross-shaped structuring element considers a cross-shaped neighborhood around each pixel during morphological operations. It treats the pixels along the horizontal and vertical axes as neighbors and ignores the diagonal pixels.

Continuing with the previous example, if we use a cross-shaped structuring element instead, the erosion operation will only remove the pixels along the horizontal and vertical axes, preserving the diagonal pixels. As a result, the object's corners will remain intact, and the final shape after erosion will be different from that obtained with a square structuring element.

Different structuring element shapes can also produce different results in dilation, opening, and closing operations.
The choice of structuring element depends on the desired effect and the characteristics of the objects in the image.

In summary, the shape of the structuring element directly influences the neighborhood considered during morphological operations, leading to variations in the resulting shape and structure of objects in the image. Selecting an appropriate structuring element shape is crucial to achieve the desired morphological transformation or analysis outcome.

2b) Explain hit or miss transformation in brief with necessary example Ans:

The hit or miss transformation is a morphological operation used to detect specific patterns or configurations in a binary image. It involves the use of two structuring elements: one for the foreground (object) and one for the background (complement of the object).

The operation aims to identify pixels in the image that match both the foreground and background structuring elements simultaneously. These pixels represent locations where the desired pattern or configuration exists.

Here's a step-by-step explanation of the hit or miss transformation:

Define the foreground structuring element (FSE) to represent the desired pattern or configuration.

Define the background structuring element (BSE) to represent the complement of the desired pattern or configuration.

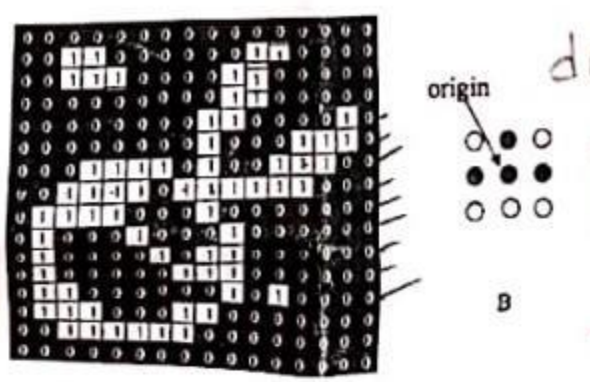
Perform erosion on the input binary image using the FSE. This operation removes pixels that do not match the FSE.

Perform erosion on the input binary image using the complement of the BSE. This operation removes pixels that do not match the complement of the BSE.

Take the intersection of the results obtained from the two erosions. This means considering only the pixels that were preserved in both erosion operations.

The resulting image will contain white pixels at locations where the desired pattern or configuration matches exactly and there is no interference from the background.

2c) Binary image X and Structuring element B are given below



X

Calculate $Y1 = X \circ B$ Where $\circ$ denotes morphological opening operation..

2d) show the histogram of a 3 bit image of the following table gray level 0 1 2 3 4 5 6 7
number of pixels 5 4 3 2 1 6 3 2

3a) “segmentation is one of the most important steps in object recognition” describe in brief

Segmentation is indeed one of the most important steps in object recognition. Object recognition involves identifying and categorizing objects or regions of interest within an image or scene. Segmentation plays a crucial role in this process by dividing the image into meaningful and coherent regions, enabling the recognition algorithms to focus on individual objects.

Here are some key reasons why segmentation is crucial in object recognition:

1. Localization: Segmentation helps in localizing objects within an image by separating them from the background or other surrounding elements. By identifying the boundaries of objects through segmentation, the recognition algorithms can precisely locate and extract the regions of interest for further analysis.

2. Feature Extraction: Once objects are segmented, various features can be extracted from each segmented region.

These features can include color histograms, texture descriptors, shape characteristics, or other relevant attributes. Segmentation provides a well-defined boundary for extracting these features specifically from the regions of interest, leading to more accurate and informative representations for object recognition.

3. Background Suppression: In many cases, the background or surrounding context in an image can contain irrelevant or distracting elements that can hinder object recognition. Segmentation helps to suppress the background by isolating the objects of interest. This allows the recognition algorithms to focus on the foreground objects and disregard the irrelevant background information, improving the overall accuracy of object recognition.

4. Object Classification: Segmentation provides individual object regions that can be treated as separate instances for classification. Once objects are segmented, recognition algorithms can analyze the extracted features and compare them to a pre-trained model or database to determine the class or category of each object. Segmentation enables the classification process to be performed at the object level rather than at the pixel level, leading to more robust and accurate object recognition.

5. Scene Understanding: Segmentation contributes to a higher level of scene understanding by providing information about the spatial arrangement and relationships between objects. By segmenting an image into distinct regions, the recognition algorithms can analyze the interactions and context between objects, leading to a more comprehensive understanding of the scene as a whole.

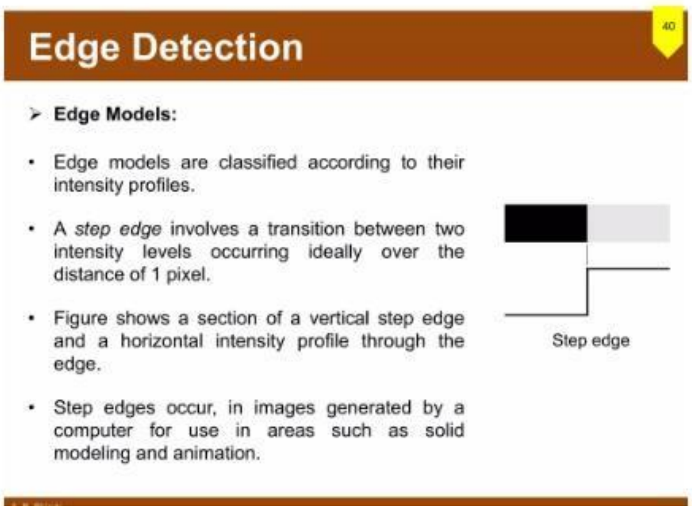
In summary, segmentation is a critical step in object recognition as it enables localization, feature extraction, background suppression, object classification, and scene understanding. It provides the necessary foundation for recognition algorithms to effectively analyze and interpret the objects within an image, leading to accurate and robust object recognition capabilities.

Or **what are edges? how different types of edges can be segmented give necessary example** Answer:

Edges in image processing refer to the boundaries or transitions between different regions or objects within an image. They represent significant changes in pixel intensity, color, or texture and often indicate important features or structures in the image.

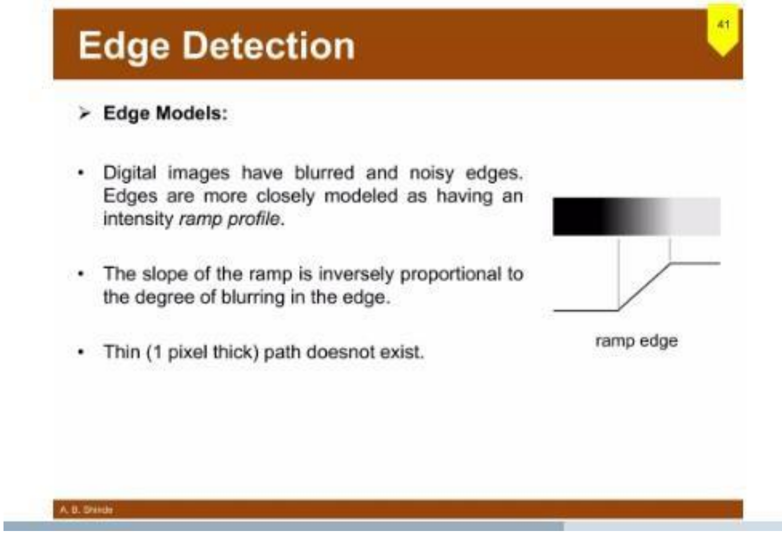
There are several types of edges that can be segmented:

1. Step Edges: Step edges occur when there is a sudden and significant change in intensity or color values across adjacent pixels. They represent sharp transitions and are often observed in images with well-defined boundaries between objects or regions.



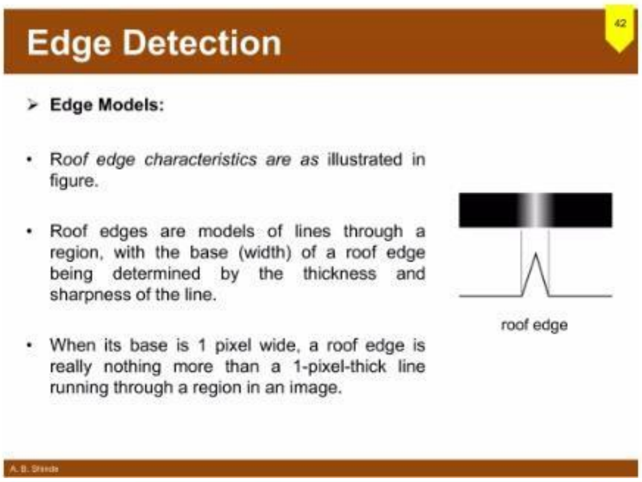
Example: The edge between the sky and the ground in an image of a landscape.

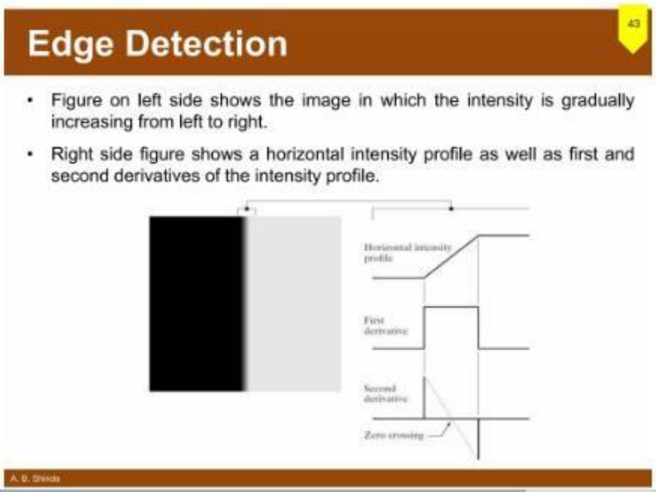
2. Ramp Edges: Ramp edges are characterized by a gradual change in intensity or color values over a region. They represent smoother transitions and are commonly found in textured or gradient regions.



Example: The transition from light to dark in a grayscale gradient image.

3. Roof Edges: Roof edges are formed when there is an intensity or color transition followed by a plateau or constant region. They represent a sudden change followed by a flat portion.





Example: The edge between a person's face and their hair in a portrait image.

4. Ridge Edges: Ridge edges occur when there is a bright or dark line along the center of a region. They represent linear structures or ridges in the image.

Example: The edge along the center of a road in an aerial image.

To segment edges in an image, various edge detection algorithms can be used. These algorithms aim to highlight or mark the pixels that correspond to edge locations. Some commonly used edge detection methods include:

- Gradient-based Methods: These methods calculate the gradient magnitude or the first derivative of the image to detect areas of rapid intensity changes.

- Laplacian of Gaussian (LoG): The LoG method convolves the image with a Gaussian filter and then applies the Laplacian operator to detect regions of intensity transitions.

- Hough Transform: The Hough transform is often used to detect and segment line or curve edges by converting the edge points into parameter space.

These techniques, along with other advanced edge detection methods, help to identify and segment different types of edges in an image, enabling further analysis and processing tasks such as object recognition, shape analysis, or image-based measurements.

3b) why is linking is important in many image processing operations? explain local edge linking algorithm?

Ans:

Linking is important in many image processing operations, particularly in the context of segmentation and object recognition, because it helps connect related pixels or regions to form coherent and meaningful objects or regions. It plays a crucial role in overcoming the challenges of image noise, discontinuities, and occlusions.

Here are some reasons why linking is important:

1. Completeness: Linking helps connect fragmented or disjointed regions or contours, allowing for the formation of complete objects or regions of interest. This is crucial in tasks like object recognition, where complete and coherent objects need to be identified.

2. Continuity: Linking establishes the connectivity between neighboring pixels or regions with similar properties, such as intensity or color, to create continuous boundaries or edges. This helps in accurately delineating the boundaries of objects or regions.

3. Contextual Information: Linking considers the spatial relationships between neighboring pixels or regions, taking into account their proximity and similarity. By linking related elements, it incorporates contextual information and improves the accuracy of segmentation.

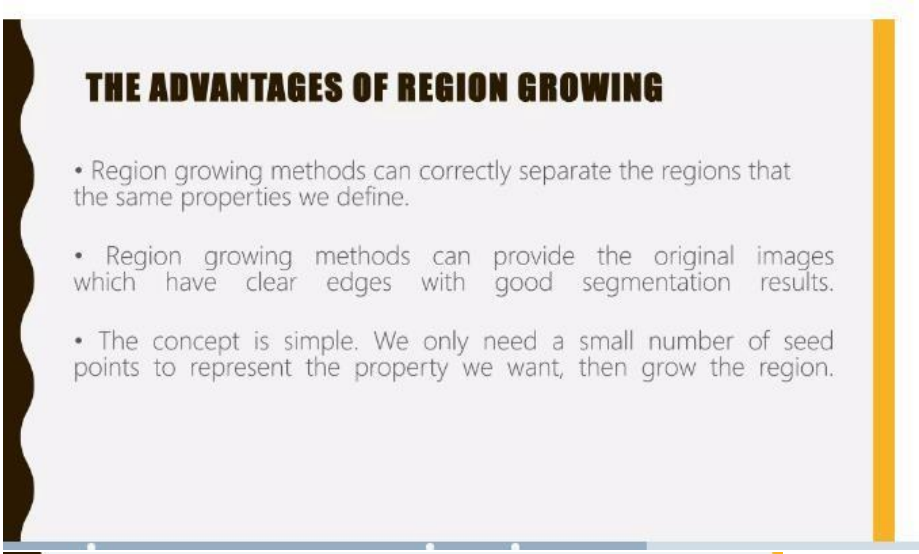
4. Noise Reduction: Linking helps filter out noise or small isolated regions by considering their connectivity to larger, more significant regions. This ensures that only meaningful structures are retained and irrelevant noise is eliminated.

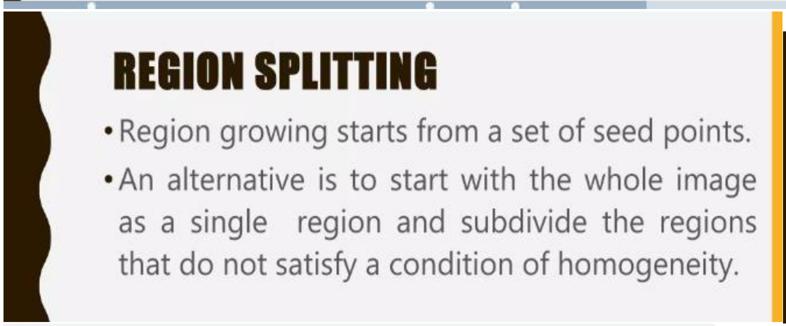
5. Occlusion Handling: Linking is crucial in handling occlusions, where one object partially obscures another. By linking relevant edges or regions, it becomes possible to separate and distinguish occluded objects, allowing for accurate recognition and segmentation.

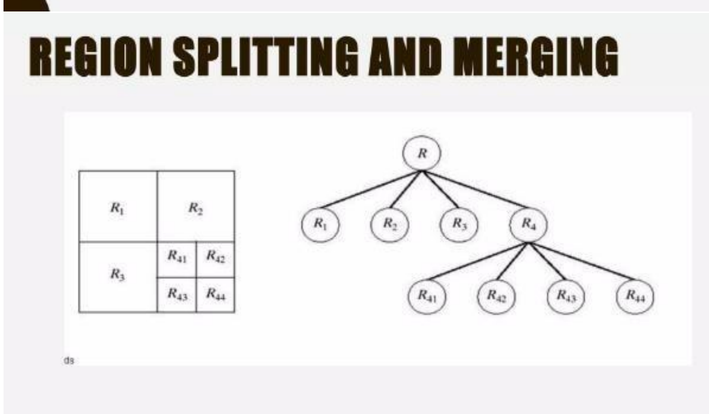
Overall, linking is essential in image processing operations as it enables the formation of complete objects, ensures the continuity of boundaries, incorporates contextual information, reduces noise, and handles occlusions. It plays a vital role in achieving accurate and meaningful segmentation, object recognition, and image analysis. **3c) what is region? how region can be segmented give any algorithm for region segmentation** Ans:

In image processing, a region refers to a connected and coherent area or group of pixels that share similar properties such as intensity, color, texture, or other visual characteristics. Regions can represent objects, backgrounds, or other distinct areas within an image. Segmenting regions involves separating or grouping pixels based on their similarities to form meaningful and distinct regions.

Important in interpreting an image because they may correspond to objects in a scene.For that an image must be partitioned into regions that correspond to objects or parts of an object.







4a)

i) Represent the boundary in figure 1 within 8 directional chain code ii) What does rotational invariant mean in this case? make the chain code in a rotational invariant?

Figure 1

Answer

In mathematics, a function defined on an inner product space is said to have rotational invariance if its value does not change when arbitrary rotations are applied to its argument.

A rotation invariant code sequence also can be obtained by using the first difference of the chain code; the difference between two consecutive digits is defined as the number of directions between them, which is taken to be positive when counterclockwise.

https://ojskrede.github.io/inf4300/notes/week_04/#:~:text=Rotation%20invariant%20chain%20code.&text=8%2Dc onnected%20case-,c8%3D(7%2C7%2C6%2C6%2C,1%2C5%2C7).

4b) what is LBP ? the steps to calculate LBP give necessary example ?

OR,

describe your method of computing skeleton of the images?

SKELETONS

- An important approach to represent the structural shape of a plane region is to reduce it to a graph.
- The reduction may be accomplished by obtaining the skeleton of the region via thinning (skeletonizing) algorithm.
- Skeleton of a region may be defined as the medial axis transformation (MAT).
- MAT of a region R with border B is as follows:
- \- For every point p in R, we find its closest neighbor in B. If p has more than such neighbor, it is said to belong to the medial axis (skeleton).
- 'prairie fire concept'
- Implementation involves calculating the distance from every interior point to every boundary point on region.

SKELETONS

- **Thinning algorithm** deals with deleting the boundary points of a region subject to condition that deleting these points:

1. Does not remove end points.
2. Does not break connectivity &
3. Does not cause excessive erosion of the region.

Thinning algorithm:

- Region points are assumed to have value 1 & background points are assumed to have value 0.

SKELETONS

p9	p2	p3
p8	p1	p4
p7	p6	p5

- Step 1

  flags a contour point for deletion if the following conditions are satisfied:

  a) $2 \leq N(p1) \leq 6$

  b) $T(p1) = 1$

  c) $p2.p4.p6 = 0$

  d) $p4.p6.p8 = 0$

SKELETONS

where N(p1) is the number of nonzero neighbors of p1;

i.e. $N(p1) = p2 + p3 + \ldots\ldots + p8 + p9$ where pi is either 0 or 1.

T(p1) is number of 0-1 transitions in the ordered sequence P2, p3, ..., p8, p9.,

In Step 2:

conditions a & b remain the same, but conditions c & d are changed to

c') $p2.p4.p8 = 0$

d') $p2.p6.p8 = 0$

0	0	1
1	p1	0
1	0	1

Step 1 is applied to every border pixel in binary region under Consideration If 1 or more of conditions a to d are violated, the value of point in question is not changed.

what kind of structural elements will you be using? what happens when you change the size of structuring element?

Structural elements are used in morphological image processing operations to analyze and modify the shapes and structures of objects within an image. They are small, predefined patterns or shapes represented as binary matrices.

When you change the size of a structuring element, several things can happen:

1. Effect on Object Size: In operations like erosion and dilation, the size of the structuring element determines the extent to which objects are eroded or dilated. Larger structuring elements result in more significant changes to object size, while smaller ones produce more localized effects. Increasing the size can cause objects to expand, while decreasing the size can cause them to shrink.

2. Preservation of Object Details: A larger structuring element may lead to the loss of finer details and smaller structures within objects. It tends to smooth out the object boundaries and merge nearby regions. Conversely, a smaller structuring element preserves finer details and allows for better differentiation between objects or regions.

3. Computational Impact: The size of the structuring element affects the computational complexity of the morphological operation. Larger structuring elements require more computational resources as they involve processing a larger number of pixels. Therefore, using larger structuring elements may result in slower processing times.

Choosing the appropriate size of the structuring element depends on the specific task, the characteristics of the objects or structures you are analyzing, and the desired outcome. It often involves a trade-off between preserving object details and achieving the desired modification. Experimentation and adjusting the size according to the specific requirements of the task are typically necessary to obtain optimal results.

It's important to note that the effects of changing the size of a structuring element are not universally predictable and may vary depending on the image content and the specific morphological operation being applied. Therefore, it is often necessary to experiment and fine-tune the size of the structuring element to achieve the desired image processing results.

4c) explain string and tree representation of any object with example

5a) Build the decision functions and decision boundary for two pattern class w1 and w2 where two mean by vectors m1 = [6.1, 1.6, 6] and m2=[1.2,0.2, 2.0] find the class of an unknown pattern X =[3.1, 0.4, 2.8]

5b) how matching by correlation works on pattern recognition? explain with necessary example
OR,
explain syntactical pattern recognition write the steps of recognize the string in brief

5c) write the applications of pattern recognition in autonomous car

Previous Solve by Tasnim (Incomplete)

Previous Solve

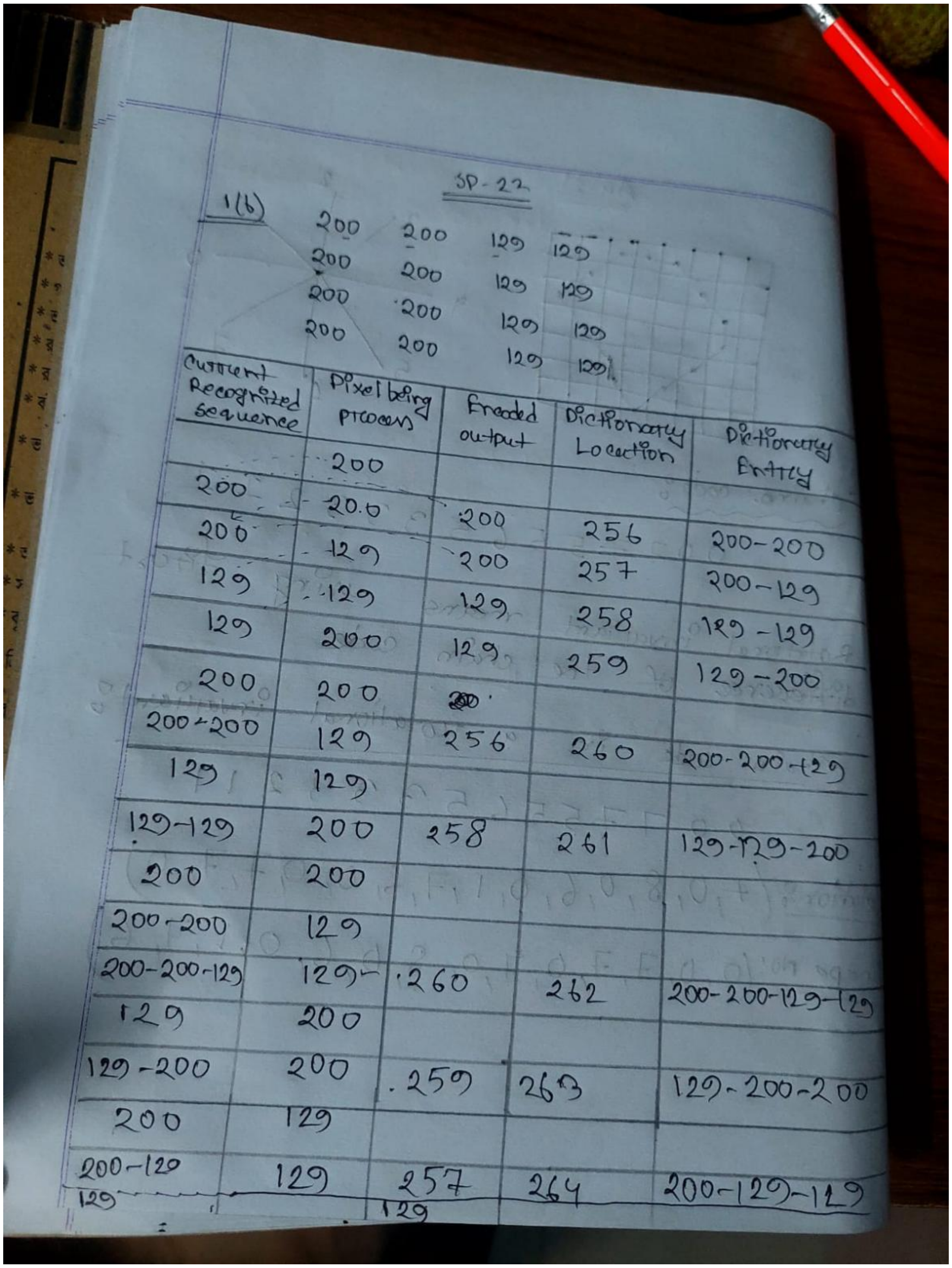
Spring 2022

Group A

1. 1(a) "Data is a combination of information and redundant data" - do you agree? Justify your answer with proper example.

1(b) Encode the following 4 x 4, 8-bit image using LZW coding:

200	200	129	129
200	200	129	129
200	200	129	129
200	200	129	129



SP-22

1(b)

200 200 129 129
200 200 129 129
200 200 129 129
200 200 129 129

Current Recognized Sequence	Pixel being processed	Encoded output	Dictionary Location	Dictionary Entry
	200			
200	200	200	256	200-200
200	129	200	257	200-129
129	129	129	258	129-129
129	200	129	259	129-200
200	200			
200-200	129	256	260	200-200-129
129	129			
129-129	200	258	261	129-129-200
200	200			
200-200	129			
200-200-129	129	260	262	200-200-129-129
129	200			
129-200	200	259	263	129-200-200
200	129			
200-129	129	257	264	200-129-129
129		129		

OR

1(b) Describe in details how color-based segmentation can be used in an attempt to segment "same-colored" object in real color images. What problems arise? What are some ways of dealing with them? 04

Ans:
Color-based segmentation is a technique used to segment objects of the same color in real color images. It involves identifying regions in an image that have similar color characteristics and grouping them together as the same-colored object. Here is a detailed description of how color-based segmentation can be applied and the problems that may arise, along with potential solutions:

1. Color Space Selection: The first step is to select an appropriate color space for representing the image. Commonly used color spaces include RGB, HSV, and Lab. Each color space has its own advantages in terms of separating color information and can affect the segmentation results.

2. Color Thresholding: Once the color space is chosen, color thresholding is performed by defining a range of acceptable color values for the desired object. Pixels falling within this range are considered part of the same-colored object. However, several challenges can arise:

- Illumination Variations: Changes in lighting conditions can cause variations in color appearance. To address this, techniques such as histogram equalization, color normalization, or color constancy algorithms can be applied to reduce the impact of illumination variations.

- Similar Color Interference: Objects with different colors may appear similar due to factors such as lighting conditions, texture patterns, or background clutter. Additional features such as texture, shape, or spatial relationships can be incorporated to differentiate between objects of similar colors.

3. Noise and Artifacts: Image noise and artifacts can negatively impact color-based segmentation. Preprocessing steps like noise reduction techniques (e.g., median filtering) or smoothing filters can help reduce the impact of noise and improve the segmentation results.

4. Parameter Selection: The selection of color thresholds and segmentation parameters is crucial for achieving accurate results. These parameters may vary depending on the specific image and object characteristics. Techniques such as adaptive thresholding or statistical approaches can be used to dynamically determine suitable thresholds.

5. Post-processing: After initial segmentation, post-processing steps can be applied to refine the results. These steps may include morphological operations (e.g., erosion, dilation) to remove small noise or gaps, region merging/splitting techniques for better connectivity, or contour smoothing to improve the object boundaries.

6. Evaluation and Validation: It is important to evaluate and validate the segmentation results using appropriate metrics and ground truth data. This helps assess the accuracy and effectiveness of the color-based segmentation method and make necessary adjustments if required.

While color-based segmentation is effective for segmenting same-colored objects, it has limitations. Some challenges include variations in lighting, similar colors of different objects, and noise interference. To address these challenges, techniques like illumination normalization, incorporating additional features, noise reduction, parameter tuning, and post-processing steps can be employed. Additionally, combining color-based segmentation with other segmentation methods or machine learning approaches can further enhance the segmentation accuracy and robustness in real-color images.

1(c) What is Bit Plane decomposition? Write the benefit of bit-plane decomposition with necessary example.

Ans:
Bit plane decomposition is a technique used in image processing to decompose an image into its individual bit planes. In this technique, the binary representation of each pixel in the image is separated into different bit planes, with each bit plane containing the binary values of a specific bit position (e.g., the most significant bit plane, second most significant bit plane, etc.).

The benefit of bit plane decomposition is that it provides a way to analyze and manipulate the image at different levels of detail, based on the significance of the bits. Here's an example to illustrate the benefit:

Let's consider a grayscale image where each pixel value ranges from 0 to 255, represented by 8 bits (1 byte). By performing bit plane decomposition, we can extract the individual bit planes for each pixel value.

For instance, consider a pixel value of 150, which in binary representation is 10010110. Through bit plane decomposition, we obtain the following bit planes:

Bit Plane 7: 1 0 0 1 0 1 1 0 (Most Significant Bit - MSB)
Bit Plane 6: 0 0 0 0 0 0 0 0
Bit Plane 5: 0 0 0 0 0 0 0 0
Bit Plane 4: 0 0 0 0 0 0 0 0
Bit Plane 3: 1 0 0 1 0 1 1 0
Bit Plane 2: 0 0 0 0 0 0 0 0
Bit Plane 1: 1 0 0 0 0 0 0 0
Bit Plane 0: 0 0 0 0 0 0 0 0 (Least Significant Bit - LSB)

Each bit plane represents a different level of information present in the image. The MSB (bit plane 7) contains the most significant information, while the LSB (bit plane 0) contains the least significant information.

The benefit of bit plane decomposition becomes apparent when we observe the impact of manipulating or discarding certain bit planes. For example:

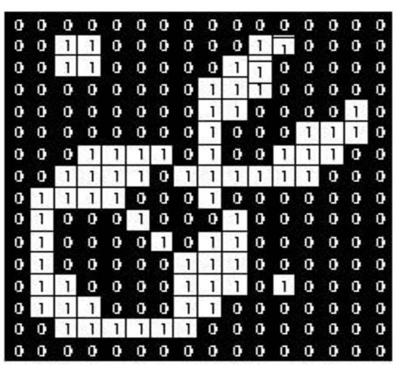
1. Image Compression: By discarding the bit planes with lower significance (e.g., bit planes 0 to 3), we can achieve lossy compression of the image, reducing the storage requirements without significant degradation in visual quality.

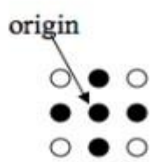
2. Image Enhancement: By manipulating or applying image processing operations on specific bit planes, we can enhance specific details or features in the image. For example, amplifying or attenuating the higher bit planes (e.g., bit planes 5 to 7) can enhance fine details or emphasize edges in the image.

3. Image Watermarking: Bit plane decomposition can be utilized in watermarking techniques. By embedding watermarks in specific bit planes, the watermark can be selectively extracted or detected without affecting the overall image appearance.

Overall, bit plane decomposition allows for analysis, manipulation, and selective processing of images at different levels of detail, based on the significance of the bits. It provides flexibility and control in various image processing applications, such as compression, enhancement, and watermarking.

2(a) Binary Image X and Structuring element B are given below.





B

X

$Y1 = X \Theta B$, Where Θ denotes morphological erosion operation.
$Y2 = X \oplus B$, Where $\oplus$ denotes morphological dilation operation.

https://www.youtube.com/watch?v=77IvijY3H8g

i) Calculate AoB= (AΘB) ⊕ B;Where AoB denotes morphological opening operation.

ii) What is pruning? Why it is important in many image processing applications?2

Answer:

Pruning is a technique used in digital image processing to remove unwanted parasitic components (spurs) from images. It is typically used as a complement to the skeleton and thinning algorithms.

Pruning is important in many image processing applications, including:

- Character recognition: Pruning can be used to remove unwanted spurs from the skeletons of characters, which can improve the accuracy of character recognition algorithms.
- Object detection: Pruning can be used to remove unwanted spurs from the edges of objects, which can improve the accuracy of object detection algorithms.
- Image segmentation: Pruning can be used to remove unwanted spurs from the boundaries of image segments, which can improve the accuracy of image segmentation algorithms.

There are several different methods for pruning images. One common method is to use a morphological operation called erosion. Erosion is a local operation that erodes away the edges of an image. This can be used to remove small spurs from the edges of objects or characters.

Another common method for pruning images is to use a thresholding operation. Thresholding is a global operation that classifies each pixel in an image as either foreground or background. This can be used to remove small spurs from the entire image.

The choice of which pruning method to use depends on the specific application. For example, erosion is typically used for character recognition, while thresholding is typically used for object detection.

Here are some of the benefits of using pruning in image processing:

- Improved accuracy: Pruning can improve the accuracy of image processing algorithms by removing unwanted spurs from images. This can lead to improved performance in applications such as character recognition, object detection, and image segmentation.
- Reduced noise: Pruning can also be used to reduce noise in images. This can improve the quality of images and make them easier to process.
- Simplified images: Pruning can simplify images by removing unnecessary detail. This can make images easier to understand and interpret.

Overall, pruning is a powerful technique that can be used to improve the accuracy, reduce noise, and simplify images in many image processing applications.

OR

2(a) i) In the pattern recognition and classification there are four basic steps 4 involved, which are sensing an image, segmenting the image, extracting the features from the segmented objects, and classification to recognize the specific object. Based on this procedure explain how you can segment and classify a specific object from an 8 bit color image.

Answer:

Ans:
To segment and classify a specific object from an 8-bit color image, we can follow the basic steps of pattern recognition and classification: sensing, segmentation, feature extraction, and classification. Here's a detailed explanation of the procedure:

1. Sensing the Image:
   - Capture the 8-bit color image using a sensing device, such as a camera or scanner.
   - Ensure proper lighting conditions and capture the object of interest with sufficient image resolution.

2. Segmentation:
   - Apply segmentation techniques to separate the object from the background:
     - Color-based Segmentation: Set a threshold on the color channels to distinguish the object from the background based on color intensity. For example, you can use color thresholding techniques to identify regions that fall within a specified color range.
     - Region-based Segmentation: Utilize algorithms such as region growing, graph cuts, or watershed segmentation to group pixels with similar color characteristics that belong to the object.
     - Edge-based Segmentation: Employ edge detection algorithms like Canny or Sobel to identify object boundaries by detecting significant changes in color intensity or gradients.

3. Feature Extraction:
   - Extract relevant features from the segmented object. These features should capture discriminative information for classification.
   - In the case of an 8-bit color image, features can be derived from color attributes such as:
     - Color Histogram: Calculate the distribution of color intensities or color channels to represent the color distribution within the object.
     - Color Moments: Compute statistical moments, such as mean, standard deviation, or higher-order moments, to describe the color properties of the object.
     - Color Texture Descriptors: Analyze texture patterns in the color channels, such as Local Binary Patterns (LBP) or Haralick features, to capture textural information.

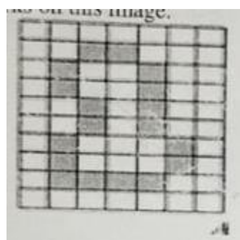
4. Classification:
   - Utilize a classification algorithm to recognize the specific object based on the extracted features.
   - Various classification techniques can be employed, including:

- Supervised Learning: Train a classifier, such as Support Vector Machines (SVM), Random Forests, or Neural Networks, using labeled training data with known object classes. Then, apply the trained classifier to classify new objects.
- Unsupervised Learning: Utilize clustering algorithms like k-means or Gaussian Mixture Models (GMM) to group similar objects together based on extracted features.
- Deep Learning: Apply deep neural network architectures, such as Convolutional Neural Networks (CNNs), for end-to-end feature learning and object classification.

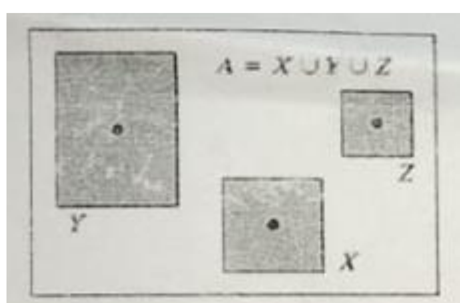
The classification step will provide the recognition or classification result, indicating the specific object present in the 8-bit color image.

It's important to note that the choice of segmentation techniques, feature extraction methods, and classification algorithms can vary depending on the specific characteristics of the objects and the application domain. Experimentation, evaluation, and fine-tuning of each step may be necessary to achieve accurate segmentation and classification results for a given 8-bit color image.

ii) Perform the region filling algorithm on the following image. Explain step by step how this algorithm works on this image.



2(b) Apply (AΘ X) n (A^cΘ (W-X)) morphological operation on the following image.



Group-B

3(a) Analytics produce a decision boundary with the equation 5.1 x_1 -0.3x2 - 8.43. The first and second decision functions are produced as 6.6x1+1.2x2-28.33 and 5.6x,+3.8x-10.0 respectively. Determine the two mean vector (m1, m2) from the above information.

3(b) Proof the equation P(Ri)= TRUE for region.

P(Ri) is a logical predicate defined over the points in set Ri, and ∅ is the null set. Describe

3(c) Canny edge detection algorithm.

The Canny edge detection algorithm is a popular and widely used method for detecting edges in digital images. It was developed by John F. Canny in 1986 and is known for its ability to accurately detect edges while minimizing noise and producing thin, well-defined edges.

The Canny edge detection algorithm consists of several steps:

1. **Image smoothing**: The algorithm starts by applying a Gaussian blur to the input image. This step helps to reduce noise and remove small details that are not relevant to the edge detection process.

2. **Gradient calculation**: The next step involves calculating the gradients of the smoothed image. The image is convolved with two separate filters (Sobel operators) in the horizontal and vertical directions. These filters highlight regions of the image with strong intensity changes, which are likely to correspond to edges.

3. **Gradient magnitude and direction**: The gradient magnitudes and directions are computed based on the horizontal and vertical gradient components. The magnitude represents the strength of the gradient at each pixel, while the direction indicates the orientation of the gradient.

4. **Non-maximum suppression**: In this step, the algorithm aims to thin out the detected edges to obtain a single-pixel-wide edge. It involves scanning the gradient magnitude image and suppressing pixels that are not local maximums along the direction of the gradient. This ensures that only the most significant edges are preserved.

5. **Double thresholding**: A double thresholding technique is employed to classify the remaining edge pixels into strong, weak, or non-edges. This is done by defining high and low thresholds. If a pixel's gradient magnitude is above the high threshold, it is marked as a strong edge. If it is below the low threshold, it is considered a non-edge. Pixels with values between the low and high thresholds are labeled as weak edges.

6. **Edge tracking by hysteresis**: The final step involves connecting weak edges to strong edges to form continuous edges. This is achieved by analyzing the connectivity of weak edges. If a weak edge pixel is adjacent to a strong edge pixel, it is promoted to a strong edge. This process continues until no more weak edges can be connected.

The output of the Canny edge detection algorithm is a binary image where the detected edges are represented by white pixels and the non-edge regions are black.

The Canny edge detection algorithm is widely used in various computer vision and image processing applications, including object detection, image segmentation, and feature extraction.

Or,

Algorithm contains following steps:

1. Filtering input image with Gaussian filter with given standard deviation (filter size should be equal or greater than 6 * sigma)

2. Determining horizontal and vertical gradients of the filtered image

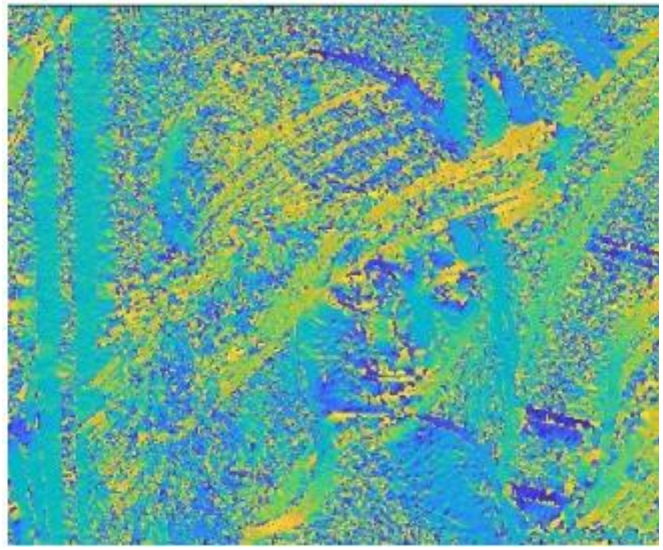
Original Image **Gradients in x direction** **Gradients in y direction**

3. Determining magnitude and angle od the gradients with following formulas:

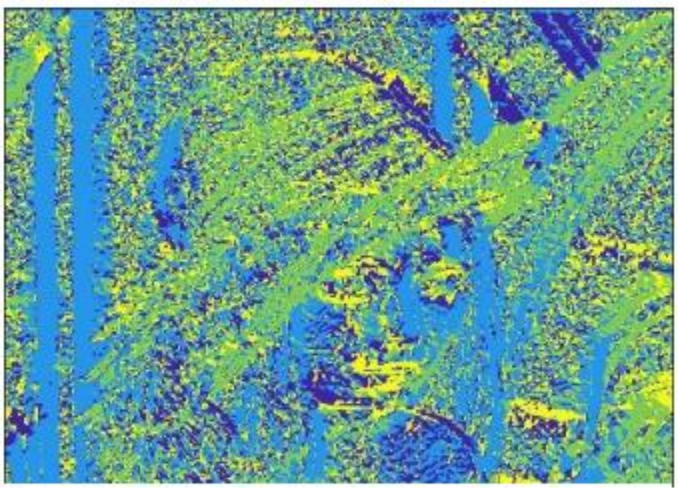
$$magnitude = \sqrt{\frac{dI}{dx}^2 + \frac{dI}{dy}^2}$$

$$angle = \operatorname{atan}\left(\frac{\frac{dI}{dy}}{\frac{dI}{dx}}\right)$$

Gradient magnitude **Gradient angle**



4. Quantization of the angle of the gradient on the following directions: 0, -45, 45, 90 degrees.



5. Supression of the gradients that do not represent local maximum.
We iterate thorugh every pixel of the magnitude of the gradient and we read qunatized value of the gradient angle for that pixel. For every direction of the gradient angle specific matrix of 0s and 1s is defined:

90 degrees

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

45 degrees

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

0 degrees

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

-45 degrees

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

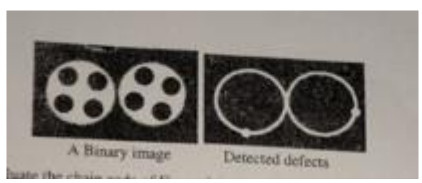
We take the matrix that corresponds to the read quantized value of the gradient angle and multiply it with 3x3 gradient magnitude surrounding of the pixel. If the maximum value in the resulting 3x3 matrix is not in the middle, the pixel value is set to zero. This must NOT be done inplace!

6. Determining the maps of weak and strong edges based on low and high threshold.
All values of the magnitude of the gradient that are higher than T_high are set to value 1 in the map of the combined edges and they go into the map of the strong edges. All values between T_low and T_high are set to value 50/250 in the map of the combined edges and they go into the map of the weak edges. All values below T_low are set to 0.

Weak edges **Strong edges** **Combined edges**

7. Determining final map of edges by including weak edges that are connected with strong edges. We iterate thorugh the combined map of the edges and if the edge is weak we check the 3x3 surrounding arround that edge. If in that surrounding exists at least 1 strong edge, the value of the weak edge is set to strong. This must NOT be done in place! This procedure repeats until there is no change from weak to strong.

4(a) Shown below A is an original binary image which has some defects. How would you detect the defects of the original image using morphological operators with different structural elements? Explain each structural element with proper example. The detected defects sample is also given below. 05


A Binary image   Detected defects

4(b)

- **i. Evaluate the chain code of Figure 1 (use 8-direction).**
- **ii. Chain code has some limitations? Make the chain code in a rotational invariant.**
- **iii. Find the area of the figure 1.**

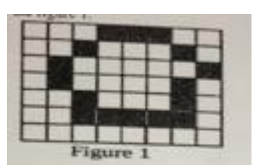


Figure 1

OR

4 (b)

i) **Describe your method of computing the skeleton of the images. What kind of structural elements will you be using? What happens when you change the size of the structuring element?**

ii) **Define Local Binary Pattern (LBP) with example.**

5(a)

In the pattern recognition we have to extract the features from the object and that features are needed to recognize through recognizing algorithm. For recognizing a specific object, is it possible to recognize the object features through neural network that features are extracted from the object through local binary pattern? Justify your answer with a 4x4 8-bit color image.

Ans:
Yes, it is possible to recognize object features through a neural network using features extracted from the object using the Local Binary Pattern (LBP) method. LBP is a widely used texture descriptor that captures local patterns and variations in an image.

To justify the answer, let's consider a 4x4 8-bit color image as an example. The LBP algorithm can be applied to this image to extract local patterns as follows:

1. **Image Conversion**: Convert the 8-bit color image to grayscale. This can be done by taking the average of the RGB values for each pixel or by using other color-to-grayscale conversion methods.

2. **LBP Calculation**: Divide the grayscale image into a grid of cells, such as 4x4. For each pixel within a cell, compare its intensity value with the surrounding neighbor pixels (usually 8 neighbors in a circular pattern). Set a binary code for each neighbor pixel based on whether its intensity is greater or smaller than the central pixel. This results in an 8-bit binary pattern for each pixel within a cell.

3. **Histogram Calculation**: Count the occurrences of each unique binary pattern within each cell and create a histogram. The histogram represents the distribution of local patterns within the image.

4. **Feature Vector**: Concatenate the histograms from all the cells to form a feature vector. In the case of a 4x4 grid, we would have a feature vector of size (16 x Number of unique binary patterns).

Once the feature vector is extracted using the LBP method, it can be used as input to a neural network for object recognition. The neural network can be trained on a dataset of labeled objects, where the extracted LBP features are paired with their corresponding object classes. The network learns to recognize patterns and correlations between the extracted features and the object classes, enabling it to predict the class of new, unseen objects based on their LBP features.

By training a neural network with LBP features extracted from objects, it can learn to recognize object classes based on the extracted local patterns and variations. The network's ability to generalize and make predictions on unseen data depends on the quality and representativeness of the training dataset, the network architecture, and the training process.

5(b) "Template matching is a popular way to identify patterns from images" 5

i. **Explain the formulation of correlation-based algorithm with example.**
ii. **Describe Pattern recognition model.**

Answer:

Ans:
1st part :
To compute the skeleton of an image, I'll describe a commonly used method known as the "thinning" algorithm. Thinning is a morphological operation that iteratively reduces the foreground regions of an image to their skeleton, a thin representation of the original shapes.

Here's an outline of the steps involved in computing the skeleton using the thinning algorithm:

1. **Thresholding**: Convert the input image to a binary image by applying an appropriate thresholding technique. This step separates the foreground objects from the background.

2. **Boundary extraction**: Extract the boundaries of the foreground objects in the binary image. This can be done using contour extraction techniques like the Moore-Neighbor Tracing algorithm or the Freeman Chain Code.

3. **Thinning iteration**: Repeat the following steps until no further changes occur in the image:
   - a. Initialize a blank output image to store the thinned skeleton.
   - b. Iterate over the boundary pixels of the foreground objects.
   - c. Identify each boundary pixel's 8-neighborhood (or 4-neighborhood, depending on the desired connectivity).
   - d. Determine if the current boundary pixel is a candidate for thinning by evaluating specific conditions based on its neighborhood. These conditions typically involve connectivity, neighborhood patterns, and preservation of topology.
   - e. If the current boundary pixel satisfies the thinning conditions, mark it for deletion in the output image.
   - f. Once all boundary pixels have been processed, delete the marked pixels from the original image.
   - g. Copy the updated image to the output image.
   - h. Repeat the iteration with the updated image.

4. **Post-processing**: After the thinning algorithm has completed, you can perform additional post-processing steps to refine the skeleton if needed. This may include removing noise, closing gaps, or improving connectivity.

The thinning algorithm continues iterating until no further changes occur, indicating that the skeleton has been extracted as much as possible.

It's important to note that the effectiveness of the thinning algorithm can depend on factors such as image quality, noise levels, and the specific characteristics of the objects in the image. Experimentation and adjustments to the thinning conditions and post-processing steps may be necessary to achieve the desired skeleton representation.

2^{nd} part:

To detect defects in an original image using morphological operators, we can use different structural elements to highlight specific characteristics or shapes of the defects. Morphological operators, such as erosion and dilation, are used to modify the shape and structure of objects in an image.

Here are a few examples of structural elements and how they can be used to detect defects:

1. **Square Structural Element**:
   A square structural element is a square-shaped neighborhood that is used in morphological operations. It can be defined by its size, such as a 3x3 square or a 5x5 square. This structural element is useful for detecting defects that have a square or block-like shape. When applied with dilation, it can expand the defect areas, making them more prominent. Similarly, applying erosion with the square structural element can help to remove small noise or smooth out irregularities.

2. **Cross Structural Element**:
   A cross structural element is a cross-shaped neighborhood with a central pixel and four arms extending horizontally and vertically. This structural element is particularly effective in detecting line-like defects. When used with dilation, it can thicken and enhance the line-like defects, making them easier to identify. Erosion with the cross structural element can remove small line-like details and thin out the defects.

3. **Disk Structural Element**:
   A disk structural element is a circular-shaped neighborhood defined by its radius. It is suitable for detecting defects that have a circular or rounded shape. By applying dilation with the disk structural element, circular defects can be expanded and emphasized. Erosion with the disk structural element can help to smooth out the defects and remove small irregularities within them.

4. **Custom Structural Element**:
   Depending on the specific characteristics of the defects you are trying to detect, you can design and use custom structural elements. For example, if you are looking for defects with a specific pattern or shape, you can create a structural element that matches that pattern. Custom structural elements offer flexibility in targeting specific defect types or shapes.

It's important to note that the selection of the appropriate structural element depends on the specific characteristics of the defects and the desired outcome. Experimenting with different structural elements and combinations of morphological operations (such as dilation, erosion, opening, and closing) can help in accurately detecting and enhancing defects in the original image.

define local binary pattern example

Local Binary Pattern (LBP) is a texture descriptor that captures the local patterns and variations within an image. It assigns a binary code to each pixel in an image based on the comparison of its intensity value with its neighboring pixels.

To define the Local Binary Pattern, let's consider a 3x3 neighborhood around a central pixel in a grayscale image. We can assign a binary code to the central pixel based on the intensity comparison with its neighbors.

1. **Central Pixel**: The central pixel is denoted by P(x_c, y_c), where (x_c, y_c) represents its coordinates.
2. **Thresholding**: Compare the intensity value of the central pixel P(x_c, y_c) with its 8 neighboring pixels in a circular pattern. For simplicity, let's assume we move clockwise starting from the top-left neighbor. If the intensity of a neighbor pixel is greater than or equal to the intensity of the central pixel, assign it a value of 1. Otherwise, assign it a value of 0.
3. **Binary Code**: Concatenate the binary values of the 8 neighboring pixels in a clockwise manner to form an 8-bit binary code. This binary code represents the local pattern around the central pixel.
4. **Decimal Conversion**: Convert the binary code to decimal. This decimal value represents the LBP value of the central pixel.

For example, let's consider a grayscale image with the intensity values of the 3x3 neighborhood around the central pixel P(x_c, y_c) as follows:

1 4 3
2 6 5
7 9 8

Let's assume the intensity of the central pixel P(x_c, y_c) is 6. We can compare the intensity values of the neighboring pixels with the central pixel and assign binary values:

1 0 0
0 1 1
1 1 1

The binary code formed by concatenating the binary values in a clockwise manner is "00111101". Converting this binary code to decimal, we get the LBP value for the central pixel P(x_c, y_c) as 61.

This process is repeated for each pixel in the image, resulting in a new image where each pixel represents the LBP value of the corresponding pixel in the original image.

Local Binary Pattern is a powerful texture descriptor that captures the local structure and texture information in an image, making it useful for various computer vision tasks such as object recognition, face recognition, and texture analysis.

OR

5(b) Describe minimum distance classifier. Build the decision boundary for two pattern classes wl and w2. Where two mean vectors $R1^T$ (5.6, 1.2, 6.2) and $R2^T$= (1.4, 0.3, 3.7). Explain how decision boundary works on classification stage.

Spring 2021

1. a) Define image compression. Write the necessity of image compression in brief. 1

Answer:

What is image compression?

Image compression is a process applied to a graphics file to minimize its size in bytes without degrading image quality below an acceptable threshold. By reducing the file size, more images can be stored in a given amount of disk or memory space. The image also requires less bandwidth when being transmitted over the internet or downloaded from a webpage, reducing network congestion and speeding up content delivery.

Why do We Need image compression?

- Storage
    - Still image
        - A page of A4 size 600dpi□100 MB
        - Single color image of camera□10-30MB
        - Scanned 3x7 inch photo 300dpi-> 30 MB
    - Video
        - 4K×2K×3 ×12 bits/pel = 48 MB/frame or 1 GB/sec or 70 GB/min.
        - 2 hours movie 720x 480 □ 208 GB
- For data TRANSMISSION
    - DVD
    - Remote Sensing
    - Video conference
    - FAX
    - Control of remotely piloted vehicle
        - The bit rate of uncompressed digital cinema data exceeds one Gbps.
- Data access:

Consider a black and white image that has a resolution of 1000*1000 and each pixel uses 8 bits to represent the intensity. So the total no of bits required = 1000*1000*8 = 80,00,000 bits per image. And consider if it is a video with 30 frames per second of the above-mentioned type images then the total bits for a video of 3 secs is: 3*(30*(8, 000, 000)) =720, 000, 000 bits as we see just to store a 3-sec video we need so many bits which is very huge. So, we need a way to have proper representation as well to store the information about the image in a minimum no of bits without losing the character of the image. Thus, image compression plays an important role.

b) **Given a four-symbol source {A, B, C, D} with source probabilities {0.2, 0.2, 0.4, 0.2}, arithmetically 2.5 encode the sequence "ACDCB".**

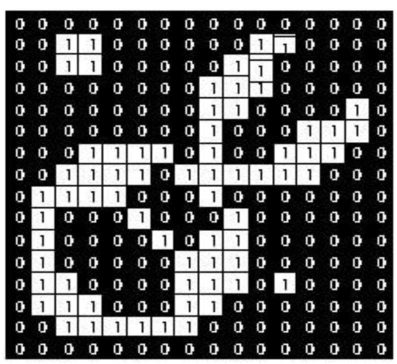
c) **Encode and decode the following 4 × 4, 8-bit image using LZW coding: 2.5**

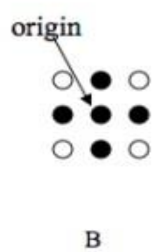
208	**208**	**129**	**129**
208	**208**	**129**	**129**
208	**208**	**129**	**129**
208	**208**	**129**	**129**

Answer:

2. a) Describe Hit-or-Miss transformation in brief? Why this operation in used?

b) Binary Image X and Structuring element B are given below. 3





X

$Y1 = X \ominus B$, Where $\ominus$ denotes morphological erosion operation.

$Y2 = X \oplus B$, Where $\oplus$ denotes morphological dilation operation.

i) **Calculate**

ii) **Calculate**

3. a) Describe the fundamental formula of a region.

Answer:

Basic definition :- A group of connected pixels with similar properties.

Important in interpreting an image because they may correspond to objects in a scene.
For that an image must be partitioned into regions that correspond to objects or parts of an object.

(a) $\bigcup_{i=1}^{n} R_i = R.$

(b) R_i is a connected region, $\mathrm{i} = 1,\ 2,\ \ldots, \mathrm{n}$

(c) $R_i \bigcap R_j = \varnothing$ for all $i = 1, 2, \ldots, n.$

(d) $P(R_i) = TRUE$ for $i = 1, 2, \ldots, n.$

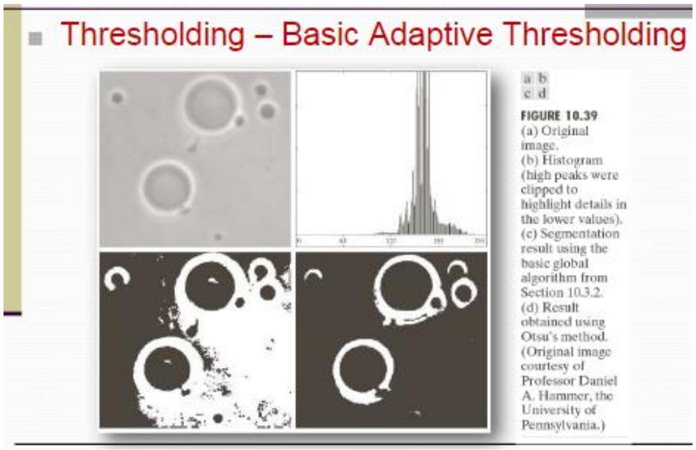
(e) $P(R_i \bigcup R_j) = FALSE$ for any adjacent region R_i and R_j.

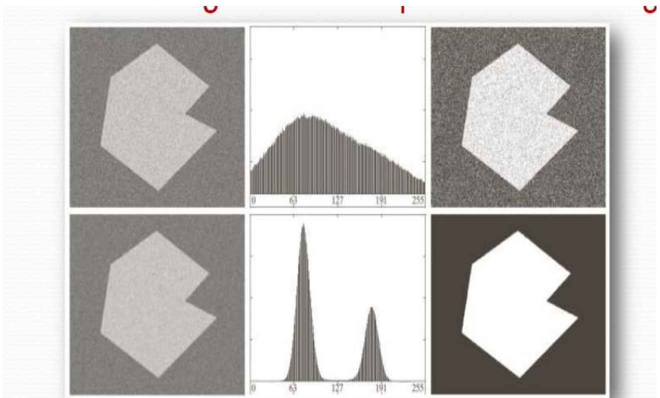
$P(R_i)$ is a logical predicate defined over the points in set R_i and $\varnothing$ is the null set.

b) Write down the applications of adaptive thresholding. 2

Answer:

- Adaptive Thresholding is used in scenes with uneven illumination where same threshold value not usable throughout complete image.
- In such case, look at small regions in the image and obtain thresholds for individual sub-images. Final segmentation is the union of the regions of sub-images.





a b c
d e f

FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Adaptive thresholding is a technique used in image processing to segment images into foreground and background objects. It is a more robust method than global thresholding, which uses a single threshold value for the entire image. Adaptive thresholding uses a different threshold value for each pixel, based on the values of its neighboring pixels. This makes it more adaptable to images with uneven illumination or noise.

Here are some of the applications of adaptive thresholding in image processing:

- Image segmentation: Adaptive thresholding can be used to segment images into foreground and background objects. This can be used for a variety of tasks, such as object detection, edge detection, and image classification.
- Noise removal: Adaptive thresholding can be used to remove noise from images. This can be done by setting the threshold value to a high enough value to remove all noise pixels.
- Image enhancement: Adaptive thresholding can be used to enhance images by increasing the contrast between foreground and background objects. This can be done by setting the threshold value to a low enough value to capture all detail in the image.
- Image restoration: Adaptive thresholding can be used to restore images that have been damaged by noise, blur, or other artifacts. This can be done by setting the threshold value to a value that is optimal for the specific image.

Here are some of the most popular adaptive thresholding algorithms:

- Mean adaptive thresholding: This algorithm calculates the mean value of a local neighborhood and uses that value as the threshold for all pixels in the neighborhood.
- Gaussian adaptive thresholding: This algorithm calculates the weighted mean of a local neighborhood, with weights that are determined by a Gaussian distribution.
- Median adaptive thresholding: This algorithm calculates the median value of a local neighborhood and uses that value as the threshold for all pixels in the neighborhood.

c) Briefly explain edge linking technique.

2

Local processing:

(1) Compute $M(x, y)$ **and** $\alpha(x, y)$ of input image f(x, y)

(2) Form binary image

$$g(x,y) = \begin{cases} 1, & \textbf{if } M(x,y) > T_M \textbf{ AND } \alpha(x,y) \in [A - T_A, A + T_A] \\ 0, & \textbf{otherwise} \end{cases}$$

(3) Scan rows of g and fill (set to 1) all gaps (sets of 0s) in each row that do not exceed a specified length K

(4) Rotate g by and apply step (3). Rotate result back by −θ.

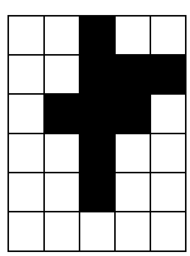
Global Processing using the Hough transform:

(1) Set all cells equal to zero

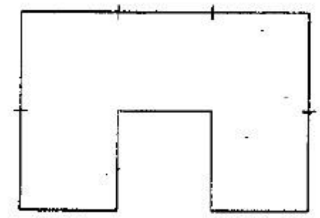
(2) For every (x_k, y_k)

   a) Let a = every subdivision on the a-axis
   b) Calculate $b = -x_k a + y_k$
   c) Round off *b* to the nearest allotted value on the *b*-axis
   d) Increment accumulator cell (*a*, *b*) with 1

4. **a) Calculate the area of the following object using chain code.** **3**



b) **Obtain the shape number for the figure.** **2**



c) **Explain String and Tree representation of any object with example. 1**
https://www.geeksforgeeks.org/binary-tree-string-brackets/

5. a) Describe minimum distance classifier. Build the decision function for two pattern classes w1 and w2. Where two mean vectors $R1^T$= (4.9, 1.4, 7.2) and $R2^T$ = (1.3, 0.2, 2.7). Find the decision boundary also.

b)Explain Artificial neural network. Draw an Artificial Neural Network for classify five pattern classes.

Explain the pattern recognition model in brief.
https://www.xenonstack.com/blog/artificial-neural-network-applications

Autumn 2021

1a) "data is a combination of Information and redundant data" do you agree? justify your answer with proper example
OR write the steps of JPEG algorithm

Ans: Yes, I agree that data is a combination of information and redundant data. Information is data that has been processed and organized in a way that makes it meaningful. Redundant data is data that is duplicated or repeated.

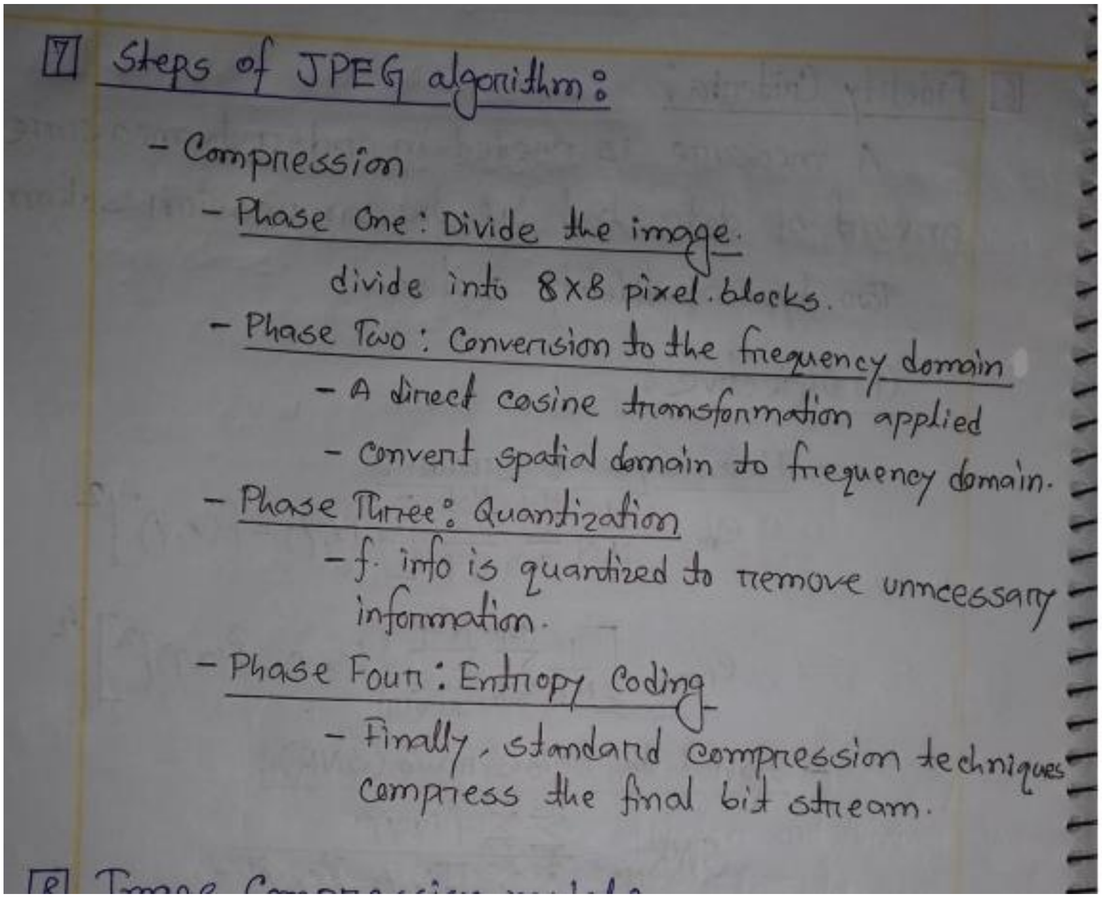
For example, a customer database contains information about customers, such as their names, addresses, and phone numbers. This information is useful for businesses to track customers and to market to them. However, the database may also contain redundant data, such as multiple copies of the same customer's name or address. This redundant data can take up space and make it difficult to update the database.

Here is an example of how data can be both information and redundant data:

A customer's name is information. It is a piece of data that can be used to identify the customer.
A customer's name can also be redundant data. For example, if a customer has purchased from a company multiple times, their name may be entered multiple times into the company's database. This is redundant data because it does not add any new information to the database.
In general, information is more valuable than redundant data. Information can be used to make decisions, to solve problems, and to communicate with others. Redundant data, on the other hand, does not add any new value. It can, however, take up space and make it difficult to update data.

It is important to minimize the amount of redundant data in a database. This can be done by carefully designing the database and by regularly reviewing the data to identify and remove any unnecessary data.

Or:



7 Steps of JPEG algorithm:

- Compression
  - Phase One: Divide the image.
    divide into 8x8 pixel blocks.
  - Phase Two: Conversion to the frequency domain
    - A direct cosine transformation applied
    - convert spatial domain to frequency domain.
  - Phase Three: Quantization
    - f. info is quantized to remove unnecessary information.
  - Phase Four: Entropy Coding
    - Finally, standard compression techniques compress the final bit stream.

1b) **considered the text “Dept of CSE”**

A. what is the probability of occurrence of each character

B. using these probabilities construct the Huffman coding table for transmitting this text

C. write the encoded sequence

D. from encoded sequence do decoding

1c) A source contains four symbols(i,i,u,c) with corresponding probability of 0.3 0.2 0.4 and 0.1 respectively construct arithmetic coding to encode and decode a specific word

Or,

Encode the following image using LZW coding

126 75 36 75

126 75 36 75

126 75 36 75

126 75 36 75

2a) the shape of structuring element has significant impact of every morphological operation. explain the statement with necessary examples

Ans:

Sure. The shape of the structuring element is one of the most important factors that determines the result of a morphological operation. Here are some examples:

Erosion: Erosion is a morphological operation that erodes away the boundaries of objects in an image. The shape of the structuring element determines how much of the object boundary is eroded away. For example, a structuring element with a circular shape will erode away a circular region from the object boundary, while a structuring element with a rectangular shape will erode away a rectangular region from the object boundary.

Dilation: Dilation is a morphological operation that expands the boundaries of objects in an image. The shape of the structuring element determines how much the object boundary is expanded. For example, a structuring element with a circular shape will expand the object boundary by a circular region, while a structuring element with a rectangular shape will expand the object boundary by a rectangular region.

Opening: Opening is a morphological operation that erodes an image and then dilates it. The shape of the structuring element determines how much of the image is eroded and how much it is dilated. For example, an opening operation with a circular structuring element will erode away small holes in an image and then dilate the remaining object boundaries.

Closing: Closing is a morphological operation that dilates an image and then erodes it. The shape of the structuring element determines how much the image is dilated and how much it is eroded. For example, a closing operation with a circular structuring element will fill in small holes in an image and then erode the remaining object boundaries.

In general, the shape of the structuring element determines the type of morphological operation that is performed. For example, a structuring element with a circular shape is used for erosion and dilation operations, while a structuring element with a rectangular shape is used for opening and closing operations.

Here are some additional examples of how the shape of the structuring element can impact the result of a morphological operation:

Morphological thinning: A morphological thinning operation is used to thin out objects in an image. This can be useful for removing noise from an image or for extracting the skeleton of an object. The shape of the structuring element determines how much of the object is removed during the thinning operation. For example, a structuring element with a narrow, vertical shape will remove more of the object's vertical edges than a structuring element with a wider, horizontal shape.

Morphological widening: A morphological widening operation is used to widen objects in an image. This can be useful for connecting disconnected components of an object or for filling in small holes in an object. The shape of the structuring element determines how much of the object is widened during the widening operation. For example, a structuring element with a wide, horizontal shape will widen the object more than a structuring element with a narrow, vertical shape.

2b) Explain hit or miss transformation in brief with necessary example

Ans:

The hit or miss transformation is a morphological operation used to detect specific patterns or configurations in a binary image. It involves the use of two structuring elements: one for the foreground (object) and one for the background (complement of the object).

The operation aims to identify pixels in the image that match both the foreground and background structuring elements simultaneously. These pixels represent locations where the desired pattern or configuration exists.

Here's a step-by-step explanation of the hit or miss transformation:

Define the foreground structuring element (FSE) to represent the desired pattern or configuration.

Define the background structuring element (BSE) to represent the complement of the desired pattern or configuration.

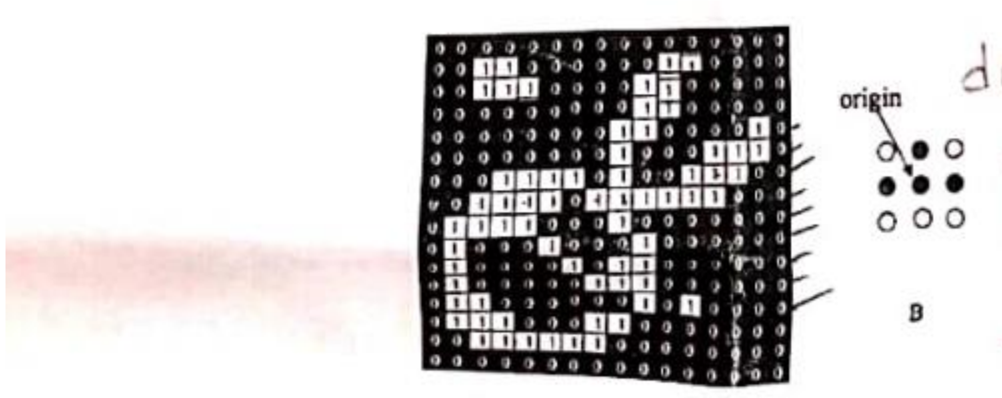
Perform erosion on the input binary image using the FSE. This operation removes pixels that do not match the FSE.

Perform erosion on the input binary image using the complement of the BSE. This operation removes pixels that do not match the complement of the BSE.

Take the intersection of the results obtained from the two erosions. This means considering only the pixels that were preserved in both erosion operations.

The resulting image will contain white pixels at locations where the desired pattern or configuration matches exactly and there is no interference from the background.

2c) Binary image X and Structuring element B are given below



X

Calculate $Y1 = X \circ B$ Where $\circ$ denotes morphological opening operation..

2d) show the histogram of a 3-bit image of the following table

gray level	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
number of pixels	**5**	**4**	**3**	**2**	**1**	**6**	**3**	**2**

3a) "segmentation is one of the most important steps in object recognition" describe in brief

Segmentation is indeed one of the most important steps in object recognition. Object recognition involves identifying and categorizing objects or regions of interest within an image or scene. Segmentation plays a crucial role in this process by dividing the image into meaningful and coherent regions, enabling the recognition algorithms to focus on individual objects.

Here are some key reasons why segmentation is crucial in object recognition:

1. Localization: Segmentation helps in localizing objects within an image by separating them from the background or other surrounding elements. By identifying the boundaries of objects through segmentation, the recognition algorithms can precisely locate and extract the regions of interest for further analysis.

2. Feature Extraction: Once objects are segmented, various features can be extracted from each segmented region. These features can include color histograms, texture descriptors, shape characteristics, or other relevant attributes. Segmentation provides a well-defined boundary for extracting these features specifically from the regions of interest, leading to more accurate and informative representations for object recognition.

3. Background Suppression: In many cases, the background or surrounding context in an image can contain irrelevant or distracting elements that can hinder object recognition. Segmentation helps to suppress the background by isolating the objects of interest. This allows the recognition algorithms to focus on the foreground objects and disregard the irrelevant background information, improving the overall accuracy of object recognition.

4. Object Classification: Segmentation provides individual object regions that can be treated as separate instances for classification. Once objects are segmented, recognition algorithms can analyze the extracted features and compare them to a pre-trained model or database to determine the class or category of each object. Segmentation enables the classification process to be performed at the object level rather than at the pixel level, leading to more robust and accurate object recognition.

5. Scene Understanding: Segmentation contributes to a higher level of scene understanding by providing information about the spatial arrangement and relationships between objects. By segmenting an image into distinct regions, the recognition algorithms can analyze the interactions and context between objects, leading to a more comprehensive understanding of the scene as a whole.

In summary, segmentation is a critical step in object recognition as it enables localization, feature extraction, background suppression, object classification, and scene understanding. It provides the necessary foundation for recognition algorithms to effectively analyze and interpret the objects within an image, leading to accurate and robust object recognition capabilities.

Or

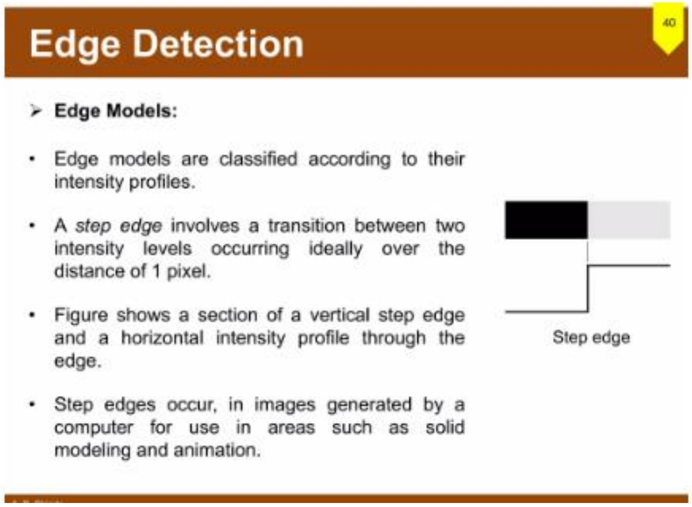
what are edges? how different types of edges can be segmented give necessary example

Answer:

Edges in image processing refer to the boundaries or transitions between different regions or objects within an image. They represent significant changes in pixel intensity, color, or texture and often indicate important features or structures in the image.

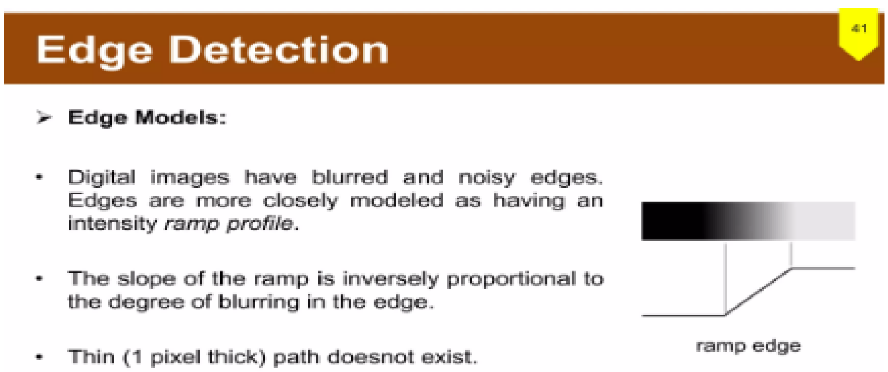
There are several types of edges that can be segmented:

1. Step Edges: Step edges occur when there is a sudden and significant change in intensity or color values across adjacent pixels. They represent sharp transitions and are often observed in images with well-defined boundaries between objects or regions.



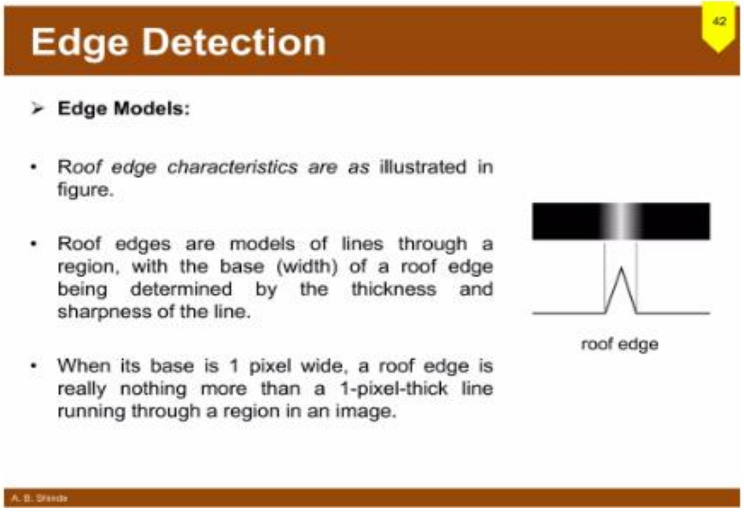
Example: The edge between the sky and the ground in an image of a landscape.

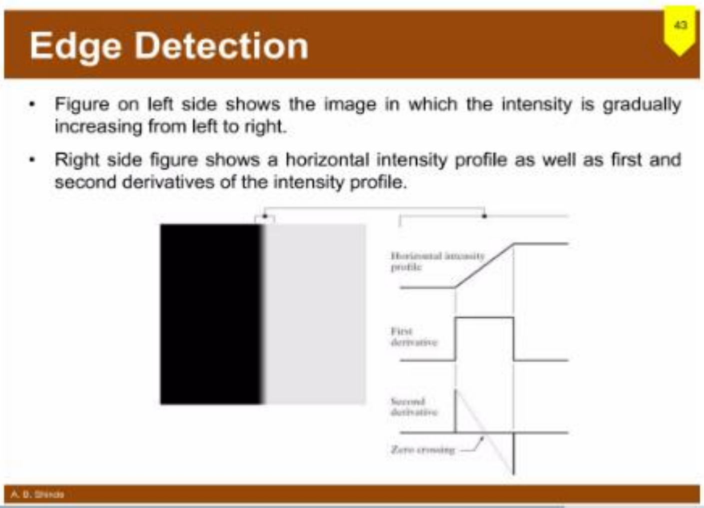
2. Ramp Edges: Ramp edges are characterized by a gradual change in intensity or color values over a region. They represent smoother transitions and are commonly found in textured or gradient regions.



Example: The transition from light to dark in a grayscale gradient image.

3. Roof Edges: Roof edges are formed when there is an intensity or color transition followed by a plateau or constant region. They represent a sudden change followed by a flat portion.





Example: The edge between a person's face and their hair in a portrait image.

4. Ridge Edges: Ridge edges occur when there is a bright or dark line along the center of a region. They represent linear structures or ridges in the image.

Example: The edge along the center of a road in an aerial image.

To segment edges in an image, various edge detection algorithms can be used. These algorithms aim to highlight or mark the pixels that correspond to edge locations. Some commonly used edge detection methods include:

- Gradient-based Methods: These methods calculate the gradient magnitude or the first derivative of the image to detect areas of rapid intensity changes.

- Laplacian of Gaussian (LoG): The LoG method convolves the image with a Gaussian filter and then applies the Laplacian operator to detect regions of intensity transitions.

- Hough Transform: The Hough transform is often used to detect and segment line or curve edges by converting the edge points into parameter space.

These techniques, along with other advanced edge detection methods, help to identify and segment different types of edges in an image, enabling further analysis and processing tasks such as object recognition, shape analysis, or image-based measurements.

3b) why is linking is important in many image processing operations? explain local edge linking algorithm?

Ans:

Linking is important in many image processing operations, particularly in the context of segmentation and object recognition, because it helps connect related pixels or regions to form coherent and meaningful objects or regions. It plays a crucial role in overcoming the challenges of image noise, discontinuities, and occlusions.

Here are some reasons why linking is important:

1. Completeness: Linking helps connect fragmented or disjointed regions or contours, allowing for the formation of complete objects or regions of interest. This is crucial in tasks like object recognition, where complete and coherent objects need to be identified.

2. Continuity: Linking establishes the connectivity between neighboring pixels or regions with similar properties, such as intensity or color, to create continuous boundaries or edges. This helps in accurately delineating the boundaries of objects or regions.

3. Contextual Information: Linking considers the spatial relationships between neighboring pixels or regions, taking into account their proximity and similarity. By linking related elements, it incorporates contextual information and improves the accuracy of segmentation.

4. Noise Reduction: Linking helps filter out noise or small isolated regions by considering their connectivity to larger, more significant regions. This ensures that only meaningful structures are retained and irrelevant noise is eliminated.

5. Occlusion Handling: Linking is crucial in handling occlusions, where one object partially obscures another. By linking relevant edges or regions, it becomes possible to separate and distinguish occluded objects, allowing for accurate recognition and segmentation.

Overall, linking is essential in image processing operations as it enables the formation of complete objects, ensures the continuity of boundaries, incorporates contextual information, reduces noise,

and handles occlusions. It plays a vital role in achieving accurate and meaningful segmentation, object recognition, and image analysis.
Another answer:

Linking is important in many image processing operations because it allows us to connect together edges that are close together in space. This can be used for a variety of tasks, such as:

Edge detection: Linking can be used to improve the accuracy of edge detection algorithms by connecting together weak edges that are close together.
Object detection: Linking can be used to identify objects in an image by connecting together edges that form the boundaries of the objects.
Segmentation: Linking can be used to segment an image into different regions by connecting together edges that form the boundaries of the regions.
Linking is important in many image processing operations because it allows us to connect together edges that are close together in space. This can be used for a variety of tasks, such as:

Edge detection: Linking can be used to improve the accuracy of edge detection algorithms by connecting together weak edges that are close together.
Object detection: Linking can be used to identify objects in an image by connecting together edges that form the boundaries of the objects.
Segmentation: Linking can be used to segment an image into different regions by connecting together edges that form the boundaries of the regions.
Local edge linking is a type of linking algorithm that is used to connect edges that are close together in an image. This can be useful for tasks such as object detection and image segmentation. The algorithm works by first detecting all of the edges in an image. Then, it uses a linking rule to connect the edges that are close together. The linking rule can be based on a variety of factors, such as the brightness of the edges, the direction of the edges, and the distance between the edges.

Here are the steps involved in local edge linking:

Detect edges: The first step is to detect all of the edges in an image. This can be done using a variety of edge detection algorithms, such as the Canny edge detector.

Find connected components: Once the edges have been detected, the next step is to find all of the connected components in the image. A connected component is a group of pixels that are all connected by edges.

Link connected components: The final step is to link the connected components together. This can be done using a variety of linking rules, such as the 4-connectivity rule or the 8-connectivity rule.
Local edge linking is a simple and efficient way to connect edges in an image. It is a commonly used algorithm in a variety of image processing applications.
Here are some of the advantages of local edge linking:

- It is simple and efficient.
- It can be used to connect edges that are close together.
- It can be used for a variety of tasks, such as object detection and image segmentation.

Here are some of the disadvantages of local edge linking:

- It can be sensitive to noise.
- It can produce false positives.
- It can be difficult to control the linking process.

3c) what is region? how region can be segmented give any algorithm for region segmentation

Ans:

A region is a contiguous area of pixels in an image that has similar properties, such as color, texture, or intensity. Region segmentation is the process of dividing an image into its constituent regions. This can be done using a variety of algorithms, each with its own strengths and weaknesses.

One common algorithm for region segmentation is region growing

REGION GROWING

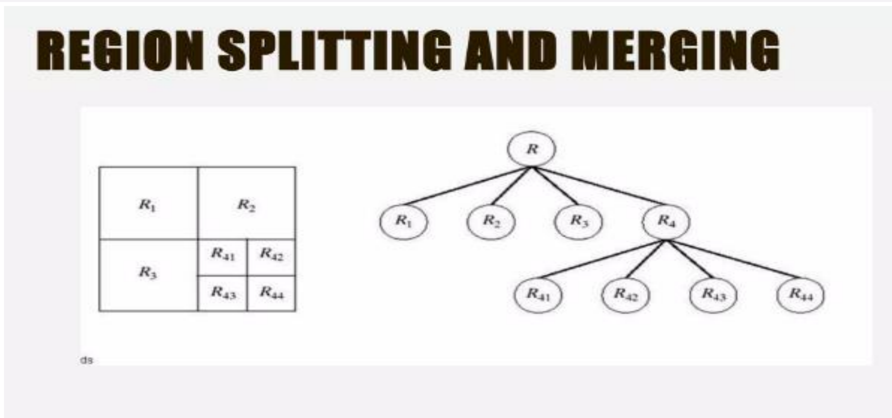
- Region growing is a procedure that groups pixels or sub regions into larger regions.
- The simplest of these approaches is pixel aggregation, which starts with a set of seed points and from these grows regions by appending to each seed points those neighboring pixels that have similar properties (such as gray level, texture, color, shape).
- Region growing based techniques are better than the edge-based techniques in noisy images where edges are difficult to detect

THE ADVANTAGES OF REGION GROWING

- Region growing methods can correctly separate the regions that the same properties we define.
- Region growing methods can provide the original images which have clear edges with good segmentation results.
- The concept is simple. We only need a small number of seed points to represent the property we want, then grow the region.

REGION SPLITTING

- Region growing starts from a set of seed points.
- An alternative is to start with the whole image as a single region and subdivide the regions that do not satisfy a condition of homogeneity.



4a)

i) Represent the boundary in figure 1 within 8 directional chain code

ii) What does rotational invariant mean in this case? make the chain code in a rotational invariant?

Figure 1

Answer

In mathematics, a function defined on an inner product space is said to have rotational invariance if its value does not change when arbitrary rotations are applied to its argument.

A rotation invariant code sequence also can be obtained by using the first difference of the chain code; the difference between two consecutive digits is defined as the number of directions between them, which is taken to be positive when counterclockwise.

https://ojskrede.github.io/inf4300/notes/week_04/#:~:text=Rotation%20invariant%20chain%20code.&text=8%2Dconnected%20case-,c8%3D(7%2C7%2C6%2C6%2C,1%2C5%2C7).

4b) what is LBP ? the steps to calculate LBP give necessary example ?

Local Binary Patterns (LBP) is a texture analysis technique that is used to describe the spatial distribution of intensity values in an image. LBP is a simple and computationally efficient method that can be used to extract features from images for a variety of tasks, such as object detection, image classification, and face recognition.

The steps to calculate LBP are as follows:

1. Select a pixel in the image.
2. Define a circular neighborhood around the selected pixel with a radius and a number of neighboring pixels.
3. Compare the intensity of each neighboring pixel to the intensity of the center pixel.
4. If the intensity of the neighboring pixel is greater than or equal to the intensity of the center pixel, assign it a value of 1. Otherwise, assign it a value of 0.
5. Convert the binary values of the neighboring pixels to a decimal value.
6. Repeat steps 3-5 for all pixels in the image.
7. Create a histogram of the decimal values to represent the texture of the image.

Generate the LBP code: The final step is to generate the LBP code. This is done by converting the binary pattern into a decimal number.

Here is an example of how LBP is calculated:

Consider a 5x5 grayscale image with the following pixel values:

```
50 50 50 40 30
50 50 50 40 30
50 50 50 40 30
60 60 60 70 80
90 90 90 70 80
```

Suppose we choose a circular neighborhood with a radius of 1 and 8 neighboring pixels. We start with the pixel at (2,2), which has a value of 50. The circular neighborhood around this pixel includes the following neighboring pixels:

```
50 50 40
50  0 40
50 40 30
```

We compare the intensity of each neighboring pixel to the center pixel (50), and assign a binary value of 1 or 0 accordingly. The resulting binary pattern is:

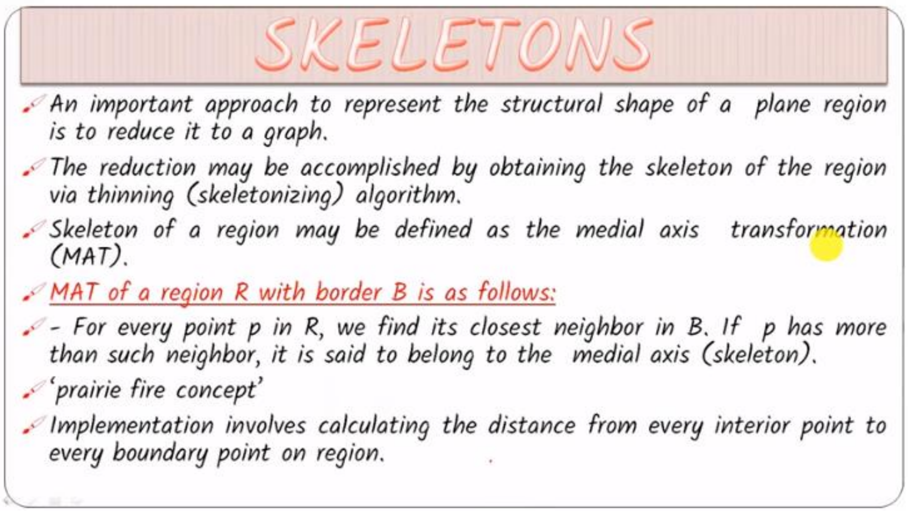
```
0 0 0
0 1 1
0 1 0
```

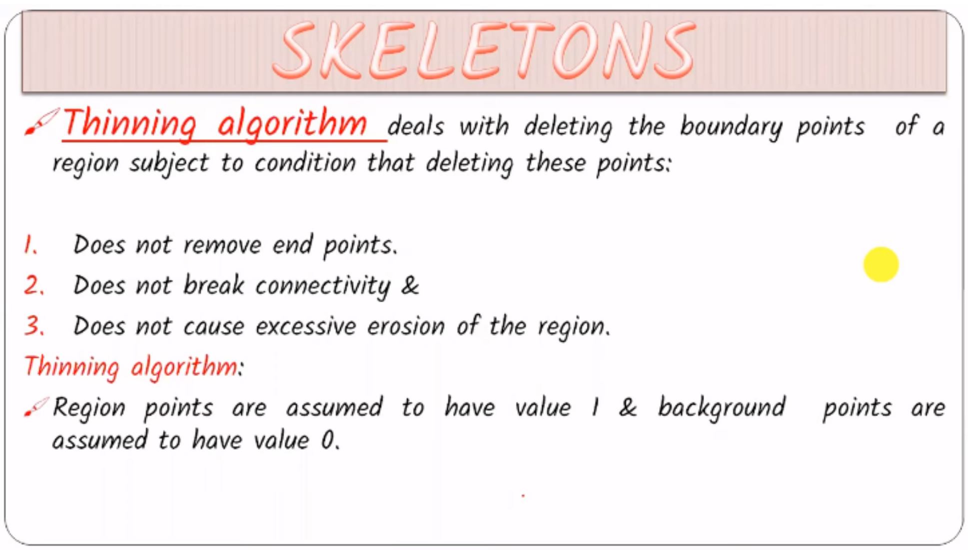
We convert this binary pattern to a decimal value, which is 2^1 + 2^2 + 2^5 = 6. We repeat this process for all pixels in the image and create a histogram of the resulting decimal values. The

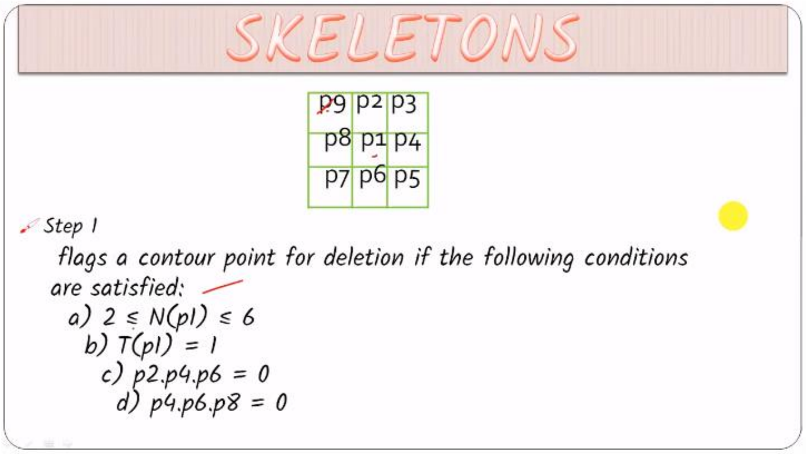
histogram represents the texture of the image, and can be used for various applications such as texture classification or object recognition.
LBP is a powerful texture analysis technique that can be used to extract features from images for a variety of tasks. It is a simple and computationally efficient method that is easy to implement. LBP has been shown to be effective for a variety of applications, such as object detection, image classification, and face recognition.

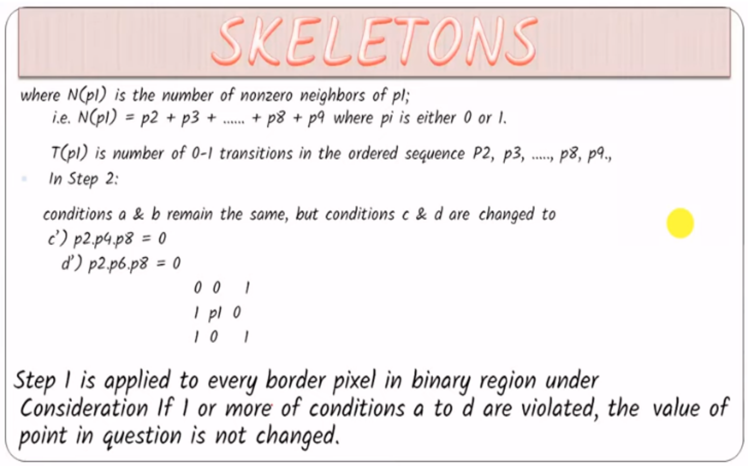
OR,

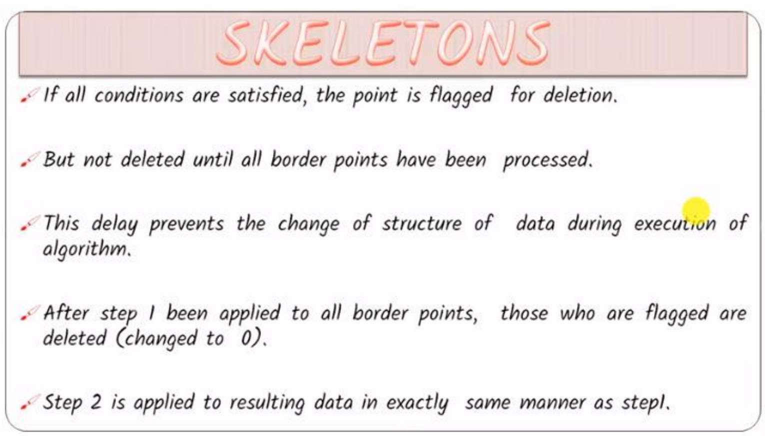
Describe your method of computing skeleton of the images?











what kind of structural elements will you be using? what happens when you change the size of structuring element?

Structural elements are used in morphological image processing operations to analyze and modify the shapes and structures of objects within an image. They are small, predefined patterns or shapes represented as binary matrices.

When you change the size of a structuring element, several things can happen:

1. Effect on Object Size: In operations like erosion and dilation, the size of the structuring element determines the extent to which objects are eroded or dilated. Larger structuring elements result in more significant changes to object size, while smaller ones produce more localized effects. Increasing the size can cause objects to expand, while decreasing the size can cause them to shrink.

2. Preservation of Object Details: A larger structuring element may lead to the loss of finer details and smaller structures within objects. It tends to smooth out the object boundaries and merge nearby regions. Conversely, a smaller structuring element preserves finer details and allows for better differentiation between objects or regions.

3. Computational Impact: The size of the structuring element affects the computational complexity of the morphological operation. Larger structuring elements require more computational resources as they involve processing a larger number of pixels. Therefore, using larger structuring elements may result in slower processing times.

Choosing the appropriate size of the structuring element depends on the specific task, the characteristics of the objects or structures you are analyzing, and the desired outcome. It often involves a trade-off between preserving object details and achieving the desired modification. Experimentation and adjusting the size according to the specific requirements of the task are typically necessary to obtain optimal results.

It's important to note that the effects of changing the size of a structuring element are not universally predictable and may vary depending on the image content and the specific morphological operation being applied. Therefore, it is often necessary to experiment and fine-tune the size of the structuring element to achieve the desired image processing results.

4c) explain string and tree representation of any object with example

String and tree representations are two common ways to represent objects in computer science.
A string representation of an object is a sequence of characters that encode the object's properties and attributes in a human-readable format. For example, consider a simple object representing a person:

```
{
  "name": "John Smith",
  "age": 35,
  "gender": "male",
  "address": {
    "street": "123 Main St",
    "city": "Anytown",
    "state": "CA",
    "zip": "12345"
  }
```

}
This object could be represented as a string using a serialization format such as JSON or XML:

```
{
  "name": "John Smith",
  "age": 35,
  "gender": "male",
  "address": {
    "street": "123 Main St",
    "city": "Anytown",
    "state": "CA",
    "zip": "12345"
  }
}
```

This string representation can be easily transmitted and stored in a variety of formats, but it is not well-suited for complex hierarchical structures or recursive data types.

A tree representation, on the other hand, is a graphical representation of an object's structure using a set of interconnected nodes. Each node represents a property or attribute of the object, and the edges between nodes indicate the relationships between them. For example, the same person object could be represented as a tree:

```
Person
├── Name: John Smith
├── Age: 35
├── Gender: Male
└── Address
  ├── Street: 123 Main St
  ├── City: Anytown
  ├── State: CA
  └── Zip: 12345
```

This tree representation clearly shows the hierarchical structure of the object and can be easily traversed and manipulated using standard algorithms and data structures. However, it may not be as intuitive or readable as a string representation for simple or flat objects.

In summary, string representations are useful for simple objects with flat structures, while tree representations are better suited for complex objects with hierarchical or recursive structures.

5a) Build the decision functions and decision boundary for two pattern class w1 and w2 where two mean by vectors m1 = [6.1, 1.6, 6] and m2=[1.2,0.2, 2.0] find the class of an unknown pattern X =[3.1, 0.4, 2.8]

5b) how matching by correlation works on pattern recognition? explain with necessary example

OR,

explain syntactical pattern recognition write the steps of recognize the string in brief

5c) write the applications of pattern recognition in autonomous car

Ans:

ere are some of the applications of pattern recognition in autonomous cars:

Object detection: Pattern recognition is used to detect objects in the environment, such as other cars, pedestrians, and cyclists. This information is used to plan the car's path and avoid collisions.
Traffic sign recognition: Pattern recognition is used to recognize traffic signs, such as stop signs, yield signs, and speed limit signs. This information is used to follow the rules of the road and avoid accidents.
Lane keeping: Pattern recognition is used to keep the car in its lane. This is done by detecting the edges of the lane and steering the car accordingly.
Pedestrian detection: Pattern recognition is used to detect pedestrians and cyclists. This information is used to slow down or stop the car to avoid a collision.
Traffic light recognition: Pattern recognition is used to recognize traffic lights. This information is used to stop at red lights and proceed at green lights.

Spring 2019

Group A

1a) A 24 bit image with a 1024 x 840 pixels is compressed by a 16 bit lossy truncation algorithm. calculate the compression ratio and data redundancy of the image.

Ans:

To calculate the compression ratio, we need to first determine the size of the original uncompressed image and compare it to the size of the compressed image.

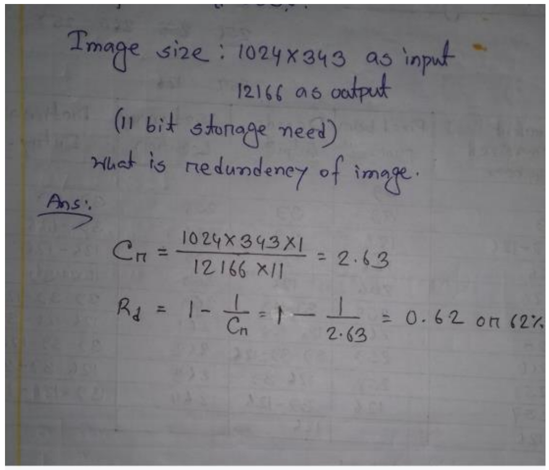


Image size : 1024×343 as input
12166 as output
(11 bit storage need)
What is redundancy of image.

Ans:

$$C_r = \frac{1024 \times 343 \times 1}{12166 \times 11} = 2.63$$

$$R_d = 1 - \frac{1}{C_r} = 1 - \frac{1}{2.63} = 0.62 \text{ or } 62\%$$

Size of the original image:
Each pixel in the original image is represented by 24 bits, or 3 bytes. Therefore, the total size of the image can be calculated as follows:

1024 pixels × 840 pixels × 3 bytes/pixel = 2,764,160 bytes

Size of the compressed image:
Each pixel in the compressed image is represented by 16 bits, or 2 bytes. Therefore, the total size of the compressed image can be calculated as follows:

1024 pixels × 840 pixels × 2 bytes/pixel = 1,771,520 bytes

Compression ratio:
The compression ratio is the ratio of the size of the original image to the size of the compressed image:

Compression ratio = Size of original image / Size of compressed image
Compression ratio = 2,764,160 bytes / 1,771,520 bytes
Compression ratio ≈ 1.56

Data redundancy:
The data redundancy is the difference between the size of the original image and the size of the compressed image, expressed as a percentage of the size of the original image:

Data redundancy = (Size of original image - Size of compressed image) / Size of original image × 100%
Data redundancy = (2,764,160 bytes - 1,771,520 bytes) / 2,764,160 bytes × 100%
Data redundancy ≈ 35.9%

Therefore, the compression ratio is approximately 1.56 and the data redundancy is approximately 35.9%.

1b) "data and information are not synonymous terms" justify the statement with example . Estimate the Huffman coding for the word 'CSEDIP '

Ans:
Data and information are not synonymous terms because while data refers to raw and unorganized facts, information refers to processed and organized data that has meaning and relevance to the user. Data is a collection of raw and unprocessed facts, figures, and symbols that have no meaning on their own. In contrast, information is data that has been processed, organized, and given context so that it can be understood and used by people.

For example, consider a database of customer orders. The database contains data such as the customer's name, order number, date of purchase, and order details. While this data is valuable to the business, it is not information until it is organized, analyzed, and presented in a meaningful way. Once the data is processed and organized, it becomes information that can be used to make informed decisions about the business.

Huffman coding is a type of variable-length data compression algorithm that assigns shorter codes to more frequently occurring characters or symbols. To estimate the Huffman coding for the word 'CSEDIP', we would first determine the frequency of each character in the word. Counting the frequency of each character, we get:

- C: 1
- S: 1
- E: 1

- D: 1
- I: 1
- P: 1

Since each character appears only once, any Huffman code assigned to each character would be the same length, such as 3 bits. Therefore, the Huffman coding for the word 'CSEDIP' would be:

- C: 000
- S: 001
- E: 010
- D: 011
- I: 100
- P: 101

In this case, the Huffman coding provides a fixed length code of 3 bits for each character, which does not provide any compression benefit. In practice, Huffman coding is most effective when there is a significant difference in the frequency of characters in the data being compressed, allowing shorter codes to be assigned to more frequently occurring characters, resulting in a more efficient compression.

1c) write the advantage of LZW coding over arithmetic coding .

Ans:

LZW (Lempel-Ziv-Welch) coding and arithmetic coding are two popular data compression algorithms. While both techniques are effective at compressing data, they have different advantages and disadvantages.

One of the advantages of LZW coding over arithmetic coding is its simplicity. LZW coding is a relatively simple and easy-to-implement algorithm that requires minimal computational resources. In contrast, arithmetic coding is a more complex algorithm that requires a lot of computational resources and is difficult to implement efficiently.

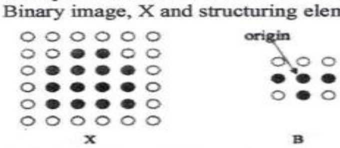
Another advantage of LZW coding is that it is a lossless compression technique, meaning that it can compress data without losing any information. In contrast, arithmetic coding is a lossy compression technique, meaning that it may discard some information during compression, which can lead to a loss of quality in the compressed data.

Additionally, LZW coding is particularly effective at compressing data with repeating patterns, as it can recognize and encode these patterns efficiently. This makes it well-suited for compressing text data and other types of data that have repeating patterns. In contrast, arithmetic coding is better suited for compressing data with a more random distribution of symbols, as it can assign shorter codes to more frequently occurring symbols.

Overall, while arithmetic coding has its advantages, such as providing better compression ratios for certain types of data, LZW coding's simplicity, lossless compression, and effectiveness at compressing data with repeating patterns make it a popular choice for many applications.

2 a) Explain the concept of Morphological Opening and closing. Explain with an example that $A\Theta B \neq B\Theta A$ where Θ denotes the morphological erosion operator. 4

b) Binary image, X and structuring element, B, are given as follows: 3



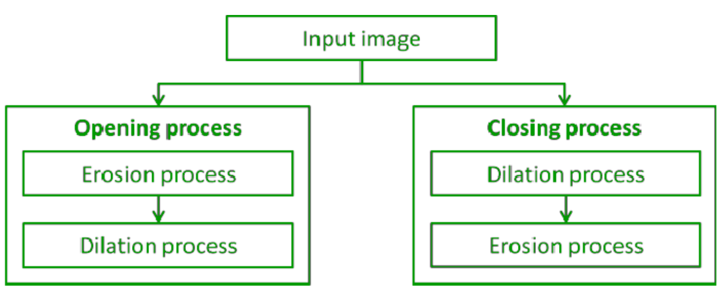
Calculate Y1 = XΘB, where Θ denotes the morphological erosion operator and Y2= X$\oplus$B where $\oplus$ denotes the morphological dilation operator.

c) Write the short note about: i) Hit-or-Miss Transformation, ii) Skeleton, and ii) Boundary filling. 3

2a) Explain the concept of Morphological Opening and closing. Explain with an example that A ⊖ B ≠ B ⊖ A where ⊖ denotes the morphological erosion operator.

Answer:

Opening and **Closing** are dual operations used in Digital Image Processing for restoring an eroded image. Opening is generally used to restore or recover the original image to the maximum possible extent. Closing is generally used to smoother the contour of the distorted image and fuse back the narrow breaks and long thin gulfs. Closing is also used for getting rid of the small holes of the obtained image. The combination of **Opening** and **Closing** is generally used to clean up artifacts in the segmented image before using the image for digital analysis.



Some of the differences between Opening and Closing are:

S.No.	Opening	Closing
1.	Opening is a process in which first erosion operation is performed and then dilation operation is performed.	Closing is a process in which first dilation operation is performed and then erosion operation is performed.
2.	Opening operation performed on X & Y is the union of all translations of Y that fit entirely within X.	Closing operation performed on X & Y is the complement of the union of all translations of Y that do not fit entirely within X.
3.	It eliminates the thin protrusions of the obtained image.	It eliminates the small holes from the obtained image.
4.	Opening operation performed on X & Y is represented by (A^oB).	Closing operation performed on X & Y is represented by ($A{\cdot}B$)
5.	Opening is used for smoothening of contour and fusing of narrow breaks.	Closing is used for removing internal noise of the obtained image

Opening is denoted by:

$$A \circ B = (A \ominus B) \oplus B$$

Properties of Opening are:

1. X^oY is a subset (subimage of X)
2. If X is a subset of Z then X^oY is a subset of Z^oY
3. $(X^oY)^oY = X^oY$

Closing is denoted by:

$$A \bullet B = (A \oplus B) \ominus B$$

Properties of Closing are:

1. X is a subset subimage of $X{\cdot}Y$
2. $(X{\cdot}Y){\cdot}Y = X{\cdot}Y$

Morphological opening and closing are two operations in mathematical morphology used for image processing and analysis. They involve the combination of morphological erosion and dilation operations with a structuring element.

Morphological Opening:
Morphological opening is performed by applying morphological erosion followed by morphological dilation to an image. It is commonly used for removing small foreground objects and noise while preserving the larger structures.
The opening operation is defined as follows:
Opening(A) = Dilation(Erosion(A, B), B)

In this equation:

A represents the input image or binary object.
B denotes the structuring element, which defines the shape and size of the neighborhood used for erosion and dilation operations.
Erosion(A, B) is the morphological erosion of A by B, which shrinks the foreground regions of A.
Dilation(Erosion(A, B), B) is the morphological dilation of the eroded image by B, which expands the remaining foreground regions to their original size.
Morphological Closing:
Morphological closing is performed by applying morphological dilation followed by morphological erosion. It is commonly used for filling in gaps or holes in an image, thereby smoothing object boundaries and connecting nearby structures.
The closing operation is defined as follows:
Closing(A) = Erosion(Dilation(A, B), B)

In this equation:

A represents the input image or binary object.
B denotes the structuring element, which defines the shape and size of the neighborhood used for dilation and erosion operations.
Dilation(A, B) is the morphological dilation of A by B, which expands the foreground regions of A.
Erosion(Dilation(A, B), B) is the morphological erosion of the dilated image by B, which restores the original size of the objects while filling in gaps.
Now, let's consider an example to demonstrate that $A \ominus B \neq B \ominus A$, where $\ominus$ represents the morphological erosion operator:

Example:
Suppose we have a binary image A and a structuring element B defined as follows:

A =
0 0 0 0 0
0 1 1 1 0
0 1 1 1 0
0 1 1 1 0
0 0 0 0 0

B =
1 1 1
1 1 1
1 1 1

Performing erosion on A with B ($A \ominus B$):
$(A \ominus B)$ =
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

0 0 0 0 0

Performing erosion on B with A (B ⊖ A):
(B ⊖ A) =
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

As we can see from the example, A ⊖ B and B ⊖ A both result in an all-zero image, indicating that they are equal. Therefore, the example contradicts the statement A ⊖ B ≠ B ⊖ A. In general, the morphological erosion operation is not commutative, meaning the order of the input image and the structuring element can affect the result.

2b)Answer:

2c)
The Hit-or-Miss transformation is a fundamental operation in digital image processing, specifically in the field of morphological image processing. It is used for pattern matching, shape detection, and image analysis tasks in binary images.

The goal of the Hit-or-Miss transformation is to identify pixels or regions in an image that match a specific pattern. This pattern consists of foreground (white) pixels, background (black) pixels, and a third category of pixels that may or may not be present.

The operation is performed using two structuring elements: J, representing the foreground, and K, representing the background. These elements define the pattern to be matched. J contains the pixels that must be present (foreground) in the image, while K specifies the pixels that must be absent (background) in the image.

The Hit-or-Miss transformation

9.5.1 Boundary Extraction

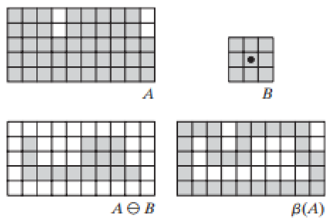
The boundary of a set A, denoted by $\beta(A)$, can be obtained by first eroding A by B and then performing the set difference between A and its erosion. That is,

$$\beta(A) = A - (A \ominus B) \tag{9.5-1}$$

where B is a suitable structuring element.

Figure 9.13 illustrates the mechanics of boundary extraction. It shows a simple binary object, a structuring element B, and the result of using Eq. (9.5-1). Although the structuring element in Fig. 9.13(b) is among the most frequently used, it is by no means unique. For example, using a 5×5 structuring element of 1s would result in a boundary between 2 and 3 pixels thick.

From this point on, we do not show border padding explicitly.



a b
c d

FIGURE 9.13 (a) Set A. (b) Structuring element B. (c) A eroded by B. (d) Boundary, given by the set difference between A and its erosion.

Skeleton:

Skeletonization is a morphological operation used in digital image processing to extract the thin, elongated representation of an object while preserving its connectivity and topology. It reduces the binary image to its skeleton or medial axis, which represents the centerlines of the objects.

Skeletonization is commonly employed in various applications, such as shape analysis, pattern recognition, and feature extraction. The process involves iteratively thinning the binary image until only the skeleton remains.

There are different algorithms for skeletonization, but the basic idea is to repeatedly apply erosion operations to the image until the desired skeleton is obtained. The erosion operation removes the boundary pixels of objects while preserving their connectivity.

During each iteration of the skeletonization process, the image is eroded using a specific structuring element, usually a 3x3 or 5x5 neighborhood. The structuring element is applied to each pixel of the image, and if the pattern matches a specific condition, the pixel is removed. The process continues until no further changes occur, indicating the convergence to the skeleton.

The specific conditions for pixel removal depend on the chosen algorithm. Generally, pixels are removed if they do not affect the connectivity or topology of the objects in the image. This iterative erosion process thins the objects, preserving their essential shape characteristics.

The resulting skeleton represents the medial axis or centerlines of the objects in the image. It provides a simplified representation that retains important information about the shape and structure of the objects while reducing the complexity of the image.

Skeleton morphological operations have proven to be useful in various applications, such as character recognition, shape matching, object tracking, and analysis of vascular structures in medical imaging.

3a)Explain Image Compression model

An image compression system is composed of two distinct functional components: an encoder and a decoder. The encoder performs compression, and the decoder performs the complementary operation of decompression.
Input image is fed into the encoder, which creates a compressed representation of the input.This representation is stored for later use, or transmitted for storage and use at a remote location. When the compressed representation is presented to its complementary decoder, a reconstructed output image is generated. In still-image applications, the encoded input and decoder output are and respectively

8.1 ■ Fundamentals 537

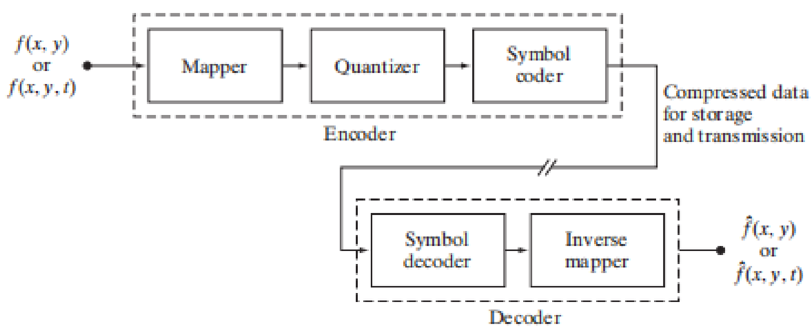


FIGURE 8.5 Functional block diagram of a general image compression system.

The encoding or compression process

In the first stage of the encoding process, a mapper transforms into a (usually nonvisual) format designed to reduce spatial and temporal redundancy

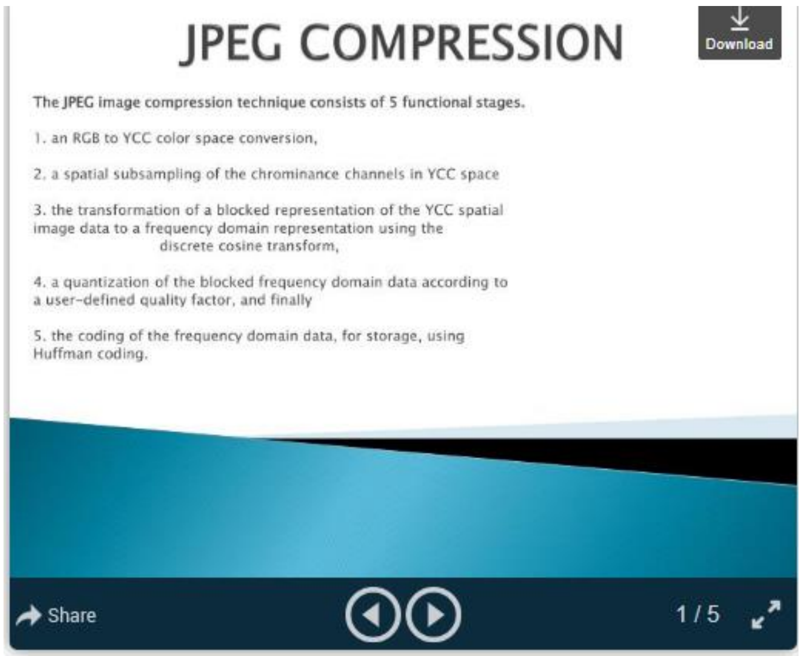
The quantizer in Fig. 8.5 reduces the accuracy of the mapper's output in accordance with a pre-established fidelity criterion. The goal is to keep irrelevant
information out of the compressed representation

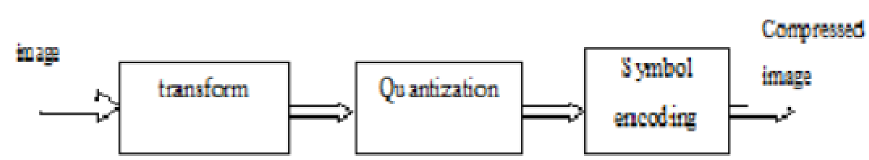
The decoding or decompression process

The decoder of Fig. 8.5 contains only two components: a symbol decoder and an inverse mapper. They perform, in reverse order, the inverse operations of the encoder's symbol encoder and mapper. Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general decoder mode

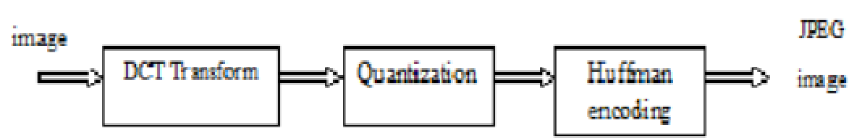
In the third and final stage of the encoding process, the symbol coder of Fig. 8.5 generates a fixed- or variable-length code to represent the quantizer output and
maps the output in accordance with the code.

3b)describe the steps of JPEG Image Compression





General Steps in Image Compression



Process Flow of JPEG image compression

3c)Describe bit plan coding? what are the differences between lossy and lossless Image Compression.

The technique, called bit-plane coding, is based on the concept of decomposing a multilevel (monochrome or color) image into a series of binary images (see Section 3.2.4) and compressing each binary image via one of several well-known binary compression methods

Table 1. Comparison between Lossy and Lossless compression

FACTORS	DATA COMPRESSION	
	LOSSLESS COMPRESSION	LOSSY COMPRESSION
Definition	Lossless compression is a class of data compression algorithms that allow the original data to be perfectly reconstructed from the compressed data[7].	Lossy compression is the class of data encoding methods that uses inexact approximations to represent the content. These techniques are used to reduce the data size for storage, handling, and transmitting content[8].
Algorithm	RLW, LZW, Arithmetic encoding, Huffman coding, Shannon-Fano coding	Transform coding, DCT,DWT, Fractal compression, Rectangle Segmentation and Sparse Matrix Storage (RSSMS).
Uses	Text or programs, images and sound	Images, audio and video.
Images	RAW, BMP, and PNG are all Lossless formats.	JPEG and GUI are lossy image formats.
Audio	WAV, FLAC, and ALAC are all Lossless formats.	MP3, MP4, and OGG are lossy audio formats.
Video	Few lossless video formats are in common consumer use, they would result in video files taking up a huge amount of space.	Common formats like H-264, MKV, and WMV are all lossy. H-264 can provides smaller files with higher qualities than previous generation of video codec because it has a “smaller” algorithm that’s better at choosing the data to throw out.
Advantages	It maintains quality. Conversion in any other format possible without loss of audio information.	It can make a multimedia file much smaller than its original size.It can reduce file sizes much more than lossless compression.
Disadvantages	It doesn’t reduce the file size as much as lossy compression. Lossless encoding technique cannot achieve high levels of compression.	Conversion to another format only with loss of audio information.It cannot be used in all types of files because it works by removing data. Text and data cannot be compressed because they do not have redundant information.

3d)What is pruning ? why is it so important?

4a)What is segmentation? Describe the necessity of segmentation

Answer:

Image segmentation is a commonly used technique in digital image processing and analysis to partition an image into multiple parts or regions, often based on the characteristics of the pixels in the image. Image segmentation could involve separating foreground from background, or clustering regions of pixels based on similarities in color or shape. For example, a common application of image segmentation in medical imaging is to detect and label pixels in an image or voxels of a 3D volume that represent a tumor in a patient’s brain or other organs.

Segmentation is a crucial step in image processing that involves dividing an image into meaningful and coherent regions or objects. It plays a vital role in various computer vision tasks, such as object recognition, image analysis, and scene understanding. The primary necessity of segmentation arises from the fact that raw images are essentially a collection of pixels, and without segmentation, it would be challenging to extract meaningful information from them.

Here are some key reasons why segmentation is necessary in image processing:

1. Object Recognition: Segmentation helps in identifying and isolating objects within an image. By separating the foreground objects from the background, it becomes easier to recognize and classify different objects present in an image. Object recognition is crucial for various applications, including autonomous driving, surveillance systems, and robotics.

2. Image Analysis: Segmentation provides a higher-level representation of the image content, allowing for more in-depth analysis. By dividing an image into distinct regions, different properties and characteristics of those regions can be studied individually. This enables tasks such as texture analysis, shape detection, and motion tracking, which are essential in medical imaging, satellite imagery analysis, and video processing.

3. Image Understanding: Segmentation helps in understanding the content and structure of an image. By partitioning the image into semantically meaningful regions, it becomes easier to comprehend the relationships between objects, their spatial arrangement, and the overall scene layout. This facilitates higher-level reasoning and decision-making processes in applications like scene understanding, object tracking, and augmented reality.

4. Image Editing and Manipulation: Segmentation is crucial for various image editing tasks, such as background removal, object manipulation, and image retouching. By segmenting the image, specific regions of interest can be isolated, allowing for selective modifications or enhancements. This is widely used in graphic design, photo editing software, and video post-production.

5. Image Compression: Segmentation can be used to exploit the redundancy present in an image and improve compression efficiency. By separating the image into meaningful regions, compression algorithms can allocate more bits to preserve important details while reducing the bits allocated to less critical regions. This leads to higher compression ratios without significant loss of visual quality.

In summary, segmentation is necessary in image processing as it provides a fundamental step for extracting meaningful information from images. It enables object recognition, image analysis, image understanding, image editing, and compression, ultimately facilitating a wide range of computer vision applications across various domains.

4b)Depict the edge linking using the local processing algorithm

Local processing

One of the simplest approaches for linking edge points is to analyze the characteristics of pixels in a small neighborhood about every point (x, y) that has been declared an edge point by one of the techniques discussed in the previous section. All points that are similar according to predefined criteria are linked, forming an edge of pixels that share common properties according to the specified criteria.

The two principal properties used for establishing similarity of edge pixels in this kind of analysis are (1) the strength (magnitude) and (2) the direction of the gradient vector. The first property is based on Eq. (10.2-10). Let S_{xy} denote the set of coordinates of a neighborhood centered at point (x, y) in an image. An edge pixel with coordinates (s, t) in S_{xy} is similar in *magnitude* to the pixel at (x, y) if

$$|M(s, t) - M(x, y)| \leq E \qquad (10.2\text{-}36)$$

where E is a positive threshold.

The direction angle of the gradient vector is given by Eq. (10.2-11). An edge pixel with coordinates (s, t) in S_{xy} has an *angle* similar to the pixel at (x, y) if

$$|\alpha(s, t) - \alpha(x, y)| \leq A \qquad (10.2\text{-}37)$$

where A is a positive angle threshold. As noted in Section 10.2.5, the direction of the edge at (x, y) is *perpendicular* to the direction of the gradient vector at that point.

A pixel with coordinates (s, t) in S_{xy} is linked to the pixel at (x, y) if both magnitude and direction criteria are satisfied. This process is repeated at every location in the image. A record must be kept of linked points as the center of the neighborhood is moved from pixel to pixel. A simple bookkeeping procedure is to assign a different intensity value to each set of linked edge pixels.

The preceding formulation is computationally expensive because all neighbors of every point have to be examined. A simplification particularly well suited for real time applications consists of the following steps:

1. Compute the gradient magnitude and angle arrays, $M(x, y)$ and $\alpha(x, y)$, of the input image, $f(x, y)$.
2. Form a binary image, g, whose value at any pair of coordinates (x, y) is given by:

$$g(x, y) = \begin{cases} 1 & \text{if } M(x, y) > T_M \text{ AND } \alpha(x, y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

where T_M is a threshold, A is a specified angle direction, and $\pm T_A$ defines a "band" of acceptable directions about A.

3. Scan the rows of g and fill (set to 1) all gaps (sets of 0s) in each row that do not exceed a specified length, K. Note that, by definition, a gap is bounded at both ends by one or more 1s. The rows are processed individually, with no memory between them.
4. To detect gaps in any other direction, θ, rotate g by this angle and apply the horizontal scanning procedure in Step 3. Rotate the result back by $-\theta$.

https://benchpartner.com/q/explain-about-the-edge-linking-procedures

4c)Define adaptive thresholding. how can thresholding be used in segmentation

Adaptive Thresholding is used in scenes with uneven illumination where same threshold value not usable throughout complete image.
In such case, look at small regions in the image and obtain thresholds for individual sub-images.
Final segmentation is the union of the regions of sub-images.
1. Thresholding Segmentation

The simplest method for segmentation in image processing is the threshold method. It divides the pixels in an image by comparing the pixel's intensity with a specified value (threshold). It is useful when the required object has a higher intensity than the background (unnecessary parts).
You can consider the threshold value (T) to be a constant but it would only work if the image has very little noise (unnecessary information and data). You can keep the threshold value constant or dynamic according to your requirements.
The thresholding method converts a grey-scale image into a binary image by dividing it into two segments (required and not required sections).
According to the different threshold values, we can classify thresholding segmentation in the following categories:

Simple Thresholding

In this method, you replace the image's pixels with either white or black. Now, if the intensity of a pixel at a particular position is less than the threshold value, you'd replace it with black. On the other hand, if it's higher than the threshold, you'd replace it with white. This is simple thresholding and is particularly suitable for beginners in image segmentation.

Adaptive Thresholding

Having one constant threshold value might not be a suitable approach to take with every image. Different images have different backgrounds and conditions which affect their properties.

Thus, instead of using one constant threshold value for performing segmentation on the entire image, you can keep the threshold value variable. In this technique, you'll keep different threshold values for different sections of an image.

This method works well with images that have varying lighting conditions. You'll need to use an algorithm that segments the image into smaller sections and calculates the threshold value for each of them.

From Chatgpt:

Thresholding is a fundamental technique used in image segmentation to separate objects or regions of interest from the background. It involves selecting a threshold value and classifying each pixel in the image as either foreground or background based on its intensity or other image characteristics.

Here's how thresholding can be used in segmentation:

1. Intensity-based Segmentation: In many cases, objects or regions of interest in an image exhibit different intensity characteristic compared to the background. Thresholding can be employed to separate these regions based on their intensity values. The basic idea is to select a threshold value that effectively distinguishes the foreground objects from the background based on intensity alone.

2. Binary Image Generation: Thresholding converts a grayscale or color image into a binary image, where pixels above the threshold are classified as foreground (object) pixels, and pixels below or equal to the threshold are classified as background pixels. This binary image can be used as a segmentation mask or a preliminary step for further analysis or processing.

3. Adaptive Thresholding: As mentioned earlier, adaptive thresholding adjusts the threshold value locally, taking into account the local characteristics of the image. This can be particularly useful when dealing with images that have non-uniform lighting conditions or variations in contrast. Adaptive thresholding allows for better adaptation to local image properties, leading to more accurate segmentation results.

4. Multiple Thresholds: In some cases, a single threshold may not be sufficient to separate different regions of interest in an image. Multiple thresholds can be utilized to segment an image into multiple classes or categories. This technique, known as multi-level thresholding or multiclass thresholding, enables the identification and separation of distinct objects or regions with different intensity ranges.

5. Feature-based Thresholding: Thresholding can also be applied based on features other than intensity, such as texture, color, gradient magnitude, or other image attributes. By selecting appropriate features and threshold values, it becomes possible to segment the image based on specific characteristics of interest, allowing for more advanced segmentation techniques.

6. Post-processing: Thresholding is often followed by post-processing operations to refine and improve the segmentation results. These operations can include morphological operations (such as erosion and dilation) to remove noise or fill gaps in the segmented regions, or contour extraction to obtain more precise boundaries of the segmented objects.

Thresholding is a versatile and widely used technique in image segmentation due to its simplicity and effectiveness. However, it is important to choose appropriate thresholding methods, adjust threshold values carefully, and consider the specific characteristics of the image and the desired segmentation goals for optimal results.

4d) Describe the segmentation process of printed characters

5a) define feature extraction how is the region features a choice
5b) explain the use of motion in segmentation
5c) calculate the area perimeter and shape factor using chain code
5d) define LBP local binary pattern give examples

Answer:
Local Binary Patterns (LBP)
LBP is one of the most commonly used features for texture discrimination tasks like face, facial expressions, gesture, scene and object recognition. LBP was conceived well before SIFT and HOG.
In this blog, we will see the calculation of LBP, how it can be expressed as a filter bank, code implementations and an example.

LBP Calculation

For calculating the LBP, the LBP code for each pixel is calculated and the histogram of LBP codes is constructed as the LBP feature.
To calculate the lbp code, for each pixel p, the 8 neighbours of the center pixel are compared with the pixel p and the neighbours x are assigned a value 1 if $x \geq p$. Fig. 1. shows how LBP code for a 3×3 area is calculated. This process is computed across the whole image.

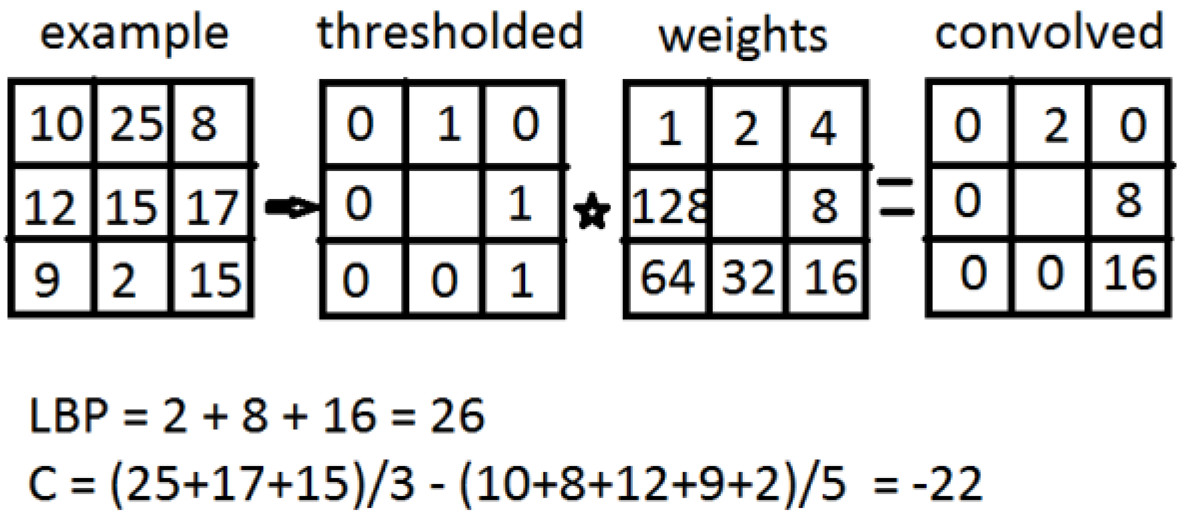


Fig 1. Example of how LBP is calculated
After computing the lbp code for entire image or image patch, calculate 256×1 histogram.
Filter Banks
Filter banks are a collection of filters which capture different scales and orientations.
The operation of comparing the center pixel with its neighbouring 8 pixels can be visualized as a weighted sum of 8 convolutions with sobel operator oriented in different directions.

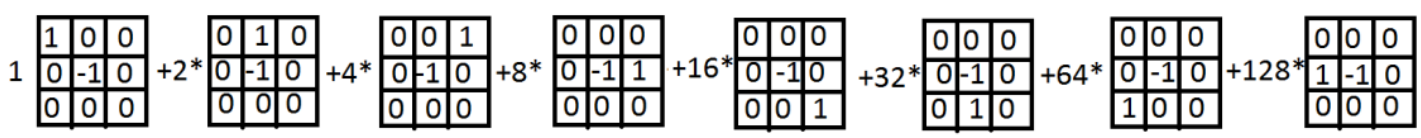
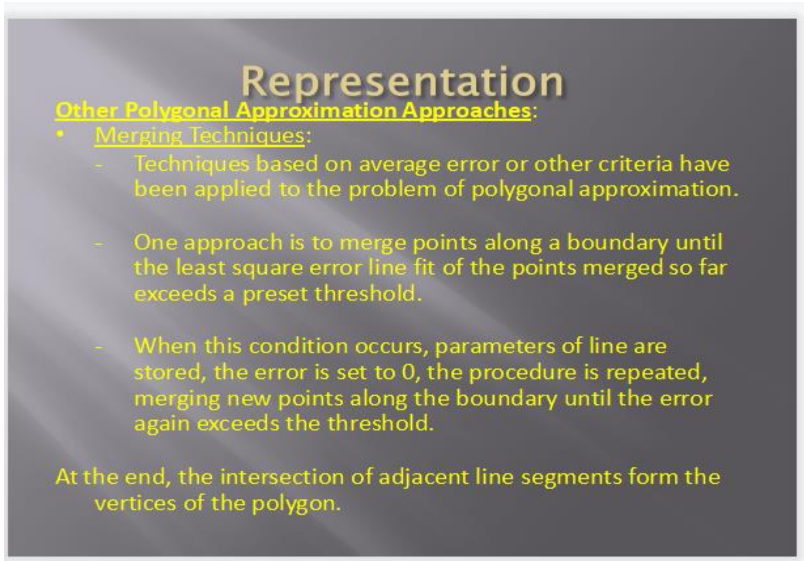
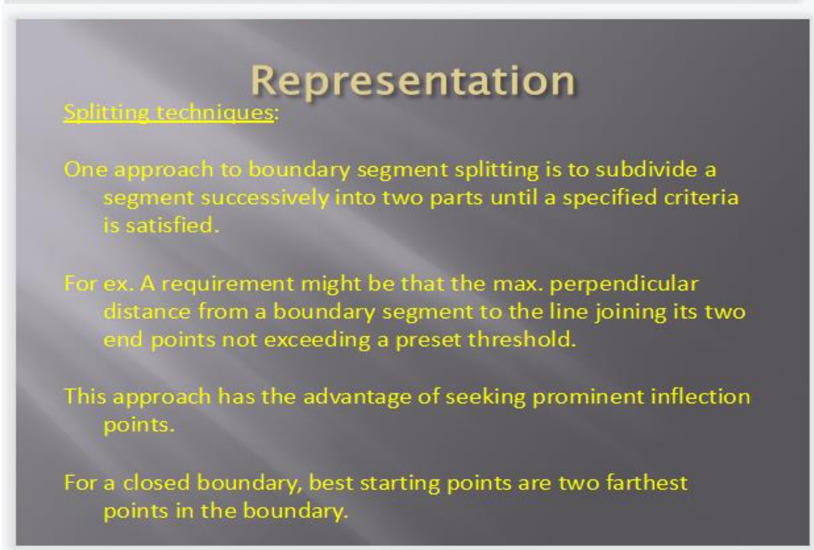


Fig. 2. LBP shown as filter bank
Visualizaing LBP as a filter bank, shifts the implementation from double nested for loop to matrix operations which effectively reduces the computation time.

6a) define pattern recognition.write down the applications of pattern recognition
6b) describe the minimum distance classifier IRDA decision functions for two pattern class w1 and w2 where to main vectors m1 = [5.1, 1.4,6.5] And m2 = [1.6, 0.4,2.1].
And find the pattern class for X1 = [4.6,1.2,5.2] and X2 = [3.5, 0.3, 1.9] using decision boundary.
6c) describe syntactic pattern recognition
6d) describe a pattern recognition system to recognize 'A' using template meaning matching

7a) describe region splitting and merging algorithm





7b) define the artificial neural network network with 10 inputs and two outputs
7c) describe the pattern recognition system

7d) what are the limitations of the chain code? how can we overcome the shortcomings?

Ans:

The limitations of chain codes in image processing include:

Sensitivity to Noise: Chain codes are highly sensitive to noise and small variations in the boundary of an object. Even minor noise or irregularities can result in significantly different chain code representations, leading to inaccurate boundary descriptions.

Lack of Scale Invariance: Chain codes are not inherently scale-invariant. If an object's size changes, the chain code representation will also change accordingly. This lack of scale invariance makes it challenging to compare or match objects at different scales using chain codes alone.

Rotation Sensitivity: Chain codes are sensitive to object rotations. When an object is rotated, the chain code representation will be different from the original, making it difficult to directly use chain codes for rotation-invariant object recognition or matching.

Autumn 2016

BismillahirRahmanir Rahim
International Islamic University Chittagong
Department of Computer Science & Engineering
B. Sc. in CSE Semester Final Examination, Autumn 2016
Course Code: CSE-4875 Course Title: Pattern Recognition and Image Processing
Total marks: 50 Time: 2 hours30 minutes

[Answer any two questions from Group-A and any three questions from Group-B; Separate answer script must
be used for Group-A and Group-B.]

G
roup-A 1. a) Define image compression. Describe image compression model.

b) **Define data redundancy. Describe various types of interpixel redundancy.**

c) **Decode the message 0.23355 using arithmetic coding technique for image compression. Coding model is given as follows.**

Symbol	Probability
a	0.2
e	0.3
i	0.1
o	0.2
u	0.1
!	0.1

d) **Describe the Fidelity Criteria.**

2. a) Define morphological image processing. What is structuring element?

b) **Describe opening and closing operation in morphological image processing. Give necessary example for both operations.**

c) **Describe Hit-or-Miss transformation in brief. Write the uses of this operation.**

d) **Define noise. Give the name of various noise removal algorithms.**

e) **Define the image processing in the Full-color and Pseudo-color.**

3. a) Encode the following sequences using LZW coding.

29	**150**	**150**	**29**
29	**150**	**150**	**29**
29	**150**	**150**	**29**
29	**150**	**150**	**29**

b) **What are the differences between lossy and lossless image compression?**

c) **Describe structuring element in morphological image processing.**

d) **Define convex hull.**

Group B

4. a) What is segmentation? Why is it so important in various industry?

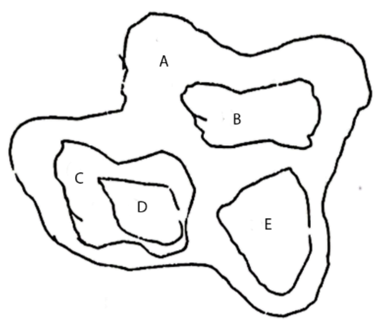
b) **Write down the techniques to segment the edges in image.**

c) **Explain how thresholding is used in segmentation.**

d) **Describe the region-oriented segmentation.**

5. a) Define chain code, skeleton, convex hull for representing object.

b) **Describe the following regions (in figure) using tree structure.**



c) **Describe ways to define regions using Euler formula.**

d) **Explain String and Tree representation of any object with example.**

6. a) Define pattern, pattern class, and pattern recognition.
b) Describe minimum distance classifier. Build the decision functions for two pattern classes w1 a w2. Where two mean vectors R1 (5.1, 1.4, 6.5)^T and R2= (1.6, 04, 2.1)^T. Find the decision boundary also. c) Draw an Artificial Neural Network for classifying five pattern classes.
d) Explain the pattern recognition model in brief.

7.a) Explain temple matching.

b) **Explain color-based segmentation.**

c) **Describe the segmentation process of character recognition in brief. d) Describe Edge Linking algorithm.**

Autumn 2015

Bismillahir Rahmanir Rahim

International Islamic University Chittagong Department of Computer Science & Engineering B. Sc. in CSE Semester Final Examination, Autumn 2015

Course Code: CSE-4875, Course Title: Pattern Recognition and Image Processing

Total marks: 50 Time: 2 hours 30 minutes

[Answer any two questions from Group-A and any three questions from Group-B; Separate answer script must be used for Group-A and Group-B.] Group-A 1. a) Define image compression. Describe image compression model.

b) **Define data redundancy. Describe various types of interpixel redundancy.**

c) **Decode the message 0.23355 using arithmetic coding technique for image compression. Coding model is given as follows.**

Symbol	Probability
a	0.2
e	0.3
i	0.1
o	0.2
u	0.1
!	0.1

d) **Describe the Fidelity Criteria.**

Morphological operations rely only on the relative ordering of pixel values, not on their numerical values" Explain.
Describe and differentiate dilation and erosion operation in morphological image processing. Give necessary example for both operations.

Describe boundary extraction and convex hull algorithms in brief? Write the uses of these operations.

What are the effects of structuring element in morphological processing?
Write down the differences between Lossy Compression and Error-Free Compression. Which one is used in document compression and why?

Explain the steps of JPEG algorithm in brief.

Explain morphological skeleton algorithm.
Why pruning is necessary in different image processing algorithms?

Group-B

4. a) What is segmentation? Why it is so important in various pattern recognition systems?

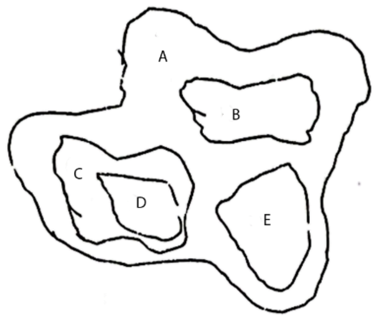
b) **Write down the Edge linking algorithm.**

c) **Define local, global and adaptive thresholding.**

d) **Describe the basic formulation of region. Explain any of the region-based segmentation algorithms.**

5. a) Define chain code, skeleton, convex hall for representing object.

b) **Describe the following regions (in figure) using tree structure.**



c) **Describe ways to define regions using Euler formula.**

d) **Explain String and Tree representation of any object with example.**

6. a) Define Pattern and pattern class. Write down the application areas of pattern recognition.

b) **Describe minimum distance classifier. Build the decision functions for two pattern class bolts and sewing needles. A set of two attribute classes (e.g. length along the major axis and head diameter). Where two mean vectors R1= (0.86, 2.34) and R2= (5.74, 5.85). Find the decision boundary also.**

c) **Define artificial neural network. Write the advantages and disadvantages of it.**

d) **Explain the pattern recognition system model in brief.**

7.a) Calculate the area, perimeter and shape factor using chain code.

•	•			
	•	•	•	•
•	•		•	
	•		•	•
	•		•	•
	•	•		•

b) **Write down the Edge linking algorithm in brief.**

c) **Define template matching with example.**

Spring 2014

Bismillahir Rahmanir Rahim

International Islamic University Chittagong

BS.C in Computer Science & Engineering

Final Examination, Spring 2014

CSE-4801, Pattern Recognition and Image Processing

Total marks: 50 Tone: 2:30 hours

[Answer any two questions from Group-Á and any three questions from Group-B; Separate answer script must be sed for Group-A and Group-B] Pattern Recognition and Image Processing

"

Group A" **1. A) What is image compression? Why do we need compression? B) Encode the following sequences using LZW coding.**

29	**160**	**160**	**29**
29	**160**	**160**	**29**
29	**160**	**160**	**29**
29	**160**	**160**	**29**

C) What are the differences between lossy and lossless image compression?

2. **A) Define morphological image processing. When dilation operation is used in image processing.**

   B) **Describe opening and closing operation in morphological image processing. In which context the two operations are applicable?**

   C) **Why pruning is necessary in different image processing algorithms?**

   D) **Define Hit-or-Miss transformation, morphological Skeleton.**

3. **A) Define data redundancy. Explain interpixel redundancy in brief.**

   B) Explain morphological skeleton algorithm.

   C) **Explain bit-plane decomposition technique for image compression. d) Describe Fidelity criteria for image compression.**

   D) **Define full color and pseudo color image processing.**

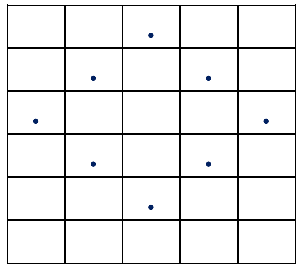
" Group B"

4. **A) Define Segmentation. Write the Edge detection algorithm in brief.**

   B) **Explain Thresholding. How Thresholding can be used in segmentation?**

   C) **What is the basic formulation of a region? How region will be selected in region splitting and merging algorithm.**

5. **A) Calculate the area, perimeter and shape factor using chain code.**



B) **Define signature, runcode and convex hill to represent any object.**

C) **Describe the tree and string descriptors to describe regions.**

6. **A) Define Pattern Recognition. Describe pattern recognition system design diagram.**

   B) **What is Artificial Neural Network? Draw a multilayer neural network to classify 10(Ten) English alphabets.**

   C) **Describe minimum distance classifier. Build the decision functions and decision boundary for two pattern classes' w1 and w2. Where two reference/mean vectors R1= $(4.5, 0.5)^T$ and R2= $(1.2, 3.5)^T$. D) Define template matching with example.**

7. **A) Define pattern class, classifier, and pattern recognition. B) Describe syntactic method of pattern recognition.**
   c) "The outputs of segmentation are attributes not an image"- Explain. D)Depict the importance of motion in segmentation.

Spring 2015

Bismillahir Rahmanir Rahiin

International Islamic University Chittagong Department of Computer Science & Engineering

B. Sc. in CSE Semester Final Examination, Spring 2015

Course Code: CSE-4801, Course Title: Pattern Recognition and Image Processing

Total marks: 50 **Time: 2 hours 30 minutes**

[Answer any two questions from Group-A and any three questions from Group-B; Separate answer script must be used for Group-A and Group-B.]

Group-A

1. **a) What is data compression? Explain different types of data redundancies**

b) Explain an image compression model.

c) Given a four-symbol source {a, b, c, d) with source probabilities {0.1, 0.4, 0.3, 0.2}, arithmetically encode the sequence bbadc.

2. **a) "Morphological image processing is based on set theory" -Explain.**

b) Describe and different opening and closing operation in morphological image processing. Give necessary example for both operations.

c) Describe Hit-or-Miss transformation in brief. Write the uses of this operation.

d) What are effects of structuring element in morphological processing?

e) What is the duality property?

3. **a) What is information theory?**

b) Explain the steps of JPEG algorithm in brief.

c) Define bit-plane decomposition technique for image compression. Explain any bit-plane coding algorithm in brief.

d) The skeleton can be expressed in terms of erosions and openings. Explain.

Group-B

4. **a) What is basic formulation of a region? Describe the region-oriented segmentation.**

b) Write down some application of image segmentation. Write down the basic concepts of motion-oriented segmentation.

c) Write the fundamental steps of edge detection.

5. **a) Define chain code, skeleton, convex hall for representing object.**

b) Define runcode. Calculate the area of the following object using run code.

c) Describe ways to define regions using Euler formula.

d) Explain Vector, String and Tree representation of any object with example.

6. **a) Define Pattern Recognition. Write down the application areas of pattern recognition.**

b) Describe minimum distance classifier. Build the decision functions for two pattern classes wl and w2. Where two mean vectors R1= $(5.1, 1.4, 6.5)^T$ and R2= $(1.6, 0.4, 2.1)^T$ Find the decision boundary also.

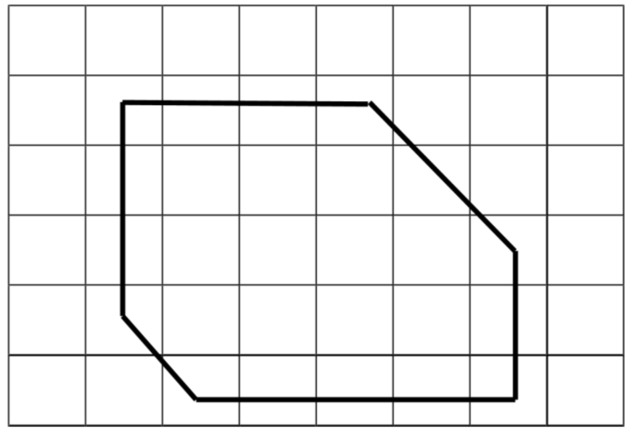
c) Explain the magnitude and direction of a point of an image (using gradient operator).

d) Explain diagonal line detection algorithm in brief.

7. **a) What is Artificial Neural Network? Discuss different types of Artificial Neural Network.**

b) Explain the syntactic method of pattern recognition.

c) How could you measure the compactness of a region? Find the compactness of the following region.



Autumn 2014

Final Examination, Autumn 2014

CSE-4801, Pattern Recognition and Image Processing

Total marks: 50 Time: 2:00 hours

[Answer any two questions from Group-A and any two questions from Group-B; Separate answer script must be

used for Group-A and Group-B]

"Group A"

1. a) Define image compression. Write the necessity of image compression in brief.

b) Calculate the relative data redundancy of the following image. When a natural 3 bit binary code 3.5 is used for each gray levels. (Symbols and their probabilities are given). Give a suitable code for each pixel to eliminate data redundancy.

Symbol	**Probability**
0	0.35
1	0.20
2	0.20
3	0.10
4	0.9
5	0.5
6	0.06
7	0.04

c) Encode the following images using LZW coding algorithm.

96	126	126	96
126	96	126	96
96	126	126	96
126	96	126	96

d) Describe information theory.

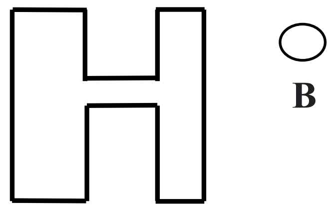
2. a) Define morphological image processing. When dilation operation is used in image processing

b) How does Erotion work to reduce the thickness of an Object? Explain with an example.

c) Describe morphological convex hull algorithm.

d) What are effects of structuring element in morphological processing.

e) Apply morphological opening and closing operations on the following image. Where A is the original image and B is the structuring element. Explain the output image.



B

A

3. a) Explain vector quantization compression.

b) Explain the steps of JPEG algorithm in brief.

c) Why pruning algorithm is necessary in many image processing applications?

d) Define bit-plane decomposition technique for image compression. Explain any bit-plane coding algorithm in brief.

"Group B"

4. a) Define Segmentation. Differentiate discontinuity and similarity-based segmentations in brief b) Write down edge detection algorithm in brief.

c) Explain diagonal line detection algorithm in brief.

d) Why edge linking algorithm is needed after edge detection algorithm in many cases? Explain.

5. a) Calculate the area, perimeter and shape factor using chain code.

		•			
	•		•		
	•		•		
	•		•	•	
•				•	
	•				•
	•			•	•
	•	•	•	•	

b) Explain Euler formula to describe topological objects.

c) Describe the vector, string and tree representations of objects.

6. a) Define Pattern Recognition, pattern classes and computer vision.

b) What is Artificial Neural Network? Draw a multilayer neural network to determine eight 3. pattern classes.

c) Describe minimum distance classifier. Build the decision functions and decision boundary for two pattern classes' w1 and w2. Where two mean vectors R1= $(5.5, 2.3, 0.5)^T$ and R2= $(1.2, 0.5, 2.5)^T$. d) Define template matching with example.

7. a) Explain decision boundary.

b) Write the concepts of motion-oriented segmentation.

c) Explain Skeleton algorithm.

d) Explain syntactic pattern recognition with necessary diagram.

IP Math

13 Given six symbol source $\{a_2, a_6, a_1, a_4, a_3, a_5\}$ with source probability $\{0.4, 0.3, 0.1, 0.1, 0.06, 0.04\}$. Give suitable code for each pixel to eliminate data redundancy.

Ans:

Symbol	Probability	code	1	2	3	4
a_2	0.4	1	0.4	0.4	0.4	0.6 0
a_6	0.3	00	0.3	0.3	0.3 00	0.4 1
a_1	0.1	011	0.1	0.2 010	0.3 01	
a_4	0.1	0100	0.1 0100	0.1 011		
a_3	0.06	01010	0.1 0101			
a_5	0.04	01011				

$$L_{avg} = 0.4 \times 1 + 0.3 \times 2 + 0.1 \times 3 + 0.1 \times 4 + 0.06 \times 5 + 0.04 \times 5$$

$$= 2.2$$

Here,
n_1 = number bits need to represent symbol
$n_2 = L_{avg}$

$$C_R = \frac{n_1}{n_2} = \frac{3}{2.2} = 1.36$$

$$\therefore R_d = 1 - \frac{1}{C_R} = 1 - \frac{1}{1.36} = 0.27 = 27\%$$ (Ans)

Chain code : 00764431

Perimeter = even count + $\sqrt{2}$. odd count

$= 6 + 3\sqrt{3}$

Value of y	code	Area
$y = y$	0	Area = Area + y
$y = y+1$	1	Area = Area + (y + 0.5)
$y = y+1$	2	Area = Area
$y = y+1$	3	Area = Area − (y + 0.5)
$y = y$	4	Area = Area − y
$y = y-1$	5	Area = Area − (y − 0.5)
$y = y-1$	6	Area = Area
$y = y-1$	7	Area = Area + (y − 0.5)

$$\text{Area} = y + y + (y - 0.5) + 0 - (y-2) - (y-2) - (y-2) - (y - 2 + 0.5) + (y - 1 + 0.5)$$

$$= 2y + y - 0.5 - y - y + 2 - y + 2 - y + 2 - y + 2 - 0.5 + y - 1 + 0.5$$

$$= 6.5 .$$

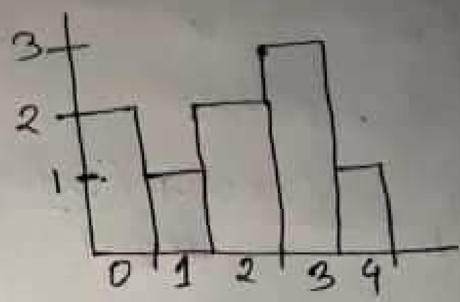
26/33

Scanned by CamScanner

CamScanner

$$\sigma_B^2 = \omega_b \omega_f (\mu_b - \mu_f)^2$$

Iteration 1:



$$\omega_b = 0$$

$$\mu_b = 0$$

$$\omega_f = \frac{2+1+2+3+1}{9} = 1$$

$$\mu_f = \frac{(2\times0)+(1\times1)+(2\times2)+(3\times3)+(1\times4)}{2+1+2+3+1}$$

$$= 2$$

$$\sigma_B^2 = \omega_b \omega_f (\mu_b - \mu_f)^2$$

$$= 0 \times 1 (0-2)^2 = 0$$

Iteration-2:

$$\omega_b = \frac{2}{9} = 0.22$$

$$\mu_b = \frac{2\times0}{2} = 0$$

$$\omega_f = \frac{1+2+3+1}{9} = 0.78$$

$$\mu_f = \frac{(1\times1)+(2\times2)+(3\times3)+(1\times4)}{1+2+3+1}$$

$$= 2.57$$

$$\sigma_B^2 = 0.22 \times 0.78 (0-2.57)^2 = 1.13$$

Iteration-3:

$$\omega_b = \frac{2+1}{9} = 0.33$$

$$\mu_b = \frac{(2\times0)+(1\times1)}{2+1} = 0.33$$

$$\omega_f = \frac{2+3+1}{9} = 0.67$$

$$\mu_f = \frac{(2\times2)+(3\times3)+(1\times4)}{2+3+1} = 2.83$$

$$\sigma_B^2 = 0.33\times0.67(0.33-2.83)^2 = 1.38$$

Iteration-4:

$$\omega_b = \frac{2+1+2}{9} = 0.56$$

$$\mu_b = \frac{(2\times0)+(1\times1)+(2\times2)}{2+1+2} = 1$$

$$\omega_f = \frac{3+1}{9} = 0.44$$

$$\mu_f = \frac{(3\times3)+(1\times4)}{3+1} = 3.25$$

$$\sigma_B^2 = 0.56\times0.44(1-3.25)^2 = 1.247$$

Iteration-5:

$$\omega_b = \frac{2+1+2+3}{9} = 0.89$$

$$\omega_f = \frac{1}{9} = 0.11$$

$$\mu_b = \frac{(2\times0)+(1\times1)+(2\times2)+(3\times3)}{2+1+2+3} = 1.75$$

$$\mu_f = \frac{1\times4}{9} = 0.44$$

$$\sigma_B^2 = 0.89\times 0.11\,(1.75-0.44)^2 = 0.168$$

Iteration-6:

$$\omega_b = \frac{2+1+2+3+1}{9} = 1$$

$$\mu_b = \frac{(2\times0)+(1\times1)+(2\times2)+(3\times3)+(1\times4)}{2+1+2+3+1} = 2$$

$$\omega_f = 0$$

$$\mu_f = 0$$

$$\sigma_B^2 = 1\times0\,(2-0)^2 = 0$$

So, Optimum thresholding value is 1.38 with partition $\{0,1\}$ & $\{2,3,4\}$

Au-21

1(b)

	Probability					
D	0.1	0.2	0.2	0.2	0.2	0.3
e	0.1	0.1	0.2	0.2	0.2	0.2
P	0.1	0.1	0.1	0.2	0.2	0.2
t	0.1	0.1	0.1	0.1	0.2	0.2
o	0.1	0.1	0.1	0.1	0.1	
f	0.1	0.1	0.1	0.1		
C	0.1	0.1	0.1			
S	0.1	0.1				
E	0.1					

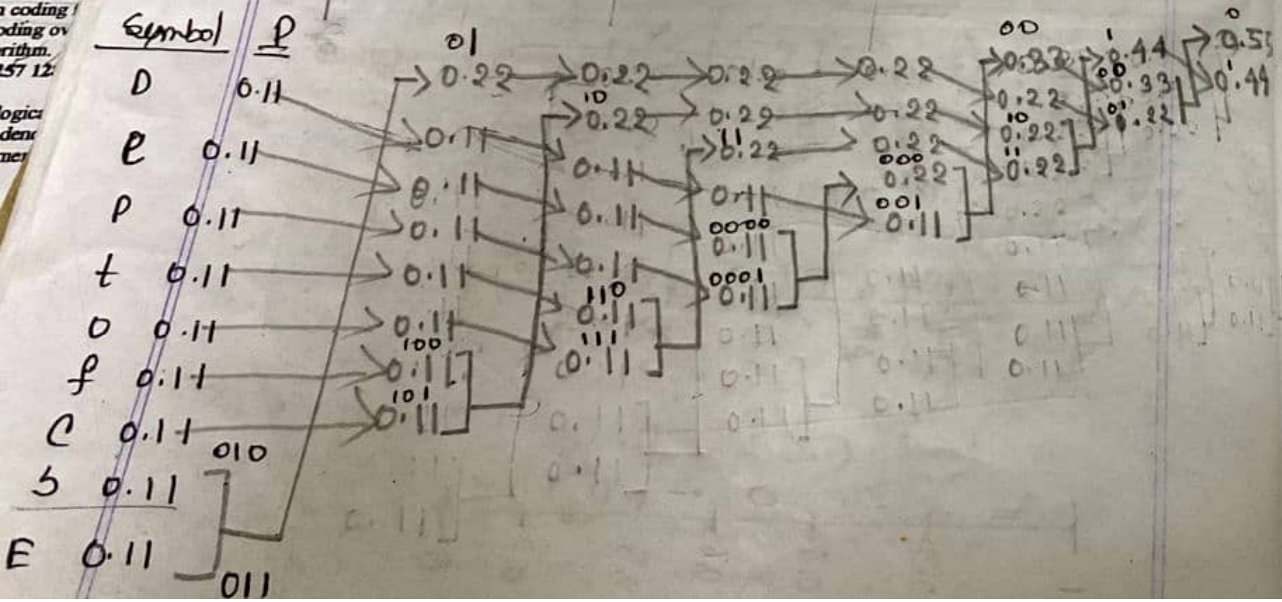
0.5

Symbols	Prob								Code
D	0.1	0.2	0.2	0.2	0.2	0.3	0.4	0.6	
e	0.1	0.1	0.2	0.2	0.2	0.2	0.3	0.4	
p	0.1	0.1	0.1	0.2	0.2	0.2	0.2		
d	0.1	0.1	0.1	0.1	0.2	0.2	01		
o	0.1	0.1	0.1	0.1	0.1				
f	0.1	0.1	0.1	0.1					
c	0.1	0.1	0.1						
S	0.1	0.1							
E	0.1								

Huffman coding.

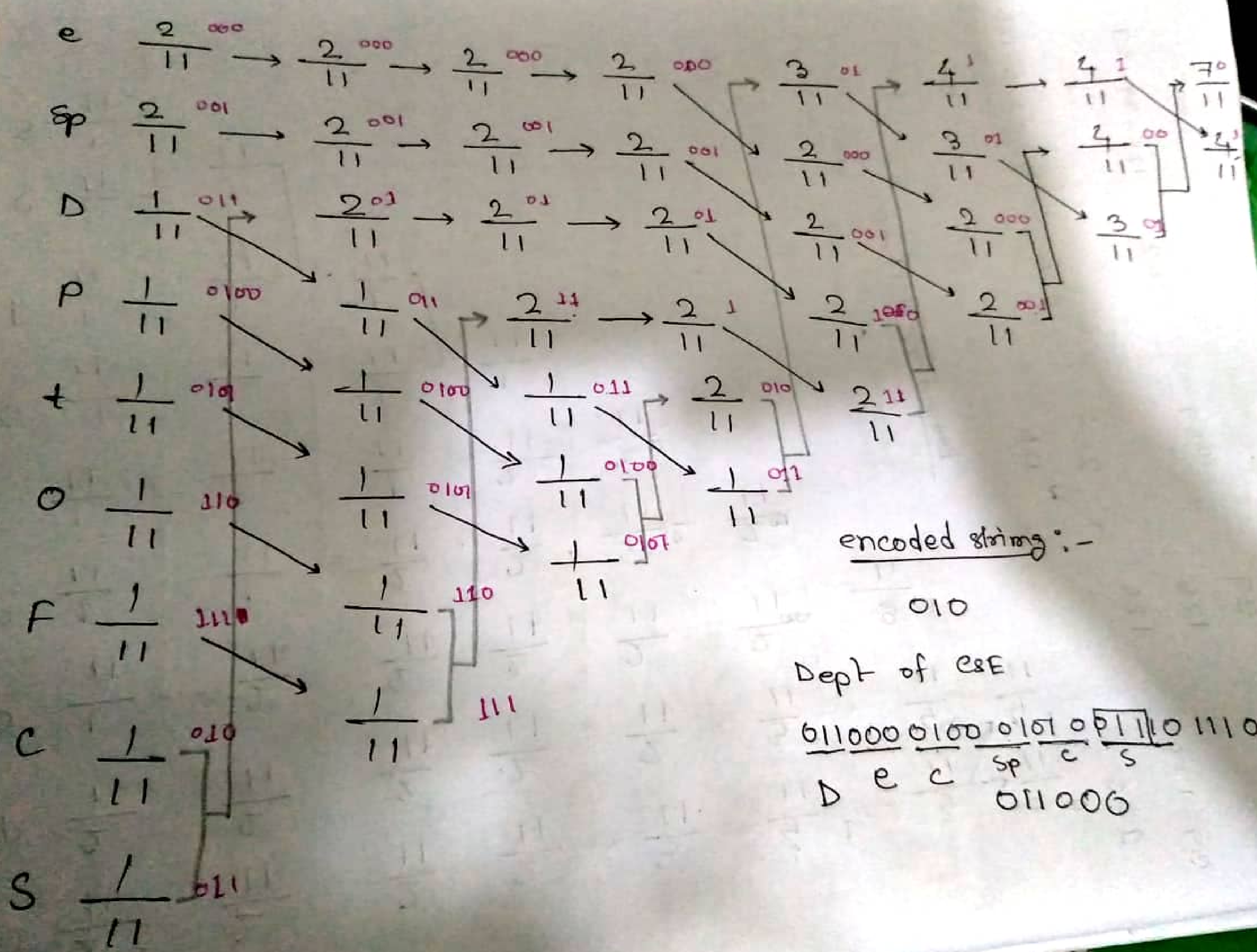
(1b) Dept of CSE

Symbol	Probability
D	1/9 = 0.11
e	1/9 = 0.11
p	1/9 = 0.11
t	1/9 = 0.11
o	1/9 = 0.11
f	1/9 = 0.11
c	1/9 = 0.11
S	1/9 = 0.11
E	1/9 = 0.11



Symbol	Probability	Codeword	length
D	0.11	001	3
e	0.11	0000	4
p	0.11	0001	4
t	0.11	110	3
o	0.11	111	3
f	0.11	100	3
c	0.11	101	3
S	0.11	010	3
E	0.11	011	3

encode 0010000 0001110 111100101 010 011



encoded string :-

010

Dept of CSE

$\underbrace{011}_{D}\underbrace{000}_{e}\underbrace{0100}_{c}\underbrace{0101}_{Sp}\ \underbrace{0}_{c}\ \underbrace{\boxed{011}}_{S}10$ 111001010

011006

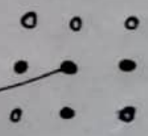
10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

Morphological Operation:

Spring 19_2(b)

* Erroshion: (Fit)

origin



B

x

B = Structural Element

$Y_1 = x \ominus B$

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

B যেখানে full fit হবে সেখানের Origin কে Count করবে, এক্ষেত্রে প্রতিটি Pixel চেক করবে।

* Dilation: (Hit, Fit)

$Y_2 = x \oplus B$

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

B কোন জায়গায় hit করলেই hit value B এর Origin এ add হবে, এক্ষেত্রে প্রতিটি Pixel চেক করবো।

* Opening : (1st Erosion then 2nd Dilation)

From $Y_1 = x \ominus B$ we get Erosion

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

$x \ominus B$

$\xrightarrow{\text{Dilation}}$

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

$(x \ominus B) \oplus B$

[Opening]

* Closing : (1st Dilation then 2nd Erosion)

From $Y_2 = x \oplus B$ we get Dilation

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

$x \oplus B$

$\xrightarrow{\text{Erosion}}$

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

$(x \oplus B) \ominus B$

[Closing]

LZW Coding: (Encoding)

SP-22
1(b) Given,

200	200	129	129
200	200	129	129
200	200	129	129
200	200	129	129

Currently Recognized Sequence	Pixel being Processed	Encoded output	Dictionary location	Dictionary Entry
	200			
200	200	200	256	200-200
200	129	200	257	200-129
129	129	129	258	129-129
129	200	129	259	129-200
200	200			
200-200	129	256	260	200-200-129
129	129			
129-129	200	258	261	129-129-200
200	200			
200-200	129			
200-200-129	129	260	262	200-200-129-129
129	200			
129-200	200	259	263	129-200-200
200	129			
200-129	129	257	264	200-129-129
129		129		

REDMI 10C | TÁQÎA 07/06/2023 22:13

∴ Encoded Output

200 200 129 129

256 258 260 259

257 129

(Ans)

REDMI 10C | TÁQÎA 07/06/2023 22:13

∴ Encoded output

200 200 129 129
256 258 260 259
257 129

(Ans)

LZW Coding: (DeCoding)

Currently Recognized Sequence	Pixel being Proceed	Decode output	Dictionary location	Dictionary Entry
	200			
200	200	200	256	200-200
200	129	200	257	200-129
129	129	129	258	129-129
129	256 DE 1st value	129	259	129-200
256	258 DE 1st value	200-200	260	200-200-129
258	260 DE 1st value	129-129	261	129-129-200
260	259 DE 1st values	200-200-129	262	200-200-129-129
259	257 DE 1st value	129-200	263	129-200-200
257	129 DE 1st value	200-129	264	200-129-129
129		129		

∴ Decode output

200 200 129 129
200 200 129 129
200 200 129 129
200 200 129 129

(Ans)

REDMI 10C | TÁQÎA 08/06/2023 14:48

Decision Function & Decision Boundary:

1

Spring: 22

5(b)

Given,

$$R_1^T = (5.6 \quad 1.2 \quad 6.2)$$

$$R_2^T = (1.4 \quad 0.3 \quad 3.7)$$

$$R_1 = \begin{pmatrix} 5.6 \\ 1.2 \\ 6.2 \end{pmatrix} \qquad R_2 = \begin{pmatrix} 1.4 \\ 0.3 \\ 3.7 \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

The Decision Functions —

$$d_1(x) = xR_1 - \frac{1}{2} R_1^T R_1$$

$$= (x_1 \; x_2 \; x_3)\begin{pmatrix} 5.6 \\ 1.2 \\ 6.2 \end{pmatrix} - \frac{1}{2}(5.6 \quad 1.2 \quad 6.2)\begin{pmatrix} 5.6 \\ 1.2 \\ 6.2 \end{pmatrix}$$

$$= 5.6x_1 + 1.2x_2 + 6.2x_3 - \frac{1}{2}(31.36 + 1.44 + 38.44)$$

$$\therefore d_1(x) = 5.6x_1 + 1.2x_2 + 6.2x_3 - 35.62$$

$$d_2(x) = xR_2 - \frac{1}{2} R_2^T R_2$$

$$= (x_1 \; x_2 \; x_3)\begin{pmatrix} 1.4 \\ 0.3 \\ 3.7 \end{pmatrix} - \frac{1}{2}(1.4 \quad 0.3 \quad 3.7)\begin{pmatrix} 1.4 \\ 0.3 \\ 3.7 \end{pmatrix}$$

$$= 1.4x_1 + 0.3x_2 + 3.7x_3 - \frac{1}{2}(1.96 + 0.09 + 13.69)$$

$$\therefore d_2(x) = 1.4x_1 + 0.3x_2 + 3.7x_3 - 7.87$$

The equation of decision boundary —

$$d_{12}(x) = d_1(x) - d_2(x)$$

$$= (5.6x_1 + 1.2x_2 + 6.2x_3 - 35.62) - (1.4x_1 + 0.3x_2 + 3.7x_3 - 7.87)$$

$$= 4.2x_1 + 0.9x_2 + 2.5x_3 - 27.75$$

$$\therefore d_{12}(x) = 4.2x_1 + 0.9x_2 + 2.5x_3 - 27.75 \quad \underline{\text{(Ans)}}$$

Autumn: 21

5(a) Given,

$m_1^T = (6.1, 1.6, 6)$ $x = (3.1\ 0.4\ 2.8)$

$m_2^T = (1.2, 0.2, 2.0)$

$$m_1 = \begin{pmatrix} 6.1 \\ 1.6 \\ 6 \end{pmatrix} \quad m_2 = \begin{pmatrix} 1.2 \\ 0.2 \\ 2.0 \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

The decision functions ——

$$d_1(x) = x m_1 - \frac{1}{2} m_1^T m_1$$

$$= (x_1\ x_2\ x_3)\begin{pmatrix} 6.1 \\ 1.6 \\ 6 \end{pmatrix} - \frac{1}{2}(6.1\ 1.6\ 6)\begin{pmatrix} 6.1 \\ 1.6 \\ 6 \end{pmatrix}$$

$$= 6.1x_1 + 1.6x_2 + 6x_3 - \frac{1}{2}(75.77)$$

$$= (6.1\times 3.1) + (1.6\times 0.4) + (6\times 2.8) - 37.88$$

$$=(36.35-37.88)$$

$$\therefore d_1(x) = -1.53$$

$$d_2(x) = x m_2 - \frac{1}{2} m_2^T m_2$$

$$= (x_1\ x_2\ x_3)\begin{pmatrix}1.2\\0.2\\2.0\end{pmatrix} - \frac{1}{2}(1.2\ 0.2\ 2.0)\begin{pmatrix}1.2\\0.2\\2.0\end{pmatrix}$$

$$= 1.2x_1 + 0.2x_2 + 2.0x_3 - \frac{1}{2}(5.48)$$

$$= (1.2\times 3.1) + (0.2\times 0.4) + (2.0\times 2.8) - 2.74$$

$$= 9.4 - 2.74$$

$$= 6.66$$

The equation of decision boundary —

$$d_{12}(x) = d_1(x) - d_2(x)$$

$$= -1.53 - 6.66$$

$$= -8.19 < 0$$

So, This Pattern from ω_2 (Ans)

$\boxed{\text{Note}}$: If $d_{12}(x) > 0$ then it will be from ω_1 Patterns.

Spring: 19

4

6(b) Given,

$m_1^T = (5.1 \ 1.4 \ 6.5)$. $x_1 = (4.6 \ 1.2 \ 5.2)$

$m_2^T = (1.6 \ 0.4 \ 2.1)$ $x_2 = (3.5 \ 0.3 \ 1.9)$

$$m_1 = \begin{pmatrix} 5.1 \\ 1.4 \\ 6.5 \end{pmatrix} \quad m_2 = \begin{pmatrix} 1.6 \\ 0.4 \\ 2.1 \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

The decision functions —

$$d_1(x) = x_1 m_1 - \frac{1}{2} m_1^T m_1$$

$$= (x_1 \ x_2 \ x_3)\begin{pmatrix} 5.1 \\ 1.4 \\ 6.5 \end{pmatrix} - \frac{1}{2}(5.1 \ 1.4 \ 6.5)\begin{pmatrix} 5.1 \\ 1.4 \\ 6.5 \end{pmatrix}$$

$$= 5.1x_1 + 1.4x_2 + 6.5x_3 - \frac{1}{2}(70.22)$$

$$= (5.1 \times 4.6) + (1.4 \times 1.2) + (6.5 \times 5.2) - 35.11$$

$$= 58.94 - 35.11$$

$$\therefore d_1(x) = 23.83$$

$$d_2(x) = x_2 m_2 - \frac{1}{2} m_2^T m_2$$

$$= (x_1 \ x_2 \ x_3)\begin{pmatrix} 1.6 \\ 0.4 \\ 2.1 \end{pmatrix} - \frac{1}{2}(1.6 \ 0.4 \ 2.1)\begin{pmatrix} 1.6 \\ 0.4 \\ 2.1 \end{pmatrix}$$

$$= 1.6x_1 + 0.4x_2 + 2.1x_3 - \frac{1}{2}(7.13)$$

$$= (1.6\times3.5)+(0.4\times0.3)+(2.1\times1.9)-3.565$$

5

$$= (9.71-3.565)$$

$$= 6.145$$

The decision boundary equation —

$$d_{12}(x) = d_1(x) - d_2(x)$$

$$= 23.83 - 6.145$$

$$= 17.685 > 0$$

So, This Pattern from ω_1 (Ans)

$\boxed{\text{Note}}$: If $d_{12}(x) < 0$ then it will be from ω_2 Patterns.

REDMI 10C | TÁQÎA 09/06/2023 11:33