All CHAPTER PPTs: https://drive.google.com/drive/folders/1TBUM5O12zV2AY-Iz8rIUeJpXmNOpKjgA

CHAPTER 1:

An image is a visual representation of something, while a digital image is a binary representation of visual data.

What is a Digital Image? : https://youtu.be/aduibjn0ehM

Pixel: Pixel is the smallest element of an image. Each pixel correspond to any one value. OR A digital image contains a finite set of elements called pixels. Each pixel represents the color at a single point in the image.

Calculation of total number of pixels: We have define an image as a two dimensional signal or matrix. Then in that case the number of Pixel would be equal to the number of rows multiply with number of columns.

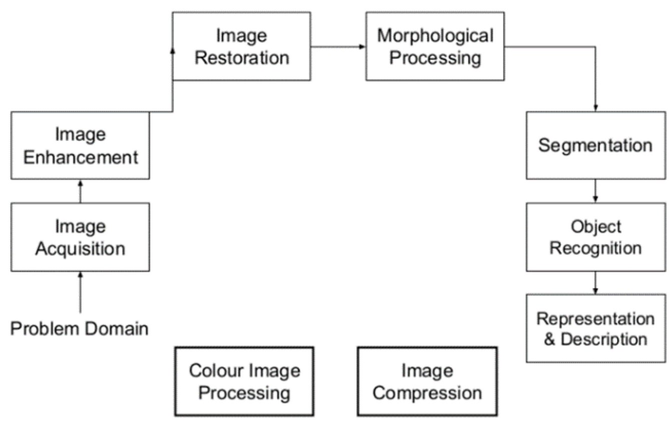
This can be mathematically represented as below:

Total number of pixels = number of rows * number of columns

Digital image processing focuses on two major tasks

- Improvement of pictorial information for human interpretation
- Processing of image data for storage, transmission and representation for autonomous machine perception

Key Stages in Digital Image:



1. Image Acquision: The Image is captured by a sensor (ex. Camera) and digitized if the output of the camera or sensor is not already in digital form, using analogue to digital converter.
2. Image Enhancement: Image enhancement is the simplest and most attractive area of DIP. In this stage details which are not known, or we can say that interesting features of an image is highlighted. Such as brightness, contrast, etc…, here basically noice from the image will be removed using wiener filter.
3. Image Restoration: Image restoration is the stage in which the appearance of an image is improved.
4. Morphological Processing: It deals with tools for extracting image components that are useful in the representation and description of shape.
5. Image Segmentation: Segmentation procedures partition an image into its constituents parts or objects. The more accurate the segmentation, the more likely the recognition is to succeed.
6. Object Recognition: The process that assigns a label to an object based on the information provided by its description.
7. Representation and Description: Representation and description follow the output of the segmentation stage. The output is a raw pixel data which has all points of the region itself. To transform the raw data, representation is the only solution. Whereas description is used for extracting information's to differentiate one class of objects from another.
8. Image Compression: Compression is a technique which is used for reducing the requirement of storing an image. It is a very important stage because it is very necessary to compress data for internet use.
9. Color image processing: Color image processing is a famous area because it has increased the use of digital images on the internet. This includes color modeling, processing in a digital domain, etc....

Application of Image Processing:

1. **Healthcare domain:** X-rays
2. **Biometrics:** DIP is utilized for facial recognition, fingerprint identification, and iris scanning in security systems and authentication processes.
3. **Document Processing:** DIP helps in tasks like text extraction, handwriting recognition, and document image enhancement, facilitating digitization and management of documents.
4. **Remote Sensing:** Monitoring environment, land use, and disasters using satellite and aerial images.
5. **Security and Surveillance:** Facial recognition, object tracking, and suspicious activity detection.
6. **Graphic Design:** Image editing, retouching, and visual effects in digital art.

7. **Entertainment:** Realistic graphics and special effects in gaming and movies.
8. **Automotive:** Object detection, lane tracking, and collision avoidance in self-driving cars.
9. **Manufacturing:** Inspecting products for defects and ensuring quality.
10. **Astronomy:** Analyzing celestial images to study the universe.
11. **Forensics:** Enhancing crime scene images and facial reconstruction.

Sampling and Quantization: https://youtu.be/KWc9SOOLfLw?si=n4oI79T6gPF3kFZJ

- **Sampling:** Sampling is the process of converting a continuous signal, like an analog signal (a smooth curve), into a discrete signal (a series of individual points). In the context of images, it's like taking a "snapshot" of the image at specific intervals. This is essential for representing analog signals in digital form, as computers can only handle discrete values.
- **Quantization:** Quantization is the process of assigning discrete values to the sampled data points. It's like deciding on the closest "allowed" value for each point on the sampled curve. In image processing, this involves mapping pixel values from a continuous range (e.g., 0 to 255 in an 8-bit image) to specific digital values.
- **In simpler terms:**

Sampling: Picking specific points from a smooth line to create a digital version of it.
Quantization: Giving those picked points exact digital values that the computer can understand.

Image Acquisition: https://youtu.be/3_3YlrMcEJE?si=MzgM7NAHcVe-0sWO

Resolution: Resolution is a measurement of the number of pixels -- picture elements or individual points of color -- that can be contained on a display screen or in a camera sensor. In practical terms, resolution describes the sharpness, or clarity, of an image or picture. It is expressed in terms of the number of pixels that can be displayed both horizontally and vertically.

Resolution is an important factor to measure the visual quality of digital images, photos and videos. A higher resolution signifies the picture contains more pixels, which means it can display more visual information. As a result, a high-resolution picture is sharper and clearer than a low-resolution one.

Spatial and Intensity Resolution: https://youtu.be/DOYRMHAST4U?si=QiD5i7ANadgGkkD2

Image Interpolation: Image interpolation occurs in all digital photos at some stage in photo enlargement. It happens anytime you resize or remap (distort) your image from one pixel grid to another. Interpolation is commonly used in various image processing tasks, including resizing images, geometric transformations, and enhancing image quality.

1. Nearest-Neighbor Interpolation: Nearest-neighbor interpolation is the simplest interpolation method, where the value of the nearest pixel is assigned to non-integer coordinates.

Algorithm:
Given a non-integer coordinate (x, y), find the nearest integer coordinates (rounding to the nearest).
Assign the value of the nearest pixel to the non-integer coordinate.
Effect: While computationally efficient, this method can result in blocky artifacts, especially when enlarging images.

2. Bilinear Interpolation: Bilinear interpolation considers the values of the four nearest pixels and computes a weighted average based on the distances to the target position.

Algorithm:
Given a non-integer coordinate (x, y), find the four nearest integer coordinates.
Compute the weighted average of pixel values based on the distances to the target position.
Effect: Bilinear interpolation provides smoother results compared to nearest-neighbor interpolation, but it may still exhibit some artifacts.

3. Bicubic Interpolation:

Bicubic interpolation is a higher-order interpolation method that considers a larger neighborhood of 16 pixels. It uses cubic polynomials to estimate pixel values.

Algorithm:
Given a non-integer coordinate (x, y), find the 16 nearest pixels in a 4x4 neighborhood.
Apply a cubic polynomial to interpolate pixel values based on the distances to the target position.
Effect: Bicubic interpolation offers higher quality results and is especially useful for image resizing and geometric transformations. It tends to produce smoother and more accurate interpolations compared to bilinear interpolation.

Comparison:
- Nearest-neighbor interpolation is computationally efficient but can result in blocky artifacts, especially during image enlargement.
- Bilinear interpolation provides smoother results than nearest-neighbor interpolation and is suitable for moderate-quality resizing.
- Bicubic interpolation is a higher-quality method that produces smoother and more accurate results, making it suitable for tasks like image resizing, geometric transformations, and anti-aliasing.

SUM based of Interpolation: https://youtu.be/dIlR8vS9T-c?si=OC5yyELzaa5b7ECz

CHAPTER 2:

Goal of Segmentation: Divide each image into pieces, where each piece represents a distinguished thing in the image. It is important that all of the pieces have approximately equal importance.
Histogram of gray level images and Histogram Equalization sum: https://youtu.be/Yd6QISby8kk
Histogram matching sum: https://youtu.be/r565euxWZBs?si=p5RhvBKyFFLhCZVI
Multilevel Thresholding: Multilevel thresholding is a process that segments a gray level image into several distinct regions. This technique determines more than one threshold for the given image and segments the image into certain brightness regions, which correspond to one background and several objects. The method works very well for objects with colored or complex backgrounds, on which bi-level thresholding fails to produce satisfactory results.

Multilevel Thresholding for Image Segmentation:
Multilevel thresholding is like sorting things into different boxes based on their shades of grey. Imagine you have a bunch of objects with different shades of gray, and you want to group them by their darkness or brightness.

- Choosing Thresholds: Start by selecting multiple threshold values. These thresholds are like lines that separate the different shades of gray.
- Grouping Pixels: Now, look at each pixel in the image. If it's darker than the first threshold, you put it in one box. If it's between the first and second thresholds, it goes in another box. Keep doing this for all the thresholds.
- Creating Segments: The boxes you've created are like segments of the image. Pixels with similar shades of gray end up in the same segment.
- Object Separation: This technique helps to separate different objects or areas based on their gray levels. It's like sorting clothes into different piles based on their colors.

In simple terms, multilevel thresholding divides a grey-level image into segments by using multiple threshold values to group pixels with similar shades of gray together. This technique is useful for segmenting objects based on their brightness or darkness in the image.

Watershed algorithm for segmenting gray level image: https://youtu.be/FLmxZaQhvsI?si=GZ2jm7HFma63_jfD
The Watershed algorithm is a computer vision and image processing technique used for image segmentation, which is the process of dividing an image into meaningful regions or objects. It is particularly useful for segmenting objects in images that have unclear or complex boundaries. Here's an explanation of how the Watershed algorithm works and its limitations:
How the Watershed Algorithm Works:
Grayscale Image: The Watershed algorithm typically works on grayscale images. If you have a color image, you can convert it to grayscale.
Gradient Calculation: It calculates the gradient of the image. This gradient represents the intensity changes in the image and helps identify potential boundaries between objects.
Marker Generation / Marker Placement: The user or an automated process needs to specify markers in the image. Markers are points or regions that indicate where different objects or regions should be separated. These markers are often labeled with different values.
Flooding: Starting from the markers, a "flooding" process occurs. The algorithm simulates the flooding of an area, where water rises from the markers. As the water level rises, it fills the image, and when the water levels from different markers meet, they form watershed lines.
Watershed Lines: Watershed lines are the boundaries that separate different objects or regions. The algorithm assigns a label to each region, and the result is a segmented image with distinct regions.

Limitations of the Watershed Algorithm:

Over-Segmentation: The Watershed algorithm is sensitive to noise and fine details in the image. This can lead to over-segmentation, where the algorithm creates too many small regions, making it challenging to interpret the segmentation result.

Marker Placement: The effectiveness of the algorithm heavily depends on the correct placement of markers. Incorrect or imprecise marker placement can lead to poor segmentation results.

Gradient Calculation: The accuracy of the gradient calculation can be influenced by the choice of gradient operator and noise in the image. Noisy images can lead to incorrect boundaries.

Computational Complexity: The Watershed algorithm can be computationally expensive, especially for large images. Efficient implementations and optimizations are required to make it practical for real-time or large-scale applications.

Overcoming Plateaus: In some cases, plateaus in the image can be misinterpreted as watershed lines, leading to inaccurate segmentation. Additional pre-processing steps may be needed to address this issue.

Tuning Parameters: The Watershed algorithm often has parameters that require tuning, such as marker placement and handling of flat regions. Finding the right parameter settings can be a challenging task.

Thresholding: https://youtu.be/DcWrbsPJEd8?si=jhQt-yc_wRwdjE4g

Region Based Segmentation: https://youtu.be/mn3nD3bEnO0?si=wuLNSUqUk5MpKc5b

OTSU Thresholding Algorithm Sum:

- Otsu's Example
- Find out threshold value using Otsu's method for the given 3 * 3 image.

154	138	163
75	74	151
70	106	118

-σ^2_B (k) is maximum for k=106
- Otsu threshold is 106
-106/255 = 0.4156

k	$P_1(k)$	m(k)	m_G	$(m_G*P_1-m)^2$	$P_1(1-P_1)$	σ^2_B (k)
70	0.111	7.7778	116.56	26.76	0.0986	271.39
74	0.222	16		98	0.1727	567.45
75	0.333	24.33		210.91	0.222	950.04
106	**0.4444**	**36.11**		**246.3**	**0.2469**	**997.56**
118	0.5556	49.22		241.33	0.2469	977.44
138	0.666	64.55		172.9	0.2223	777.77
151	0.7777	81.33		87	0.1728	503.47
154	0.8888	98.44		26.70	0.0988	270.24
163	1	116.55		0	0	0

CHAPTER 3:

Basic Idea About Edge Detection:

- Edges are significant local changes of intensity in a digital image. An edge can be defined as a set of connected pixels that forms a boundary between two disjoint regions. There are three types of edges:

Horizontal edges

Vertical edges

Diagonal edges

- Edge detection is a fundamental technique in image processing that aims to identify boundaries or transitions between different objects or regions in an image. The edges in an image represent significant changes in intensity, color, or texture and are crucial for various computer vision and image analysis applications

Edge Detection Operators are of two types:

Gradient – based operator which computes first-order derivations in a digital image like, Sobel operator, Prewitt operator, Robert operator

Gaussian – based operator which computes second-order derivations in a digital image like, Canny edge detector, Laplacian of Gaussian

Some Real-world Applications of Image Edge Detection:

medical imaging, study of anatomical structure

locate an object in satellite images

automatic traffic controlling systems
face recognition, and fingerprint recognition

Basic Overview of All edge Operators: https://www.geeksforgeeks.org/image-edge-detection-operators-in-digital-image-processing/
First order and second order edge operators: https://youtu.be/fhDBy-wV3ic
Canny's edge detection algorithm,: https://youtu.be/zCcj8onPrOE

Why do we Care about edge? / Goal Of edge Detection:
Edge detection is a crucial step in image processing and computer vision with several important applications. Here are some reasons why edge detection is performed:
Object Recognition: Edge detection helps identify the boundaries of objects within an image, making it a fundamental step in object recognition. Recognizing objects based on their shapes and contours relies on accurately detecting edges.
Image Segmentation:Edge detection plays a key role in image segmentation, where the goal is to partition an image into meaningful regions or objects. Edges provide natural boundaries for segmentation algorithms.
Feature Extraction: Edges often represent important features in an image. Extracting these features is essential for various computer vision tasks, such as pattern recognition, image matching, and analysis.
Image Analysis: Edge information is vital for understanding the structure and content of an image. Analyzing the arrangement and patterns of edges helps in interpreting the visual content.
Object Tracking: In video analysis and object tracking, detecting and tracking edges can help follow the movement of objects over time. Edges provide distinctive features for tracking algorithms.
Shape Analysis: Detecting edges is a fundamental step in shape analysis, allowing for the characterization of different shapes based on their contour information. This is particularly useful in medical imaging, industrial inspection, and robotics.
Image Enhancement: Edge detection is used in image enhancement to emphasize important features and structures. Enhancing edges can lead to clearer visual representation of objects in an image.
Computer Vision Applications: Many computer vision applications, such as autonomous vehicles, facial recognition, and augmented reality, rely on edge information for making intelligent decisions and interacting with the environment.

Hough transform for detecting lines and curves: https://youtu.be/t1GXMvK9m84
edge linking: https://users.cs.cf.ac.uk/dave/Vision_lecture/node30.html
Edge Linking in Simple Words:
Edge linking in image processing is like connecting the dots to make complete lines. Just like when you join the dots in a puzzle to see the whole picture, edge linking connects the edge points detected in an image to create meaningful edges and shapes.
Types of Edge Linking:
Local Edge Linking:
Local edge linking is like connecting nearby dots that seem related. It's as if you're linking dots that are close to each other to form smaller parts of a picture. This approach works well for objects with simple shapes.
Global Edge Linking:
Global edge linking takes a broader view. It's like connecting dots that might be far apart but still belong to the same larger object. This method considers the overall structure of the image and can handle more complex shapes and scenes.
In essence, edge linking is about completing edges and shapes by connecting the detected edge points. Local edge linking focuses on nearby points, while global edge linking takes into account the larger context of the image to form more complex shapes.

CHAPTER 4:

Image enhancement and its Two techniques: Image enhancement is like using special magic glasses to make a picture look better. Just as you might adjust the brightness, contrast, or colors of a photo to make it more pleasing to your eyes, image enhancement techniques do something similar but with the help of computers.
Types of Image Enhancement:
Spatial Domain Enhancement:
Imagine you have a painting, and you want to highlight certain areas with brighter or darker colors. In spatial domain enhancement, you directly change the pixel values of the image. It's like gently touching the painting with your magic brush to make it more captivating. This includes techniques like adjusting brightness, contrast, and applying filters to sharpen or blur the image.
Frequency Domain Enhancement:

Think of your image as a combination of different patterns and textures. Frequency domain enhancement works with these patterns. It's like using a special lens to see hidden details. Instead of directly changing pixel values, this technique transforms the image into a frequency representation. Then, you can enhance certain patterns and reduce others. This can help reveal hidden textures or remove unwanted noise.
In essence, image enhancement is all about making pictures clearer, more vibrant, or easier to understand.

SUM FROM PPT: https://drive.google.com/file/d/1OIgypf6XWCSfYrdlL5MxUWcP_FMYiypn/view

Logarithmic Transformation and Power- law Transformation: https://youtu.be/jfoXoFoDb48?si=PLaCddQ_h6x0i7At
Point operations in digital image processing with examples: https://youtu.be/FMDmXz6ynvk?si=8LINiN6yoZr_Y8aF

Image Enhancement vs Image Restoration:

Aspect	Image Enhancement	Image Restoration
Goal	Improve the visual quality of an image.	Restore an image to its original or better state.
Purpose	Enhance certain features, details, or appearance.	Repair and fix degraded or damaged images.
Approach	Directly modify pixel values or appearance.	Use various techniques to recover lost information.
Focus	Enhance specific visual aspects for better viewing.	Recover lost or damaged information or features.
Scope	Often subjective and user-dependent.	Can involve objective measurements and analysis.
Examples	Adjusting brightness, contrast, color balance.	Removing noise, fixing scratches, restoring colors.
Usage	Used to make images look more appealing or clearer.	Used when images have suffered from degradation.
Tools/Methods	Filters, histograms, spatial transformations.	Deconvolution, interpolation, noise reduction.
Result Impact	May change the original content to enhance it.	Aims to restore the original content accurately.

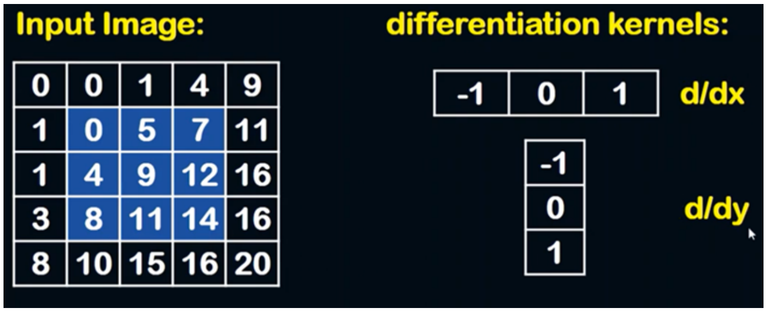
CHAPTER 5:

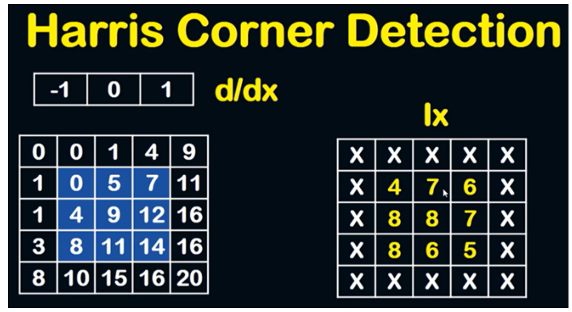
NOTES LINK: https://drive.google.com/file/d/16b-hmb-mvBir8SvQBeRJuKgl5P5aHaUD/view?usp=drive_link

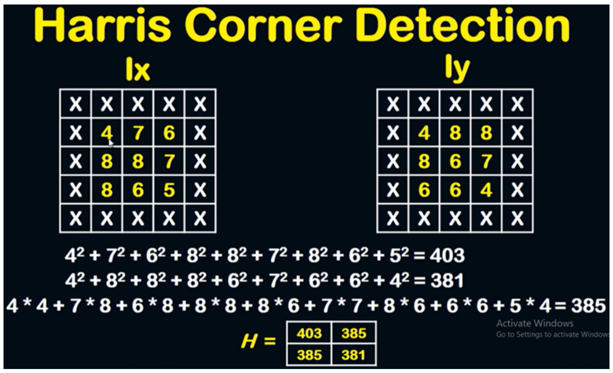
CHAPTER 6:

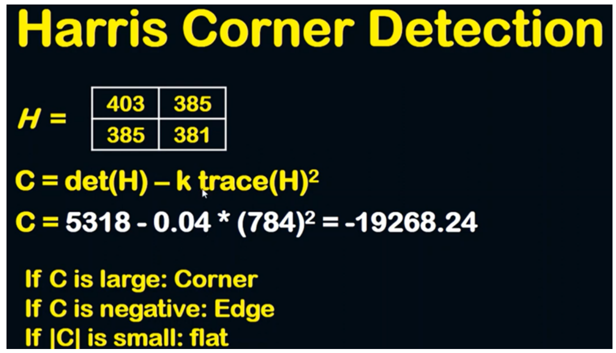
A feature is a piece of information which is relevant for solving the computational task related to a certain application. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighborhood operation or feature detection applied to the image.
One of the Main component
Detection: Identify the **Interest Point**
Interest point *or* **Feature Point** is the point which is expressive in texture. Interest point is the point at which the direction of the boundary of the object changes abruptly or intersection point between two or more edge segments.

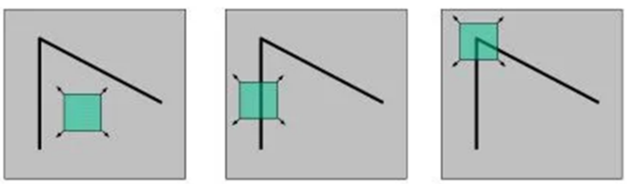
Harris Corners: https://youtu.be/vgSmxhXXF_4?si=SzrObv2lpcfYlIXl
Harris Corner detection algorithm was developed to identify the internal corners of an image. The corners of an image are basically identified as the regions in which there are variations in large intensity of the gradient in all possible dimensions and directions.
Corners are the important features in the image, and they are generally termed as interest points which are invariant to translation, rotation, and illumination.











"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

<u>Invariant feature point detector:</u>

SIFT: SIFT stands for Scale-Invariant Feature Transform

Q. Individual pixel color values are not an adequate feature to determine correspondences (why?).

One could try matching patches around the salient feature points – but these patches will themselves change if there is change in object pose or illumination.

• So these patches will lead to several false matches/correspondences.

Note :- SIFT provides features characterizing a salient point that remain invariant to changes in scale or rotation

SIFT is a feature extraction method that reduces the image content to a set of points used to detect similar patterns in other images. This algorithm is usually related to computer vision applications, including image matching and object detection.

<u>Steps :-</u>

<u>1. Feature Detection :-</u>

The image is convolved with Gaussian filters at different scales, and then the difference of successive Gaussian-blurred images are taken.

Keypoints are then taken as maxima/minima of the Difference of Gaussians (DoG) that occur at multiple scales.

This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate keypoint.

<u>2. Feature Description :-</u>

Create histogram of local gradient directions computed at selected scale. Assign canonical orientation at peak of smoothed histogram. Construct SIFT descriptor.

<u>3. Feature matching :-</u>

For each feature in image A, find nearest neighbor in image B. If the given feature matches than find an object in the scene.

SURF: The SURF method (Speeded Up Robust Features) is a fast and robust algorithm for local, similarity invariant representation and comparison of images.
The main interest of the SURF approach lies in its fast computation of operators using box filters
SURF is composed of two steps
1.) Feature Extraction :- The approach for interest point detection uses a very basic Hessian matrix approximation.
2.) Feature Description :- The creation of SURF descriptor takes place in two steps. The first step consists of fixing a reproducible orientation based on information from a circular region around the keypoint. Then, we construct a square region aligned to the selected orientation and extract the SURF descriptor from it.

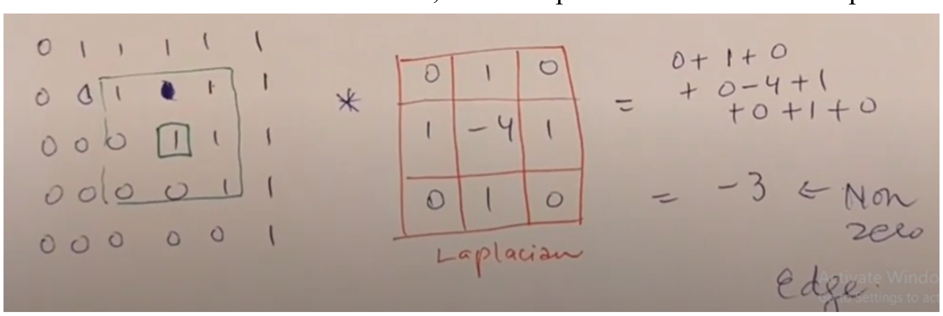
RANSAC for point matching:
Todo

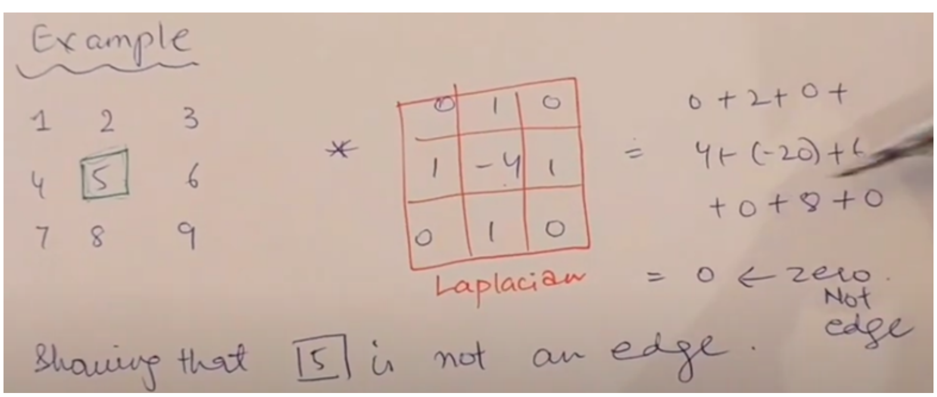
Edge detection:

*__Edge detection__ is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed *edges*.

*Edge detection is one of the fundamental steps in image processing, image analysis, image pattern recognition, and computer vision techniques.

1. **Laplacian of Gaussian (LoG):** It is a gaussian-based operator which uses the Laplacian to take the second derivative of an image. This really works well when the transition of the grey level seems to be abrupt. It works on the zero-crossing method i.e when the second-order derivative **crosses zero**, then that particular location corresponds to a maximum level. **It is called an edge location.**





Advantages:

1. Easy to detect edges and their various orientations
2. There is fixed characteristics in all directions

Limitations:

1. Very sensitive to noise
2. The localization error may be severe at curved edges
3. It generates noisy responses that do not correspond to edges, so-called "false edges"

2. Canny Operator :-

- It is a gaussian-based operator in detecting edges.
- This operator is not susceptible to noise.
- It extracts image features without affecting or altering the feature.

It detects edges based on three criteria:

1. Low error rate
2. Edge points must be accurately localized
3. There should be just one single edge response

Advantages:

1. It has good localization
2. It extract image features without altering the features
3. Less Sensitive to noise

Limitations:

1. There is false zero crossing
2. Complex computation and time consuming

3. Difference of Gaussian (DOG) :-

The Difference of Gaussian module is a filter that identifies edges.

- The DOG performs edge detection by performing a Gaussian blur on an image at a specified theta (also known as sigma or standard deviation).
- The resulting image is a blurred version of the source image.
- The module then performs another blur with a sharper theta that blurs the image less than previously.
- The final image is then calculated by replacing each pixel with the difference between the two blurred images and detecting when the values cross zero, i.e. negative becomes positive and vice versa. The resulting zero crossings will be focused at edges or areas of pixels that have some variation in their surrounding neighborhood.

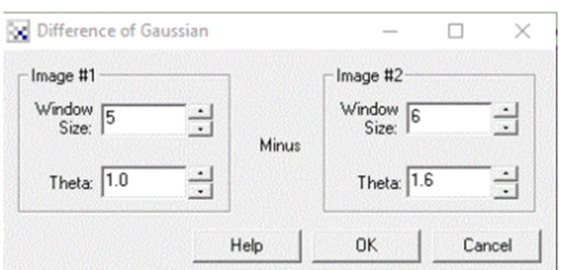
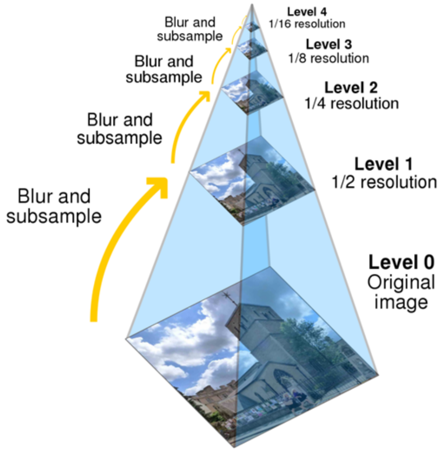


Image Pyramid

There are two kinds of Image Pyramids.

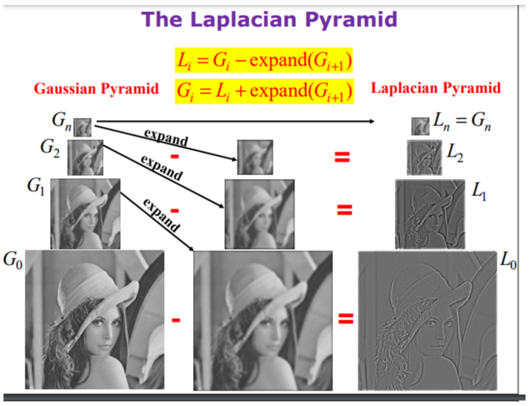
1) Gaussian Pyramid :- Gaussian Pyramid is a multi-scale representation of an image. It is created by applying a series of Gaussian filters to an original image to produce a set of images at different levels of resolution.

After applying gaussian filter , a M×N image becomes M/2×N/2 image.

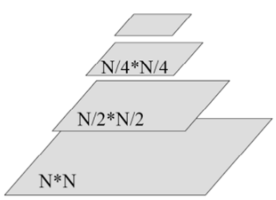


2) Laplacian Pyramids :- A Laplacian Pyramid is a multi-scale image representation that is derived from a Gaussian Pyramid. Laplacian Pyramids are formed from the Gaussian Pyramids.

A level in Laplacian Pyramid is formed by the difference between that level in Gaussian Pyramid and expanded version of its upper level in Gaussian Pyramid.



Space Required for Pyramids



$$N^2 + \frac{1}{4}N^2 + \frac{1}{16} + N^2 + \cdots = 1\frac{1}{3}N^2$$

<u>Line, circle & ellipse detectors (Hough Transform)</u>

The Hough transform in its simplest form is a method to **detect straight lines** but it can also be used to **detect circles or ellipses.**

The **Hough Transform** is a powerful tool in image processing and computer vision for detecting parametric shapes, and it can be adapted for circle and ellipse detection.

CHAPTER 7:

Classification models:

Image classification refers to a process in computer vision that can classify an image according to its visual content.

Structure for performing Image Classification.

- Image Pre-processing:
- Detection of an object:
- Feature extraction and training:
- Classification of the object:

Supervised classification uses classification algorithms and regression techniques to develop predictive models. The algorithms include linear regression, logistic regression, neural networks, decision tree, support vector machine, random forest, naive Bayes, and k-nearest neighbour.

Some of the most common algorithms used in unsupervised learning include cluster analysis, anomaly detection, neural networks, and approaches for learning latent variable models.
https://medium.com/analytics-vidhya/image-classification-techniques-83fd87011cac

Dimensionality Reduction (Principal component analysis)
Principal component analysis, or PCA, is a dimensionality reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

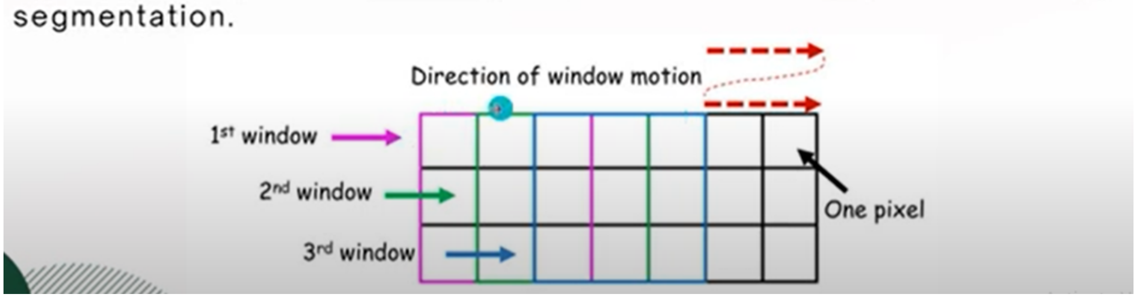
Face detection with sliding window

Here is an overview of face detection in terms of image processing and computer vision:

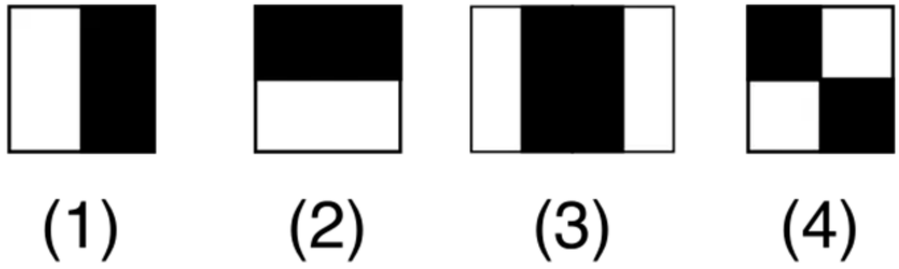
1. **Objective**: Detecting and locating human faces within digital images or video frames.
2. **Preprocessing**: Image enhancements like resizing and grayscale conversion may be applied.
3. **Feature Extraction**: Identify facial features or patterns (e.g., Haar-like, LBP, HOG) to capture distinctive face characteristics.
4. **Classifiers**: Use machine learning algorithms (SVM, AdaBoost, CNNs) to distinguish faces from non-faces.
5. **Sliding Window**: A window moves across the image at various scales and positions to identify candidate regions.
6. **Non-maximum Suppression**: Eliminate redundant detections to keep only one detection per face.

7. **Post-processing:** Refine results by filtering based on size, aspect ratio, or confidence scores. Perform face alignment if needed.
8. **Output:** Provide location (bounding box) and optional information like facial landmarks or detection confidence.
9. **Applications:** Used in real-time systems (e.g., facial recognition) and batch processing (e.g., image collections).
10. **Challenges:** Face variations (illumination, pose, expression, occlusion) require robust algorithms and large datasets.
11. **Ethical Concerns:** Address issues related to consent, privacy, surveillance, and data protection when using face detection technology.

In IPCV, the term "sliding window" refers to a technique used to analyze an image by systematically moving a small rectangular window or region-of-interest (ROI) across the image in a predefined pattern. The sliding window technique is used for various tasks, including object detection, feature extraction, and image segmentation.



The key aspect in face recognition is detecting relevant features in human face like eyes, eyebrows, nose, lips. So how do we detect these features in real time/in an image ? The answer is **Haar Wavelets** or **Haar Features** and the algorithm used is called as **Viola-Jones Algorithm**
Haar features are sequence of rescaled square shape functions.



Example of Haar Features: Square shaped kernels. Source

Each feature specifies a set of rectangles in the image window. A rectangle can be marked as white or black. The feature value is computed as the **difference between the sum of the pixel values in the white areas and the sum of pixel values in the black areas.**

Viola-Jones algorithm: Viola jones algorithm uses a 24x24 window as the base window size to start evaluating these features in any given image.

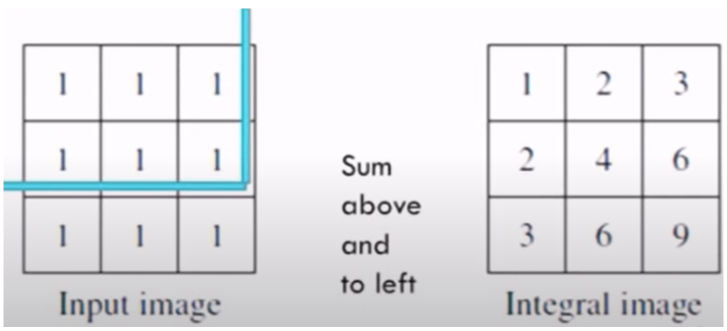
The algorithm has four stages:

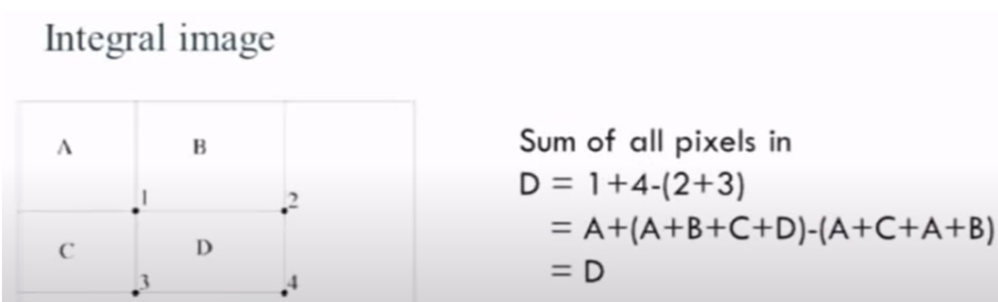
<u>1. Haar-like features selection :-</u>

All the possible parameters like position, scale and type we end up with many Haar Features over here but only few of them will be useful among all the features.

<u>2. Creating an integral image :-</u>

The integral image is a data structure for efficiently computing the sum of pixel values in a rectangular image window.





<u>3. Adaboost training :-</u>

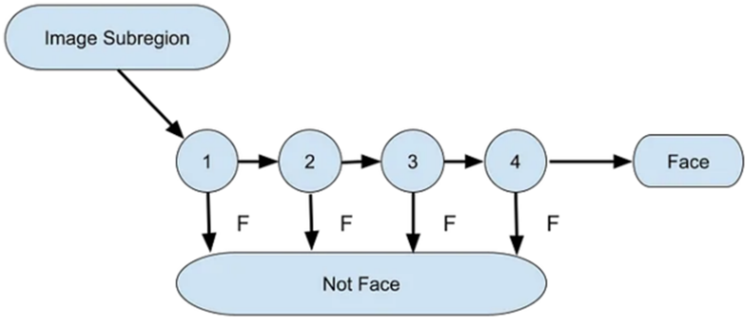
As discussed only few set of features will useful

Adaboost is a machine learning algorithm which helps in finding only the best features among all these features. After these features are found a weighted combination of all these features in used in evaluating and deciding any given window has a face or not. Each of the selected features are considered okay to be included if they can atleast perform better than random guessing (detects more than half the cases).

<u>4. Cascading classifiers :-</u>

Organize the weak classifiers into a cascade structure. The cascade consists of multiple stages, where each stage has a set of weak classifiers. The stages are ordered in such a way that simple and fast classifiers come first, and more complex ones come later.

Strong features are formed into a binary classifier. : Positive matches are sent along to the next feature. Negative matches are rejected and exit computation.



Bag of visual words

Bag Of Visual Words(also known as Bag Of Features) is a technique to compactly describe images and compute similarities between images. It is used for image classification.

- Feature Extraction: Extract local features (e.g., SIFT, SURF, ORB) from images.
- Feature Vector Quantization: Create a visual vocabulary by clustering these features into visual words.
- Feature Encoding: Encode local features in each image using the visual vocabulary.
- Histogram Creation: Create histograms based on the encoded features for each image.
- Image Classification or Retrieval: Use these histograms as feature vectors for tasks like image classification or retrieval.

People detection with sliding window:

People detection using a sliding window approach with an SVM (Support Vector Machine) classifier is a common technique in computer vision for object detection tasks, including pedestrian detection. This method involves systematically scanning an image with a sliding window of varying sizes and aspect ratios to identify potential regions of interest (ROIs) where people might be present. The SVM classifier is then used to classify each of these ROIs as either "person" or "non-person" based on a set of pre-trained features.

Here's a step-by-step guide on how you can implement people detection using a sliding window and an SVM classifier:

1. Feature Extraction: Extract features from the images within each sliding window. Common feature extraction methods include Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP). These features should capture the relevant characteristics of pedestrians.
2. Positive and Negative Samples: Create a dataset of positive samples (windows containing people) and negative samples (windows without people) using the annotated data from your dataset.
3. Training the SVM: Train an SVM classifier using the positive and negative samples. You can use libraries like OpenCV, scikit-learn, or specialized libraries like Dlib for this purpose. The SVM should learn to distinguish between features representing pedestrians and those that do not.
4. Sliding Window: Slide a window of various sizes and aspect ratios across the input image in a systematic grid pattern. For each position and scale of the window, extract the same features used during training.
5. Classification: Pass each extracted feature vector through the trained SVM classifier. The output will indicate whether the window contains a person or not.

CHAPTER:8:

CNN: https://drive.google.com/file/d/1rsB8pz8mT7DTOEDJUfjQHLOJeow7-Xau/view?usp=sharing

QUESTION BANK:

CHAPTER:1:

Q1. Consider a 3-bit gray scale image with dimension 128 × 128.
What will be possible range of values comprised by pixels in this image?
Ans :- A 3-bit grayscale image can represent 2^3 = 8 different shades of gray. The possible values for each pixel in the image will range from 0 to 7.
So, in a 3-bit grayscale image with dimensions of 128x128, each pixel can have a value between 0 (black) and 7 (white), with 6 intermediate shades of gray in between.

Q2. Consider a 16-bit gray scale image with dimension 10 × 10.
What will be possible range of values comprised by pixels in this image?
Ans :- A 16-bit grayscale image can represent 2^16 = 65,536 different shades of gray. In a 10x10 image, each of the 100 pixels can have a value ranging from 0 to 65,535.
So, in a 16-bit grayscale image with dimensions of 10x10, each pixel can have a value between 0 (black) and 65,535 (white), with 65,534 intermediate shades of gray in between.

Q3. Consider following image:

10	05	03	07	09
06	14	15	07	00
06	14	15	15	00
09	09	11	11	00
07	07	10	10	01

I. What will be the range of values in its X-axis?

Ans :- The range of values along the X-axis can be determined by considering the minimum and maximum values from all the rows. So, the range for the X-axis is from 0 to 15.

II. How many total bits are required to represent the given image in binary? (Consider uncompressed image)

Ans :- The image you provided is a 5x5 grid of values, and each value is given in decimal notation. To represent these values in binary, you need to consider that the range of values is from 0 to 15. Since the maximum value that needs to be represented is 15, you need 4 bits to represent each pixel in binary (since 2^4 = 16, which can represent values from 0 to 15).

So, for each pixel, you would need 4 bits. Since there are 25 pixels in the 5x5 image, you would require a total of 4 bits/pixel * 25 pixels = 100 bits to represent the entire image in binary.

Q4. Discuss image acquisition using i. A single sensor ii. A line sensor and iii. Array sensor.
- Image Acquisition is the first step in any image processing system.
- The general aim of any image acquisition is to transform an optical image (real-world data) into an array of numerical data which could be later manipulated on a computer.
- Image acquisition is achieved by suitable cameras.
- Now the incoming energy is transformed into a voltage by the combination of input electrical power and sensor material of the camera.
- Image Acquisition Sensors: 1) Single Sensor 2) Line Sensor 3) Array Sensor

1) <u>Image Acquisition using Single Sensor:</u>

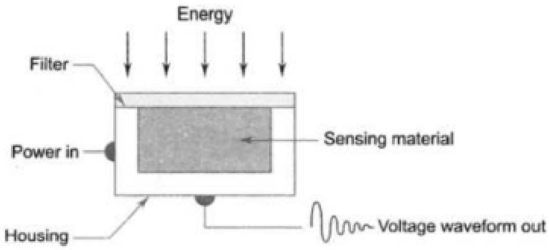


Fig: Single image sensor

An Example of a single sensor is a photodiode. Now to obtain a two-dimensional image using a single sensor, the motion should be in both x and y direction.
- Rotation provides motion in one direction.
- Linear motion provides motion in the perpendicular direction

This is an in expensive method and we can obtain high-resolution images with high precision control.

2) <u>Image Acquisition using a line sensor:</u>
- The sensor strip provides imaging in one direction.
- Motion perpendicular to the strip provides imaging in other direction.
- This type of arrangement is used in most of the flat bed sensors.

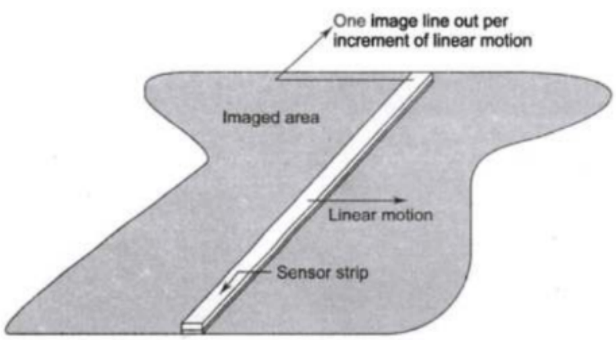


Fig: Linear sensor strip

3) <u>Image Acquisition using an array sensor:</u>
- In this, individual sensors are arranged in the form of 2-D array.
- This type of arrangement is found in digital cameras CCD array.
- In this, the response of each sensor is proportional to the integral of the light energy projected onto the surface of the sensor.
- Noise reduction is achieved by letting the sensor integrate the input light signal over minutes or ever hours.

Advantage: Since sensor array is 2D, a complete image can be obtained by focusing the energy pattern onto the surface of the array.

Q5. Explain image acquisition process with sampling and quantization steps.

https://www.codingninjas.com/studio/library/image-sampling-and-quantization

Q6. Define Spatial resolution.

Spatial resolution states that the clarity of an image cannot be determined by the pixel resolution. The number of pixels in an image does not matter.

Spatial resolution in image processing refers to the level of detail or granularity in an image, specifically how finely the image can represent the details of the objects

In other words, spatial resolution quantifies how clear and sharp the image appears

Higher spatial resolution means that the image contains more fine details and can represent smaller objects or features. Lower spatial resolution, on the other hand, implies that the image may be more pixelated and less capable of representing fine details.

In digital imaging, spatial resolution is often measured in pixels per inch (PPI) or dots per inch (DPI) or Lines per inch(LPI) and is an important consideration when determining the quality and clarity of images, especially in fields like photography, medical imaging, remote sensing, and computer vision.

Measuring spatial resolution:

Since the spatial resolution refers to clarity, so for different devices, different measure has been made to measure it.

1) Dots per inch: Dots per inch or DPI is usually used in monitors.
2) Lines per inch: Lines per inch or LPI is usually used in laser printers.
3) Pixel per inch: Pixel per inch or PPI is measure for different devices such as tablets , Mobile phones e.t.c.

Q7. Define Intensity resolution. https://youtu.be/DOYRMHAST4U?si=0VFMINuW2V4Q6HRm

Intensity resolution in image processing refers to the ability of an imaging system to differentiate between different levels of intensity or brightness within an image. It is a measure of how finely the system can represent variations in the intensity of light or color across the image. In simpler terms, intensity resolution quantifies the system's ability to distinguish between different shades, tones, or colors in an image.

Higher intensity resolution means that the imaging system can represent a larger number of distinct intensity levels, resulting in more subtle and precise variations in brightness or color within the image. Lower intensity resolution may lead to banding or loss of detail, especially in images with smooth gradients or fine variations in intensity or color.

Q8. Comment on how much spatial resolution and intensity resolution is required for at least four different application scenarios.

Spatial and intensity resolution requirements vary depending on the specific application scenario and the level of detail or precision required. Here are four different application scenarios in image processing, along with their typical spatial and intensity resolution needs:

1. Medical Imaging:

Spatial Resolution: In medical imaging, especially for applications like X-rays, CT scans, and MRIs, high spatial resolution is crucial. Physicians need to see fine details in anatomical structures and abnormalities. The spatial resolution may need to be in the order of micrometers.

Intensity Resolution: Intensity resolution must be high to differentiate subtle differences in tissue density or contrast agents. Medical images often require high bit-depth (e.g., 12 to 16 bits) to represent a wide range of intensity levels.

2. Satellite and Aerial Imagery:

Spatial Resolution: In remote sensing, the spatial resolution depends on the intended use. High-resolution satellite images used for urban planning and land use require sub-meter or even sub-decimeter spatial resolution. Lower-resolution images may suffice for larger-scale environmental monitoring.

Intensity Resolution: The intensity resolution should be sufficient to distinguish between various land cover types, which can range from vegetation to buildings. This typically requires 8 to 16 bits of intensity resolution.

3. Digital Photography:

Spatial Resolution: For general digital photography, a spatial resolution of 12 to 24 megapixels (or more) is common. Higher resolutions are useful for professional and artistic photography.

Intensity Resolution: Photography often uses 8-bit or 12-bit color channels to capture a wide range of colors and details in images.

4. Security and Surveillance:

Spatial Resolution: Surveillance cameras used for security applications might have varying spatial resolution needs. In some cases, high-resolution cameras are used to capture facial details and license plate numbers, while lower resolutions may be sufficient for monitoring larger areas.

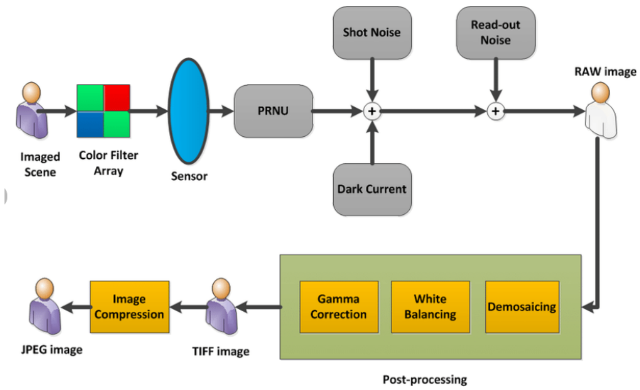
Intensity Resolution: Intensity resolution should be adequate to recognize objects and individuals, which often means using 8-bit or 12-bit grayscale or color images.

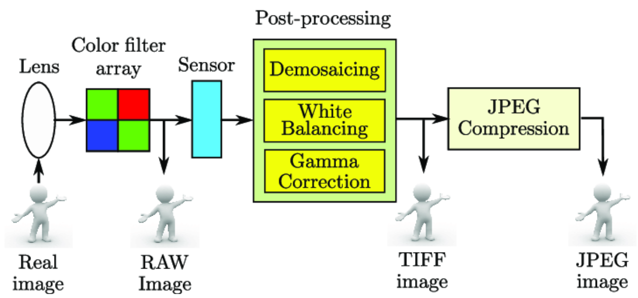
In summary, the spatial and intensity resolution requirements in image processing are highly application-specific. They depend on the nature of the objects or features being observed, the desired level of detail.

Q9. Sketch digital coordinate system from world coordinates to pixel coordinates.

Todo

Q10. Illustrate digital camera pipeline and comment on simple model vs sophisticated model digital camera effects.





Q11. Compare CCD vs CMOS camera sensor.

- CCD sensors create high-quality, low-noise images. CMOS sensors are generally more susceptible to noise.
- CMOS sensors traditionally consume little power. CCDs, on the other hand, use a process that consumes lots of power
- CCDs consume as much as 100 times more power than an equivalent CMOS sensor.
- CCD sensors have been mass produced for a longer period of time, so they are more mature. They tend to have higher quality pixels, and more of them.
- In many consumer and portable devices, CMOS sensors are now the more common choice due to their lower power consumption and cost-effectiveness.

Feature	CCD Sensor	CMOS Sensor
Full Form	Charge-Coupled Device	Complementary Metal-Oxide-Semiconductor
Working Principle	Electrons are transferred through a sequence of capacitors.	Photodiodes convert light into voltage; readout circuits amplify and convert voltage to digital data.
Power Consumption	Typically consumes more power, especially during readout.	Generally consumes less power, making them more energy-efficient.
Speed and Readout	Slower readout due to the serial transfer of charge.	Faster readout due to parallel readout capabilities.
Noise Levels	Typically lower noise levels, especially in low-light conditions.	May have higher noise levels, particularly in older CMOS designs, but newer designs have improved noise characteristics.
Image Quality	Generally provides high-quality images with good dynamic range.	Improved over the years and can also offer high-quality images and good dynamic range.
Sensitivity to Light	Can be more sensitive to light, making them suitable for low-light conditions.	May have lower sensitivity to light, but newer designs have improved low-light performance.
Cost	Traditionally more expensive to manufacture.	Typically less expensive to manufacture, leading to cost-effective camera solutions.
Integration with Other Electronics	May require additional components for signal processing.	Can integrate signal processing within the same chip, leading to compact camera systems.
Rolling Shutter vs. Global Shutter	Often has a global shutter, reducing motion artifacts.	Can have both global and rolling shutter modes, depending on the design.
Image Sensor Use	Historically used in high-end applications such as professional photography and scientific imaging.	Widely used in consumer electronics, mobile phones, webcams, and many other applications.

Q12. Which factors affect the performance of digital cameras and how?

1) Shutter Speed: Shutter speed determines how long the camera's sensor is exposed to light. Faster shutter speeds freeze motion and reduce motion blur, while slower speeds capture more light and create a sense of motion.

Impact: Shutter speed affects the ability to capture fast-moving subjects, avoid motion blur, and control exposure. In image processing, it influences the sharpness and clarity of images, especially in dynamic scenes.

2) Sampling Pitch: Sampling pitch refers to the distance between individual pixels on the camera sensor. Smaller pitch values lead to higher spatial resolution.

Impact: Smaller sampling pitch allows for capturing more detail in the image. It affects the level of detail, sharpness, and the ability to resolve fine structures. In image processing, higher spatial resolution provides more data for processing and analysis.

3) Chip Size: The physical size of the camera sensor (chip) affects its light-gathering ability. Larger sensors can capture more light and perform better in low-light conditions.

Impact: Chip size influences image quality, particularly in low-light situations, and determines the depth of field and field of view. In image processing, it affects the level of noise and dynamic range in images.

4) Analog Gain: Analog gain amplifies the electrical signal from the sensor to increase sensitivity to light. It can introduce noise as well.

Impact: Analog gain is used to enhance the camera's performance in low-light conditions. However, excessive gain can lead to increased sensor noise. In image processing, the choice of gain affects image quality and noise levels.

5) Sensor Noise: Sensor noise includes various sources of unwanted variations in pixel values, such as thermal noise, readout noise, and electronic noise.

Impact: Reducing sensor noise is crucial for image quality. Lower noise levels result in cleaner images with better detail and color accuracy. In image processing, noise reduction techniques are applied to improve image quality.

6) Analog to Digital Converter (ADC): The ADC converts the analog signal from the sensor into a digital format. The bit depth of the ADC determines the number of possible intensity levels.

Impact: A higher bit-depth ADC provides finer intensity resolution, allowing for more accurate representation of image details. In image processing, a greater bit depth offers more flexibility and accuracy during post-processing.

Q13. Distinguish color image and gray scale image with examples.

Color Image:

Representation: Color images represent a wide spectrum of colors and can contain multiple color channels. Common color spaces include RGB (Red, Green, Blue) and CMYK (Cyan, Magenta, Yellow, Key/Black).

Pixel Information: Each pixel in a color image typically consists of multiple color channels (e.g., RGB) that combine to produce a specific color. For example, a pixel in an RGB image can have values like (255, 0, 0) to represent pure red.

Examples: A photograph of a landscape with a blue sky, green trees, and various other colors.

A digital painting with vibrant colors and intricate details.

An image of a colorful product catalog.

Common Uses: Used in photography, web images, multimedia, computer graphics, and most visual content.

Grayscale Image:

Representation: Grayscale images represent visual information using shades of gray, ranging from black to white. They contain a single intensity channel.

Pixel Information: In a grayscale image, each pixel has only one value that represents its brightness or intensity. A value of 0 is black, while a value of 255 is white in an 8-bit grayscale image.

Examples: A black-and-white photograph from the early 20th century with various shades of gray.

A medical X-ray image displaying structures in different shades of gray.

A handwritten document or a page from a book, which is typically in black and white.

Common Uses: Commonly used in medical imaging, document processing, and certain image analysis tasks where color is unnecessary.

Q14. Enlist common image file formats.

JPEG/JFIF: JPEG (Joint Photographic Experts Group) is a lossy compression method in which data is lost to reduce the size of the image. Due to compression, some data is lost but that loss is very less. It is a very common format and is good for digital cameras, nonprofessional prints, E-Mail, Powerpoint, etc., making it ideal for web use.

- JPEG-compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format.
- The JPEG/JFIF filename extension is JPG or JPEG. Nearly every digital camera can save images in the JPEG/JFIF format, which supports eight-bit grayscale images and 24-bit color images (eight bits each for red, green, and blue).

- (JPEG also provides lossless image storage, but the lossless version is not widely supported.)
TIFF: Tagged Image File Format this format store image data without losing any data. It does not perform any compression on images, and a high-quality image is obtained but the size of the image is also large, which is good for printing, and professional printing.
- It usually uses either the TIFF or TIF filename extension.
- The TIFF (Tagged Image File Format) format is a flexible format that normally saves eight bits or sixteen bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively,
GIF: GIF or Graphics Interchange Format files are used for web graphics. They can be animated and are limited to only 256 colors, which can allow for transparency. GIF files are typically small in size and are portable.
- filename extension is .gif.
- as it uses lossless compression, which is more effective when large areas have a single color, and less effective for photographic images
Bitmap: Bit Map Image file is developed by Microsoft for windows. Typically, BMP files are uncompressed, and therefore large and lossless; their advantage is their simple structure and wide acceptance in Windows programs.
PNG: PNG or Portable Network Graphics files are a lossless image format. It was designed to replace gif format as gif supported 256 colors unlike PNG which support 16 million colors. The PNG file format supports eight-bit images and 24-bit true color (16 million colors) or 48-bit true color with and without alpha channel - while GIF supports only 256 colors and a single transparent color.
- PNG is designed to work well in online viewing applications like web browsers and can be fully streamed with a progressive display option. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors. Animated formats derived from PNG are MNG and APNG. This supported by Mozilla Firefox and Opera and is backwards compatible with PNG.
- file extension is .png
WebP: WebP is a new open image format that uses both lossless and lossy compression. It was designed by Google to reduce image file size to speed up web page loading
Exif: The Exif (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information.
PPM, PGM, PBM, and PNM: Netpbm format is a family including the portable pixmap file format (PPM), the portable graymap file format (PGM) and the portable bitmap file format (PBM). These are either pure ASCII files or raw binary files with an ASCII header that provide very basic functionality and serve as a lowest common denominator for converting pixmap, graymap, or bitmap files between different platforms. Several applications refer to them collectively as PNM (Portable aNy Map).

Type	Extension	Colors
Portable BitMap	.pbm	0–1 (white & black)
Portable GrayMap	.pgm	0–255 (gray scale)
Portable PixMap	.ppm	0–255 (RGB)

Q15. If an image represented by the following matrix is scaled up by a factor of 2 in both X and Y direction, what will be the scaled image with
i. Nearest neighbor interpolation
ii. Bilinear interpolation

3	3	3	3	3
3	5	5	5	3
3	5	7	5	3
3	5	5	5	3
3	3	3	3	3

Ans: Method: https://youtu.be/dIlR8vS9T-c?si=yoA6Fv-KcuGSJV8K

<u>Using Nearest Neighbour interpolation</u>

Step 1 :- Duplicate the columns

3	3	3	3	3	3	3	3	3	3
3	3	5	5	5	5	5	5	3	3
3	3	5	5	7	7	5	5	3	3
3	3	5	5	5	5	5	5	3	3
3	3	3	3	3	3	3	3	3	3

Step 2 :- Duplicate the rows

3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	5	5	5	5	5	5	3	3
3	3	5	5	5	5	5	5	3	3
3	3	5	5	7	7	5	5	3	3
3	3	5	5	7	7	5	5	3	3
3	3	5	5	5	5	5	5	3	3
3	3	5	5	5	5	5	5	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3

Hence the scaled image using Bilinear interpolation interpolation is 10x10

<u>Using Bilinear interpolation</u>

Bilinear interpolation takes a weighted average of the four nearest neighboring pixels to calculate the new pixel values.

3	2.4	3	2.5	3	2.5	3	2.4	3	
2.4	3.5	3.9	4	4	4	3.9	3.5	2.4	
3	3.9	5	4.9	5	4.9	5	3.9	3	
2.5	4	4.9	5.5	5.8	5.5	4.9	4	2.5	
3	4	5	5.8	7	5.8	5	4	3	
2.5	4	4.9	5.5	5.8	5.5	4.9	4	2.5	
3	3.9	5	4.9	5	4.9	5	3.9	3	
2.4	3.5	3.9	4	4	4	3.9	3.5	2.4	
3	2.4	3	2.5	3	2.5	3	2.4	3	

Hence the scaled image using Bilinear interpolation is 10x10

Q16. Consider the following image comprising 512 rows and 512 columns, How many total bits are required to represent the given image in binary?(Consider uncompressed image)

Ans :- TODO

Q17. Which of the following image file formats is most suitable for cartoon style image?

(A) JPEG (B) TIFF (C) GIF (D) Exif

Ans :- GIF (Graphics Interchange Format)

Q18. Which if the following image file formats is uncompressed lossless image?

(A) JPEG (B) TIFF (C) BMP (E) Exif

Ans :- BMP (Bitmap Image)

Q19. How PNG is different from GIF?

GIF	PNG
It stands for graphics interchange format.	It stands for portable network graphics.
It supports animations.	It doesn't support animations.
MIME type is image/gif.	MIME type is image/png.
The file size is generally less.	The file size is large as compared to a GIF.
It supports one bit transparency in the images.	It supports transparency with an elegance.
The extensions used are .gif and .gfa.	The extension used is .png.
Mostly used when animation is needed.	It is mostly used in image creation.
It provides a limited color range of 256 colors.	It provides thousand of colors.
It supports layers and multi-paging.	While it does not supports layers and multi-paging.
It is best suited for screenshots, typography, etc.	It is suitable for small vectors, crispy edged visuals, logos, etc.

Q20. Which of the following format is used by PCB softwares?

(A) CGM (B) Gerber format (C) SVG (D) ESP

Ans :- **Gerber format** is the de facto standard format used by printed circuit board or PCB software.

Q21. Which of the following formats is a compound format?

(B) CGM (B) Gerber format (C) SVG (D) ESP

Ans :- EPS (Encapsulated PostScript)

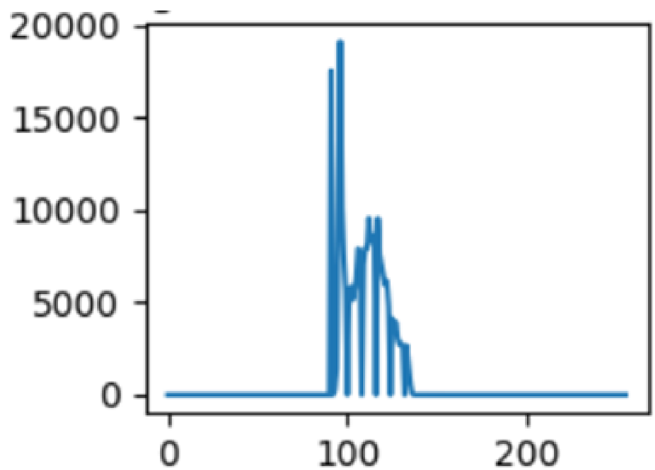
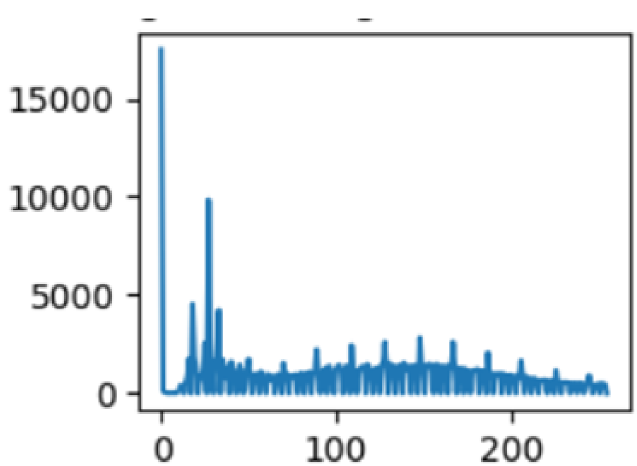
Q22. Contrast bitmap images with vector images.

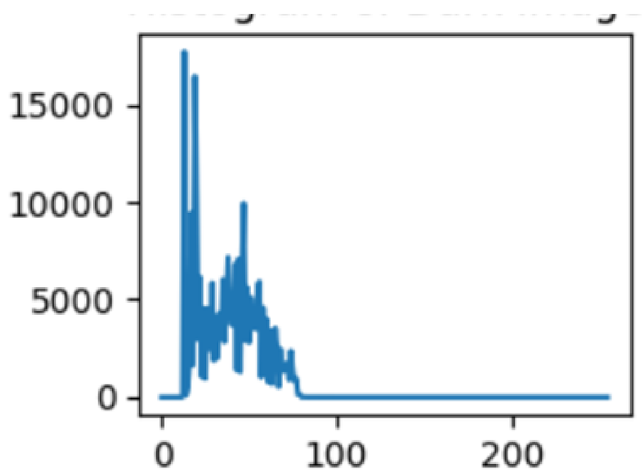
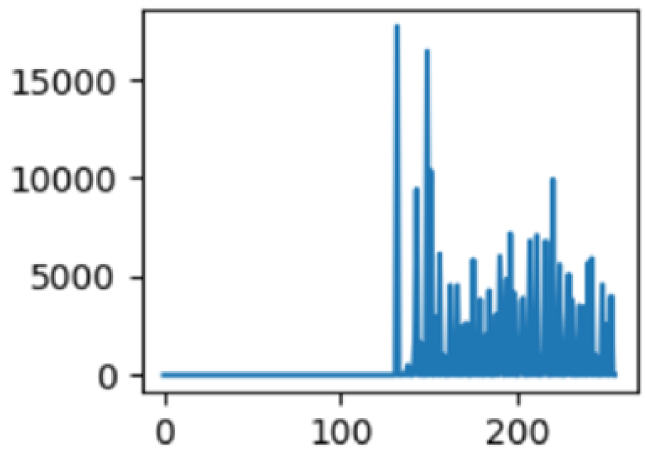
1) - Bitmap (or raster) images are stored as a series of tiny dots called pixels. Each pixel is actually a very small square that is assigned a color, and then arranged in a pattern to form the image. When you zoom in on a bitmap image you can see the individual pixels that make up that image. Bitmap graphics can be edited by erasing or changing the color of individual pixels using a program such as Adobe Photoshop.
- Unlike bitmaps, vector images are not based on pixel patterns, but use mathematical formulas to draw lines and curves that can be combined to create an image from geometric objects such as circles and polygons. Vector images are edited by manipulating the lines and curves that make up the image using a program/software such as Adobe Illustrator.
2) Vector images tend to be smaller than bitmap images. That's because a bitmap image has to store color information for each individual pixel that forms the image. A vector image just has to store the mathematical formulas that make up the image, which take up less space.
3) Vector images are also more scalable than bitmap images. Because When a vector image is scaled up, the image is redrawn using the mathematical formula. The resulting image is just as smooth as the original. Where bitmap might result in blurry sometimes when we scale up.
4) Unfortunately, vector formats are not well supported on the web. The two most popular image formats used on the Web, GIF and JPEG are bitmap formats. Most vector images must first be converted into bitmaps images (or rasterized) before they can be used on the Web.

5) Bitmap formats are best for images that need to have a wide range of color gradations, such as most photographs. Vector formats, on the other hand, are better for images that consist of a few areas of solid color. Examples of images that are well suited for the vector format include logos and type.
6) Raster graphics - Image formats: BMP, GIF, JPEG, PNG
Vector graphics - Image formats: Flash, Scalable vector graphics (SVG), CDR (corelDraw), AI (Adobe Illustrator)

CHAPTER:2

Q1. Match the following with reference to histograms of images:

1	 **Ans: Low contrast image**	**A**	**Dark image**
2	 **Ans: High contrast image**	**B**	**Light image**

3	 **Ans: Dark image**	**C**	**Low contrast image**
4	 **Ans: Light image**	**D**	**High contrast image**

Q2. Define histogram of an image. https://youtu.be/RjmmyRrmVRQ?si=_bR5tAW1FUB8TDX6

Histogram

- It indicates dynamic range of the image i.e. how well the image is spread out.
- It is a plot of the number of occurrences of gray levels in the image against the gray level values.
- It provides a convenient summary of the intensities in the image, but it is unable to convey any information regarding spatial relationships between pixels.
- It provides more insight about image contrast and brightness.
  - Dark image: clustered towards lower gray level.
  - Bright image: clustered towards higher gray level.
  - Low contrast image: not spread equally, (narrow)
  - High contrast image: equal spread
- Image can be improved by modifying the histogram of the image.

Q3. For following transformation functions given in image i. and ii., Comment on whether these functions are suitable to perform histogram equalization with justification.

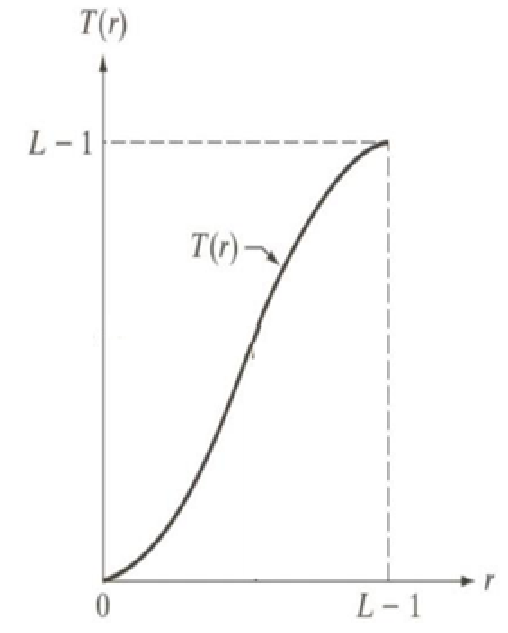
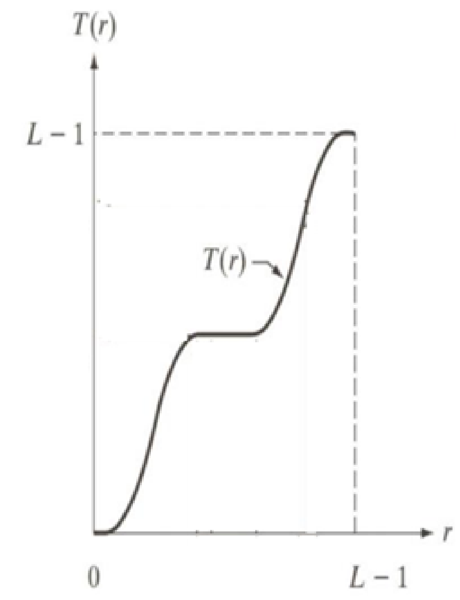
	

Image i.	Image ii.

Q4. Perform Histogram Equalization on following image:

4	4	4	4	4
3	4	5	4	3
3	5	5	5	3
3	4	5	4	3
4	4	4	4	4

Ans :- Method: https://youtu.be/Yd6QISby8kk?si=4hw4A69hdXRtddQb

Histogram equalization

n bit image, L=2^n= 8,			MXN image= 25			
Gray level (r_k)	No. of pixels (n_k)	CDF(Running sum) $CDF = \sum_{j=0}^{k} nj$ (k=0,1,2 ....L-1)	$P_r(rk)$ $PDF = \frac{nk}{MN}$	$S_k = \frac{CDF*(L-1)}{MN}$	rounded s_k	$P_s(sk)$
0	0	0	0/25	0	0	0
1	0	0	0/25	0	0	
2	0	0	0/25	0	0	
3	6	6	6/25	1.68	2	6/25
4	14	20	14/25	5.6	6	14/25
5	5	25	5/25	7	7	5/25
6	0	25	0/25	7	7	
7	0	25	0/25	7	7	

Q5. Perform Histogram Equalization on image with following details:

Gray Level	Number of Pixels
0	850
1	790
2	1023
3	656
4	122
5	81
6	245
7	329

Ans :- Method: https://youtu.be/Yd6QISby8kk?si=4hw4A69hdXRtddQb

Q.5 Histogram Equalization

Gray levels	0	1	2	3	4	5	6	7
No. of pixels	850	790	1023	656	122	81	245	329

$n = 4096$

Gray levels	No. of pixels (n_K)	(PDF) $r_K = n_K/n$	S_K (CDK)	$S_K \times 7$	Histogram Equalization
0	850	0.2075	0.2075	1.452	1
1	790	0.1929	0.4004	2.802	3
2	1023	0.2498	0.6502	4.55	5
3	656	0.1602	0.8104	5.67	6
4	122	0.0298	0.8402	5.88	6
5	81	0.0198	0.8600	6.02	6
6	245	0.0598	0.9198	6.44	6
7	329	0.0803	1.0001	7	7

$n = 4096$

Gray levels	0	1	2	3	4	5	6	7
No. of pixels	0	850	0	790	0	1023	1104	329

Q6. Perform Histogram Matching on image with following details:

Gray Level	Number of Pixels
0	790
1	1023
2	850
3	656
4	329
5	245
6	122
7	81

As per Target image with following details:

Gray Level	Number of Pixels
0	0
1	0
2	0
3	614
4	819
5	1230
6	819
7	614

ANS: Method: https://youtu.be/r565euxWZBs?si=lwFGwq8FhKnst7No

Gray level (r_k)	Number of pixels (n_k)	CDF	$P_r(r_k)$	S_k	rounded S_k	$P_s(s_k)$
0	790	790	0.193	1.35	1	0.193
1	1023	1813	0.25	3.09	3	0.25
2	850	2663	0.208	4.55	5	0.208
3	656	3319	0.160	5.67	6	0.240
4	329	3648	0.080	6.23	6	
5	245	3893	0.06	6.65	7	0.108
6	122	4015	0.029	6.86	7	
7	81	4096	0.019	7	7	

Z_q	n_q	$P_z(z_q)$	CDF	V_q	Rounded V_q	Z_k	$Pz(z_k)$
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	614	0.15	614	1.04	1	3	0.19
4	819	0.20	1433	2.44	2	4	0.25
5	1230	0.30	2663	4.55	5	5	0.21
6	819	0.20	3482	5.95	6	6	0.24
7	614	0.15	4096	7	7	7	0.11

s_k	z_q	$Ps(s_k)$	$Pz(z_k)$
1	3	0.19	0.19
3	4	0.25	0.25
5	5	0.21	0.21
6	6	0.24	0.24
7	7	0.11	0.11

Q7. Enlist limitations of Global histograms.

Loss of Local Information: Global histograms treat the entire image as a single entity and do not take into account local variations in intensity. This can lead to a loss of important local information, especially in images with varying lighting conditions or complex textures.

Limited Adaptability: Global histograms do not adapt to changes in lighting or scene conditions.

Difficulty with Multimodal Distributions: Global histograms struggle with images that have multiple peaks or modes in their intensity distribution. They may not accurately represent such images and can lead to poor results in applications like thresholding or image segmentation.

Sensitivity to Outliers: Outliers, such as noise or extreme pixel values, can have a significant impact on the global histogram. They can skew the histogram and affect the performance of image processing algorithms that rely on it.

Computational Overhead: Computing a global histogram for a large image can be computationally expensive, particularly in real-time applications. For video processing, the overhead of repeatedly computing a global histogram for each frame may not be feasible.

Q8. Compare thresholding of image using Global histograms and local histograms.

Thresholding is a fundamental image processing technique used to segment an image into regions or objects of interest based on pixel intensity values. Global histogram thresholding and local histogram thresholding are two different approaches to perform this task, and they have distinct characteristics and use cases. Let's compare the two:

Global Histogram Thresholding:

Overview: Global histogram thresholding computes a single threshold value for the entire image based on its global intensity histogram.
Simplicity: It is a simple and computationally efficient method.
Advantages: It works well when the foreground and background have distinct and consistent intensity values throughout the entire image.
- Suitable for images with uniform lighting and contrast.
Limitations: It is not robust when the image has varying lighting conditions, shadows, or complex textures.
- It may not perform well in the presence of outliers or non-uniform backgrounds.
- Cannot handle images with multimodal intensity distributions or regions with varying contrasts.

Local Histogram Thresholding:

Overview: Local histogram thresholding, or adaptive thresholding, divides the image into smaller local regions and computes a threshold for each region based on its local intensity histogram.
Complexity: It is more computationally intensive than global thresholding due to the need to process multiple local histograms.
Advantages: It is robust to varying lighting conditions and is suitable for images with non-uniform illumination.
- Effective in handling images with varying contrast and textures.
- Can adapt to images with multimodal intensity distributions.
Limitations: It may not perform well in cases where there are abrupt changes in illumination within local regions.
- Choosing the size of local regions and the method to combine local thresholds can be challenging.

Q9. Discuss Bayesian classification of foreground and background pixels.

Q10. Write Otsu's thresholding algorithm.

- Otsu's Algorithm

1. compute the normalized histogram of the input image.
2. compute the cumulative sums, P1(k) for k=0,1,2…L-1 using eq. (1)

$$P_1(k) = \sum_{i=0}^{k} p_i \quad \text{--------(1)}$$

3. compute the cumulative means, m(k) for k=0,1,2…L-1 using eq. (4)

$$m(k) = \sum_{i=0}^{k} i p_i \quad \text{-------(4)}$$

4. compute the global intensity mean mG using eq. (5) $m_G = \sum_{i=0}^{L-1} i p_i$
5. compute the between class variance σ^2_B (k) for k=0,1,2…L-1 using eq. (14)

$$\sigma_B^2(k) = \frac{(m_G P_1(k) - m(k))^2}{P_1(k)(1-P_1(k))} \text{-----(14)}$$

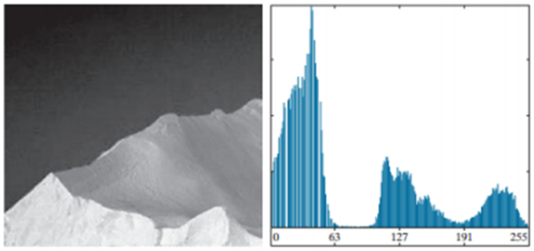
6. obtain the Otsu threshold k*, as the value of k for which σ^2_B (k) is maximum. If the maximum is not unique, obtain k* by averaging the values of k corresponding to the various maxima detected.
7. obtain the separability measure, η*, by evaluating eq. (13) at k = k*.

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2} \text{-----(13)}$$

Q11. Find out threshold value using Otsu's method for given 3*3 image:

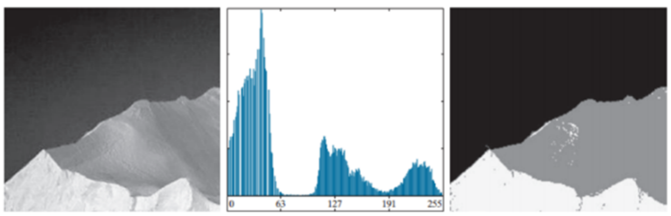
120	75	135
185	54	160
140	32	210

Q12. Apply multi-level thresholding using Otsu's method on following image:



Ans :-

Figure 10.42(a) shows an image of an iceberg. The objective of this example is to segment the image into three regions: the dark background, the illuminated area of the iceberg, and the area in shadows. It is evident from the image histogram in Fig. 10.42(b) that two thresholds are required to solve this problem. The procedure discussed above resulted in the thresholds $k_1^* = 80$ and $k_2^* = 177$, which we note from Fig. 10.45(b) are near the centers of the two histogram valleys. Figure 10.42(c) is the segmentation that resulted using these two thresholds in Eq. (10-76). The separability measure was 0.954. The principal reason this example worked out so well can be traced to the histogram having three distinct modes separated by reasonably wide, deep valleys. But we can do even better using superpixels, as you will see in Section 10.5.



a b c

FIGURE 10.42 (a) Image of an iceberg. (b) Histogram. (c) Image segmented into three regions using dual Otsu thresholds. (Original image courtesy of NOAA.)

Q13. Explain Watershed algorithm with its limitations. https://youtu.be/FLmxZaQhvsI?si=GZ2jm7HFma63_jfD

The Watershed algorithm is a computer vision and image processing technique used for image segmentation, which is the process of dividing an image into meaningful regions or objects. It is particularly useful for segmenting objects in images that have unclear or complex boundaries. Here's an explanation of how the Watershed algorithm works and its limitations:

How the Watershed Algorithm Works:

Grayscale Image: The Watershed algorithm typically works on grayscale images. If you have a color image, you can convert it to grayscale.

Gradient Calculation: It calculates the gradient of the image. This gradient represents the intensity changes in the image and helps identify potential boundaries between objects.

Marker Generation / Marker Placement: The user or an automated process needs to specify markers in the image. Markers are points or regions that indicate where different objects or regions should be separated. These markers are often labeled with different values.

Flooding: Starting from the markers, a "flooding" process occurs. The algorithm simulates the flooding of an area, where water rises from the markers. As the water level rises, it fills the image, and when the water levels from different markers meet, they form watershed lines.

Watershed Lines: Watershed lines are the boundaries that separate different objects or regions. The algorithm assigns a label to each region, and the result is a segmented image with distinct regions.

Limitations of the Watershed Algorithm:

Over-Segmentation: The Watershed algorithm is sensitive to noise and fine details in the image. This can lead to over-segmentation, where the algorithm creates too many small regions, making it challenging to interpret the segmentation result.

Marker Placement: The effectiveness of the algorithm heavily depends on the correct placement of markers. Incorrect or imprecise marker placement can lead to poor segmentation results.

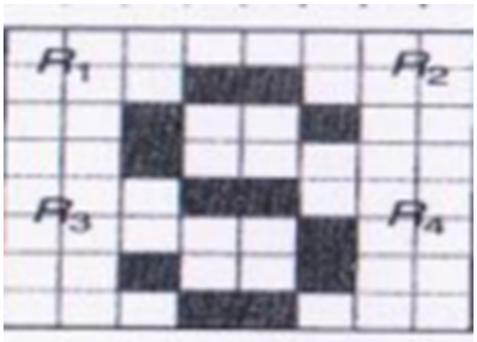
Gradient Calculation: The accuracy of the gradient calculation can be influenced by the choice of gradient operator and noise in the image. Noisy images can lead to incorrect boundaries.

Computational Complexity: The Watershed algorithm can be computationally expensive, especially for large images. Efficient implementations and optimizations are required to make it practical for real-time or large-scale applications.

Overcoming Plateaus: In some cases, plateaus in the image can be misinterpreted as watershed lines, leading to inaccurate segmentation. Additional pre-processing steps may be needed to address this issue.

Tuning Parameters: The Watershed algorithm often has parameters that require tuning, such as marker placement and handling of flat regions. Finding the right parameter settings can be a challenging task.

Q14. Segment following image using Region Splitting/Merging algorithm:



Ans :-

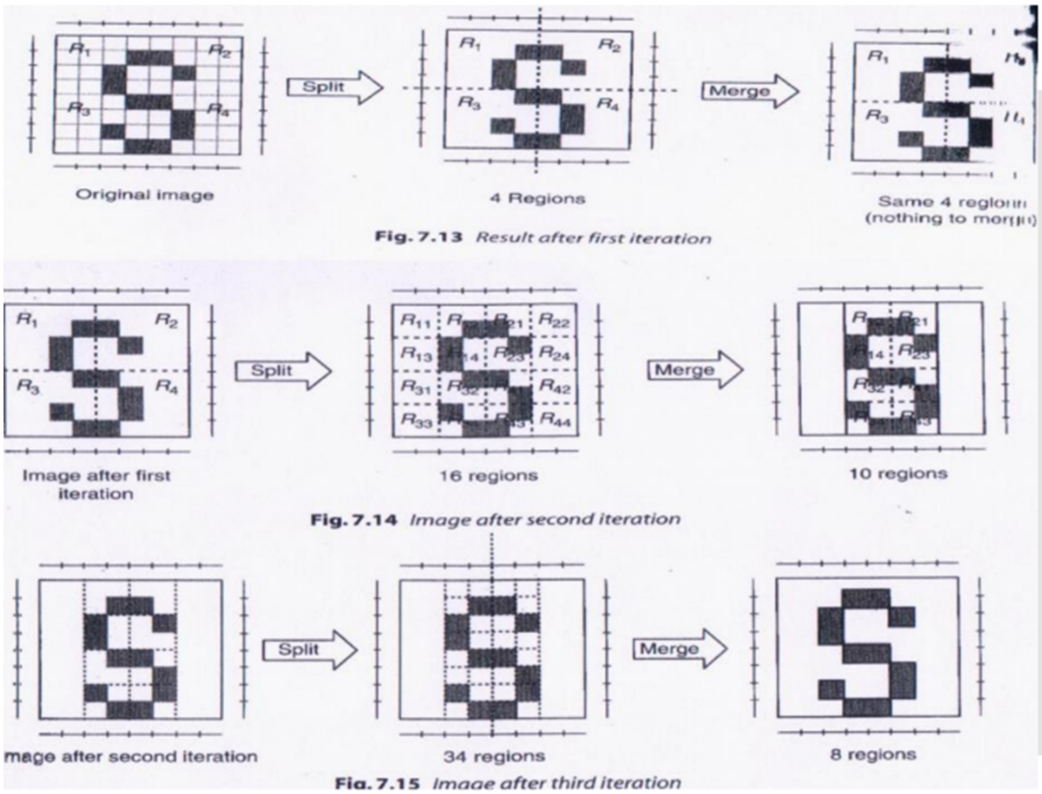


Fig. 7.13 *Result after first iteration*

Fig. 7.14 *Image after second iteration*

Fig. 7.15 *Image after third iteration*

Q15. Consider an 8 * 8 image, the gray levels range from 0 to 7. segments this image using the region growing technique.

Assume

- **Threshold = 2 (T<=2) and Seed pixel = 6**
- **Threshold = 3 (T<=3) and Seed pixel = 4**

5	6	6	6	7	7	6	6
6	7	6	7	5	5	4	7
6	6	4	4	3	2	5	6
5	4	5	4	2	3	4	6
0	3	2	3	3	2	4	7
0	0	0	0	2	2	5	6
1	1	0	1	0	3	4	4
1	0	1	0	2	3	5	4

Ans :-

For seed pixel= 6 and T≤2

5	6	6	6	7	7	6	6
6	7	6	7	5	5	4	7
6	6	4	4	3	2	5	6
5	4	5	4	2	3	4	6
0	3	2	3	3	2	4	7
0	0	0	0	2	2	5	6
1	1	0	1	0	3	4	4
1	0	1	0	2	3	5	4

For seed pixel= 4 and T≤3

5	6	6	6	7	7	6	6
6	7	6	7	5	5	4	7
6	6	4	4	3	2	5	6
5	4	5	4	2	3	4	6
0	3	2	3	3	2	4	7
0	0	0	0	2	2	5	6
1	1	0	1	0	3	4	4
1	0	1	0	2	3	5	4

Q16. What are the limitations of region growing segmentation techniques?

Sensitivity to Seed Selection: The choice of the initial seed point or region significantly impacts the result. If the seed is poorly chosen, it can lead to under-segmentation (missing important details) or over-segmentation (creating too many small regions).

Noise Sensitive: Noise in images can lead to inaccurate segmentation results, as region growing can misinterpret noise as similar regions. Pre-processing to reduce noise is often necessary.

Parameter Sensitivity: The success of region growing depends on setting appropriate criteria for similarity between pixels. Selecting the right thresholds for these criteria, such as intensity or color differences, can be challenging and can vary between images.

Difficulty Handling Complex Boundaries: Region growing tends to produce homogeneous regions, which makes it challenging to segment objects with complex or irregular boundaries.

Slow Execution: Region growing can be computationally expensive, especially for large images, because it involves iterative pixel-wise comparisons. This limits its efficiency for real-time or large-scale applications.

Propagation of Errors: If a seed region contains errors or inconsistencies, region growing can propagate these errors throughout the segmentation, making it difficult to correct.

Stopping Criteria: Deciding when to stop the region growing process can be challenging. Setting inappropriate stopping criteria can result in either under-segmentation (stopping too early) or over-segmentation (continuing too long).

Q17. What is the goal of segmenting an image?

Object Recognition and Identification: Image segmentation helps identify and delineate individual objects or regions within an image. It allows the system to distinguish between different entities, such as people, cars, buildings, and other objects of interest, which is crucial in various applications like object tracking, object recognition, and computer vision.

Feature Extraction: Once an image is divided into segments or regions, various features and characteristics of these regions can be extracted. These features can include color, texture, shape, size, and other properties, which are useful for subsequent analysis and understanding of the image content.

Image Enhancement: Segmenting an image can help in enhancing the visibility or quality of specific objects or regions within the image. After segmentation, it becomes possible to apply specific image processing operations selectively to different segments, improving the overall image quality or emphasizing certain aspects.

Object Localization: Image segmentation helps locate and determine the precise boundaries of objects within an image. This information is valuable for tasks like object tracking, robot navigation, and autonomous driving, where knowing the location of objects in the scene is critical.

In summary, image segmentation plays a vital role in image analysis and computer vision by breaking down complex images into distinct and interpretable regions or objects. It is a fundamental step in various applications, including object recognition, feature extraction, image understanding, image enhancement, and object localization, among others.

Q18. Consider the following 3-bit grey scale image

3	1	2	3
1	7	6	4
2	1	7	5
0	1	5	6

When contrast enhancement using histogram equalization is used, to which intensity is the intensity 5 mapped to?

Q.18

Gray level	(n_k) No. of pixels	PDF (r_k) = $\frac{n_k}{n}$	S_k (CDK)	$S_k \times 7$	
0	1	1/16 = 0.0625	0.0625	0.4375	0
1	4	4/16 = 0.25	0.3125	2.1875	2
2	2	2/16 = 0.125	0.4375	3.0625	3
3	2	2/16 = 0.125	0.5625	3.9375	4
4	1	1/16 = 0.0625	0.6250	4.375	4
5	2	2/16 = 0.125	0.7500	5.25	5
6	2	2/16 = 0.125	0.8750	6.125	6
7	2	2/16 = 0.125	1	7	7

$n = 16$

Hence the mapping of 5 is done to 5.

Q19. Consider two images I1 and I2 with dimensions 16 * 2 and 4 * 16 respectively. I1 consists of 16 background pixels and I2 consists of 4 background pixels. Rest pixels are foreground pixels. Suppose, a pixel is selected at random and is found to be a background pixel. What is the probability that the selected pixel is from image I2?

CHAPTER:3

Q1. Consider the following 3-bit grayscale image

0	1	2	3
4	5	6	7
0	1	2	5
4	1	5	6

What of the following can be the value when vertical Sobel operator and horizontal Sobel operator are applied on the orange colored pixel?

Ans: Method**:** https://youtu.be/fhDBy-wV3ic?si=XGDfg74D7aNcIpxH

Chapter 3

Q.1

Vertical Sobel operator:-

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Given Matrix:-

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

Horizontal Sobel operator:-

$$M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

i) Value when vertical Sobel operator are applied on orange colored

$\therefore 4(-1) + 5(0) + 6(1) + 0(-2) + 1(0) + 2(2) + 4(-1) + 1(0) + 1(5)$

$\Rightarrow -4 + 0 + 6 + 0 + 0 + 4 - 4 + 0 + 5$

$\Rightarrow \boxed{7}$

ii) Value when horizontal sobel operator are applied

$\therefore -1(4) + 5(-2) + 6(-1) + 0(0) + 1(0) + 2(0) + 4(1) + 1(2) + 5(1)$

$\Rightarrow -4 - 10 - 6 + 0 + 0 + 0 + 4 + 1 + 5$

$\Rightarrow \boxed{-10}$

Q2. Consider the following 3-bit grayscale image

0	1	2	3
4	5	6	7
0	1	2	5
4	1	5	6

What of the following can be the value when vertical Prewitt operator and horizontal Prewitt operator are applied on the orange colored pixel?

Ans: Method**:** https://youtu.be/fhDBy-wV3ic?si=XGDfg74D7aNcIpxH

Q.2 Vertical Prewitt operator:-

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Given Matrix:-

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

Horizontal Prewitt operator

$$M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

i) Value when vertical prewitt operator is applied on $\boxed{6}$

$\therefore -1(1) + 2(0) + 3(1) + 5(-1) + 6(0) + 7(1) + 1(-1) + 2(0) + 5(1)$

$\Rightarrow -1 + 0 + 3 - 5 + 0 + 7 - 1 + 0 + 5$

$\Rightarrow \boxed{8}$

ii) Value when horizontal prewitt operator is applied on $\boxed{6}$

$\therefore 1(-1) + 2(-1) + 3(-1) + 5(0) + 6(0) + 7(0) + 1(1) + 2(1) + 5(1)$

$\Rightarrow -1 - 2 - 3 + 0 + 0 + 0 + 1 + 2 + 5$

$\Rightarrow \boxed{2}$

Q3. Classify origins of edges in images.

In image processing, edges in images can be classified based on their origins or causes. The origins of edges are typically associated with different underlying factors or discontinuities in the image. Here's a classification of edge origins:

Surface Normal Discontinuity (Depth Discontinuity):

Origin: This type of edge arises from a change in the orientation of the underlying surface, which can be related to a depth or surface normal discontinuity.

Cause: When two surfaces with different orientations meet in the scene, it results in variations in the depth or surface normals at the point of contact. This leads to a sharp intensity or color transition in the image.

Example: The edge between a flat wall and a corner, where the wall meets the floor, is an example of a depth discontinuity edge.

Surface Color Discontinuity (Color Edge):

Origin: Surface color edges are caused by differences in color or texture on the surface.

Cause: Changes in surface color, texture, or reflectance properties can lead to variations in pixel values, resulting in edges. These edges may not be indicative of depth or normal changes but rather differences in material properties.

Example: The boundary between two regions of different colors on an object, such as the border between a red and blue region on a toy, is an example of a color edge.

Illumination Discontinuity (Shading Edge):

Origin: Illumination discontinuities occur due to variations in lighting conditions.

Cause: Changes in illumination, including shadows, highlights, or shading, can create edges in the image. These edges are not related to changes in surface properties but rather to the way light interacts with the scene.

Example: The boundary between a well-lit area and a shadowed area on an object's surface results in an illumination discontinuity edge.

Q4. Consider the following 3-bit grayscale image:

0	3	7	3	0
0	3	7	3	0
0	3	7	3	0
0	3	7	3	0
0	3	7		0

Calculate i. First order gradient in both X-axis and Y-axis.

Ans. TODO

Q5. What is the effect of noise image on gradients calculated on images and how it can affect extraction of edges from images?

Effect of Noise on Gradients: Noise in an image introduces variations in pixel values, which can lead to erroneous gradient calculations. Gradients represent intensity or color changes in an image, and noise can be misinterpreted as edges, causing false gradients.

Conversely, noise can also obscure or hide true edges in the image. The presence of noise can make it more challenging to identify real edges

The presence of noise can lead to inaccurate localization of edges. Edges may be detected with some uncertainty, and their positions may be shifted or blurred due to the effects of noise.

Impact on Edge Extraction: Noise-induced gradients can result in the detection of numerous false or spurious edges, reducing the accuracy and reliability of edge extraction algorithms. These additional edges can clutter the image and make it challenging to distinguish true object boundaries from noise-induced artifacts.

In summary, noise in images can corrupt gradient calculations, leading to the detection of false edges. This complicates the process of extracting meaningful edges from noisy images in image processing, requiring the use of noise reduction techniques or robust edge detection methods to mitigate the impact of noise.

Q6. How to solve the problem of noise present in images in edge detection using gradients?

To address the problem of noise in edge detection using gradients in image processing, you can employ several techniques to improve the accuracy and robustness of edge detection.

Smoothing (Noise Reduction): Apply a noise reduction technique, such as Gaussian blurring or median filtering, to the image before performing edge detection. Smoothing reduces the impact of noise by averaging pixel values in the local neighborhood, resulting in a cleaner image for gradient calculations.
Gradient Computation Methods - Use Canny Edge Detector: Instead of simple gradient operators like Sobel or Prewitt, consider using more advanced gradient operators, such as the Canny edge detector, which applies Gaussian smoothing and non-maximum suppression to improve edge detection accuracy while reducing sensitivity to noise.
Edge Linking: Implement edge linking or edge tracking techniques, such as the Hough transform or contour following, to connect and refine edge segments. These methods help consolidate fragmented edge information and reduce the impact of noisy gradients.
Thresholding: Apply an appropriate thresholding technique to the gradient magnitude or gradient direction to filter out weak gradient responses caused by noise. Adaptive thresholding methods can be especially useful in handling varying levels of noise.
Scale-Space Analysis: Use multi-scale or scale-space analysis to detect edges at multiple scales. This approach can help distinguish between genuine edges and noise artifacts, as true edges tend to persist across different scales.
Non-Maximum Suppression: Apply non-maximum suppression to the gradient magnitude to thin detected edges, preserving only the local maxima in the gradient direction. This helps remove redundant edge responses and improves the localization of true edges.

Q7. Define derivative of Gaussian in X-axis and Y-axis.

Q8. Discuss criteria of an "Optimal edge detector".
Good Detection: The optimal detector must minimize the probability of false positives(detecting spurious edges caused by noise) as well as that of false negatives(missing real edges)
Good localization: the edges detected must be as close as possible to the true edges.
Single Response: the detector must return one point only for each true edge point, that is minimize the number of local maxima around the true edge.
Low Sensitivity to Noise: The edge detector should be robust to noise in the image. It should be able to differentiate true edges from noise-induced fluctuations in pixel values. Noise-sensitive detectors can produce false edges and should be avoided.
Computational Efficiency: An optimal edge detector should be computationally efficient, particularly for real-time or large-scale applications. It should provide fast results without excessive computational resources.
Minimal False Positives: The detector should minimize the detection of false edges or non-edges. It should avoid highlighting non-edge features, such as textures or patterns, as edges, as this can lead to inaccurate results.

Q9. Consider the following 3-bit grayscale image

0	1	2	3
4	5	6	7
0	1	2	5
4	1	5	6

What of the following can be the value when Robert operators are applied on the orange colored pixel?

Ans: Method: https://youtu.be/fhDBy-wV3ic?si=6wEDjFP0rmjGlfH5

Q.9 Robert Operator: $M_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ $M_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

Given Image Matrix: $\begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & \boxed{6} & 7 \\ 0 & 1 & 2 & 5 \\ 4 & 1 & 5 & 6 \end{bmatrix}$

$\therefore$ Using M_x mask:

$\therefore$ $6(1) + 7(0) + 2(0) + 5(-1)$

$\Rightarrow 6 - 5$

$\Rightarrow 1$

$\therefore$ Using M_y Mask:

$\therefore$ $6(0) + 1(7) + 2(-1) + 5(0)$

$\Rightarrow 7 - 2$

$\Rightarrow 5$

Q10. Explain in detail the Canny edge detector.
https://youtu.be/zCcj8onPrOE?si=Zk0TtNE4NhW6Jb-r
https://www.educative.io/answers/what-is-canny-edge-detection

Q11. Write Canny edge detector algorithm.
1. Noise reduction (Apply Gaussian Smoothing):
   - Convolve the input image with a Gaussian filter to reduce noise. This step helps in reducing false edges caused by noise.
2. Calculate Gradient Magnitude and Direction:
   - Compute the gradient of the smoothed image using gradient operators (e.g., Sobel, Prewitt).
   - Calculate the gradient magnitude and direction at each pixel.
3. Non-Maximum Suppression:
   - Suppress non-maximum gradient values by only keeping the local maxima in the gradient direction.
   - Set non-maximum values to zero, creating a "thin" edge map.
4. Double Thresholding:
   - Define two thresholds, a high threshold (T_high) and a low threshold (T_low).
   - Pixels in the gradient magnitude image above T_high are marked as strong edge pixels.
   - Pixels between T_low and T_high are marked as weak edge pixels.
5. Edge Tracking by Hysteresis:
   - Trace edges by connecting strong edge pixels to weak edge pixels in the vicinity.
   - If a weak edge pixel is connected to a strong edge pixel, it is considered part of the edge.
   - Repeat this process until no more weak edge pixels can be connected to strong edge pixels.
6. Result:
   - The final output is a binary edge map, where strong and connected weak edge pixels form the detected edges.
7. Post-processing (Optional):
   - Optionally, apply further post-processing, such as edge smoothing or refinement, to improve the quality of the detected edges.
8. Return the Edge Map:
   - The output of the Canny edge detector is an edge map with edges represented as white pixels on a black background.

Q12. Discuss Non-maximum Suppression.
Q13. Illustrate the edge linking with an example.
Q14. Explain in detail Hysteresis thresholding.
Q15. Explain in detail Line detection using Hough transform.
Q16. Explain in detail Ellipse detection using Hough transform.

Q17. Write an algorithm for Generalized Hough Transform.
The Generalized Hough Transform is an extension of the standard Hough Transform that can be used to detect not only lines and circles and ellipses but also arbitrary shapes or objects in an image.

Input:
- Edge image (resulting from edge detection)
- Template image (the object or pattern to be detected)
Output:
- List of detected object positions

Algorithm:
1. Initialize an accumulator array (voting space) with dimensions matching the edge image.

2. For each edge point (x, y) in the edge image:
   a. Compute the local gradient direction (θ) at point (x, y).
   b. For each point (u, v) in the template image:
      i. Compute the relative coordinates (dx, dy) of (u, v) with respect to the center of the template.
      ii. Compute the corresponding template orientation (φ) for (u, v).
      iii. Calculate the difference between the template orientation (φ) and the local gradient direction (θ).
      iv. Convert (dx, dy) to polar coordinates (r, φ).
      v. Increment the accumulator cell at coordinates (x - r*cos(θ - φ), y - r*sin(θ - φ)).

3. Apply non-maximum suppression on the accumulator array to identify the peaks (positions with maximum votes).

4. Threshold the peaks to determine the positions with a sufficient number of votes, removing false positives.

5. Return the list of detected object positions (x, y) based on the remaining peaks.

The Generalized Hough Transform is versatile and can be adapted to detect objects of arbitrary shapes. It relies on the comparison of local gradient orientations between the edge image and the template to cast votes in the accumulator array. Detected object positions correspond to the peaks in the accumulator array. The choice of thresholds and other parameters may depend on the specific application.

CHAPTER:4

Q1. Give classification of Image Enhancement operations.

Image Enhancement: processing an image so that the result is more suitable than for a specific application.

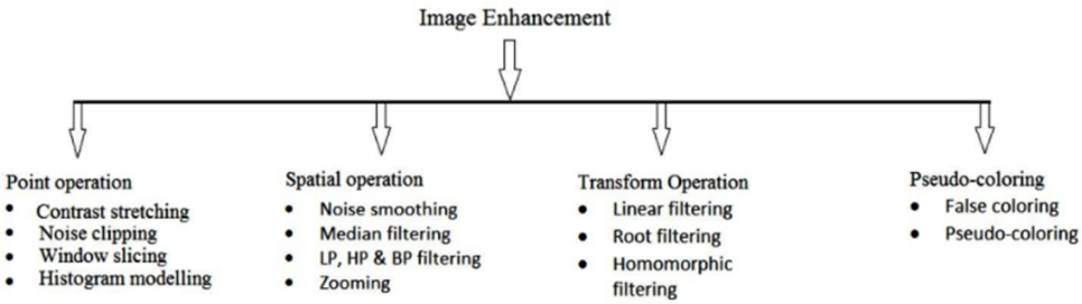
These operations can be broadly classified into various categories based on their objectives and methods. Here's a classification of image enhancement operations:

Spatial Domain Enhancement: Operations in this category are applied directly to the pixel values of the image and involve spatial filtering techniques.

Examples include histogram equalization, contrast stretching, and various spatial filters such as Gaussian, Laplacian, and sharpening filters.

Frequency Domain Enhancement: These operations involve transforming the image into the frequency domain (e.g., using the Fourier transform) and then modifying the frequency components.

Common techniques include filtering in the frequency domain to enhance or suppress certain frequency components.



Q2. Which of the following point processing operations is zero-memory operation?

(A) Image negative (B) Contrast Stretching (C) Lazy Man (D) Thresholding

Ans :- Contrast Stretching and **Thresholding** are point processing operations is zero-memory operation.

Q3. Consider following 3-bit grayscale image:

1	5	7
2	3	0
0	1	4

Evaluate output image of Image negative operation applied on given image.

Ans: Method: https://youtu.be/FMDmXz6ynvk?si=WIpBoK_lIGYY4Hsw

To apply the image negative operation, subtract each pixel value from the maximum possible pixel value (7 in this case):

So, the output image after applying the image negative operation will be:

6	**2**	**0**
5	**4**	**7**
7	**6**	**3**

Q4. Consider following 4-bit grayscale image:

12	15	14
7	5	6
4	3	2

Evaluate output image of Image negative operation applied on given image.

Ans: Method: https://youtu.be/FMDmXz6ynvk?si=WIpBoK_lIGYY4Hsw

To apply the image negative operation, subtract each pixel value from the maximum possible pixel value (15 in this case):

So, the output image after applying the image negative operation will be:

3	**0**	**1**
8	**10**	**9**
11	**12**	**13**

Q5. Consider following 8-bit grayscale image:

200	215	255
0	127	0
100	50	180

Evaluate output image of Image negative operation applied on given image.

Ans: Method: https://youtu.be/FMDmXz6ynvk?si=WIpBoK_lIGYY4Hsw

To apply the image negative operation, subtract each pixel value from the maximum possible pixel value (255 in this case):

So, the output image after applying the image negative operation will be:

55	**40**	**0**
255	**128**	**255**
155	**205**	**75**

Q6. Consider following 8-bit grayscale image:

200	215	255
0	127	0
100	50	180

Evaluate output image of Thresholding operation applied on given image with global threshold value 111.

Ans: Method: https://youtu.be/FMDmXz6ynvk?si=WIpBoK_lIGYY4Hsw

The output image after applying the Thresholding operation will be:

255	**255**	**255**
0	**255**	**0**
0	**0**	**255**

Q7. Consider following 8-bit grayscale image:

200	215	255
0	127	0
100	50	180

Evaluate output image of Contrast Stretching operation applied on a given image with following transformation function:

$s = \ 0.5r \ for \ 0 \le r < 100, \ 5r \ for \ 100 \le r < 140, \ 0.5r \ for \ 140 \le r \le 255$

For the pixel at (0,0) with an input value of 200:

- Since 140 ≤ 200 ≤ 255, we use s = 0.5r:
- s = 0.5 * 200 = 100

Similarly Do for each pixel

100	**108**	**128**
0	**635**	**0**
50	**25**	**90**

Q8. Consider following 8-bit grayscale image:

200	215	255
0	127	0
100	50	180

Evaluate output image of Clipping operation applied on a given image with following transformation function:

$$s = 0.5r \ for \ 0 \leq r \lessdot 100, 5r \ for \ 100 \leq r \lessdot 140, 0 \ for \ 140 \leq r \leq 255$$

0	0	0
0	635	0
50	25	0

Q9. Distinguish Log transformation and Power law transformation.

Log transformation and power-law (gamma) transformation are two common point processing operations used in image enhancement to modify pixel values.

Log Transformation:

Mathematical Formulation: Log transformation, also known as logarithmic transformation, is expressed as:
s = c * log(1 + r)
where:
r is the input pixel value.
s is the output pixel value.
c is a constant that scales the result.

Purpose: Log transformation is primarily used to expand the dynamic range of an image, especially in cases where the original image has a wide range of pixel intensities. It enhances the details in the darker regions of an image and compresses the brighter regions. Log transformation is often applied for images with a large variation in lighting, such as astronomical or medical images.

Effect: It brightens the dark areas of an image while reducing the intensity of the brighter regions. This can reveal more details in the shadows.

Power-Law (Gamma) Transformation:

Mathematical Formulation: The power-law transformation, also known as gamma correction, is expressed as:
s = c * r^γ
where:
r is the input pixel value.
s is the output pixel value.
c is a constant for scaling.
γ (gamma) is a parameter that controls the shape of the transformation curve.

Purpose: The power-law transformation is used for gamma correction, which adjusts the brightness and contrast of an image. It allows you to fine-tune the tone reproduction curve to match the display characteristics and improve the visibility of details. Gamma correction is commonly used in computer graphics and imaging to correct for non-linear display devices.

Effect: A higher value of γ (>1) increases the contrast, making the image appear brighter, while a lower value of γ (<1) decreases the contrast, making the image appear darker. A γ value of 1 results in no change in the image.

In summary, the key difference between log transformation and power-law (gamma) transformation lies in their mathematical expressions and the specific purposes they serve. Log transformation is used to expand the dynamic range of an image and enhance details in dark regions, while power-law transformation allows for fine-tuning of brightness and contrast and is often used for gamma correction to adapt images to specific display characteristics.

Q10. Consider following 8-bit grayscale image:

200	215	255
0	127	0
100	50	180

Evaluate output image of Gray Level operation applied on a given image with following transformation function:

$$s = \quad 0\, for\, 0 \leq r \preccurlyeq 100, 150\, for\, 100 \leq r \preccurlyeq 140, 0\, for\, 140 \leq r \leq 255$$

0	**0**	**0**
0	**150**	**0**
150	**0**	**0**

Q11. Apply bitplane slicing following 3-bit grayscale image:

1	5	7
2	3	0
0	1	4

Sketch each of the bitplanes .

Ans: Method: https://youtu.be/FMDmXz6ynvk?si=WIpBoK_llGYY4Hsw

Q.11 Bit plane Slicing.

3-bit grayscale

1	5	7
2	3	0
0	1	4

001	101	111
010	011	000
000	001	100

MSB

0	1	1
0	0	0
0	0	1

Middle

0	0	1
1	1	0
0	0	0

LSB

1	1	1
0	1	0
0	1	0

Q12. Explain gray level slicing and bit plane slicing with an example.

Gray Level Slicing

Gray level slicing is a technique used in image processing to create a new image from an existing image. The new image is created by thresholding the original image, which means that only pixels that meet a certain criteria are included in the new image. Gray level slicing is often used to highlight certain features in an image, or to remove background noise.

Bit Plane Slicing

The three main goals of bit plane slicing is:

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

Example :-

Bit plane slicing:

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

110	111	110	110	111
000	000	000	001	010
001	001	001	010	011
100	101	101	100	010
110	110	110	111	111

Separating the bit planes, we obtain

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

MSB plane

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

Centre bit plane

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

LSB plane

Q13. Demonstrate steps of operation of Spatial FIltering with an example.

https://youtu.be/eDyJsFHYFFI?si=J5LG0KqhFYxKOgc3

- Filtering refers to passing or rejecting certain frequency components.
• Low pass filter: a filter that passes low frequencies, to blur (smooth) an image.
• High pass filter: a filter that passes high frequencies, to detect edges in the image.
- There is one to one correspondence between linear spatial filters and filters in the frequency domain.
- However, spatial filters offer considerably more versatility because, they can be used also for nonlinear filtering, that we can not do in the frequency domain.
- Spatial filters consists of
• 1. neighborhood
• 2. predefined operation
- Filtering creates a new pixel with coordinates equal to the coordinates of the center of the neighborhood, and whose value is the result of the filtering operation.

- A processed (filtered) image is generated as the center of the filter visits each pixel in the input image.
- If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter, otherwise nonlinear.

Q14. Which of the following is most common approach to remove gaussian noise from image.
A) Apply High Pass FIlter on image
B) Apply Low Pass Filter on image
C) Apply Band Pass Filter on an image
D) Apply Band Stop Filter on an image
ANS: To remove Gaussian noise from an image, the most common approach is to apply a **Low Pass Filter.**

Q15. Define Smoothing Spatial FIlters. https://youtu.be/x6zoQ-a7A9U?si=vM3rlldQth1Eg0SP
Smoothing filter is used for blurring and noise reduction in the image. Blurring is pre-processing steps for removal of small details and Noise Reduction is accomplished by blurring.

Q16. Define Sharpening Spatial FIlters.
https://youtu.be/ElE_XcYxvCY?si=FPfUQb1zFMk6np_2

Q17. What is the effect changing mask size while performing spatial filtering?
Changing the mask size (also known as the kernel size or filter size) when performing spatial filtering in image processing has several effects on the filtering operation and the resulting image. The mask size determines the extent of the neighborhood used to calculate the output pixel value. Here are the effects of changing the mask size:
Smoothing or Blurring Effect: Increasing the mask size by using a larger kernel results in a stronger smoothing or blurring effect on the image. This is because a larger kernel considers a wider area, leading to a greater averaging or blurring of pixel values. As a result, fine details and noise are reduced.
Increased Computational Complexity:Using a larger mask size requires more computations to process each pixel, which increases the computational complexity of the filtering operation. Larger kernels involve more multiplications and additions for each pixel, potentially slowing down the process.
Sharper Edges and Enhanced Features: Reducing the mask size by using a smaller kernel preserves more image details and features. Smaller kernels emphasize local image information, resulting in sharper edges and more detailed structures. However, they may not be as effective in reducing noise.
Noise Amplification with Small Kernels:When using very small kernels, noise can be amplified because the filter has a limited ability to smooth or reduce noise. Smaller kernels are more sensitive to variations in pixel values, which can lead to emphasizing noise in the output.
Artifact Minimization: Large mask sizes can help in reducing artifacts such as blockiness or aliasing in images, which can be especially relevant in image compression and scaling operations.

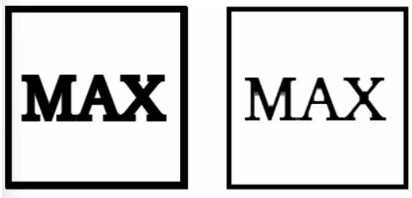
In summary, changing the mask size in spatial filtering has a significant impact on the filtering result. Smaller masks preserve details but may not be as effective in noise reduction, while larger masks are better at noise reduction but can blur fine details. The choice of mask size should be made based on the specific image characteristics and the desired outcome

Q18. Define max filter, min filter, median filter.
The Minimum Filter(erosion filter):
The transformation replaces the central pixel with the darkest one in the running window.
Purpose: The min filter is used for tasks such as noise reduction, edge detection, and morphological operations. It is effective at reducing noise and preserving dark features.
Effect: The output of the min filter reduces the intensity of bright regions, emphasizing dark areas and potentially removing noise.
For example, if you have text that is lightly printed, the minimum filter makes letters thicker.

Original Image

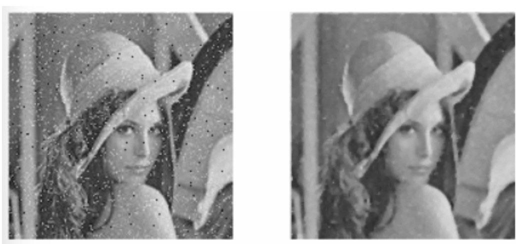
with Minimum Filter

The Maximum Filter(dilation filter):
The maximum and minimum filters are shift-invariant. Whereas the minimum filter replaces the central pixel with the darkest one in the running window, the maximum filter replaces it with the lightest one.
Purpose: The max filter is used to enhance bright features or to expand the size of image components. It's particularly useful for morphological operations and image segmentation.
Effect: The output of the max filter highlights the brightest features in the image, making them stand out.
For example, if you have a text string drawn with a thick pen, you can make the sign skinnier.



Original Image with Maximum Filter

The Noise Filter: Median:
The median filter is a very popular image transformation which allows the preserving of edges while removing noise.
Purpose: The median filter is widely used for noise reduction, especially salt-and-pepper noise. It effectively preserves image details while removing outliers.
Effect: The output of the median filter smooths the image by replacing each pixel with the median value in its local neighborhood. It is particularly effective at removing impulse noise.


Original Image with Median Filter

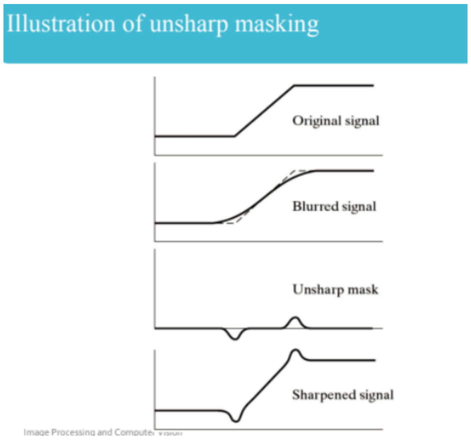
Q19. Consider following 8-bit grayscale image:

200	215	255
0	127	0
100	50	180

i. Apply 3*3 Spatial Law Pass Filter (Averaging Box Filter) :- **https://youtu.be/L7odWbdRutE?si=0AdI9Ub_2IVXr**
ii. Apply 3*3 Spatial High Pass Filter :- **https://youtu.be/Ow-i41-Tz9k?si=To_qfjNtZd_NKat1**
iii. Apply 3*3 Min Filter
iv. Apply 3*3 Max Filter
v. Apply 3*3 Median Filter
Min , Max , Median :- https://youtu.be/OPDmj6Er-dM?si=_2ik8Pmk5PtmsNu7
on a given input image and generate output images while presenting steps of calculations. (Note: Use Zero padding when required)

Q20. Compare performance of first order derivative and second order derivative for sharpening operation on images with an example.

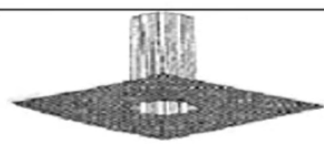
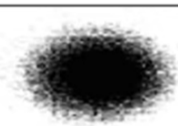
Q21. Explain Unsharp Masking and High Boost Filtering. https://youtu.be/Hutnurzr59o?si=8IHJhGvZQTQokKfv



Q22. Explain the process of filtering images in the frequency domain.
https://youtu.be/guPIbJIZ7QI?si=-fUBOn800BsJ3b1a

Q23. What is aliasing? How does it affect image quality?
Literally, the word alias means "**a false identity.**" In the field of signal processing, aliasing refers to sampling phenomena that cause different signals to become indistinguishable from one another after sampling; or, viewed another way, for one signal to "masquerade" as another.

Q24/Q25. Compare performance of Ideal, Butterworth and Gaussian Low pass filters.

Parameters	Ideal Low Pass Filter	Butterworth Low Pass Filter	Gaussian Low Pass Filter.
Cut-off frequency D_o	The filter retains all the frequency components within the radius D_o. All the frequency components outside the radius of the circle shall be eliminated.	It passes all the frequencies inferior to the D_o, whereas it cuts off all the frequencies above the D_o. The transition is smooth in case of the Butterworth filter.	The transition is much smoother because of inclusion of σ.
Ringing Effects	Definitely affected.	Affected, but not as much as ILPF. When the order gets increased, the ringing effect appears.	Totally eliminated due to inclusion of σ. σ is a measure for the spreadness.
Transfer Function			
Filter view			

https://www.slideserve.com/eyal/image-processing

1) Low pass - *image Smoothing*

Smoothing filters:

1.1) Ideal low pass filters (ILPF)

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

1.2) Butterworth low pass filters (BLPF)

$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$

1.3) Gaussian low pass filters (GLPF)

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

Q26. What is the effect of changing order of filter on output images while performing frequency domain filtering?

Q27. Which of the following filters never generates a Ringing effect in output images?
(A) Sixth order Ideal Filter (B) Sixth order Butterworth Filter (C) Sixth order Gaussian Filter
(D) Sixth order Chebyshev Filter
ANS: The filter that never generates a ringing effect in output images is the *Sixth order Gaussian Filter (C)*

Q28. Which of the following filters generates maximum Ringing effect in output images?
(A) Sixth order Ideal Filter (B) Sixth order Butterworth Filter (C) Sixth order Gaussian Filter
(D) Sixth order Chebyshev Filter
ANS: The filter that generates the maximum ringing effect in output images is the *Sixth order Ideal Filter (A).*

Q29. Where are applications of Homomorphic filtering? Explain Homomorphic filtering with an example.
https://youtu.be/NIM78sVsEVg?si=vMcnEfbM7f9oDGKR

Q30. What are the applications of Notch filters?
Notch filters, also known as band-stop filters, are designed to attenuate or reject specific frequencies while allowing others to pass. They find applications in various fields where the suppression of certain frequencies is necessary. Here are some common applications of notch filters:
Electronics and Communications: Signal Interference Rejection: Notch filters are used in electronic circuits and communication systems to reject specific interference frequencies, such as power line hum (50 or 60 Hz) or radio frequency interference (RFI).
Audio Processing: Elimination of Specific Tones: Notch filters can be employed in audio processing to eliminate unwanted tones or frequencies, such as feedback frequencies in audio systems or specific harmonics.
Biomedical Signal Processing: Power Line Noise Removal: In biomedical signal processing, notch filters are used to remove power line noise (50 or 60 Hz) from physiological signals, such as electrocardiograms (ECGs) or electromyograms (EMGs).
Image Processing: Artifact Removal in Imaging: Notch filters can be applied in image processing to remove unwanted periodic artifacts, such as interference patterns or noise caused by specific lighting conditions.
Power Systems and Control Systems: Harmonic Filtering: In power systems and control systems, notch filters are utilized to filter out specific harmonic frequencies generated by non-linear loads or devices, ensuring the stability and quality of the electrical signal.
Acoustic Engineering: Noise Reduction in Acoustics: Notch filters find applications in acoustics to reduce or eliminate specific frequencies that may cause noise or unwanted resonance in structures or systems.
Aerospace and Defense: Jamming Signal Rejection: In radar and communication systems, notch filters are employed to reject jamming signals or unwanted frequencies, enhancing the system's ability to detect and communicate.

Q31. Write an algorithm for performing adaptive median filtering.

This algorithm has three principal objectives:
->to remove salt-and-pepper (impulse) noise,
->to provide smoothing of other noise that may not be impulsive
->to reduce distortion, such as excessive thinning or thickening of object boundaries.

The adaptive median-filtering algorithm uses two processing levels, denoted level A and level B, at each point (x, y):

Level A: If $z_{min} < z_{med} < z_{max}$, go to Level B
Else, increase the size of S_{xy}
If $S_{xy} \leq S_{max}$, repeat level A
Else, output z_{med}.

Level B: If $z_{min} < z_{xy} < z_{max}$, output z_{xy}
Else output z_{med}.

where S_{xy} and S_{max} are odd, positive integers greater than 1. Another option in the last step of level A is to output z_{xy} instead of z_{med}. This produces a slightly less blurred result, but can fail to detect salt (pepper) noise embedded in a constant background having the same value as pepper (salt) noise.

CHAPTER:6:

QUESTIONS:
Q1. Define measure of corner response for Harris corner
Q2. Explain working of Harris corner detector.
Q3. Discuss properties of Harris corner detector.
ANSWER:
https://drive.google.com/file/d/1W12I5mKBA7SN45WTtT8SlKDws4e_71rH/view?usp=sharing

QUESTION BANK ALL SUM ANSWER:

https://drive.google.com/file/d/1Uzdlh4CXo7iZl3GRvHOhfP3oF4q4JpVu/view?usp=sharing