RESULT:

REVIEW QUESTIONS:

1. What is the fundamental difference between image enhancement in the spatial domain and the frequency domain?
2. How does the Fast Fourier Transform (FFT) enable image enhancement in the frequency domain?
3. What are some common filters used for image enhancement in the frequency domain, and how do they affect image features?
4. Can you explain the concept of convolution or point-wise multiplication in the context of frequency domain filtering for image enhancement?
5. What are the advantages of using frequency domain filtering for image enhancement, and in what scenarios is it particularly useful?

Ex. No.	IMAGE SEGMENTATION – EDGE DETECTION, LINE DETECTION AND POINT DETECTION	Date

AIM:
The aim of this project is to perform image segmentation, specifically focusing on edge detection, line detection, and point detection in digital images, enabling the extraction of important features and boundaries.

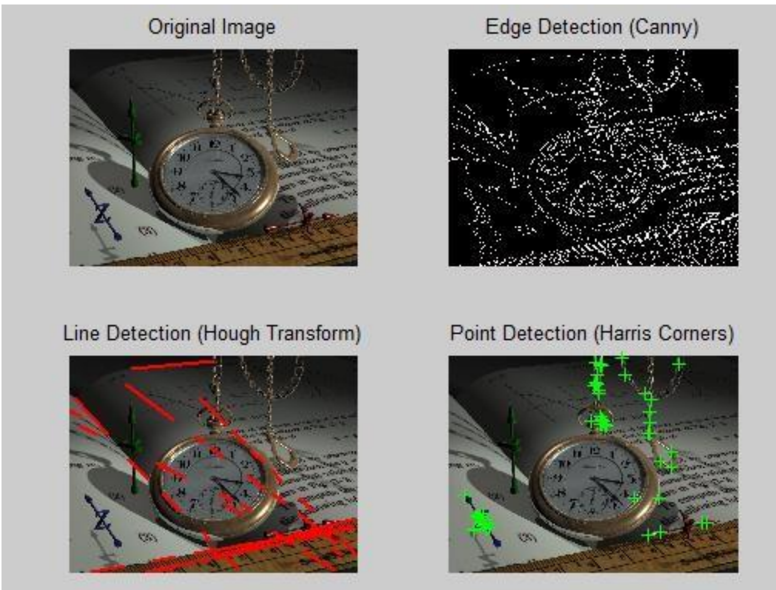
SOFTWARE REQUIRED:
MATLAB 2013b
THEORY:
Image segmentation is a crucial image processing task that aims to partition an image into meaningful regions. The techniques used in this project involve:
1. Edge Detection: This technique highlights significant transitions or boundaries in an image. It is commonly used to identify object boundaries or regions of interest.
2. Line Detection: Utilizing the Hough Transform, this technique identifies straight lines in an image, which is useful for applications such as object tracking and lane detection.
3. Point Detection: The Harris corner detection method identifies key points or corners in an image, providing critical feature points for various computer vision tasks.
PROCEDURE:

1. Load the input image onto which image segmentation techniques will be applied.
2. Apply edge detection methods, such as the Canny edge detector, to highlight image edges and transitions.
3. Employ the Hough Transform to identify straight lines in the image, displaying their positions and orientations.
4. Utilize the Harris corner detection technique to find key points or corners in the image.
5. Display the original image and the results of edge detection, line detection, and point detection.
6. Save the images resulting from each segmentation technique for further analysis or use.

PROGRAM:
```
% Load the image
inputImage = imread('your_image.jpg');
% Edge Detection using Canny
edgeImage = edge(rgb2gray(inputImage), 'Canny');
% Line Detection using Hough Transform
```

```
[H, theta, rho] = hough(edgeImage);
peaks = houghpeaks(H, 10);  % Adjust the number of peaks as needed
lines = houghlines(edgeImage, theta, rho, peaks);
% Point Detection using the Harris corner detector
corners = detectHarrisFeatures(rgb2gray(inputImage));
% Display the results
figure;
subplot(2, 2, 1);
imshow(inputImage);
title('Original Image');
subplot(2, 2, 2);
imshow(edgeImage);
title('Edge Detection (Canny)');
subplot(2, 2, 3);
imshow(inputImage);
hold on;
for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1), xy(:,2), 'LineWidth', 2, 'Color', 'red');
end
title('Line Detection (Hough Transform)');
subplot(2, 2, 4);
imshow(inputImage);
hold on;
plot(corners.selectStrongest(50));
title('Point Detection (Harris Corners)');
% Save the images if needed
imwrite(edgeImage, 'edge_detection.jpg');
imwrite(inputImage, 'line_detection.jpg');
imwrite(inputImage, 'point_detection.jpg');
```

OUTPUT:



RESULT:

REVIEW QUESTIONS:

1. What is the significance of edge detection in image segmentation?
2. How does the Hough Transform assist in line detection in image analysis?
3. What are the key objectives of point detection using methods like the Harris corner detector?
4. In what scenarios might image segmentation techniques like edge detection, line detection, and point detection be particularly valuable?
5. Can you discuss potential challenges or limitations associated with these image segmentation techniques?

Ex. No.	IMPLEMENTATION OF REGION BASED SEGMENTATION	Date

AIM:

The aim is to partition an input image into meaningful regions or segments based on color, intensity, or texture characteristics, thus improving image analysis and object recognition.

SOFTWARE REQUIRED:

MATLAB 2013b

THEORY:

Region-based segmentation is a computer vision technique that groups pixels or image regions into meaningful segments based on their similarity in terms of color, intensity, or texture. This segmentation technique involves the following key steps:

1. Preprocessing: The input image is preprocessed to enhance its quality, reduce noise, and normalize intensity values.
2. Feature Extraction: Relevant features, such as color histograms, texture features, or intensity gradients, are computed for each pixel or region in the image.
3. Region Growing or Split-and-Merge: The segmentation algorithm selects seed points and iteratively grows or splits regions based on feature similarity. Region growing starts with seed pixels and expands by including neighboring pixels that meet certain similarity criteria. Split-and-merge, on the other hand, recursively divides large regions into smaller ones if dissimilarity criteria are met.
4. Region Merging: After regions have been created, regions that are too small or similar to their neighbors are merged.
5. Postprocessing: Additional postprocessing steps can be applied to refine the segmentation results, such as removing small noise regions or smoothing region boundaries.

PROCEDURE:

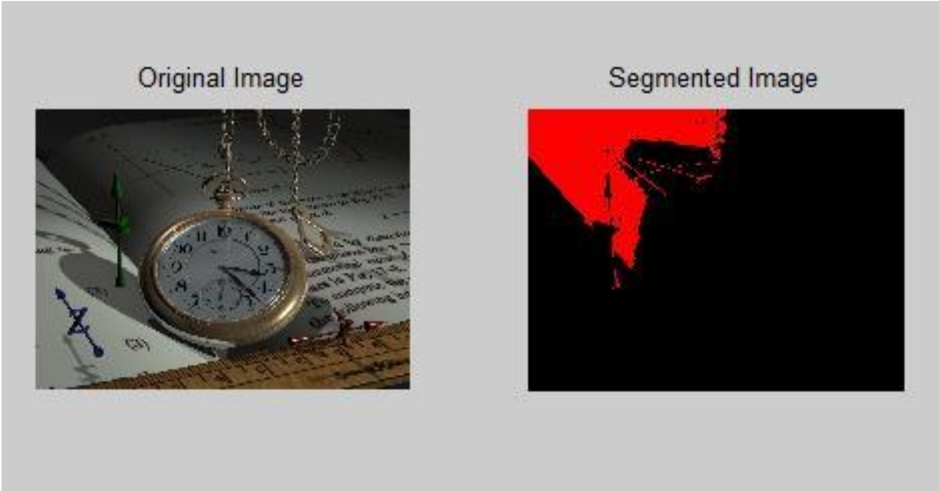
1. Import the input image.
2. Preprocess the image (e.g., noise reduction, intensity normalization).
3. Compute relevant image features (e.g., color histograms, texture features).
4. Initialize seed points or regions.
5. Implement the region growing or split-and-merge algorithm.
6. Apply region merging to refine the segmentation.
7. Perform postprocessing as needed.
8. Visualize and analyze the segmented image.

9. Optimize parameters and iterate for better results.
10. Save or export the segmented image.

PROGRAM:

```
% Read the input image
inputImage = imread('clock.jpg');
% Define a seed point (you can choose this point manually)
seedPoint = [100, 100]; % [row, column]
% Set the intensity similarity threshold
intensityThreshold = 20;
% Initialize the segmented image
segmentedImage = zeros(size(inputImage));
% Create a stack for region growing
stack = [];
stack = [stack; seedPoint];
% Define the region mean and region size
regionMean = double(inputImage(seedPoint(1), seedPoint(2)));
regionSize = 1;
% Define connectivity for 8-connected neighbors
connectivity = [-1, -1; -1, 0; -1, 1; 0, -1; 0, 1; 1, -1; 1, 0; 1, 1];
% Region growing loop
while ~isempty(stack)
   currentPoint = stack(1, :);
   stack(1, :) = [];
for i = 1:8
      neighbor = currentPoint + connectivity(i, :);
      if neighbor(1) > 0 && neighbor(1) <= size(inputImage, 1) && neighbor(2) > 0 &&
neighbor(2) <= size(inputImage, 2)
         if segmentedImage(neighbor(1), neighbor(2)) == 0 &&
abs(double(inputImage(neighbor(1), neighbor(2))) - regionMean) <= intensityThreshold
            regionSize = regionSize + 1;
            regionMean = (regionMean * (regionSize - 1) + double(inputImage(neighbor(1),
neighbor(2)))) / regionSize;
            stack = [stack; neighbor];
            segmentedImage(neighbor(1), neighbor(2)) = 1;
         end
      end
   end
end
% Display the segmented image
figure;
subplot(1, 2, 1);
imshow(inputImage);
title('Original Image');
subplot(1, 2, 2);
imshow(segmentedImage, []);
title('Segmented Image');
% You can save the segmented image if needed
imwrite(segmentedImage, 'segmented_image.jpg');
```

OUTPUT:



RESULT:

REVIEW QUESTIONS:
1. What is the primary objective of region-based image segmentation?
2. Describe the key steps involved in region-based segmentation.
3. What are some preprocessing techniques that can be applied to improve segmentation results?
4. How does region growing differ from split-and-merge in image segmentation?
5. What postprocessing steps can be employed to refine the segmented image?

Ex. No.	BASIC MORPHOLOGICAL OPERATIONS	Date

AIM:
The aim is to understand and implement basic morphological operations, including dilation, erosion, opening, and closing, using digital images.

SOFTWARE REQUIRED:
MATLAB 2013b

THEORY:
Morphological operations are image processing techniques based on the shape and structure of objects within an image. They are often applied to binary or grayscale images and rely on a structuring element (also known as a kernel) to modify the pixels in the image.
Dilation is used to enlarge the boundaries of objects in a binary image. It involves sliding the structuring element over the image and replacing the center pixel with the maximum value found in the neighborhood defined by the structuring element.
Erosion is used to shrink the boundaries of objects in a binary image. It involves sliding the structuring element over the image and replacing the center pixel with the minimum value found in the neighborhood defined by the structuring element.
Opening is a combination of erosion followed by dilation. It is used to remove small noise in binary images or separate objects that are close to each other.