PROJECT

ADDING WATERMARK TO AN IMAGE

# Aim

To Add a watermark to an image in MATLAB.

# Software Used:

**Theory:**

MATLAB

# Steps to Add a Watermark Using MATLAB

## Here’s a general process for adding a watermark using MATLAB:

1. **Load the Images**: Load the main image and the watermark (logo or text) image.

## **Resize the Watermark**: Adjust the watermark’s dimensions to fit the main image appropriately, depending on the desired size and coverage.

1. **Set Transparency**: Blend the watermark and main image pixels to create a semi-transparent effect, usually by combining the pixel values at a 50-50 ratio or adjusting as needed.

## **Positioning the Watermark**: Determine the location (e.g., bottom-right corner) and apply the watermark only to this region of the main image.

1. **Display or Save the Watermarked Image**: Show the final image with the watermark or save it to prevent unauthorized use or modification.

### PROGRAM :-

% Digital Image Watermarking in MATLAB

% This project demonstrates how to embed a visible watermark into a digital image using MATLAB.

% Step 1: Load the main image

mainImage = imread('/MATLAB Drive/1/nature.jpg'); % Replace with the path to your main image

if size(mainImage, 3) == 1

mainImage = cat(3, mainImage, mainImage, mainImage); % Convert grayscale to RGB if needed

end

% Step 2: Load the watermark image (logo or text watermark) watermarkType = 'image'; % Choose 'image' for logo or 'text' for text watermark

if strcmp(watermarkType, 'image')

watermark = imread('/MATLAB Drive/1/dhoni.jpg'); % Replace with the path to the watermark image

if size(watermark, 3) == 1

watermark = cat(3, watermark, watermark, watermark); % Convert to RGB if necessary

end

elseif strcmp(watermarkType, 'text')

watermarkText = '© Your Brand Name'; % Text watermark watermark = insertText(zeros(size(mainImage, 1), size(mainImage,

2), 3), [10, 10], watermarkText, 'FontSize', 48, 'TextColor', 'white'); end

% Step 3: Resize watermark to fit within the main image scaleFactor = 0.2; % Set the scale factor for resizing the watermark [wmRows, wmCols, ~] = size(watermark);

if wmRows > size(mainImage, 1) || wmCols > size(mainImage, 2) watermark = imresize(watermark, [size(mainImage, 1) \* scaleFactor,

size(mainImage, 2) \* scaleFactor]); end

% Step 4: Set the transparency (alpha blending)

alpha = 0.4; % Transparency factor (0 = fully transparent, 1 = fully opaque)

% Step 5: Determine watermark position

position = 'bottom-right'; % Options: 'top-left', 'top-right', 'bottom-left', 'bottom-right', 'center'

switch position case 'top-left'

rowPos = 10;

colPos = 10; case 'top-right'

rowPos = 10;

colPos = size(mainImage, 2) - wmCols - 10; case 'bottom-left'

rowPos = size(mainImage, 1) - wmRows - 10; colPos = 10;

case 'bottom-right'

rowPos = size(mainImage, 1) - wmRows - 10; colPos = size(mainImage, 2) - wmCols - 10;

case 'center'

rowPos = floor((size(mainImage, 1) - wmRows) / 2); colPos = floor((size(mainImage, 2) - wmCols) / 2);

end

% Step 6: Create a watermark with transparency watermarkedImage = mainImage;

for i = 1:3 % Process each color channel (R, G, B) if strcmp(watermarkType, 'image')

% Blend image watermark

watermarkedImage(rowPos:rowPos + wmRows - 1, colPos:colPos

+ wmCols - 1, i) = ...

alpha \* watermark(:, :, i) + (1 - alpha) \* watermarkedImage(rowPos:rowPos + wmRows - 1, colPos:colPos + wmCols - 1, i);

elseif strcmp(watermarkType, 'text')

% Blend text watermark (inserted as an image)

textRegion = watermarkedImage(rowPos:rowPos + wmRows - 1, colPos:colPos + wmCols - 1, :);

watermarkedImage(rowPos:rowPos + wmRows - 1, colPos:colPos

+ wmCols - 1, :) = ...

alpha \* watermark(rowPos:rowPos + wmRows - 1, colPos:colPos

+ wmCols - 1, :) + ...

(1 - alpha) \* textRegion;

end end

% Step 7: Display the original and watermarked images

figure; subplot(1, 2, 1);

imshow(mainImage); title('Original Image');

subplot(1, 2, 2); imshow(watermarkedImage); title('Image with Watermark');

% Step 8: Save the watermarked image

outputImagePath = 'watermarked\_image.jpg'; % Specify output path imwrite(watermarkedImage, outputImagePath);

% Step 9: Add additional optional enhancements

% - Apply noise to the watermarked image for testing robustness noiseImage = imnoise(watermarkedImage, 'gaussian', 0, 0.02); figure;

imshow(noiseImage); title('Watermarked Image with Noise');

% - Apply resizing or compression (test robustness) compressedImage = imresize(watermarkedImage, 0.5); % Resize to simulate compression

figure; imshow(compressedImage);

title('Compressed Watermarked Image');

% - Add ability to embed watermark with other transformations (e.g., rotation, contrast adjustments)

% Final Notes:

% The watermark is visible and embedded with transparency, and the watermark's position and transparency can be easily adjusted.

% The code supports both image and text-based watermarking, allowing flexibility in watermark customization.

# OUTPUT:







### RESULT:-

Thus the given experiment adding a watermark to an image has been completed successfully

BY

KEERTHI AACHUTHAN K 221701030 KARTHICK N 221701028