**Aim: Implementation of image restoring techniques**

% Load a sample image

originalImage = imread('cameraman.tif'); % Load a sample image

% Add noise to the image (you can adjust the noise level as needed)

noiseLevel = 25;

noisyImage = imnoise(originalImage, 'gaussian', 0, (noiseLevel/255)^2);

% Display the noisy image

subplot(1, 2, 1);

imshow(noisyImage);

title('Noisy Image');

% Image restoration using Total Variation regularization

lambda = 0.1; % Regularization parameter (adjust as needed)

numIterations = 100; % Number of iterations (adjust as needed)

% Perform TV regularization-based image restoration

restoredImage = tvdenoise(noisyImage, lambda, numIterations);

% Display the restored image

subplot(1, 2, 2);

imshow(restoredImage);

title('Restored Image');

% Define the TV denoising function (you can use an existing MATLAB implementation)

function denoisedImage = tvdenoise(inputImage, lambda, numIterations)

u = double(inputImage);

[Dux, Duy] = gradient(u);

for k = 1:numIterations

u\_old = u;

% Update u using TV denoising

u = u - lambda \* divergence(Dux, Duy);

% Compute gradient

[Dux, Duy] = gradient(u);

% Projection step

normDu = sqrt(Dux.^2 + Duy.^2);

normDu(normDu == 0) = 1;

Dux = Dux ./ normDu;

Duy = Duy ./ normDu;

% Ensure the solution is within the range [0, 255]

u = min(max(u, 0), 255);

% Check for convergence

if norm(u - u\_old, 'fro') / norm(u\_old, 'fro') < 1e-6

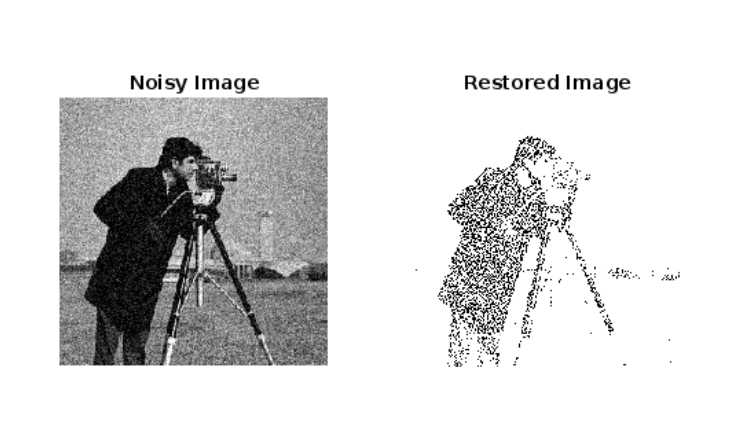
break;

end

end

denoisedImage = u;

end

****