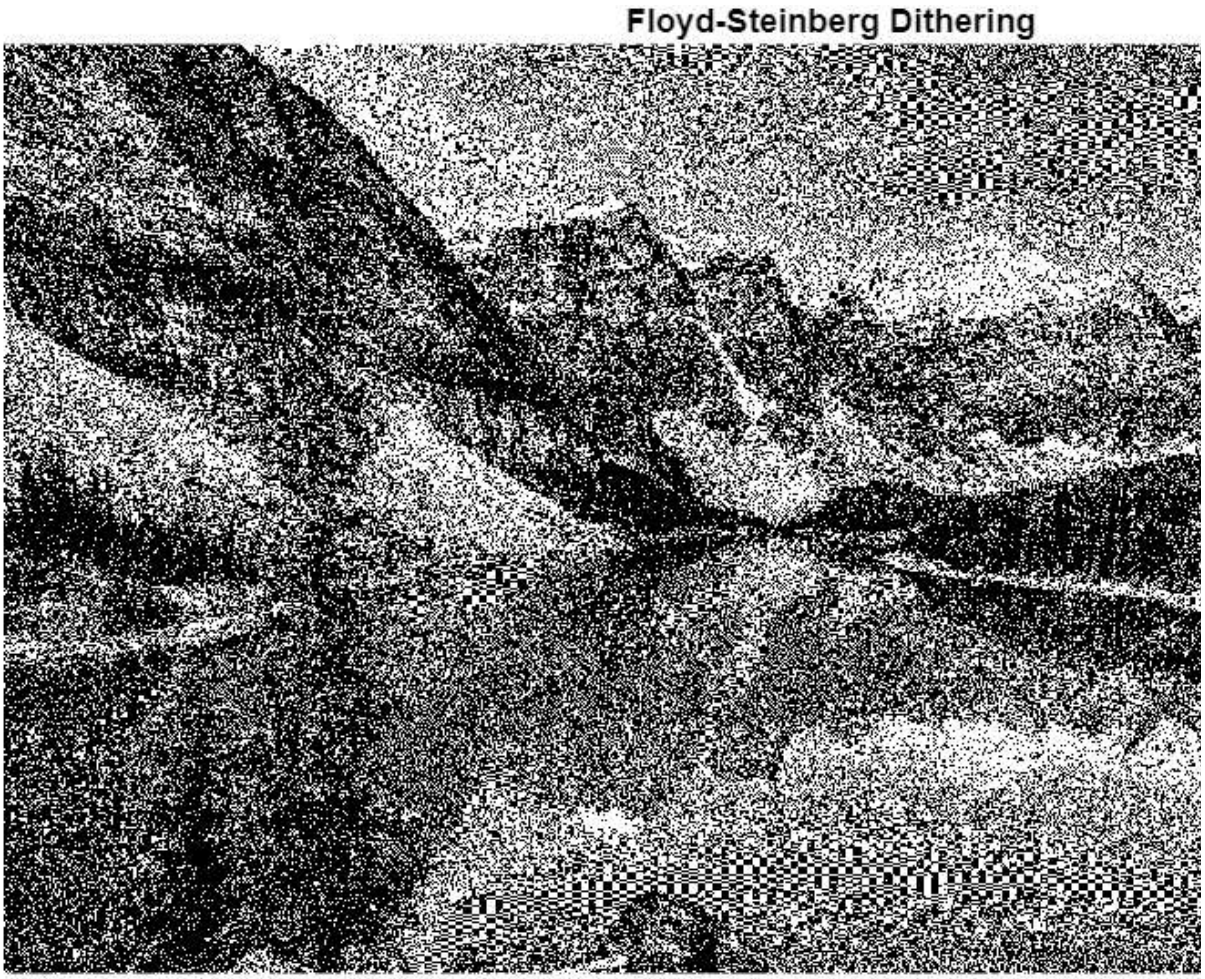
```
% Load the image
img = imread('image.jpg');
imshow(img);
title('Original Image');
```

Implement Floyd-Steinberg Dithering

```
% Convert the grayscale image to double
gray_img = im2double(gray_img);

% Floyd-Steinberg Dithering
floyd_img = gray_img;
[rows, cols] = size(gray_img);

for i = 1:rows
    for j = 1:cols
        old_pixel = floyd_img(i, j);
        new_pixel = round(old_pixel);
        floyd_img(i, j) = new_pixel;
        quant_error = old_pixel - new_pixel;

        if j < cols
            floyd_img(i, j+1) = floyd_img(i, j+1) + quant_error * 7/16;
```

```
        end
        if i < rows && j > 1
            floyd_img(i+1, j-1) = floyd_img(i+1, j-1) + quant_error * 3/16;
        end
        if i < rows
            floyd_img(i+1, j) = floyd_img(i+1, j) + quant_error * 5/16;
        end
        if i < rows && j < cols
            floyd_img(i+1, j+1) = floyd_img(i+1, j+1) + quant_error * 1/16;
        end
    end
end

% Display Floyd-Steinberg result
figure;
imshow(floyd_img);
title('Floyd-Steinberg Dithering');
```



Implement Jarvis-Judice-Ninke Dithering

```
% Convert the grayscale image to double
gray_img = im2double(gray_img);

% Get the size of the image
[rows, cols] = size(gray_img);
```

```
% Copy the grayscale image to modify
jjn_img = gray_img;

% Define the JJN error diffusion kernel
jjn_kernel = [0, 0, 0, 7, 5;
              3, 5, 7, 5, 3;
              1, 3, 5, 3, 1] / 48; % Normalize by dividing by 48

% Iterate over the image pixels
for i = 1:rows
    for j = 1:cols
        % Original pixel value
        old_pixel = jjn_img(i, j);

        % Quantize the pixel value (to either 0 or 1)
        new_pixel = round(old_pixel);
        jjn_img(i, j) = new_pixel;

        % Calculate the quantization error
        quant_error = old_pixel - new_pixel;

        % Distribute the error using the JJN kernel
        for ki = 1:3
            for kj = -2:2
                if (i+ki <= rows) && (j+kj > 0) && (j+kj <= cols)
                    jjn_img(i+ki, j+kj) = jjn_img(i+ki, j+kj) + quant_error * jjn_kernel(ki, kj+3);
                end
            end
        end
    end
end

% Display the dithered image
figure;
imshow(jjn_img);
title('Jarvis-Judice-Ninke Dithered Image');
```

Jarvis-Judice-Ninke Dithered Image