

```

% Phase 1: Load and Convert Image to Grayscale
input_image = imread('dog.jpg');
grayscale_image = rgb2gray(input_image); % Convert to grayscale
grayscale_image = double(grayscale_image) / 255; % Normalize to range [0, 1]

% Function to perform Floyd–Steinberg Dithering
function floyd_steinberg_dithered = floyd_steinberg_dithering(img)
    [rows, cols] = size(img);
    floyd_steinberg_dithered = img; % Initialize with the grayscale image

    for i = 1:rows
        for j = 1:cols
            old_pixel = floyd_steinberg_dithered(i, j);
            new_pixel = round(old_pixel); % Quantize to either 0 or 1
            floyd_steinberg_dithered(i, j) = new_pixel;
            quant_error = old_pixel - new_pixel;

            % Distribute the quantization error using Floyd–Steinberg
weights
            if j+1 <= cols
                floyd_steinberg_dithered(i, j+1) =
floyd_steinberg_dithered(i, j+1) + quant_error * 7/16;
            end
            if i+1 <= rows && j-1 >= 1
                floyd_steinberg_dithered(i+1, j-1) =
floyd_steinberg_dithered(i+1, j-1) + quant_error * 3/16;
            end
            if i+1 <= rows
                floyd_steinberg_dithered(i+1, j) =
floyd_steinberg_dithered(i+1, j) + quant_error * 5/16;
            end
            if i+1 <= rows && j+1 <= cols
                floyd_steinberg_dithered(i+1, j+1) =
floyd_steinberg_dithered(i+1, j+1) + quant_error * 1/16;
            end
        end
    end
end

% Function to perform Jarvis–Judice–Ninke (JJN) Dithering
function jjn_dithered = jarvis_judice_ninke_dithering(img)
    [rows, cols] = size(img);
    jjn_dithered = img; % Initialize with the grayscale image

    % Define JJN error diffusion weights
    jjn_weights = [0 0 0 7 5; 3 5 7 5 3; 1 3 5 3 1] / 48;

    for i = 1:rows
        for j = 1:cols

```

```

        old_pixel = jjn_dithered(i, j);
        new_pixel = round(old_pixel); % Quantize to either 0 or 1
        jjn_dithered(i, j) = new_pixel;
        quant_error = old_pixel - new_pixel;

        % Distribute the quantization error using JJN weights
        for dx = 0:2
            for dy = -2:2
                if i+dx <= rows && j+dy >= 1 && j+dy <= cols
                    jjn_dithered(i+dx, j+dy) = jjn_dithered(i+dx, j+dy)
+ quant_error * jjn_weights(dx+1, dy+3);
                end
            end
        end
    end
end

% Phase 2: Apply Floyd–Steinberg Dithering
floyd_steinberg_result = floyd_steinberg_dithering( grayscale_image );

% Phase 3: Apply Jarvis–Judice–Ninke Dithering
jjn_result = jarvis_judice_ninke_dithering( grayscale_image );

% Phase 4: Display the Results
figure;
subplot(1, 3, 1), imshow( grayscale_image ), title('Original Grayscale Image');
subplot(1, 3, 2), imshow( floyd_steinberg_result ), title('Floyd–Steinberg Dithered');
subplot(1, 3, 3), imshow( jjn_result ), title('Jarvis–Judice–Ninke Dithered');

```

**Original Grayscale Image**



**Floyd-Steinberg Dithered**



**Jarvis-Judice-Ninke Dithered**

