```
% Read image and convert to grayscale
I = imread('cat.jpeg');
grayI = rgb2gray(I);
I = double(grayI) / 255;  % Normalize the grayscale image to [0, 1]

% Initialize variables for dithering
[m, n] = size(I);
FS_dithered = I;  % Floyd-Steinberg result
JJN_dithered = I; % Jarvis-Judice-Ninke result

%% Floyd-Steinberg Dithering
for y = 1:m
    for x = 1:n
        old_pixel = FS_dithered(y, x);
        new_pixel = round(old_pixel);  % Quantize to 0 or 1
        FS_dithered(y, x) = new_pixel;
        error = old_pixel - new_pixel;

        % Spread error using Floyd-Steinberg weights
        if x+1 <= n
            FS_dithered(y, x+1) = FS_dithered(y, x+1) + error * 7/16;
        end
        if y+1 <= m && x-1 >= 1
            FS_dithered(y+1, x-1) = FS_dithered(y+1, x-1) + error * 3/16;
        end
        if y+1 <= m
            FS_dithered(y+1, x) = FS_dithered(y+1, x) + error * 5/16;
        end
        if y+1 <= m && x+1 <= n
            FS_dithered(y+1, x+1) = FS_dithered(y+1, x+1) + error * 1/16;
        end
    end
end

%% Jarvis-Judice-Ninke Dithering
for y = 1:m
    for x = 1:n
        old_pixel = JJN_dithered(y, x);
        new_pixel = round(old_pixel);  % Quantize to 0 or 1
        JJN_dithered(y, x) = new_pixel;
        error = old_pixel - new_pixel;

        % Spread error using Jarvis-Judice-Ninke weights
        if x+1 <= n
            JJN_dithered(y, x+1) = JJN_dithered(y, x+1) + error * 7/48;
        end
        if x+2 <= n
            JJN_dithered(y, x+2) = JJN_dithered(y, x+2) + error * 5/48;
        end
        if y+1 <= m
```

```
            if x-2 >= 1
                JJN_dithered(y+1, x-2) = JJN_dithered(y+1, x-2) + error *
3/48;
            end
            if x-1 >= 1
                JJN_dithered(y+1, x-1) = JJN_dithered(y+1, x-1) + error *
5/48;
            end
            JJN_dithered(y+1, x) = JJN_dithered(y+1, x) + error * 7/48;
            if x+1 <= n
                JJN_dithered(y+1, x+1) = JJN_dithered(y+1, x+1) + error *
5/48;
            end
            if x+2 <= n
                JJN_dithered(y+1, x+2) = JJN_dithered(y+1, x+2) + error *
3/48;
            end
        end
        if y+2 <= m
            if x-2 >= 1
                JJN_dithered(y+2, x-2) = JJN_dithered(y+2, x-2) + error *
1/48;
            end
            if x-1 >= 1
                JJN_dithered(y+2, x-1) = JJN_dithered(y+2, x-1) + error *
3/48;
            end
            JJN_dithered(y+2, x) = JJN_dithered(y+2, x) + error * 5/48;
            if x+1 <= n
                JJN_dithered(y+2, x+1) = JJN_dithered(y+2, x+1) + error *
3/48;
            end
            if x+2 <= n
                JJN_dithered(y+2, x+2) = JJN_dithered(y+2, x+2) + error *
1/48;
            end
        end
    end
end

%% Display the Results
figure;
subplot(1, 3, 1);
imshow(I);
title('Grayscale');

subplot(1, 3, 2);
imshow(FS_dithered);
title('Floyd-Steinberg');
```

```
subplot(1, 3, 3);
imshow(JJN_dithered);
title('Jarvis-Judice-Ninke');
```