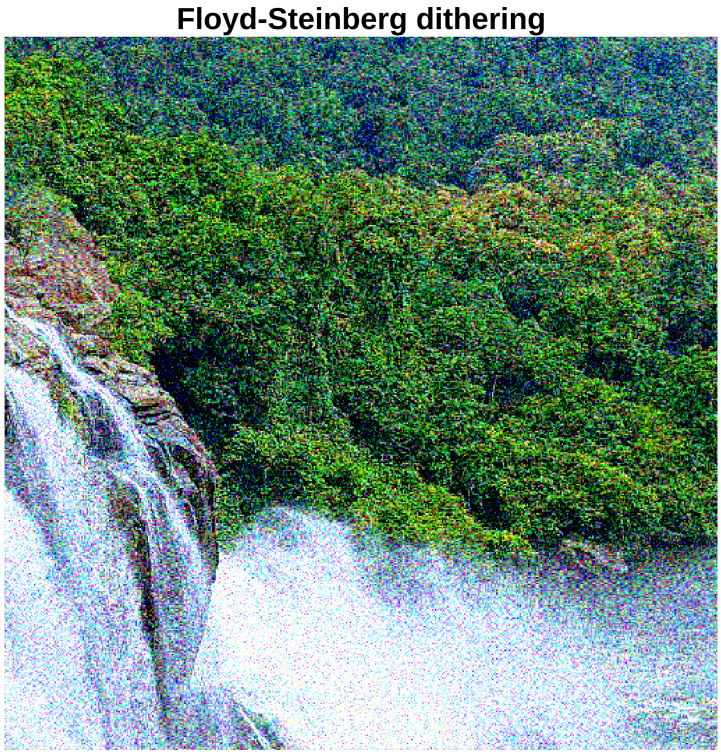
Implement both the Floyd-Steinberg and Jarvis-Judice-Ninke dithering algorithms on the image (MATLAB), then compare the results obtained from each method.

```
input_img=imread('image.jpeg');

%Floyd-Steinberg dithering

floyd_steinberg_dithered=Floyd_Steinberg_Dithering(input_img);
%Display results
imshow(floyd_steinberg_dithered),title('Floyd-Steinberg dithering');
```




```
%Jarvis-Judice-Ninke dithering
JJND_imag=double(input_img)/255;
% get image dimension
[R,C]=size(JJND_imag);
diffusionmatx = [
        0 0 0 7 5;
        3 5 7 5 3;
        1 3 5 3 1
    ] / 48;
```

```
for i=1:R-3
    for j=3:C-2
        oldpixel=JJND_imag(i,j);
        newpixel=round(oldpixel);
        JJND_imag(i,j)=newpixel;

        %error calculation
        q_error=oldpixel-newpixel;

        for diffuse_i=1:3
            for diffuse_j=-2:2
                ni=i+diffuse_i;
                nj=j+diffuse_j;

                if(ni<=R)&&(nj>=1)&&(nj<=C)
                    JJND_imag(ni,nj)=JJND_imag(ni,nj)
+q_error*diffusionmatx(diffuse_i,diffuse_j+3);
                else
                    % Print invalid index if out of bounds
                    fprintf('Invalid index at (i=%d, j=%d) -> (ni=%d, nj=%d)
\n', i, j, ni, nj);
                end
            end
        end
    end
end
JJND_imag=uint8(max(min(JJND_imag*255,255),0));


%Display results
imshow(JJND_imag),title('Jarvis-Judice-Ninke dithering');
```

Jarvis-Judice-Ninke dithering