

```

% Loading the image from the specified path and converting it to grayscale
image = imread('C:\Users\USER\Desktop\Assignment Image.JPG');
gray_image = rgb2gray(image);
gray_image = double(gray_image);

% Floyd-Steinberg dithering function
function fs_dithered = floyd_steinberg_dither(image)
    [rows, cols] = size(image);
    for y = 1:rows
        for x = 1:cols
            old_pixel = image(y, x);
            new_pixel = 255 * (old_pixel > 127); % Quantizing the pixel
            image(y, x) = new_pixel;
            error = old_pixel - new_pixel;

            % Spreading the error to neighboring pixels
            if x + 1 <= cols
                image(y, x + 1) = image(y, x + 1) + error * 7 / 16;
            end
            if y + 1 <= rows
                if x - 1 >= 1
                    image(y + 1, x - 1) = image(y + 1, x - 1) + error * 3 / 16;
                end
                image(y + 1, x) = image(y + 1, x) + error * 5 / 16;
                if x + 1 <= cols
                    image(y + 1, x + 1) = image(y + 1, x + 1) + error * 1 / 16;
                end
            end
        end
    end
    fs_dithered = uint8(image);
end

% Jarvis-Judice-Ninke dithering function
function jjn_dithered = jarvis_judice_ninke_dither(image)
    [rows, cols] = size(image);
    kernel = [0 0 0 7 5;
              3 5 7 5 3;
              1 3 5 3 1] / 48; % Defining the Jarvis-Judice-Ninke kernel

    for y = 1:rows
        for x = 1:cols
            old_pixel = image(y, x);
            new_pixel = 255 * (old_pixel > 127); % Quantizing the pixel
            image(y, x) = new_pixel;
            error = old_pixel - new_pixel;

            % Spreading the error using the Jarvis-Judice-Ninke kernel
            for ky = 1:3
                for kx = 1:5

```

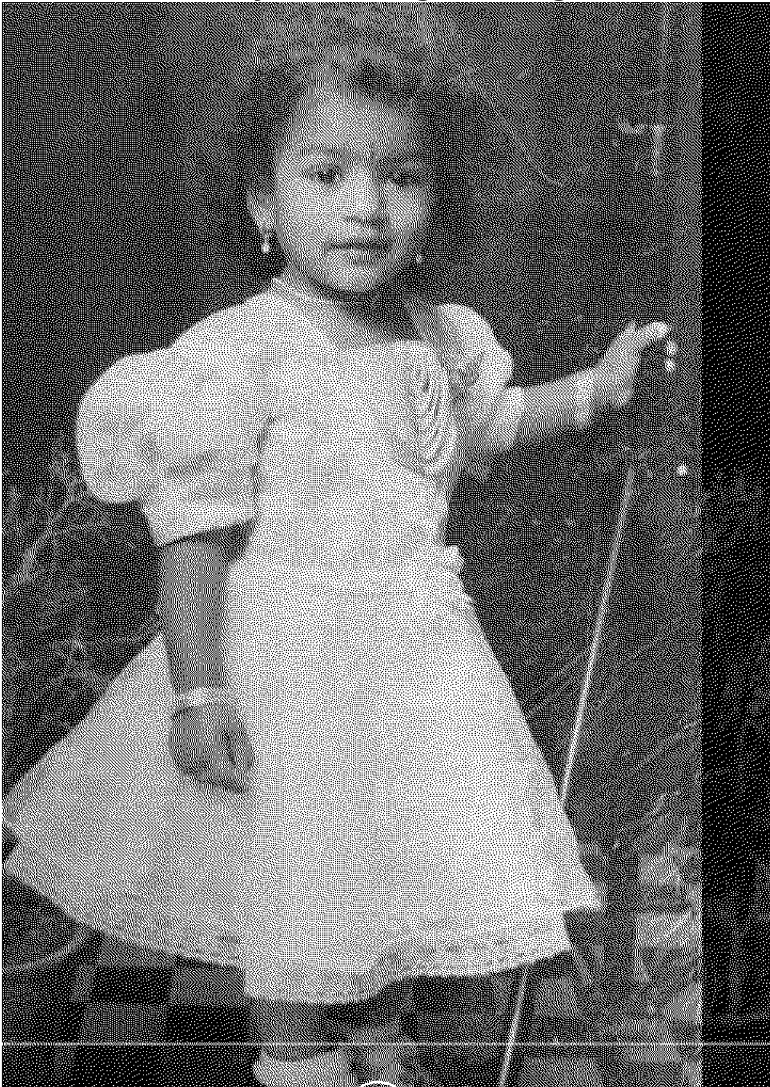
```

        if (y + ky - 1 <= rows) && (x + kx - 3 >= 1) && (x + kx - 3 <=
cols)
            image(y + ky - 1, x + kx - 3) = image(y + ky - 1, x + kx -
3) + error * kernel(ky, kx);
        end
    end
end
end
end
    jjn_dithered = uint8(image);
end

% Applying Floyd-Steinberg Dithering and saving the result
floyd_dithered = floyd_steinberg_dither(gray_image);
imwrite(floyd_dithered, 'C:\Users\USER\Desktop\floyd_dithered.png');
figure, imshow(floyd_dithered), title('Floyd-Steinberg Dithering');

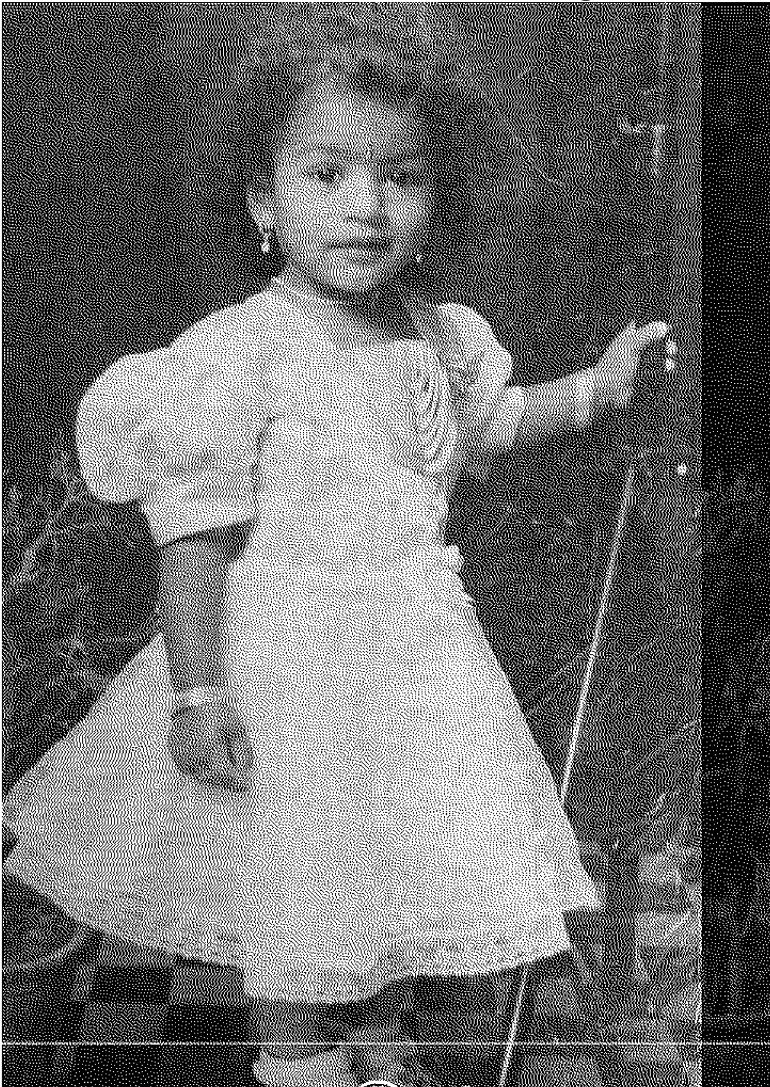
```

Floyd-Steinberg Dithering



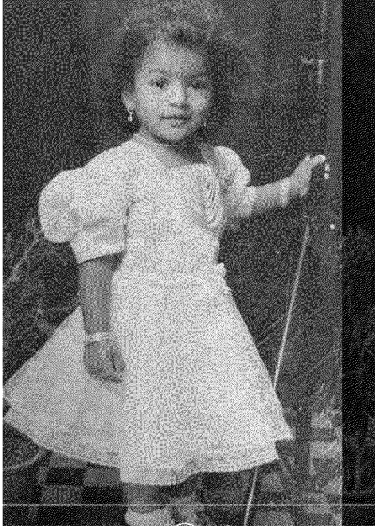
```
% Applying Jarvis-Judice-Ninke Dithering and saving the result
jarvis_dithered = jarvis_judice_ninke_dither(gray_image);
imwrite(jarvis_dithered, 'C:\Users\USER\Desktop\jarvis_dithered.png');
figure, imshow(jarvis_dithered), title('Jarvis-Judice-Ninke Dithering');
```


Jarvis-Judice-Ninke Dithering



```
% Comparing the results side-by-side
figure;
subplot(1, 2, 1);
imshow(floyd_dithered), title('Floyd-Steinberg Dithering');
subplot(1, 2, 2);
imshow(jarvis_dithered), title('Jarvis-Judice-Ninke Dithering');
```

Floyd-Steinberg Dithering



Jarvis-Judice-Ninke Dithering



```
% Calculating difference between the two dithered images  
difference = abs(double(floyd_dithered) - double(jarvis_dithered));  
figure, imshow(uint8(difference)), title('Difference Between Dithering Methods');
```

Difference Between Dithering Methods

