
```
% Local Binary Pattern
% Created on: 02/04/25
% By Author: Supratit Datta, BT22ECE127

clear;
close all;
clc;

% Read the input image
input_image = imread('input.jpeg');

% Convert the image to grayscale
gray_image = rgb2gray(input_image);
[rows, cols] = size(gray_image);

% Initialize the LBP image
lbp_image = zeros(rows, cols, 'uint8'); % Store results in an 8-bit unsigned
integer image

% Calculate the Local Binary Pattern (LBP) for each pixel
for i = 2 : (rows - 1)
    for j = 2 : (cols - 1)
        % Center pixel value
        center_pixel = gray_image(i, j);

        % Gather the 8 neighboring pixels in a specific order
        neighbors = [ ...
            gray_image(i - 1, j - 1), ... % top-left
            gray_image(i - 1, j), ... % top
            gray_image(i - 1, j + 1), ... % top-right
            gray_image(i, j + 1), ... % right
            gray_image(i + 1, j + 1), ... % bottom-right
            gray_image(i + 1, j), ... % bottom
            gray_image(i + 1, j - 1), ... % bottom-left
            gray_image(i, j - 1) ... % left
        ];

        % Initialize a binary pattern container
        binary_pattern = zeros(1, 8);

        % Compare each neighbor to the center pixel
        for idx = 1 : 8
            if neighbors(idx) >= center_pixel
                binary_pattern(idx) = 1;
            else
                binary_pattern(idx) = 0;
            end
        end

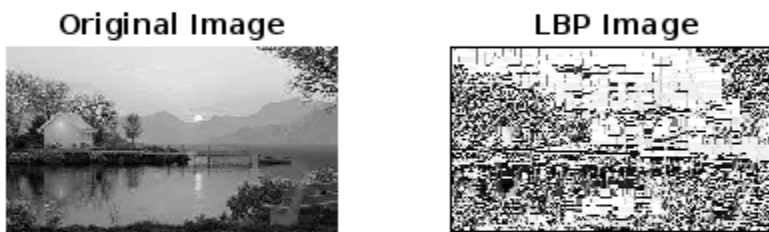
        % Convert the 8-bit binary pattern to a decimal value
        % The leftmost comparison corresponds to the highest-order bit
        lbp_value = 0;
```

```
for k = 1 : 8
    lbp_value = lbp_value + binary_pattern(k) * 2^(8 - k);
end

% Assign the computed LBP value to the pixel
lbp_image(i, j) = lbp_value;
end
end

% Display the original and LBP images
figure;
subplot(1, 2, 1);
imshow(gray_image);
title('Original Image');

subplot(1, 2, 2);
imshow(lbp_image);
title('LBP Image');
```



Published with MATLAB® R2024b