Contents


```
%Code written by Soham Roy

clear;
close all;
clc;
```

Read the input image

```
input_image = imread('lena_color.tiff');
```

Convert the image to grayscale

```
gray_image = rgb2gray(input_image);
```

Get the dimensions of the image

```
[rows, cols] = size(gray_image);
```

Initialize the Local Binary Pattern image

```
lbp_image = zeros(rows, cols);
```

Define the 8 neighbors

```
neighbors = [
    -1 -1; -1 0; -1 1;
     0 -1;       0 1;
     1 -1;  1 0;  1 1;
];
```

Calculate the Local Binary Pattern for each pixel

```
for i = 2:rows-1
    for j = 2:cols-1
        center_pixel = gray_image(i, j);
        binary_pattern = zeros(1, 8);
        for k = 1:8
            neighbor_pixel = gray_image(i + neighbors(k, 1), j + neighbors(k, 2));
```

```
            binary_pattern(k) = neighbor_pixel >= center_pixel;
        end
        % Convert the binary pattern to a decimal value
        lbp_image(i, j) = sum(binary_pattern .* 2 .^ (7:-1:0));
    end
end
```

Display the original and LBP images

```
figure;
subplot(1, 2, 1);
imshow(gray_image);
title('Original Image');
subplot(1, 2, 2);
imshow(uint8(lbp_image));
title('Local Binary Pattern Image');
```

Published with MATLAB® R2021a