

## Descriptor LBP (Local Binary Patterns)

Este descriptor emplea una codificación binaria para realizar clasificación de texturas, asimismo se basa en un cálculo del vecindario del pixel de interés. Como resultado se obtiene un histograma que permite describir una imagen de forma concisa.

Dada una imagen original, se procede a trabajar con un kernel de 3x3 (según indican los autores), y se aplica la codificación binaria.

### Etapa 1: Codificación

Imagen Original				
202	160	150	227	245
149	110	238	150	174
218	24	52	6	73
114	179	135	76	133
128	129	13	191	218



Imagen Original				
202	160	150	227	245
149	110	238	150	174
218	24	52	6	73
114	179	135	76	133
128	129	13	191	218



Código Binario		
1	1	1
1		1
1	0	0

Como primer paso, se compara el pixel central con los 8 vecinos que tiene alrededor, y se coloca un 1 cuando el pixel central es menor o igual que el vecino, y se coloca 0 en caso contrario. Con esto se obtiene un código binario, que luego debe convertirse en un número decimal.

Código Binario	→	$\begin{matrix} 1_6 & 1_7 & 1_0 \\ 1_5 & & 1_1 \\ 1_4 & 0_3 & 0_2 \end{matrix}$	x	$\begin{matrix} 1_7 & 1_6 & 1_5 & 1_4 & 0_3 & 0_2 & 1_1 & 1_0 \end{matrix}$	$\begin{matrix} 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{matrix}$	= 243
----------------	---	---	---	---	---	-------

Con el valor que se convierte a decimal (partiendo siempre desde la misma posición para todas las operaciones), se coloca en la imagen resultante:

Imagen Resultante				
	243			

A continuación se muestra el proceso para calcular los códigos de los pasos 2 y 3:

Paso 2	Imagen Original					Código Binario	→	0												
	202	160	150	227	245				0	0	0	0	0	0	0	0	0	0	0	0
	149	110	238	150	174															
	218	24	52	6	73															
	114	179	135	76	133															
	128	129	13	191	218															

Paso 3	Imagen Original					Código Binario	→	0												
	202	160	150	227	245				1	1	1	1	1	1	0	0	0	1	1	1
	149	110	238	150	174															
	218	24	52	6	73															
	114	179	135	76	133															
	128	129	13	191	218															

Los 3 primeros valores obtenidos son los que se ilustran a continuación:

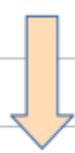
Imagen Resultante				
243	0	227		

Los valores de los bordes no se pueden calcular, ya que no existe información suficiente para hacerlo. Con ello, la información de los bordes se pierde. Esto no afecta en gran medida a imágenes medianas y grandes.

### Etapa 2: Mapeo

En base a los códigos binarios obtenidos, se procede a realizar un mapeo a un conjunto de posibles valores que señalan Ojala et al.

202	160	150
149	110	238
218	24	52



Código Binario		
1 <sub>6</sub>	1 <sub>7</sub>	1 <sub>0</sub>
1 <sub>5</sub>		1 <sub>1</sub>
1 <sub>4</sub>	0 <sub>3</sub>	0 <sub>2</sub>



2 cambios

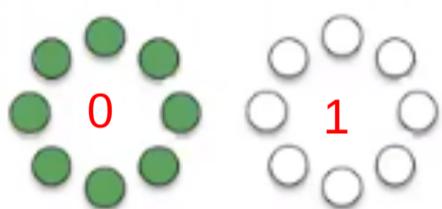


Con este resultado, se mapea un número dado:

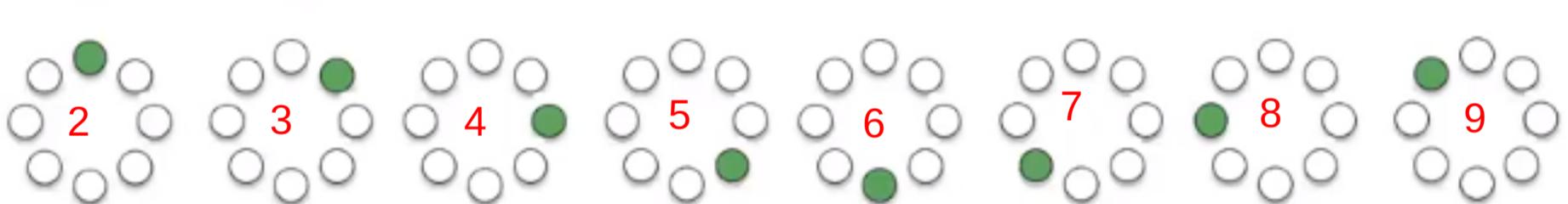
## Uniform patterns

2 + 56 = 58  
patterns

U = 0  
2  
patterns



U = 2  
8x7 = 56  
patterns



:

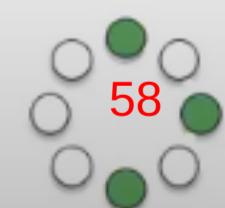
## Non-uniform patterns

256 - 58 = 198  
patterns

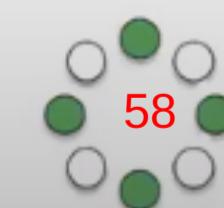
U = 4



U = 6



U = 8



:

:

:

Con los códigos binarios y valores decimales, mapeamos al código que corresponde según señalan Ojala et al:

Imagen Mapeada				
23	1	21		

### Etapa 3: Cálculo del histograma

Como paso final, calculamos el histograma, es decir contamos cuántas veces aparece cada código mapeado:

