

# LOCAL BINARY PATTERN



0	50	14	34	6	10	28	48	106	16
106	6	124	191	191	130	264	166	16	6
66	157	251	237	209	239	238	227	67	7
166	167	233	235	214	234	239	236	66	1
66	179	209	189	219	211	166	180	76	2
97	186	64	10	168	124	13	30	62	62
166	181	191	168	227	178	149	182	166	36
74	155	252	236	231	149	178	226	43	6
214	176	149	234	167	64	162	79	66	191
234	147	166	222	230	127	122	36	131	201
214	128	66	162	142	76	60	2	166	241
66	250	75	0	81	47	0	4	217	10

Local Binary Pattern



# WHERE LOCAL BINARY PATTERNS USED

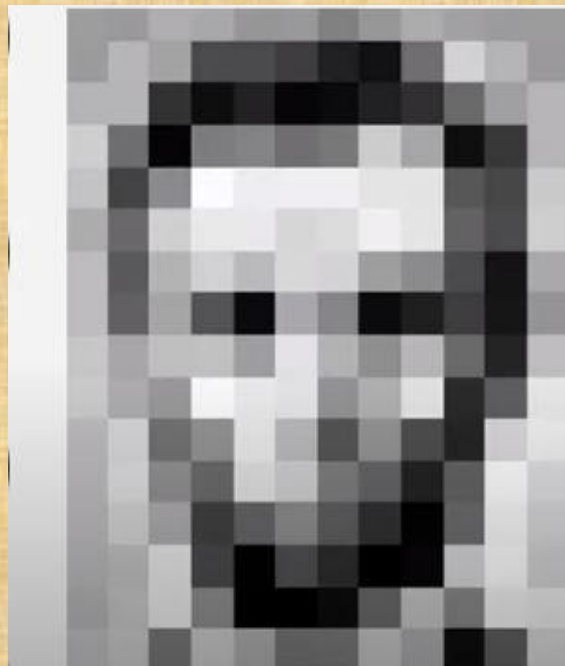
- Local binary pattern (LBP) is a popular technique used for image/face representation and classification.
- Local Binary Patterns (LBPs) have been used for a wide range of applications
  1. Face detection
  2. Face recognition
  3. Facial expression recognition
  4. Remote sensing
  5. Texture classification
  6. Object detection systems



# HOW LOCAL BINARY PATTERN WORKS?

- The most common approach however, dictates that each  $3 \times 3$  window in the image is processed to extract an LBP code.
- The processing involves thresholding the center pixel of that window with its surrounding pixels using the window mean, window median or the actual center pixel, as thresholds.

# How LBP Works ?!



187	163	174	168	150	162	129	183	172	161	165	166
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	105	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	230	239	228	98	74	204
188	88	179	209	185	216	211	168	139	75	20	169
189	97	165	84	10	168	124	11	31	62	22	168
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	96	150	79	58	218	241
190	224	147	108	227	210	127	103	36	101	255	224
190	214	173	66	103	163	95	80	2	109	249	219
187	196	236	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	263	236
196	206	123	207	177	121	123	200	175	13	96	218

187	163	174	168	150	162	129	183	172	161	165	166
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	105	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	230	239	228	98	74	204
188	88	179	209	185	216	211	168	139	75	20	169
189	97	165	84	10	168	124	11	31	62	22	168
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	96	150	79	58	218	241
190	224	147	108	227	210	127	103	36	101	255	224
190	214	173	66	103	163	95	80	2	109	249	219
187	196	236	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	263	236
196	206	123	207	177	121	123	200	175	13	96	218




# OUTPUT VALUE OF **LBP OPERATOR** CAN BE OBTAINED AS FOLLOWS:

---

$$\text{LBP} = \sum_{n=0}^7 s(i_n - i_c) 2^n$$

Here,  $i_c$  = Center Pixel Value  
 $i_n$  = Neighbour Pixel Values



A diagram showing a 3x3 grid of pixels. The center pixel is highlighted with a black circle. A mouse cursor is pointing at the center pixel. The function  $s(z)$  is defined as:

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0. \end{cases}$$

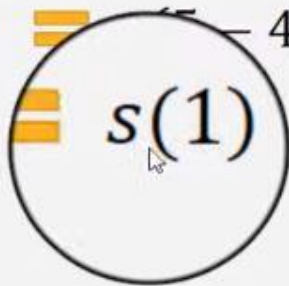
# FOR EXAMPLE

Here,  $i_c$  = Center Pixel Value  
 $i_n$  = Neighbour Pixel Value

$$\sum_{n=0}^7 s(i_n - i_c)2^n$$

For,  $n = 0$

$$s(i_0 - i_c)2^0 = s(5 - 4)2^0$$



$i_0$	$i_1$	$i_2$
$i_7$	$i_c$	$i_3$
$i_6$	$i_5$	$i_4$

5	9	1
4	4	6
7	2	3


# FOR EXAMPLE

Here,  $i_c$  = Center Pixel Value  
 $i_n$  = Neighbour Pixel Value

$$\sum_{n=0}^7 s(i_n - i_c)2^n$$

For,  $n = 0$

$$s(i_0 - i_c)2^n = s(5 - 4)2^0$$

$$= s(1)$$


$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0. \end{cases}$$

$i_0$	$i_1$	$i_2$
$i_7$	$i_c$	$i_3$
$i_6$	$i_5$	$i_4$

<b>5</b>	<b>9</b>	<b>1</b>
<b>4</b>	<b>4</b>	<b>6</b>
<b>7</b>	<b>2</b>	<b>3</b>

# FOR EXAMPLE

Here,  $i_c$  = Center Pixel Value  
 $i_n$  = Neighbour Pixel Value

For,  $n = 0$

$1 > 0$   
1

<b>1</b>	<b>9</b>	<b>1</b>
<b>4</b>	<b>4</b>	<b>6</b>
<b>7</b>	<b>2</b>	<b>3</b>

$i_0$	$i_1$	$i_2$
$i_7$	$i_c$	$i_3$
$i_6$	$i_5$	$i_4$

<b>5</b>	<b>9</b>	<b>1</b>
<b>4</b>	<b>4</b>	<b>6</b>
<b>7</b>	<b>2</b>	<b>3</b>





# FOR EXAMPLE

Here,  $i_c$  = Center Pixel Value  
 $i_n$  = Neighbour Pixel Values

$$\sum_{n=0}^7 s(i_n - i_c) 2^n$$

For,  $n = 1$

$$s(i_1 - i_c) 2^1 = s(9 - 4) 2^1$$

$$= s(10)$$

$10 > 0$   
 $1$

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0. \end{cases}$$

$i_0$	$i_1$	$i_2$
$i_7$	$i_c$	$i_3$
$i_6$	$i_5$	$i_4$

# FOR EXAMPLE

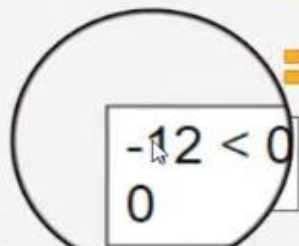
Here,  $i_c$  = Center Pixel Value  
 $i_n$  = Neighbour Pixel Values

$$\sum_{n=0}^7 s(i_n - i_c) 2^n$$

For,  $n = 2$

$$s(i_2 - i_c) 2^2 = s(1 - 4) 2^2$$

$$= s(-12)$$



$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0. \end{cases}$$

$i_0$	$i_1$	$i_2$
$i_7$	$i_c$	$i_3$
$i_6$	$i_5$	$i_4$

1	1	0
1		1
1	0	0

## FOR EXAMPLE : METHOD 2

10	12	18
7	9	6
9	2	4

If neighbor pixel

$\geq$	9	= 1
$<$	9	= 0

1	1	1
0		0
1	0	0

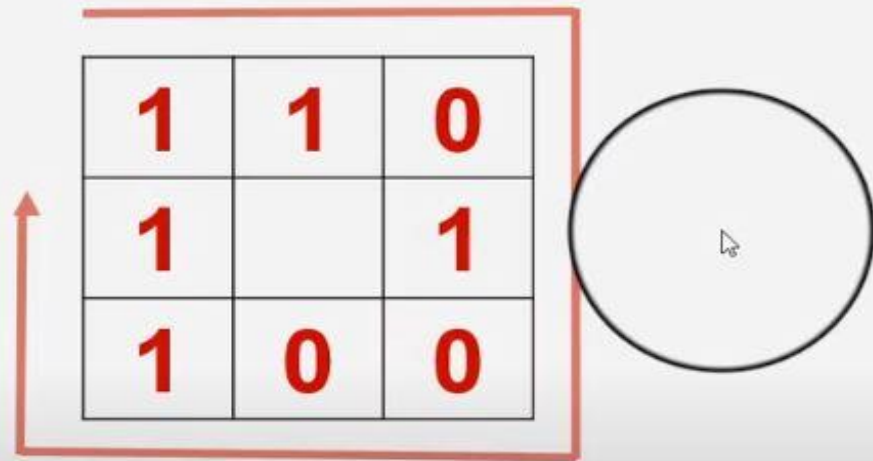
Binary Number Generated

1	1	1	0	0	0	1	0
---	---	---	---	---	---	---	---

Decimal Number = 226

# FOR EXAMPLE

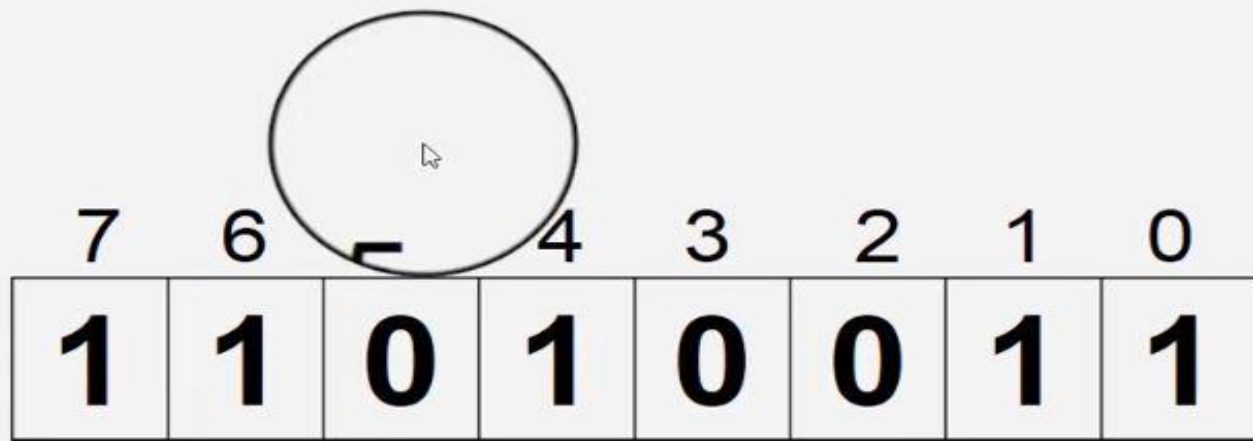
---



1	1	0	1	0	0	1	1
---	---	---	---	---	---	---	---

# BINARY NUMBER INTO DECIMAL

1	1	0
1		1
1	0	0



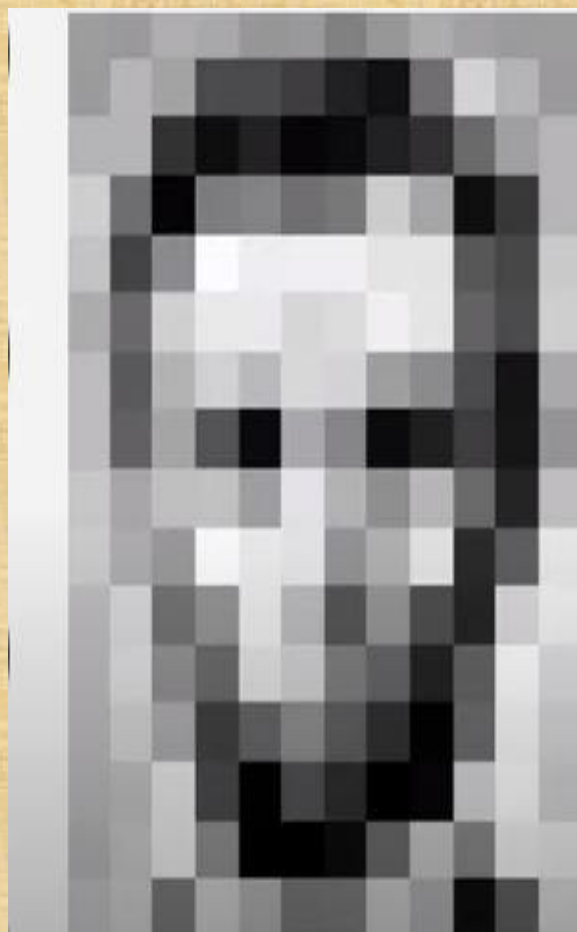
Binary Number Generated

$$\begin{aligned} &= 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^4 + 1 \times 2^6 + 1 \times 2^7 \\ &= 1 + 2 + 16 + 64 + 128 \\ &= 211 \text{ (LBP Code Generated)} \end{aligned}$$



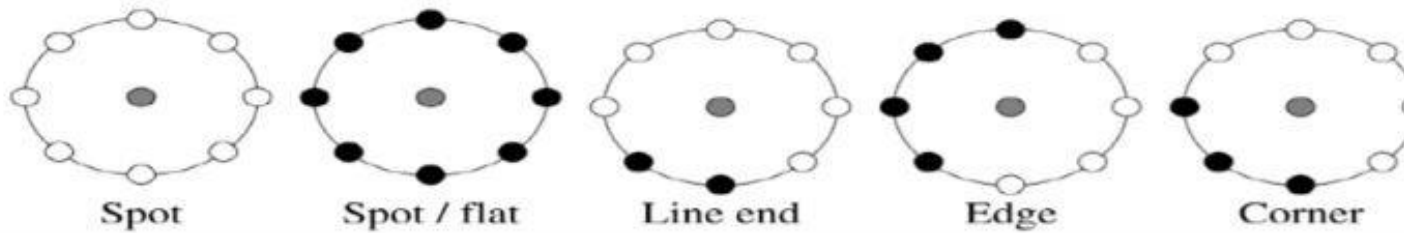
# LOCAL BINARY PATTERNS





187	153	174	168	190	182	129	183	172	161	155	156
195	182	163	74	75	62	33	17	110	210	180	164
180	180	50	14	34	6	10	33	48	105	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	230	239	224	68	74	204
188	88	178	209	185	216	211	168	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	146
199	168	191	193	158	227	178	143	182	105	36	195
205	174	155	252	235	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	103	36	101	255	224
190	214	173	66	103	163	35	50	2	199	249	216
187	196	235	75	1	81	47	0	6	217	258	211
183	202	237	145	0	0	12	108	200	138	263	236
196	206	123	207	177	121	133	200	175	19	96	218

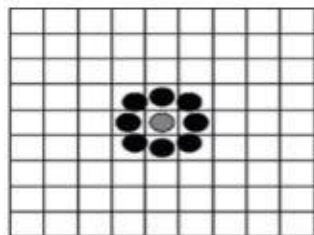
187	153	174	168	190	182	129	183	172	161	155	156
195	182	163	74	75	62	33	17	110	210	180	164
180	180	50	14	34	6	10	33	48	105	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	230	239	224	68	74	204
188	88	178	209	185	216	211	168	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	146
199	168	191	193	158	227	178	143	182	105	36	195
205	174	155	252	235	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	163	35	50	2	199	249	216
187	196	235	75	1	81	47	0	6	217	258	211
183	202	237	145	0	0	12	108	200	138	263	236
196	206	123	207	177	121	133	200	175	19	96	218



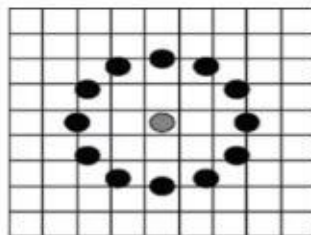
**Fig. 2.3** Different texture primitives detected by the LBP

# Advanced LBP $(P,R)$

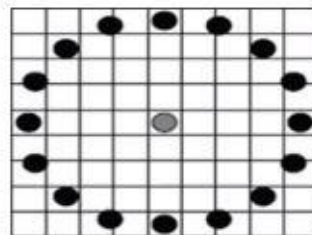
P = Pixels  
R = Radius



LBP(8,1)



LBP(16,2)



LBP(20,4)

## uLBP

- Uniformity measure  $U$  (“pattern”) is the number of bitwise transitions from 0 to 1 or vice versa.
- A local binary pattern is called uniform if its **uniformity measure is at most 2**. i.e transitions between 0 and 1  $\leq 2$

## Example

- 00000000 (0 transitions)
- 01110000 (2 transitions)
- 11001111 (2 transitions)
- 11001001 (4 transitions)
- 01010011 (6 transitions)



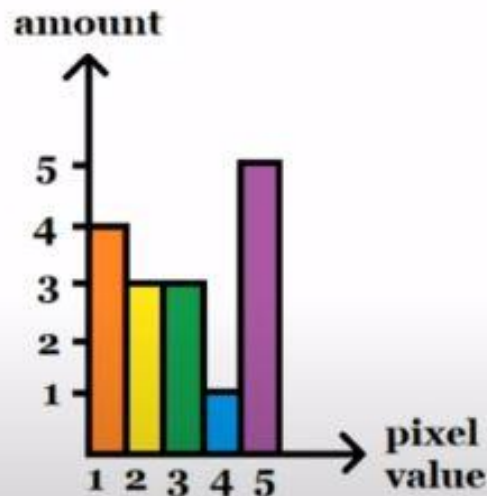
## Reasons for omitting non-uniform patterns

- most of the local binary patterns in natural images are uniform
- Ojala et al. noticed that in texture images, uLBP account for
  - 90% of all patterns using the (8,1)
  - 70% in the (16, 2) neighborhood.
- Facial images
  - 90.6% of the patterns in the (8, 1)
  - 85.2% of the patterns in the (8, 2)

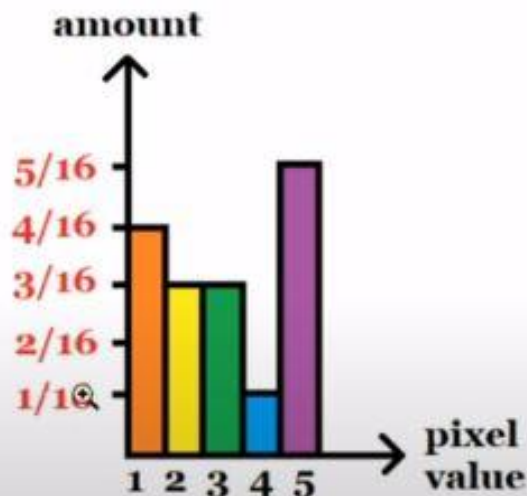
# Histogram of LBP (1)

3	1	5	3
5	2	5	1
4	3	1	2
2	1	5	5

LBP



Histogram



Normalized  
histogram