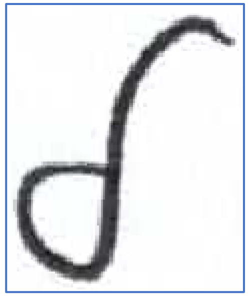
Texture features

Feature detection, description and extraction

- Feature detection – detect features
  - e.g. detect edges in an image
- Feature description – describe the features (assign quantitative attributes to the detected features)
  - e.g. orientation, magnitude
  - A feature descriptor converts an image to a feature vector of size n
- Feature extraction is both feature detection and description

→ [4, 0.25, 1.7, …..]

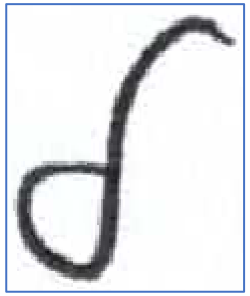


→ [1.44, 10.015, 0.65, …..]

Feature extraction

- A feature descriptor simplifies the image by extracting useful information
- Typically, a feature descriptor converts an image/image region to a feature vector of length n
- The feature vector is very useful for tasks like image recognition and object detection
  - The feature vectors can be fed into an image classification algorithms (e.g. Support Vector Machine (SVM))

Feature extraction pipeline



Input image

Feature extraction

[1.44, 10.015, 0.65, …..]

Feature vector

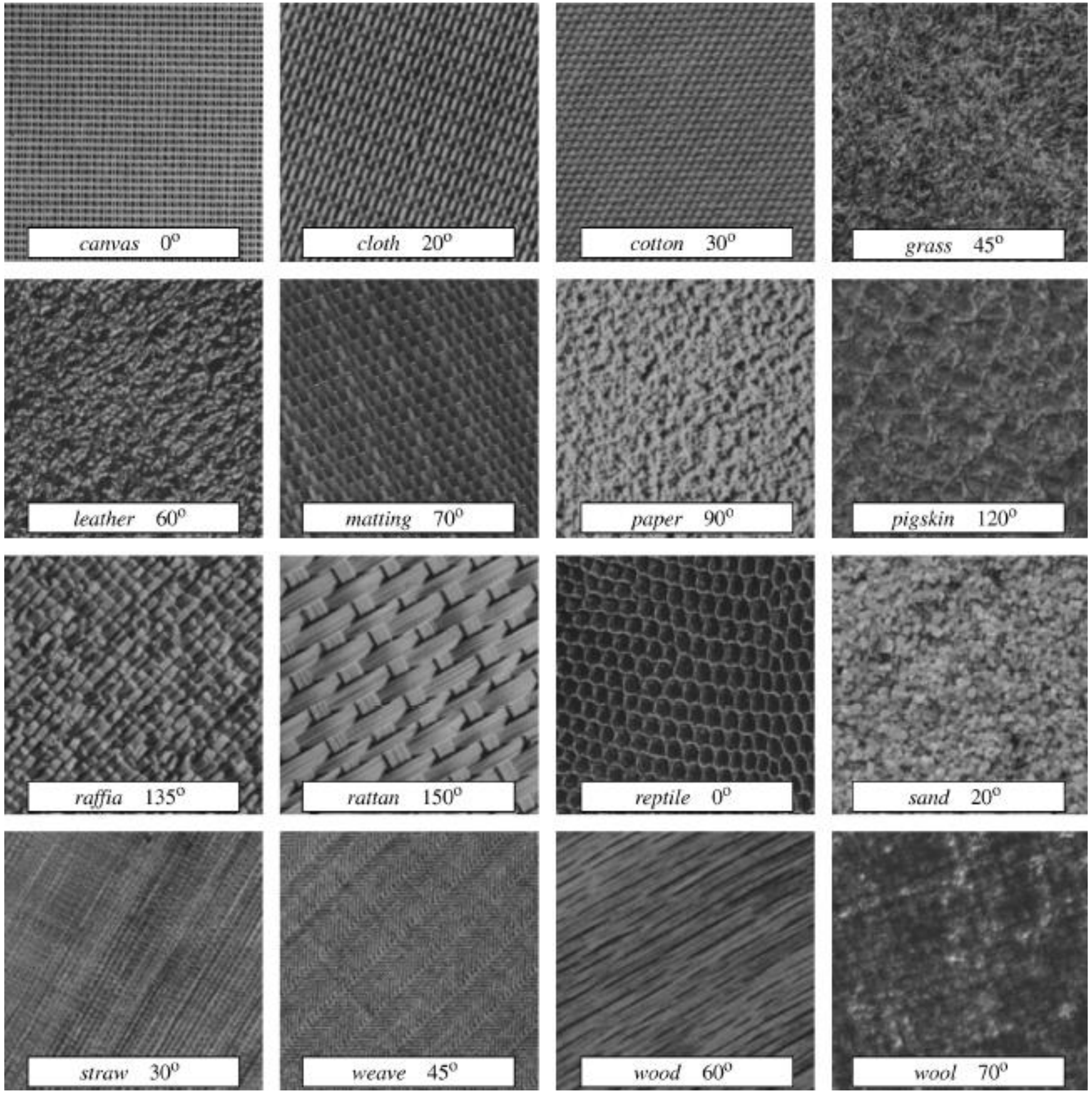
Feature Vector: An abstraction of an image (or image region) used to characterize and numerically quantify the contents of an image

Examples of well-known features

- Local Binary Patterns (LBP)
- Scale-Invariant Feature Transform (SIFT)
- Oriented FAST and Located BRUEF(ORB)
- Histogram of Oriented Gradients (HOG)
- and many more

What is texture

- No formal definition exist
- Intuitively:
    - Surface characteristics (smooth, coarse, regular, grainy, etc.)



Where we can use texture features?

- Distinguish between rough and smooth surfaces
  - Distinguish between sand, water or rock surfaces
- Satellite image analysis (e.g. analyze surface of Mars)
- Autonomous vehicles (determining if a road is paved)
- and many more

Which texture descriptors will we learn?

- Local Binary Patterns (LBP)

Local Binary Patterns (LBP)

- Introduced by Ojala el. al. in their paper "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", 2002
- Depend on the local region around each pixel.
- Compare each pixel with its surrounding neighborhood

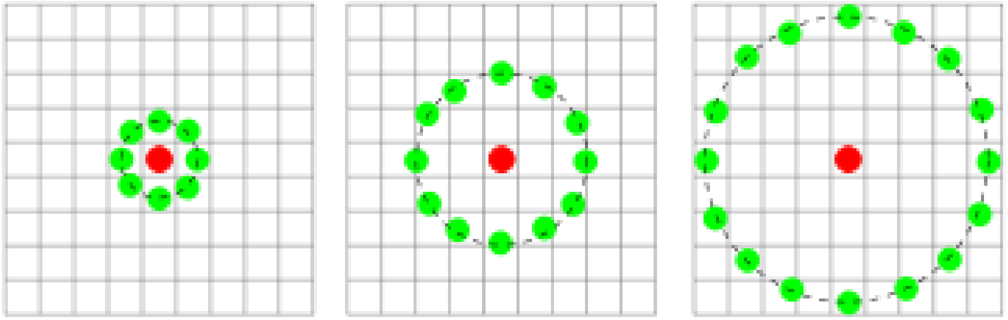
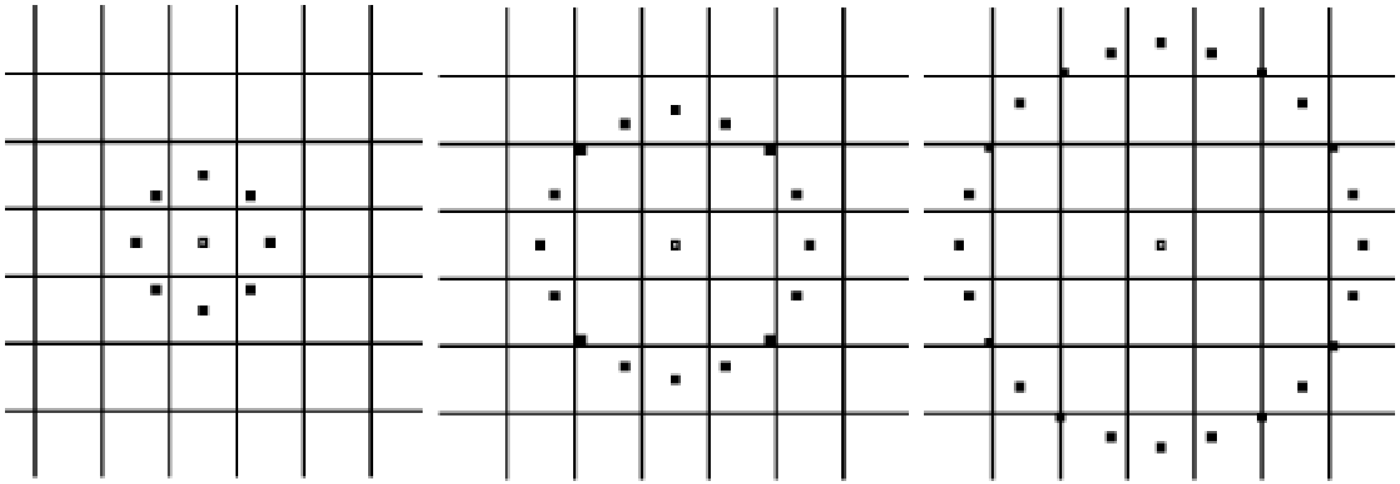


Image reference: Wikipedia

Local Binary Patterns (LBP)

- Input: a **grayscale** image
- Choose the radius r and number p of pixels on the circle.
- Compare the gray values of pixels on the circle to the central pixel



(P=8,R=1.0) (P=16,R=2.0) (P=24,R=3.0)

Local Binary Patterns (LBP)

- Choose the radius r and number p of pixels on the circle.
- Compare the gray values of pixels on the circle to the central pixel

255	**0**	**5**
85	15	124
3	2	125

→

1	**0**	**0**
1		1
0	0	1

- If the center pixel is greater or equal to its neighbor, set the value to 0;
- Otherwise, set the value to 1

Local Binary Patterns (LBP)

- With p =8 neighbors, there are 2^8 combinations
- Look at the values as a binary number

1	**0**	**0**
1		1
0	0	1

Most significant bit

$$11001100_2 = 204$$

- When the image is rotated , the value of MSB will correspondingly move along the perimeter of the circle around central point = > different LBP values

Local Binary Patterns (LBP)

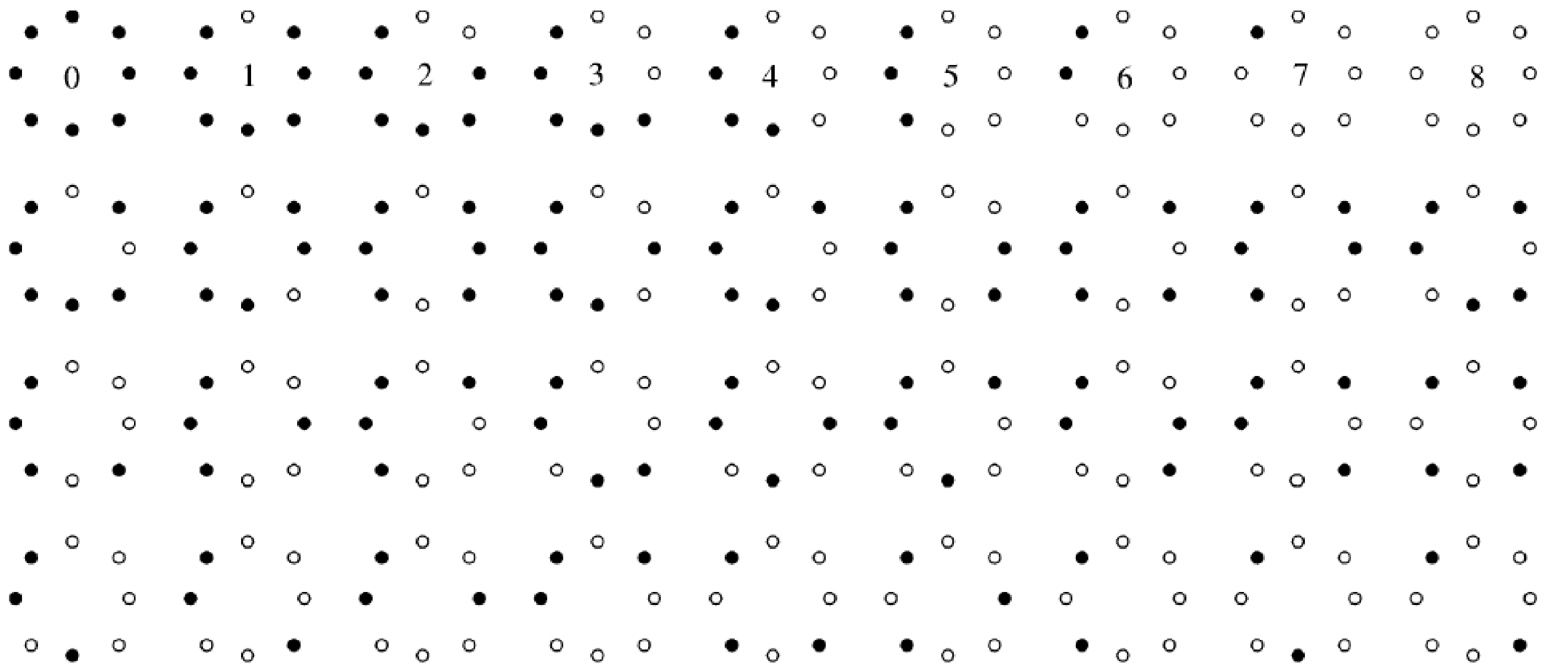
- To remove the effect of rotation, i.e., to assign a unique identifier, the LBP is defined to be minimum value

1	**0**	**0**
1		1
0	0	1

00110011
01100110
11001100
10011001
….

Local Binary Patterns (LBP)

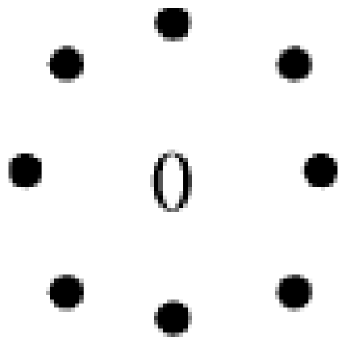
- LBP quantifies the occurrences statistics of individual rotation invariant patterns corresponding to certain micro-features in the image

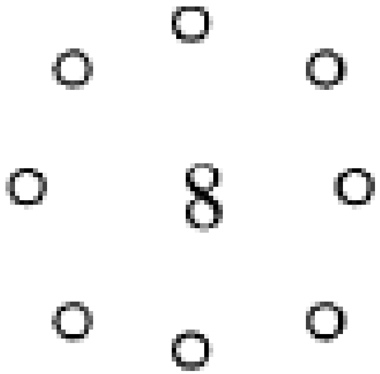


The 36 unique rotation invariant binary patterns that can occur in the circularly symmetric neighbor set of LBP. Black and white circles correspond to bit values of 0 and 1 in the 8-bit output of the operator. The first row contains the nine uniform patterns and the numbers inside them correspond to their unique LBP codes.

Local Binary Patterns (LBP)

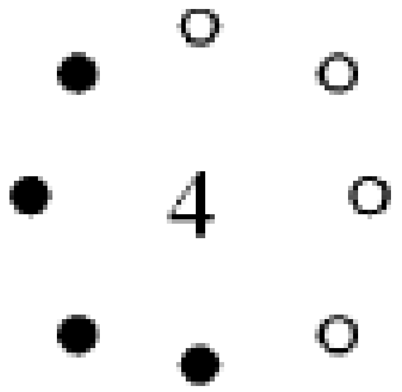
- Bright region: center pixel surrounded by intensities smaller than itself
- Dark region: center pixel surrounded by intensities greater (or equal) than itself





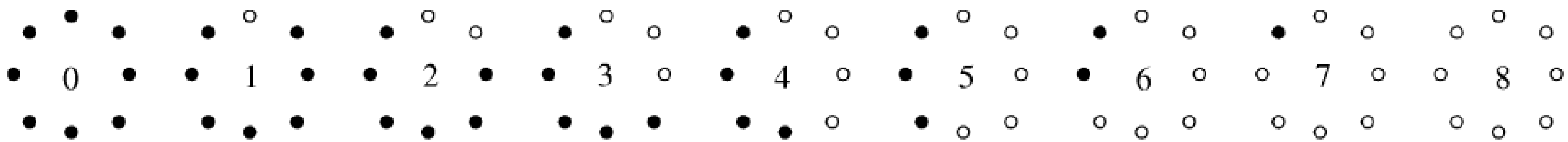
Local Binary Patterns (LBP)

- Edge : transition from dark to light



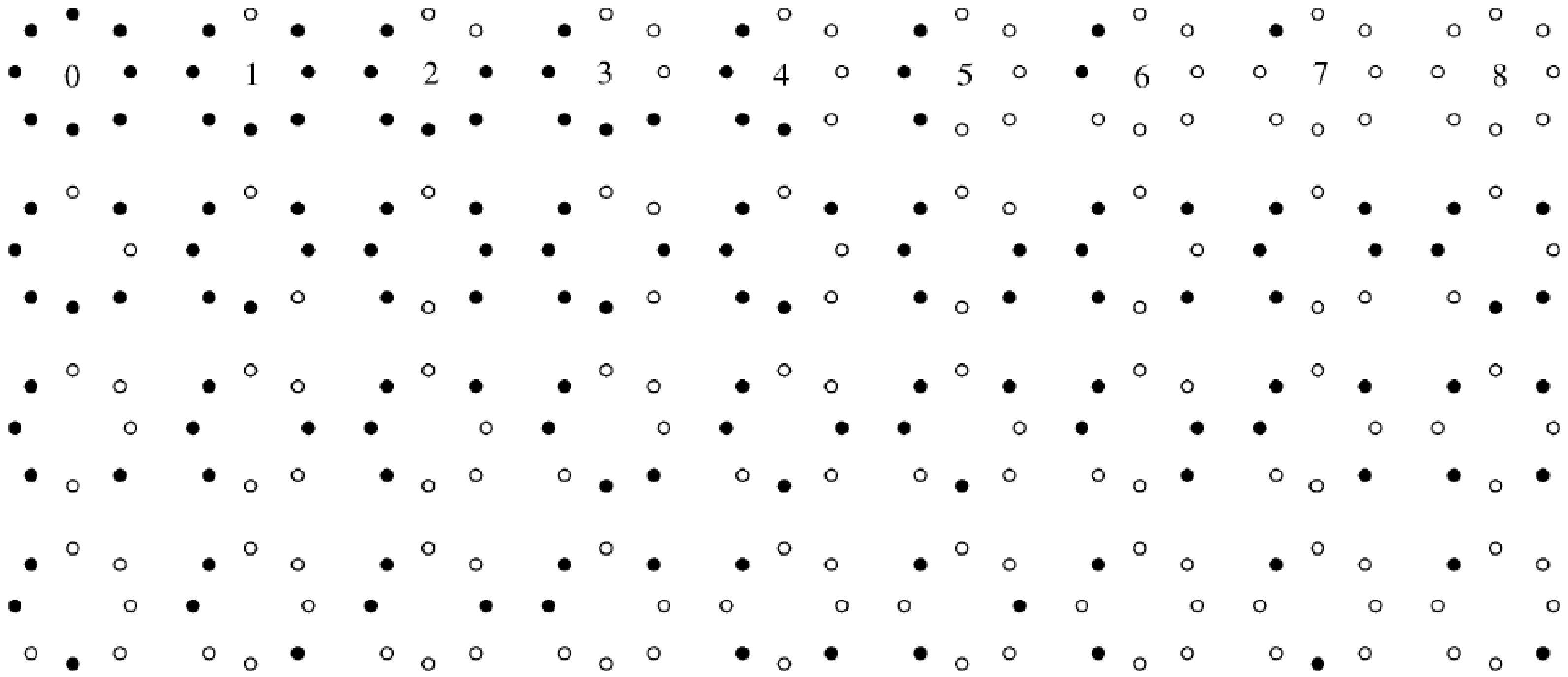
Local Binary Patterns (LBP)

- Ojala el. al. observed that curtain local binary patterns are fundamental properties of texture, providing the vast majority (up to 90%) of all patterns in the observed textures
- They call them “*uniform*” patterns



Local Binary Patterns (LBP)

- A LBP is uniform if it has at most 2 transition from 0 to 1 or from 1 to 0



- Ojala el. al. observed that if we have p points on the circle, there are p+1 uniform LBP

Local Binary Patterns (LBP)

- Ojala el. al. observed that if we have p points on the circle, there are p+1 uniform LBP
- Finally, they build a histogram with p+2 bins (one bin per each of the p+1 uniform patterns and the last bin for all other patterns)
- For certain applications, it is all important to capture spatial information of LBP.
  - Divide the image into cells, compute histogram in each cell, then concatenate the histograms

Where do LBP implemented?

- LBP implemented both in [mahotas] and [scikit-image] libraries

```
from skimage import feature

# extract the histogram of Local Binary Patterns
lbp = feature.local_binary_pattern(im, numPoints, radius, method="uniform")

(hist, _) = np.histogram(lbp.ravel(), bins=range(0, numPoints + 3),
                         range=(0, numPoints + 2))

# optionally normalize the histogram
eps = 1e-7
hist = hist.astype("float")
hist /= (hist.sum() + eps)
```