Feature Extraction from image- Wavelet texture

Uday Mewada(CS171089)
[udaymewada1@gmail.com]

Karina Rathore(CS183D08)
[karinarathore0705@gmail.com]

December 13, 2020

Contents

1. **Introduction**
    1.1 What is Texture
    1.2 What is GLCM?
2. **Mathematical Formulation**
    2.1 Calculating Contrast
    2.2 Calculating Correlation
    2.3 Calculating Energy
    2.4 Calculating Entropy
    2.5 Calculating Homogeneity
    2.6 Calculating Maximum Probability
3. **Algorithm**
4. **Documentation of API**
    4.1 Package organization
    4.2 Methods
5. **Examples**
6. **Learning Outcome**
7. **References**

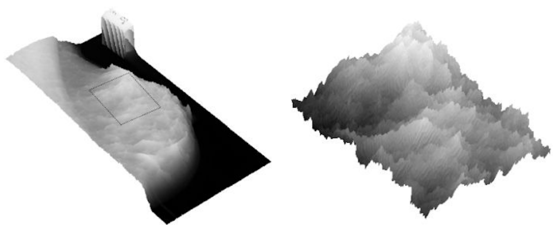
Chapter 1
Introduction

1.1 What is Texture

Texture is that innate property of all surfaces that describes visual patterns, and that contain important information about the structural arrangement of the surface and its relationship to the surrounding environment. Texture consists of primitive or texture elements called texels.

Texture is a repeating pattern of local variations in image intensity

Texture cannot be defined for any particular point.



In image processing, the texture can be defined as a function of spatial variation of the brightness intensity of the pixels.The texture represents the variations of each level, which measures characteristics such as smoothness, smoothness, coarseness and regularity of each surface in different order directions. Textural images in the image processing and machine vision refer to the images in which a specific pattern of distribution and dispersion of the intensity of the pixel illumination is repeated sequentially throughout the image.

For example in below image, the image of a wall of wood with a fully repetitive texture. This pattern is repeated throughout the image. The bamboo pattern is also shown

Fig: An example of a textural image: (a)texture image; (b) repetitive pattern of texture

Another example, an image has a 50% black and 50% white distribution of pixels.

Two different images with the same intensity distribution, but with different textures.

Texture is an important feature of an image. To describe the texture of the region three approaches are used in image processing; these are statistical, structural and spectral. Statistical approaches specify the characterization of the textures by smooth, coarse, grainy, and silky and so on. The common second order statistic is the gray level co- occurrence matrix.

1.2 What is Gray Level Co-occurrence Matrix (GLCM)?

GLCM is performed on Haar Wavelet because in Haar wavelet transform the resulting wavelet bands are strongly correlated with the orientation elements in the GLCM computation. The second reason is because the total pixel entries for Haar wavelet transform is always minimum. Thus, the GLCM computation burden can be reduced.

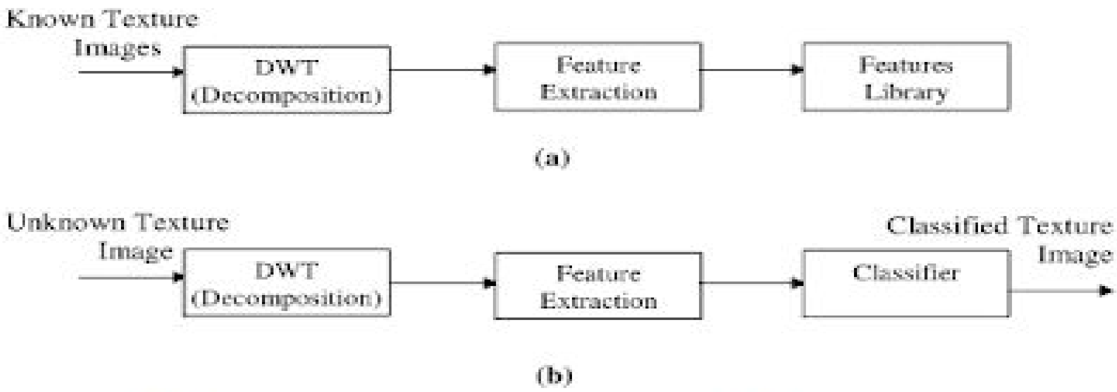
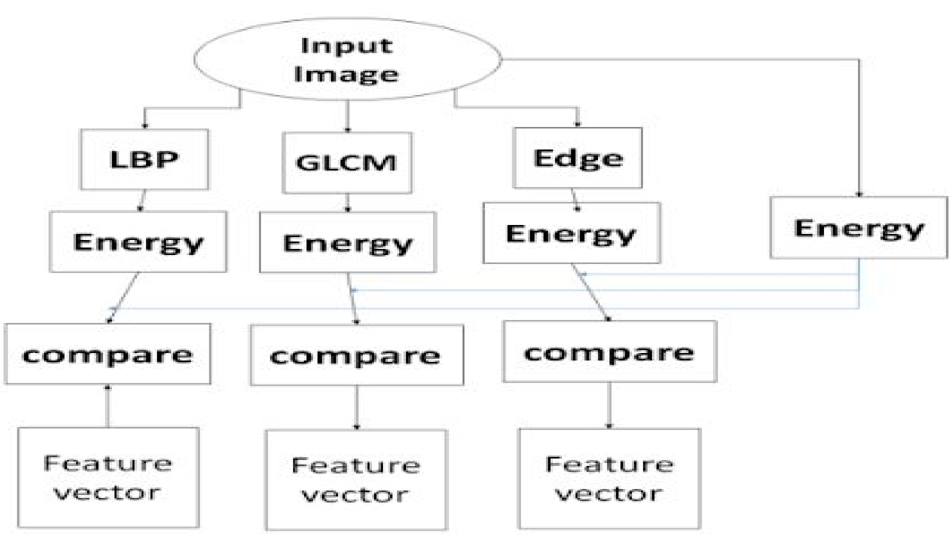


Fig. 2. (a) Texture training steps, (b) texture classification steps.

In order to use information contained in the GLCM, there are defined some statistical measures to extract textual characteristics. Some of these features are entropy, contrast, correlation etc. Contrast measures the local variation in the gray level of GLCM. Correlation measures the joint probability of occurrence of pixel pairs of GLCM. Energy gives the sum of squared pixel values of GLCM. Homogeneity refers to the closeness of distribution of elements to the GLCM diagonal. Homogeneous textures contain ideal repetitive structures. Weak homogeneity refers to variation in texture elements in their spatial arrangements.
The GLCM element P (i, j, d, Θ) represent probability of the pair of pixels, which are located with an inter sample distance d and a direction Θ, have a gray level i and a gray level j. If the inter-pixel distance is set to 1 or 2 GLCM measure the local high frequency information.



Chapter 2
Mathematical Formulation

2.1 Calculating Contrast

The contrast of image can be calculated as-

Contrast = $\sum_{i,j} |i-j|^2 p_{i,j}$

2.2 Calculating Correlation

The correlation of image can be calculated as-

Correlation=$\sum_{i,j} (i-\mu i)(j-\mu j) p_{i,j} / \sigma_i \sigma_j$

2.3 Calculating Energy

The energy of image can be calculated as-

Energy E = $\sum_{i,j} p_{i,j}^2$

2.4 Calculating Entropy

The entropy of image can be calculated as-

Entropy =$-\sum_{i,j} p_{i,j} \log_2 p_{i,j}$

2.5 Calculating Homogeneity

The homogeneity of image can be calculated as-

Homogeneity = $\sum_{i,j} p_{i,j} / 1+|i-j|$

2.6 Calculating Maximum Probability

The maximum probability can be calculated as-

Maximum probability = $\max(p_{i,j})$

Chapter 3
Algorithm

Input : image

Output : here we contrast, homogeneity,energy, correlation of the given image as output

`skimage.feature.`**`greycomatrix`**(*image*, *distances*, *angles*, *levels=None*, *symmetric=False*, *normed=False*)

Calculate the grey-level co-occurrence matrix.

Parameters:

Image array_like

Integer typed input image.

distances array_like

List of pixel pair distance offsets.

angles array_like

List of pixel pair angles in radians.

levels int, optional

symmetric bool, optional

If True, the output matrix *P[:, :, d, theta]* is symmetric

normed bool, optional

Default is false

Returns

P : 4-D ndarray

`skimage.feature.`**`greycoprops`**(*P*, *prop='contrast'*)

Compute a feature of a grey level co-occurrence matrix to serve as a compact summary of the matrix.

Parameters

P : nd array

Input array. *P* is the grey-level co-occurrence histogram for which to compute the specified property.

prop{‘contrast’, ‘dissimilarity’, ‘homogeneity’, ‘energy’, ‘correlation’, ‘ASM’}, optional

The property of the GLCM to compute. The default is ‘contrast’.

Returns

results 2-D ndarray

Parameters

P : ndarray

Input array. *P* is the grey-level co-occurrence histogram for which to compute the specified property.

prop{‘contrast’, ‘dissimilarity’, ‘homogeneity’, ‘energy’, ‘correlation’, ‘ASM’}, optional

The property of the GLCM to compute. The default is ‘contrast’.

Returns

results2-D ndarray

2-dimensional array. `results[d, a]` is the property ‘prop’ for the d’th distance and the a’th angle

Chapter 4
Documentation of API

4.1 Package organization

skimage
numpy
greycoprops
greycomatrix
io
color

4.2 Methods

io.imread() Load an image from file.

color.rgb2gray()This example converts an image with RGB channels into an image with a single grayscale channel.The value of each grayscale pixel is calculated as the weighted sum of the corresponding red, green and blue pixels.

img_as_ubyte() Convert an image to unsigned byte format, with values in [0, 255]

skimage.feature.greycoprops() Calculate texture properties of a GLCM.

Chapter 5
Examples

```
import numpy as np
from  skimage.feature import greycomatrix, greycoprops
from skimage import io,color, img_as_ubyte

def contrast_feature(x):
    contrast=greycoprops(x,'contrast')
    return "Contrast =",contrast

# Load image=
img=io.imread('C:/ProgramData/imgg.png')
gray = color.rgb2gray(img)
image = img_as_ubyte(gray)
max_value = img.max()
matrix_coocurrence = greycomatrix(image, [1], [0, np.pi/4, np.pi/2],
         levels=max_value+1, normed=False, symmetric=False)

print(contrast_feature(matrix_coocurrence))
```

Output :

Contrast = array([[88.85209577, 172.36065734, 87.05059215]])

Chapter 6
Learning Outcome

- ➔ Understand wavelet transform
- ➔ Decomposition of image at level 1
- ➔ Extracting the texture of image

Chapter 7
References

[1] Wavelet tutorial by Robi Polikar.
[2] https://arxiv.org/pdf/1904.06554.pdf
[3] http://www.ijiset.com/v1s6/IJISET_V1_I6_64.pdf
[4]https://www.researchgate.net/publication/5606814_Wavelet-based_Texture_Retrieval_using_Generalized_Gaussian_Density_and_Kullback-Leibler_Distance