Contents

Chapter 1

Introduction

1.1 What is texture?

Although there is no strict definition of the image texture, it is easily perceived by humans and is believed to be a rich source of visual information about the nature and three-dimensional shape of physical objects. Generally speaking, textures are complex visual patterns composed of entities, or sub-patterns, that have characteristic brightness, color, slope, size, etc. Thus texture can be regarded as a similarity grouping in an image The local sub-pattern properties give rise to the perceived lightness, uniformity, density, roughness, regularity, linearity, frequency, phase, directionality, coarseness, randomness, fineness, smoothness, granulation, etc., of the texture as a whole.

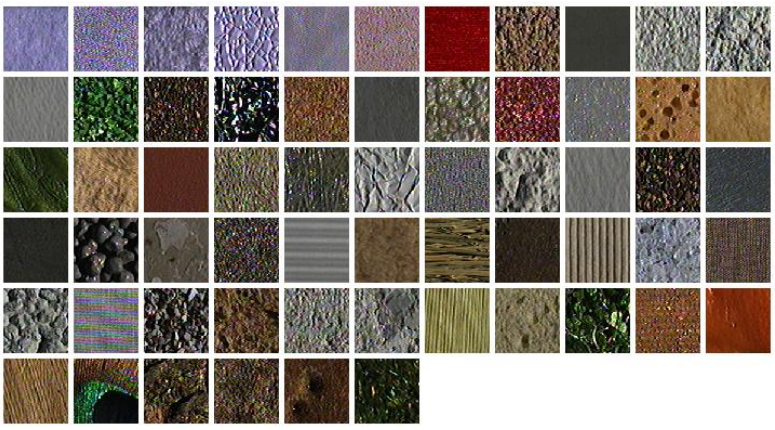


Figure 1.1: Different Textures

1.2 Texture Applications

There are four major issues in texture analysis:

1. Feature extraction: to compute characteristic of a digital image, able to numerically describe its texture properties
2. Texture discrimination: to partition a textured image into regions, each corresponding to a perceptually homogeneous texture (leads to image segmentation)
3. Texture classification: to determine to which of a finite number of physically defined classes (such as normal and abnormal tissue) a homogeneous texture region belongs
4. Shape from texture: to reconstruct 3D surface geometry from texture information.

Feature extraction is the first stage of image texture analysis. Results obtained from this stage are used for texture discrimination, texture classification or object shape determination.

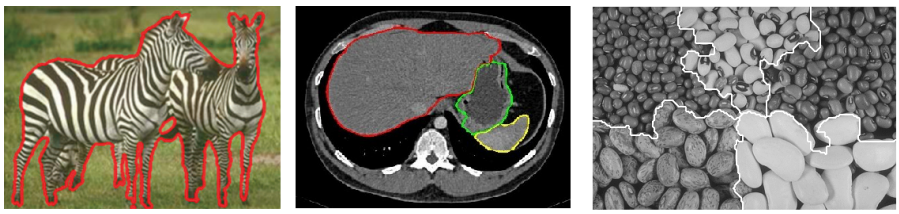


Figure 1.2: Using textural features for image segmentation

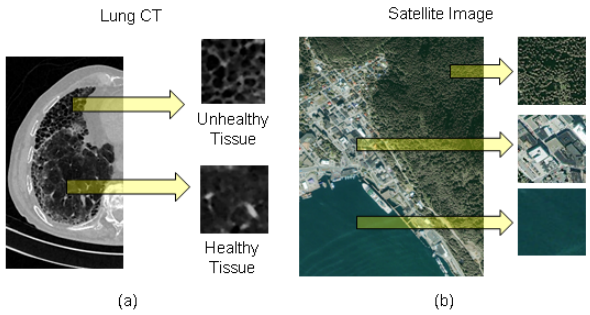


Figure 1.3: Using textural features for image classification

1.3 Motivation

The biggest motivation for choosing textural features for image classification was that my thesis will be mainly about medical image segmentation based on texture. This project was a good starting point for me to get familiar with GLCM textural features and gave me the chance to successfully classify different textures using several classifiers and have a comparison on the results.

Chapter 2

Machine Learning Procedure

In general,for the task of image classification we have the following basic steps: Data acquisition and pre-processing, Feature extraction, Classification, Deriving the accuracy assessment measures. The purpose of feature extraction is the measurement of those attributes of patterns that are most pertinent to a given classication task.

2.1 Data Acquisition

For this project, I have used the "'Kylberg Texture Dataset v. 1.0"' database[1]. The data set comes in two versions: *without* rotated texture patches and *with* rotated texture patches.

However, not all the texture classes are being used. I've chosen a subset of 6 different textures including: "Cushion", "Grass", "Lin seeds", "Rice",

Data set with rotations	
number of texture classes	28
number of rotations	12
rotation increments	30 degree
number of unique samplesclass	1920
total number of samples	53760
texture patch size	576 by 576 pixels
image format	8bit gray scale PNG
total size of data set	10.3 GB

Table 2.1: Properties for the texture dataset with rotated texture patches

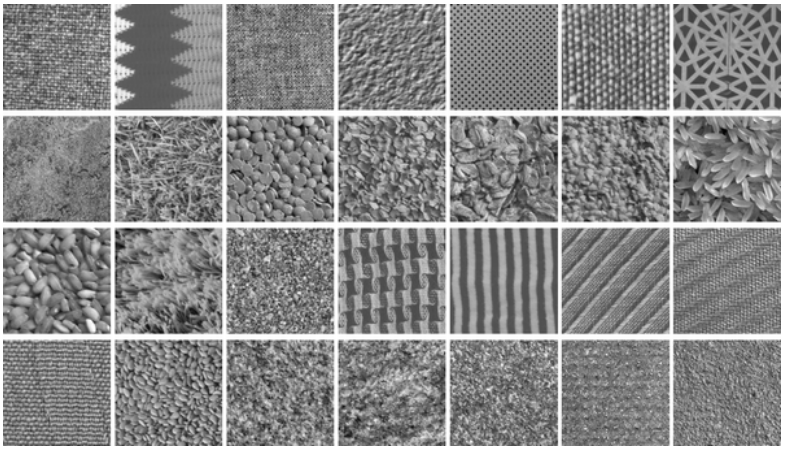
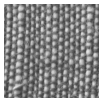
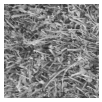


Figure 2.1: Kylberg Texture Dataset



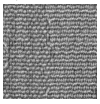
(a) Cushion



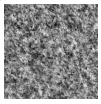
(b) Grass

(c) Linseeds

(d) Rice



(e) Seat



(f) Stone

Figure 2.2: Data set used in this project: 6 different texture classes

"Seat" and "Stone". (Including 2232 training examples and 648 testing examples)

2.2 Feature Extraction

Approaches to texture analysis are usually categorized into:

- structural,
- statistical,
- model-based and
- transform

Structural approaches represent texture by well defined primitives (microtexture) and a hierarchy of spatial arrangements (macrotexture) of those primitives. To describe the texture, one must define the primitives and the placement rules. The choice of a primitive (from a set of primitives) and the probability of the chosen primitive to be placed at a particular location can be a function of location or the primitives near the location. The advantage of the structural approach is that it provides a good symbolic description of the image; however, this feature is more useful for synthesis than analysis tasks. The abstract descriptions can be ill defined for natural textures because of the variability of both micro- and macrostructure and no clear distinction between them. A powerful tool for structural texture analysis is provided by mathematical morphology. It may prove to be useful for bone image analysis, e.g. for the detection of changes in bone microstructure.

In contrast to structural methods, statistical approaches do not attempt to understand explicitly the hierarchical structure of the texture. Instead, they represent the texture indirectly by the non-deterministic properties that govern the distributions and relationships between the grey levels of an image. Methods based on second-order statistics (i.e. statistics given by pairs of pixels) have been shown to achieve higher discrimination rates than the power spectrum (transform-based) and structural methods (Weszka 1976). Human texture discrimination in terms of texture statistical properties is investigated in (Julesz 1975). Accordingly, the textures in grey-level images are discriminated spontaneously only if they differ in second order moments. Equal secondorder moments, but different third-order moments require deliberate cognitive effort. This may be an indication that also for automatic processing, statistics up to the second order may be most important (Niemann 1981). The most popular second-order statistical features for texture analysis are derived from the so-called co-occurrence matrix (Haralick 1979).

They were demonstrated to feature a potential for effective texture discrimination in biomedical-images (Lerski 1993, Strzelecki 1995). The approach based on multidimensional co-occurrence matrices was recently shown to outperform wavelet packets (a transform-based technique) when applied to texture classification (Valkealathi 1998).

In this project, the features derived from co-occurrence matrix are used and the classification is done based on these features.

2.2.1 Gray Level Co-occurance Matrices

In this section, there will be a brief explanation of "Gray level co-occurance matrices" also called "Gray tone spatial dependetnt matrices" which give us certain textural features. Features generated using this technique are usually called "Haralick features". An essential component of Haralick framework of texture is a measure, or more precisely, four closely related measures from which all of our texture features are derived. These measures are arrays termed angular nearest-neighbor gray-tone spatial-dependence matrices. Haralick method assumes that the texture-context information in an image I is contained in the overall or "average" spatial relationship which the gray tones in image I have to one another. More specifically, it assumes that this texture- context information is adequately specified by the matrix of relative frequencies Pij with which two neighboring resolution cells separated by distance d occur on the image, one with gray tone i and the other with gray tone j. Such matrices of gray-tone spatial-dependence frequencies are a function of the angular relationship between the neighboring resolution cells as well as a function of the distance between them. Figure 2.5 shows the gray tone matrices for rotations 0, 45, 90, 135 and the unite distance.

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Figure 2.3: A 4 by 4 pixels image with 4 gray levels

	0	1	2	3
0	#(0,0)	#(0,1)	#(0,2)	#(0,3)
1	#(1,0)	#(1,1)	#(1,2)	#(1,3)
2	#(2,0)	#(2,1)	#(2,2)	#(2,3)
3	#(3,0)	#(3,1)	#(3,2)	#(3,3)

Figure 2.4: Corresponding gray tone matrix

0^o PH =
$$\begin{matrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{matrix}$$

135^o PLD =
$$\begin{matrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{matrix}$$

90^o PV =
$$\begin{matrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{matrix}$$

45^o PRD =
$$\begin{matrix} 6 & 0 & 0 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{matrix}$$

Figure 2.5: Corresponding gray tone matrix for different rotations: Horizontal, Vertial and two Diagonals

2.2.2 Haralick features

1. Angular Second Moment:(Also called Uniformity, Energy or ASM)

   $f_1 = \sum_{i=1} \sum_{j=1} P(i,j)^2$

2. Contrast:

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \sum_{i=1}^{N_g} \sum_{\substack{j=1 \\ |i-j|=n}}^{N_g} P(i,j)$$

3. Correlation:

$$f_3 = \frac{\sum_i \sum_j (ij) P(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

4. Variance:

$$f_4 = \sum_{i=1} \sum_{j=1} (i-\mu)^2 P(i,j)$$

5. Inverse Different Moment:

$$f_5 = \sum_i \sum_j \frac{1}{1+(i-j)^2} P(i,j)$$

6. Sum Average:

$$f_6 = \sum_{i=2}^{2N_g} i P_{x+y}(i)$$

7. Sum Variance:

$$f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 P_{x+y}(i)$$

8. Sum Entropy:

$$f_8 = -\sum_{i=2}^{2N_g} P_{x+y}(i) \log P_{x+y}(i)$$

9. Entropy:

$$f_9 = -\sum_i \sum_j P(i,j) \log P(i,j)$$

10. Difference Variance:

$$f_{10} = varianceof P_{x-y}$$

11. Difference Entropy:

$$f_{11} = -\sum_{i=0}^{N_g-1} P_{x-y}(i) \log P_{x-y}(i)$$

12. Information Measures of Correlation:

$$f_{12} = -\frac{HXY - HXY1}{maxHX,HY}$$

$$f_{13} = 1 - e^{-2(HXY2 - HXY)^{1/2}}$$

Where,

$$HXY = -\sum_i \sum_j P(i,j) \log P(i,j)$$

$$HXY1 = -\sum_i \sum_j P(i,j) \log P_x(i) P_y(j)$$

$$HXY2 = -\sum_i \sum_j P_x(i) P_y(j) \log P_x(i) P_y(j)$$

	Grass	Cushion	Linseeds	Rice	Seat	Stone
Energy (ASM)	0.019823	0.055581	0.047523	0.035675	0.020017	0.033243
Contrast	2.617138	0.473954	0.521309	0.772448	2.284532	0.839385
Variance	77.90769	78.03481	78.05843	77.98608	78.0529	78.05824
Entropy	4.331385	3.409118	3.474453	3.780832	4.255987	3.824554
Correlation	0.136335	0.726553	0.727663	0.636488	0.183452	0.594405

Figure 2.6: Haralick features calculated for these 6 textures

2.3 Classification

The features extracted in the previous part are used for the next stage which is the classification stage. In this project I have used three different classifiers: KNN, SVM and Naive Bayes.

2.3.1 KNN Classifier

As we know, the KNN classifier gets the k closest training examples to a new object, in the feature space. The object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. For this project, KNN classifier has been used with the Euclidean distance metric. The cross validation has been done on the data set for choosing the best possible value of K and the results show that K=5 is the optimum K for this set of training examples. Figure 2.7 shows the resulted testing error using cross validation.

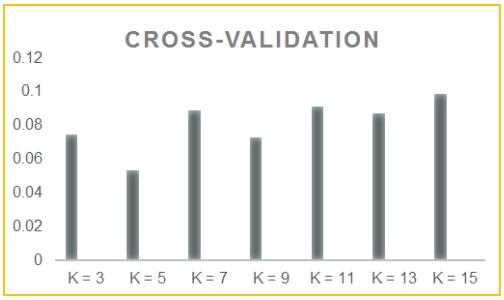


Figure 2.7: Testing error versus K, optimum K is 5.

2.3.2 SVM Classifier

Support Vector Machine (SVM) is part of a group of kernel based methods which are used for pattern classification and regression.Let xi, i =1, 2, , M be feature vectors of a training set X, which belong to either of two classes and . Using this training data, SVM finds an optimal hyperplane. with maximum margin that separates the unknown input patterns into 2 classes as

shown in Figure 1. Many hyperplanes separating the feature vectors are possible, SVM finds the one that has maximum margin and better generalization performance for classification.

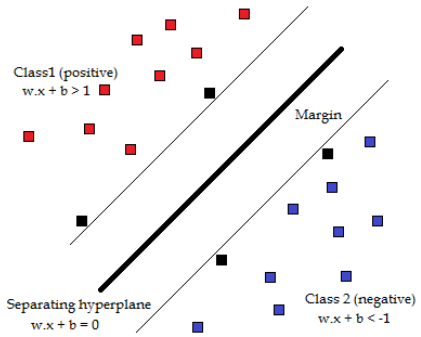


Figure 2.8: SVM classifiers chooses the hyperplane with maximum margin

Given training vectors $x_i \in R^n, i = 1, .., l$, in two classes, and an indicator vector $y \in R^l$ such that

$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$

subject to: $y_i w^T \phi(x_i) + b) \geq 1 - \xi_i$
$xi_i \geq 0, i = 1, .., l$

where ϕx_i maps x_i into a higher-dimensional space and $C > 0$ is the regularization parameter. Due to the possible high dimensionality of the vector variable w, usually we solve the following dual problem.

$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$

subject to: $y^T \alpha = 0$
$0 \leq \alpha_i \leq C, i = 1, .., l$

where $e = [1, ..., 1]^T$ is the vector of all ones, Q is an l by l positive semidefintie matrix. $Q_{ij} \equiv y_i y_j K(x_i, x_j)$ and $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$

After problem (?) is solved, using the primal dual relationship, the optmial w satisfies:

$w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i)$

and the decision function is:

$sgn(w^T \phi(x) + b) = sgn(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b)$

SVM is basically a linear classifier that classify linearly separable data, but in general , the feature vectors might not be linearly separable. To overcome this issue, kernel trick is used. The original input space is mapped into a high-dimensional feature space using kernel functions where it becomes linearly separable. The performance of an SVM classifier is dependent on the choice of a proper kernel function. Different kernel functions have been employed for different classification tasks. In this project I have used the SVM classifier with three different kernels and compared their performance

1. Linear kernel
   $K(x_i, x_j) = (\gamma {x_i}^T x_j + 1)^d, \gamma > 0$
2. Polynomial kernel
   $K(x_i, x_j) = \gamma {x_i}^T x_j + c$
3. Radial Basis Function
   $K(x_i, x_j) = exp(-\gamma ||x_i - x_j)||^2), \gamma > 0$

SVM with linear kernel is indeed one of the most simplest classifiers, but it won't be surprising if we get very high performance accuracy when the data distribution is linearly separable.Linear kernel is generally faster to train in comparison with non-linear kernels such as RBF.

In this project, I have used the "LibSVM" library [2]. Here is the output of the program for the using training set:

optimization finished, #iter = 372
nu = 1.000000
obj = -744.000000, rho = 0.000000
nSV = 744, nBSV = 744
Total nSV = 2232

2.3.3 Naive Bayes Classifier

Chapter 3

Experimental Results and Conclusion

I have used the commonly used measure of performance which is the accuracy of the classifiers. This accuracy is defined as the number of truly classifier examples over the total number of examples:

$$Accuracy = \frac{\#correctlypredicteddata}{\#totaltestingdata} \times 100$$

As the conclusion, we can see that among these three classifiers, SVM with polynomial Kernel has the highest accuracy (92.9 %). Although RBF kernel is the most popular kernel among the non-linear ones, in this case, the polynomiall kernel gives higher accuracy. In addition, although the GLCM features are not independent, the Naive Bayes yields good results. This confirm the fact that despite their naive design and apparently oversimplified assumptions, Naive Bayes classifiers have worked quite well in many complex real-world situations. An advantage of naive Bayes is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix. Finally,Generative models (e.g. Nave Bayes) will give more accurate results than Discriminative models (SVM), if the assumption about the kernel is incorrect.

Learning Alg.	Accuracy
KNN	80.9 %
SVM (Linear Kernel)	82.1 %
SVM (Polynomial)	92.9 %
SVM (Radial Basis Function)	81.5 %
Naïve Bayes	86.4 %

Figure 3.1: Classification accuracy using KNN, SVM and Naive Bayes classifiers

Bibliography

[1] G. Kylberg. The Kylberg Texture Dataset v. 1.0, Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, External report (Blue series) No. 35. Available online at: http://www.cb.uu.se/ gustaf/texture/

[2] Chih-Chung Chang, Chih-Jen Lin Department of Computer Science National Taiwan University, Taipei, Taiwan Last updated: March 4, 2013 Availabe online at: http://www.csie.ntu.edu.tw/ cjlin/libsvm/