Textures

Texture Analysis: Representation and Matching

Anil K. Jain and Kalle Karu

Department of Computer Science
Michigan State University
East Lansing, MI 48824, USA
jain@cps.msu.edu, karukall@cps.msu.edu

Abstract. Texture has found many applications in computer vision. Examples where texture analysis methods are being used include: (i) classifying images and browsing images based on their texture; (ii) segmenting an input image into regions of homogeneous texture; (iii) extracting surface shape information from 'texture gradient'; and (iv) synthesizing textures that resemble natural images for various computer graphics applications. Image texture is characterized by the gray value or color 'pattern' in a neighborhood surrounding the pixel. Different methods of texture analysis capture this gray-level pattern by extracting textural features in a localized input region. Practical texture-based image processing methods define texture in a manner that is most appropriate for achieving a given goal and ignore the issue whether the input image really contains any texture. This paper describes attempts to learn 'optimal' texture discrimination masks using neural networks.

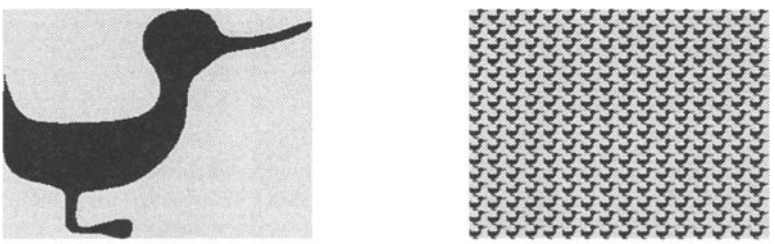
1 Introduction

Texture characterizes local variations of image color or intensity. Although texture-based methods have been widely used in computer vision and graphics, there is no single commonly accepted definition of texture. Each texture analysis method defines texture according to its own model. Texture is often described to consist of primitives that are arranged according to some placement rule. This definition gives rise to structural methods of texture analysis which explicitly attempt to recover the primitives and the replacement rules. Statistical methods, on the other hand, consider texture as a random phenomenon with identifiable local statistics. What is common in all descriptions of texture is that texture is a neighborhood property. Image regions that are perceived to have a similar texture have a similar pattern of color or intensity variation, which may be independent of scale, rotation, or illumination.

The following four areas of texture analysis have been studied most extensively: texture segmentation, classification, synthesis, and 'shape from texture' [23]. The purpose of texture segmentation is to divide the input image into homogeneously textured regions, without knowing *a priori* what the textures are. Texture classification methods, on the other hand, attempt to assign a known texture class to each image region. Figure 1(a) shows an image which can be segmented into five homogeneous textured regions. Closely related to texture

(a) (b) (c)

Fig. 1. Example of textured image: (a) a composition of natural images; (b) segmentation using Gabor filtering; (c) segmentation using masks from a neural network.



(a) (b)

Fig. 2. Does the image contain texture? (a) an image without texture; (b) a textured image with the same texture primitive as in (a).

classification is the newly emerging area of image database retrieval based on texture [5, 19]. Elaborate classification methods are often simplified in image database applications to achieve higher processing speed on a large number of images. Texture synthesis methods are used in computer graphics to generate naturally looking surfaces from a few parameters. Changes in the granularity of texture, or in the size of its primitives, give cues about the 3D shape of a textured surface. Shape from texture algorithms use these cues to reconstruct the 3D shape, often with the help of other information, such as stereopsis or shading [17].

Texture analysis methods usually assume that the input images really have texture. In a practical application, it may not be known if the image has any texture and if texture-based methods are appropriate for processing it. Figure 2 shows two images composed of the same primitive. Texture analysis algorithms should not be applied to the first image which contains no texture [14].

In this paper we consider the texture segmentation and classification problems. Section 2 gives background information on different texture analysis methods. Section 3 describes a neural network model to optimize the feature extraction and classification tasks. Section 4 presents experimental results, and Section 5 concludes the paper.

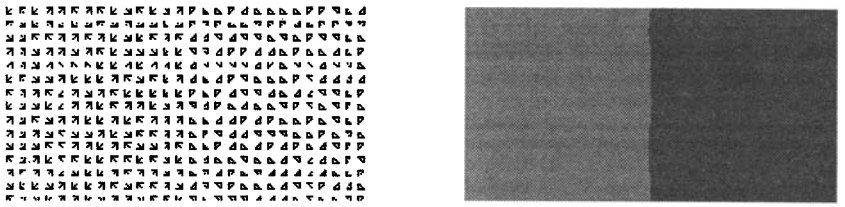


Fig. 3. A texture pair with identical second-order statistics: (a) input texture pair; (b) its segmentation using masks learned by a neural network.

2 Texture Analysis Methods

Texture is a popular image attribute which has been successfully used in a variety of applications (remote sensing, medical imaging, industrial inspection). The main goal of texture analysis is to replicate human ability to segment textures. Psychological and psychophysical studies have been conducted to understand human perception of texture and its limits [1]. Tamura *et al.* [22] selected six perceptual characteristics of texture – coarseness, contrast, directionality, line-likeness, regularity, roughness, and proposed algorithms for automatically determining numerical values for these features from given input images. The same texture features have recently been used in image database retrieval [5].

Figure 3 (a) contains a part of synthetic texture images used by Julesz and other researchers to establish the limits of our ability to segment textures preattentively. Julesz [13] conjectured that people cannot segment textures that agree in second-order statistics, that is, probabilities of observing gray values at points that are a certain displacement from each other. The image in Figure 3 (a) is a counterexample to this conjecture. Co-occurrence matrices estimate the second-order statistics by counting the frequencies for all pairs of gray values and all displacements in the input image. Since the number of co-occurrence matrices is very large, they are usually combined in a more compact representation. Haralick has proposed several textural features that can be extracted from co-occurrence matrices – uniformity of energy, entropy, maximum probability, contrast, inverse difference moments, correlation, and probability of run lengths [7]. When a large number of textural features are available, feature selection becomes an important issue. Schistad Solberg and Jain [21] have studied feature selection and combination for SAR image segmentation.

Structural texture analysis methods consider texture as a composition of primitive elements arranged according to a placement rule. Recovering these primitives from natural images is often very difficult, and this severely limits the applicability of structural methods. Model-based texture analysis methods are used both in computer vision and computer graphics. In texture recognition, a model (Markov random field [2], autoregressive [15], fractal [18], or other) is fitted to the image data. The estimated model parameters can then be used to segment or classify the image, or to synthesize new images.

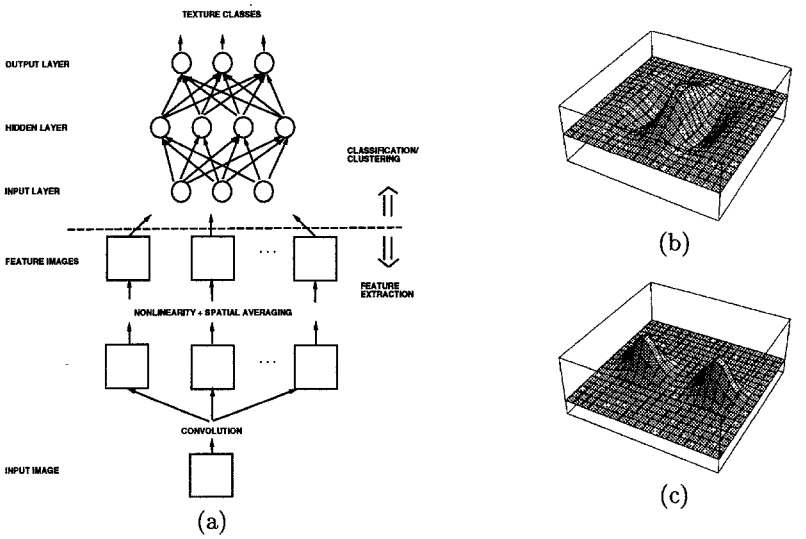


Fig. 4. Multichannel filtering: (a) a general scheem; (b),(c) a Gabor filter in spatial and spatial frequency domains, respectively

The statistical methods described above have the common property that processing is divided into two parts: (i) feature extraction and (ii) classification/clustering. Similar two-phase computation can be found in multichannel filtering methods (Figure 4 (a)) where features are extracted by convolving the input image with a bank of filters. Filters with different band-pass characteristics extract different properties of input texture. The resulting feature values are often called textural energies. The second stage of the multichannel filtering method classifies or clusters pixels based on the extracted features. Statistical classification and clustering methods (nearest neighbor classifier or k-means clustering, for instance) can be used to segment the feature vectors. In Figure 4 (a), we have drawn a neural network as a universal classification or clustering device.

Gabor features have been used to extract textural features corresponding to different orientations and frequencies [3, 11]. In the experiments by Jain and Farrokhnia [11], 20 even-symmetric Gabor filters were used to form the filter bank, $tanh(\beta x^2)$ nonlinearity was applied to each output pixel, and the results were averaged with a Gaussian filter. Figures 4 (b) and (c) show an even symmetric Gabor filter in spatial and spatial frequency domains. The other 19 filters can be obtained by rotating and stretching this kernel. The advantages of Gabor filtering include its multi-scale representation from which invariant features could be computed. Complex Gabor filters have been shown to minimize the joint localization in spatial and spatial frequency domains [6, 3]. Figure 1 (b) shows segmentation of the textured image in Figure 1 (a) using Gabor filters. Sixteen Gabor filters (with four orientations and four radial frequencies) were used for filtering. The clustering program CLUSTER [10] was used to segment the image. The number of clusters was set to five. The classification is quite

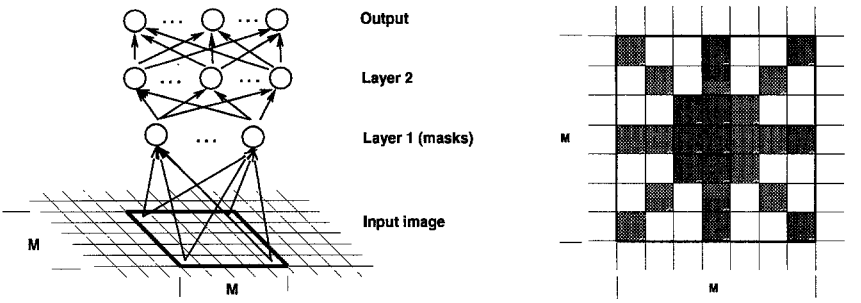


Fig. 5. Neural network for texture classification: (a) neural network architecture; (b) input configuration; shaded pixels are used as inputs to the network for the classification of the center pixel.

accurate except near border areas where the averaged features from different textures cause confusion.

3 Neural Network Classifier

A texture classifier can be obtained by specifying a set of filters and the nonlinearity in Figure 4 (a). A multi-layer perceptron can then be trained in a supervised mode to classify the extracted feature vectors. For an optimal classifier, not only the best set of filters has to be found, but the network has to be trained to classify with minimal error. When considering the convolution and nonlinearity operations as part of the neural network (Figure 5), the two tasks – filter specification and classifier training – can be performed at the same time. The input to the multi-layer perceptron in Figure 5 (a) comes directly from a $M \times M$ window in the image. In this architecture, the weights in the first layer correspond to the mask coefficients of a filter bank in the general multichannel filtering scheme. Rather than use the entire $M \times M$ window for input values, we have constrained the network's input to the shaded pixels in Figure 5(b). This configuration helps to reduce the number of network parameters when the mask size M grows larger.

The neural network is trained with the back propagation method [20]. We have used a constant learning rate and momentum term. About one million steps were sufficient for convergence. During training the network weights are updated in all the layers, including the first one, whereas in Figure 4 (a) only the network classifier is trained with fixed filter coefficients. Therefore, training the unified neural network involves performing two optimization tasks at the same time – finding the best mask coefficients and the best classifier, so that the classification error is minimized. The selection of a suitable number of filters can also be included in the network training by growing or pruning the neural network. We have used the node pruning method by Mao *et al.* [16]. Since we are interested in the selection of masks, we only prune nodes in the first layer.

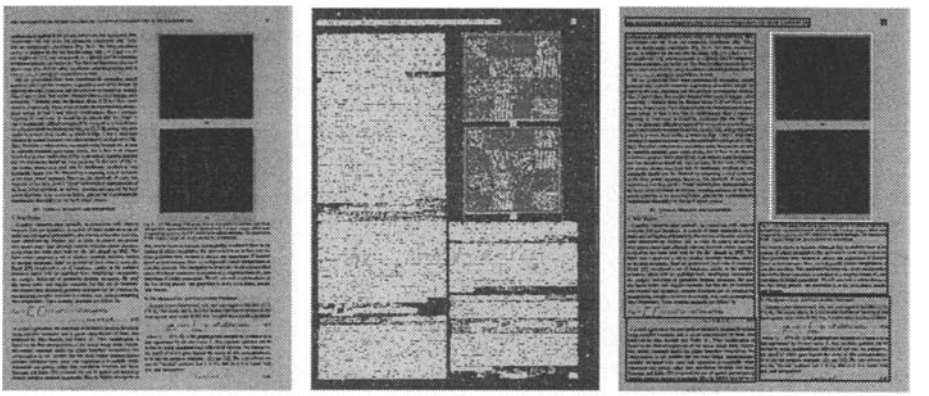


Fig. 6. Page layout segmentation: (a) a 1000 × 800 document image scanned at 100 dpi; (b) segmentation into text, graphics, and background; (c) the post-processed image with bounding boxes surrounding different regions.

4 Experimental Results

In this section, we show classification results from several applications. Figure 3 (b) shows a classification result of segmenting the synthetic image in Figure 3 (a) by a two-layer network with only three masks of size 5×5. The masks learned in a supervised neural network can also be used in unsupervised texture segmentation. A two-layer neural network with sixteen 11×11 masks was trained on nine different textures, including four textures from the image in Figure 1 (a). The learned sixteen masks were then separated from the network and used to perform unsupervised segmentation of the five-textured image in Figure 1 (a). The input image was convolved with the masks, output pixel values were squared and smoothed by a Gaussian function, and the resulting feature vectors were clustered using the CLUSTER algorithm [10]. Figure 1(c) shows the unsupervised segmentation result.

In a document processing system, a scanned page needs to be segmented into regions of text, graphics and background, after which character recognition or other image processing algorithms are used with the segmented parts. This segmentation can be done based on the texture of the different regions [8]. Figure 6(a) shows a digitized image of a page containing both text and graphics. A three-layer neural network with 20 nodes in each hidden layer was trained to classify the input pixels into three classes – background, text, and graphics. The result of applying the network with ten masks (pruning ten nodes in the first layer did not decrease the performance) to the page is shown in Figure 6(b). Due to the small mask sizes (11×11 in this experiment), the network is accurate in locating the texture boundaries, and finding even such small regions as the page numbers. Figure 6(c) shows the image after applying the post-processing steps described in [12].

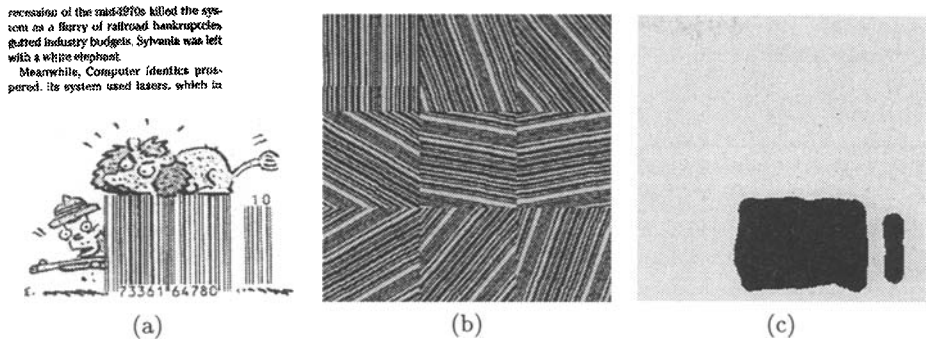


Fig. 7. Locating barcode: (a) a barcode image; (b) training patterns; (c) segmentation.

The problem of barcode localization is to find the barcode in an input image, which may be present in any orientation, position, and scale [9]. In the experiments shown in Figure 7, a three-layer network with ten hidden nodes was trained to classify input image regions into three categories: (i) barcode, (ii) text, and (iii) graphics and uniform gray-value combined. The training patterns for barcode were taken from a different image, and they were rotated to ensure rotational invariance of the classifier. The result of applying the trained network to the input image and median filtering the output class labels is shown in Figure 7 (c).

5 Conclusions

We have shown how a neural network can be trained for supervised as well as unsupervised texture segmentation. Practical problems, such as barcode localization and page layout segmentation, can be solved more efficiently with specifically designed masks than with a general filter set. The use of neural network makes the mask selection automatic. For a new problem, instead of manually choosing a system and testing its performance, one can simply train a neural network (or several of them) on known images. The neural network approach is particularly suitable in texture classification applications where the input textures are known. Compared with general filtering methods, a neural network that is optimized for the specific input textures is able to achieve higher classification accuracy and processing speed.

References

1. Caelli, T.: Visual Perception: Theory and Practice. Pergamon Press. (1981)
2. Cross, G.C, Jain, A.K.: Markov Random Field Texture Models. IEEE Trans. on PAMI. **PAMI-5** (1983) 25-39
3. Daugman, J.G.: Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters. J. Opt. Soc. Amer. **2(7)** (1985) 1160-1169

4. Elfadel, I.M., Picard, R.W.: Gibbs random fields, cooccurrences, and texture modeling. IEEE Trans. on PAMI. **PAMI-16** (1994) 24-37
5. Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., Equitz, W.: Efficient and Effective Querying by Image Content. JIIS. **3** (1994) 231-262
6. Gabor, D.: Theory of communication. J. IEE (London). **93** (1946) 429-457
7. Haralick, R.M.: Statistical and Structural Approaches to Texture. Proc. of the IEEE **5** (1979) 786-804
8. Jain, A.K., Bhattacharjee, S.: Text Segmentation Using Gabor Filters for Automatic Document Processing. Machine Vision and Appl. **5(3)** (1992) 169-184
9. Jain, A.K., Chen, Y.: Barcode Localization Using Texture Analysis. Proc. Sec. Int'l Conf. on Document Anal. and Rec., Tsukuba city, Japan. (1993) 41-44
10. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, New Jersey. (1988)
11. Jain, A.K., Farrokhnia, F.: Unsupervised Texture Segmentation Using Gabor Filters. Pat. Rec. **12** (1991) 1167-1186
12. Jain, A.K., Zhong, Y.: Page Layout Segmentation based on Texture Analysis. (under review)
13. Julesz, B., Gilbert, E.N., Shepp, L.A., Frisch, H.L.: Inability of humans to discriminate between textures that agree in second-order statistics – revisited. Perception. **2** (1973) 391-405
14. Karu, K., Jain, A.K., Bolle, R.M.: Is there any texture on the image? Tech. report, Michigan State University. (1995)
15. Mao, J., Jain, A.K.: Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models. Pat. Rec. **2** (1992) 173-188
16. Mao, J., Mohiuddin, K., Jain, A.K.: Minimal Network Design and Feature Selection Through Node Pruning. Proc. 12^{th} ICPR, Jerusalem. **2** (1994) 622-624
17. Pankanti, S., Jain, A.K.: Integrating Vision Modules: Stereo, Shading, Grouping, and Line Labeling. IEEE Trans. on PAMI. (to appear)
18. Pentland, A.: Fractal-based description of natural scenes. IEEE Trans. on PAMI. **PAMI-9** (1984) 661-674
19. Picard, R.W., Minka, T.P.: Vision texture for annotation. Multimedia Systems. **3** (1995) 3-14
20. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In D.E. Rumelhart and J.L. McClelland, editors, Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Cambridge, MA: MIT Press. **1** (1986) 318-362
21. Schistad Solberg, A.H., Jain, A.K.: Texture Analysis of SAR Images: A comparative Study. IEEE Trans. on Geoscience and Remote Sensing. (under review)
22. Tamura, H., Mori, S., Yamawaki, Y.: Textural Features Corresponding to Visual Perception. IEEE Trans. on SMC. **SMC-8** (1978) 460-473
23. Tuceryan, M., Jain, A.K.: Texture Analysis. Chapter 2.1 in Handbook of Pattern Recognition and Computer Vision, C.H. Chen, L.F. Pau, P.S.P. Wang (eds.) World Scientific Publishing Co. (1993) 235-276