An Autoassociator for Automatic Texture Feature Extraction

S. Kulkarni and B. Verma
School of Information Technology
Griffith University, Gold Coast Campus, Qld 9726, Australia
E-mail: {S.Kulkarni, B.Verma}@gu.edu.au
Web: http://intsun.int.gu.edu.au

Abstract

This paper presents an autoassociator neural network for texture feature extraction. Texture features are extracted through the hidden layer of an autoassociator. The Resilient Propagation (RP) algorithm was employed to train the autoassociator with the texture input and output patterns. The performance of the feature extractor was evaluated on Brodatz benchmark database. A detail analysis of the results is included. The results and analysis showed that the autoassociator is capable of extracting texture features better than the other traditional techniques.

1. Introduction

1.1 Background

In order to determine the texture features, which are useful in classification process, it is necessary to have an intuition as to what constitutes a texture. Texture is a blanket term used to describe the brightness variations observed in almost all surfaces; it conveys information about pixel and its surrounding. The main purpose of texture feature extraction is to obtain relationships among the pixels that belong to a similar texture. Many studies have been conducted about the performance of various families of computational methods for texture feature extraction [1][2]. A texture feature extraction method is a computational method that analyzes the pixels contained in a certain region of an input image and generates a single value that represents the contents of that window.

There are three principal approaches to describe the texture of an image region: they are statistical, structural and spectral [3]. Statistical techniques attempt to describe texels by attributing them with statistical distribution of pixels, while structural techniques use the arrangement of image primitives and shapes to classify and recognize texels. Spectral techniques, based on the Fourier transform, are used mainly to detect global periodicities. In the spatial domain, textures are characterized by a two step procedure. First a window is slided across the image and at each location the local structure is represented by a vector. Next, the mean and the covariance of these vectors are used as texture representation of the entire image. In a spatial domain the 2D Gabor functions are complex sinusoidal gratings modulated by 2D Gaussian functions. In the spatial frequency domain they correspond to 2D bandpass filters. A typical Gabor filter bank design is described in [4]. The MRSAR method [4] models the texture as a second order noncasual Markov random field. The four parameters of the underlying autoregressive model are estimated independently at these resolutions using different window sizes. Tamura et al. [6] explored the correlation between human visual perception and the computational measures for six textural features, coarseness, contrast, directionality, line-likeness, regularity and roughness. The experiments were performed with 16 patterns. Liu and Picard [7] presented an image model with a set of features, which is based on the theory of a 2-D Wold decomposition of homogenous fields, periodicity, directionality and randomness. Ma and Manjunath [8] presented an image retrieval system to use texture features for aerial photographs. Photos were segmented into
homogenous regions and the texture features were computed by filtering the image with a bank of a Gabor filters. A hybrid neural network algorithm was used to cluster texture classes in the feature space. J. Smith [9] presented the algorithm of binary texture feature sets extracted from image spatial-frequency analysis. Belongie et al. [10] presented an image representation, “blobworld” which is based on segmentation using Expectation-Maximisation algorithm on combined color and texture features. In [11], Lerner et al. compared different neural networks for feature extraction and found that a neural network could be a good feature extractor. In this paper, we propose and design an auto-associator for extraction of texture features.

1.2 Organisation of the Paper

The rest of the paper is organised as follows: Section 2 describes the feature extraction process using an AANN. The preparation of training and testing sets from Brodatz texture patterns is presented in Section 3. The experimental results in form of feature graphs, number of hidden units, number of iterations, RMS error, etc. are presented in Section 4. The results, analysis and comparisons are discussed in Section 5. The conclusion is presented in Section 6.

2. Proposed Auto-associator Feature Extractor

The auto-associator feature extractor is based on applying the same patterns at input and output. The auto-associator learns the same pattern at input and output and provides the feature vectors through its hidden layer. Only one hidden layer is used to extract the features as shown in the figure 1. The network is trained using a supervised training. After training is finished, the values of the hidden layer are extracted and taken as a feature vector. These feature vectors could be used by a classifier to classify features into texture classes.

![Figure 1. Auto-associator as a feature extractor](image)

To check the performance of our proposed auto-associator, we used a simple MLP texture feature classifier to classify the features extracted from auto-associator. The number of inputs to classifier must be same as the number of hidden units used for feature extractor. The output of the hidden layer from auto-associator should be used as input to classifier. The number of classes should be equal to the number of classes.
3. Preparation of Training and Testing Sets

The Brodatz texture database [12] was used to evaluate the performance of the proposed techniques detailed in previous section for texture features extraction. The database contains 96, 512 × 512 texture images. In order to create a number of small images which belong to the same class, we partition each of the 512 × 512 images into 128 × 128 sub-images, thus forming 16 sub-images from each image. To reduce the size of input vector to neural network, the mean (µ) and standard deviation (σ) was calculated for each row (128 pixels). First 12 sub-images were used for the training of auto-associator and the last 4 images were used as testing data set. These images were normalized in the range of 0 and 1. There were total 96 texture patterns in the database, which were grouped into 32 similar classes, each of them containing 1-5 texture classes. All the texture sub-images belonging to the same similarity class are visually similar [8].

4. Experimental Results

The experimental results were conducted separately in two stages for an auto-associator-classifier technique. Firstly the training of the auto-associator and then the training of the classifier. The auto-associator was trained by keeping the same number of hidden units and by increasing the number of iterations to achieve better results. The RMS error was decreased as the number of iterations increased. Table 1 shows some of the results obtained after the training of the auto-associator. The number of inputs and outputs were 256. The values of momentum and learning rate used for the experiments were 0.7 and 0.8 respectively. The number of pairs for training and testing were 640 and 160 respectively.

<table>
<thead>
<tr>
<th>Table 1. Auto-associator feature extractor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden Units</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>32</td>
</tr>
</tbody>
</table>

The classifier was trained after obtaining the output from the hidden layer from auto-associator. Experiments were conducted by varying the number of iterations and the results are shown in Table 2. The best results were obtained after training the auto-associator with 25000 iterations and feeding the output of the hidden layer to the classifier.

<table>
<thead>
<tr>
<th>Table 2. MLP texture feature classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden Units</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>16</td>
</tr>
</tbody>
</table>

5. Analysis of the Results

The experiments were conducted for various numbers of hidden units and iterations. The best results were obtained with 32 hidden units and 25000 iterations. To check the performance of the proposed autoassociator, the output (texture features) of the hidden layer were extracted and the graphs were plotted for different texture classes. The following
The graph shows the texture features for class 1 and class 6. Similarly the graphs were plotted for other classes to see the difference in the texture features.

The graphs were plotted for testing sets and few results are shown in the following graphs for class 1 and class 6.

**Figure 2. Texture features for class 1 and class 6 (training set)**

It is difficult to compare texture features shown in graphs (Figures 2 & 3) directly with other techniques so we implemented a simple MLP classifier. The classification results obtained after extracting the features from an auto-associator and classified through classifier were compared with other techniques. The comparison was performed with other researchers who uses the same brodatz texture database for classification of texture patterns. In [13] Jones and Jackway used granold technique and achieved the classification rate of 76.9% on Brodatz texture database. In [14], Wang and Liu compared the Nearest Linear Combination (NLC) with Nearest Neighbor (NN) Classification and listed the classification rate 95.48% with NLC and 93.47 % with NN. Our techniques achieved 97.5% classification rate, which is better than other techniques.

**Figure 3. Texture features for class 1 and class 6 (test set)**
6. Conclusions

In this paper, we have investigated an automatic texture feature extraction technique based on auto-associator neural network. We have shown texture feature graphs from training and testing sets in Figures 2 and 3 respectively, which show that the auto-associator is capable of separating two classes very well and the whole process of feature extraction is automatic. The feature extraction technique was tested on large database of texture patterns such as Brodatz texture database. The results are very promising.

References