Abstract. We describe an approach to finding regions of a texture of interest in arbitrary images. Our texture detectors are trained only on positive examples and are implemented as autoassociative neural networks trained by backward error propagation. If a detector for texture $T$ can reproduce an $n \times n$ window of an image with a small enough error then the window is classified as $T$. We have tested our detectors on a range of classification and segmentation problems using 12 textures selected from the Brodatz album. Some of the detectors are very accurate, a small number are poor. The segmentations are competitive with those using classifiers trained with both positive and negative examples. We conclude that the method could be used for finding some textured regions in arbitrary images.

1 Introduction

As digital cameras, mobile phones and computers become more common, more and more people are taking and storing digital images. The problem of finding a particular stored image is becoming more and more difficult. Content Based Image Retrieval (CBIR) systems [1] are attempting to solve this problem using a wide variety of image features. Many images contain textured regions, for example, surf breaking on a beach, a waterfall, a brick building, grass on a sports field and trees in a forest. We are interested in whether using texture can increase the accuracy of retrieval for queries like “Find the picture of the family in front of the waterfall that we took on our holiday a few years ago” or “Find that photo of our son playing hockey when he was in primary school”. While some CBIR systems include some texture features in the hundreds of features they use, there is little evidence that the texture features are particularly useful.

Our use of texture will be quite different to the way it is used in CBIR systems. Such systems typically compute a number of Gabor or Wavelet features [2] on the whole image (sometimes image segments) and use feature values in finding a matching database image to the query image. Our approach will be to learn a classifier for a small window of the texture of interest ($T$), to apply the classifier as a moving window to a database image and report that an image in the database contains $T$ only if a large enough region of $T$ is detected.
There is no generally agreed definition of texture. For our purposes we consider a texture to be an image, or a region in an image, in which a basic visual unit is repeated. In synthetic textures the repeating unit is replicated exactly, for example a checkerboard. In a natural texture the repeating unit is repeated with some variation, for example, blades of grass or pebbles. A considerable amount of work on texture is done with the Brodatz album [3]. The album contains a large variety of natural textures, some highly regular, some very irregular. We have chosen a number of the more regular textures. Some of these are shown in Figure 1.

Most work on learning a classifier requires examples of the class of interest and examples that are not in the class of interest. The learnt classifier is required to distinguish the two classes. A number of texture classification systems of this kind have been built, for example, [2,5]. There is a major problem in using this approach to find a texture \( T \) in an arbitrary image since there are many more examples in the non \( T \) class. This leads to unbalanced training sets and increased classification error as many of the test/unseen examples are nothing like the ones in the training set. Our goal is to solve these kinds of problems with classifiers by using only examples of the texture of interest in the training data.

1.1 Research Questions

Our overall goal is to determine whether a texture detector for a texture \( T \), trained on only examples of \( T \), is able to find regions of \( T \) in arbitrary images, as in Figure 3 for example. In particular, we will address the following research questions:

1. How can a neural network be used to recognize a texture using only examples of the class of interest? What is a suitable neural network architecture?
2. How accurately can the texture of interest be located in a mosaic of other textures? How does this accuracy compare with classifiers that have been trained with examples of all textures?
3. How accurately can the texture of interest be located in arbitrary images?