Discovery of Human-Competitive Image Texture Feature Extraction Programs Using Genetic Programming

Brian Lam[1] and Vic Ciesielski[1]

School of Computer Science and Information Technology
RMIT University
GPO Box 2476V, Melbourne, Vic Australia 3001
{blam, vc}@cs.rmit.edu.au

Abstract. In this paper we show how genetic programming can be used to discover useful texture feature extraction algorithms. Grey level histograms of different textures are used as inputs to the evolved programs. One dimensional K-means clustering is applied to the outputs and the tightness of the clusters is used as the fitness measure. To test generality, textures from the Brodatz library were used in learning phase and the evolved features were used on classification problems based on the Vistex library. Using the evolved features gave a test accuracy of 74.8% while using Haralick features, the most commonly used method in texture classification, gave an accuracy of 75.5% on the same problem. Thus, the evolved features are competitive with those derived by human intuition and analysis. Furthermore, when the evolved features are combined with the Haralick features the accuracy increases to 83.2%, indicating that the evolved features are finding texture regularities not used in the Haralick approach.

1 Introduction

Human-competitive methods can be defined as automatically generated methods that perform equal or better than those derived by human intuition, that require simple input from humans in their generation and are general in that they can be readily applied to new problems in the same domain [1]. Koza et al [1] have identified a number of instances where genetic programming has produced human-competitive solutions in such domains as analog electrical circuits and controllers, synthesis of networks of chemical reactions and antenna design. In this paper we describe an approach to the evolution of texture features that are human-competitive.

Apart from size, shape and colour, texture is an important property used in image classification. Texture can be defined as a function of the spatial variation in pixel intensities [2]. Typically repetition of some basic pattern is involved. Examples of textures are given in figures 8 and 9. The usual approach to texture classification involves extracting texture features from two or more classes of

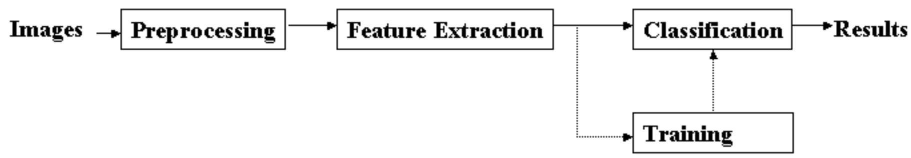

Fig. 1. Image Texture Classification

texture images to train a classifier. The features and classifier can then be used on new images. This process is shown in Figure 1. Currently, algorithms for texture features are developed by human intuition and analysis and there is a large number of approaches and theories. Texture classification is a significant economic problem and has been applied in many domains, for example, remote sensing [3], automatic inspection [4], medical image processing [5] and document processing [6].

There are two texture libraries that are used in most of the research in texture analysis - the Brodatz album and the Vistex data set. The Brodatz album consists of homogeneous categories of naturally occurring textures. The Vistex set consists of heterogeneous categories of texture images, that is, each class may have more than one type of texture. For example, the flower category may have flower images at three different resolutions, thus making the Vistex set more difficult to classify.

Our conjecture is that it may be possible to discover general feature extraction algorithms using an evolutionary search technique such as genetic programming if suitable fitness evaluation is provided. The overall process is shown in figure 2. Our specific research questions are:

1. What inputs, pixels or histograms, should be used as inputs to the genetic programs?
2. How can feature extraction algorithms be evolved from two textures?
3. Will features learnt from the Brodatz dataset generalize to the Vistex data set?
4. Have the evolved features detected any texture regularities not used in human developed methods.

In this paper we use the following terminology. A *learning set* is the set of images used by the evolutionary process to evolve feature extraction algorithms. These algorithms are then used on a different *training set* of images to get a nearest neighbour classifier which is evaluated against a different *test set*.

2 Related Work

2.1 Genetic Programming and Computer Vision

Genetic programming has been applied to a variety of image classification problems. The work so far can be grouped into three approaches. The first approach

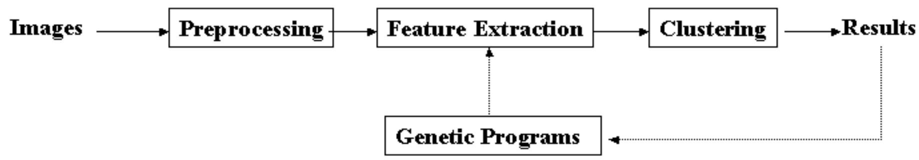

Fig. 2. Feature Extraction Discovery Using Genetic Programming

involves pre-processing images with low-level feature extraction algorithms followed by genetic programming designed to discover the classification rule. Tackett [7] used statistical features such as mean and standard deviation of grey levels within a window to classify military targets in infrared images. Poli [8] used pixel grey levels and moving averages of different size convolution masks to segment medical images. Agnelli [9] used statistical features to classify documents. Ross et al [10] used statistical features to classify microscopic images of minerals. Lin and Bhanu [11] used a co-evolutionary approach to build composite features from primitive features. This approach involves the steps of training and classification shown in Figure 1.

The second approach involves not using features and working directly from image pixels. Koza [12] discovered a detector for two alphabetic characters by moving a 3 x 3 matrix over two 6 x 4 binary pixel grids. Building on Koza's work, Andre [13] used moving templates to identify digits. Song et al [14] developed a pixel based approach to classify grey scale texture images. Martin [15] evolved image classification processes for robot navigation. This approach combines the feature extraction and classification steps shown in Figure 1.

The algorithms mentioned in the two previous approaches are problem-specific, that is, the derived algorithms only work with the types of images they were trained on. The learning process needs to be repeated for new types of images.

In the third approach, genetic programming is used to discover general algorithms that can be applied to new types of images without prior training. Harris [16] evolved edge detectors that performed well on real and synthetic data. This approach involves only the feature extraction step in Figure 1.

2.2 Conventional Texture Features

There has been considerable work on texture features since 1960 and many theoretical models have been proposed. A list of some of these is in table 1. The most well known texture feature extraction algorithm is the Grey Level Co-occurrence Matrix method developed by Haralick [17]. Assuming we are working with images that have 256 grey levels, this method involves first generating a grey level co-occurrence matrix with $256(i)$ columns and $256(j)$ rows. An entry in the matrix is the frequency of occurrence of pixels with grey level i and j level separated by a displacement d in a particular orientation. Most work uses a displacement of 1. There are four principal orientations namely $0^o, 45^o, 90^o$ and 135^o so four matrices are generated. Thirteen second order statistical features are then calculated from each matrix. The features for the four principal orientations are then averaged giving a total of 13 Haralick features. Most new texture feature extraction methods are benchmarked against the GLCM method.

In our experiments we have followed the methodology of Wagner [18] who compared a large number of conventional methods on classification problems based on the Vistex dataset. This permits direct comparison of classification accuracy with the other methods. The image size used was 64×64.

3 Configuration of Genetic Programming

3.1 Inputs

Determining the most appropriate inputs to use for texture feature discovery is a major issue. Using grey levels of individual pixels will result in a very large number of terminals which can lead to memory and search problems. For example a 64×64 image would have 4096 terminals. We have determined empirically that 256 terminals is about the limit for reasonable performance in our system. For images larger than 16×16 some kind of aggregation of pixel values needs to be used. We have experimented with two forms of inputs – pixel values and histograms. In the case of pixel values, the grey level of each pixel is a terminal. Image size was 16×16 making a total of 256 terminals. In the case of histograms, the image size was 64×64 and the number of pixels at each grey level is a terminal in the program, making a total of 256 terminals. It is important to note that image texture has important spatial characteristics. These are preserved in the pixel representation, but lost in the histogram representation.

3.2 Functions

Originally we used the function set $\{+, -, *, /\}$. However we found that using just + gave feature extraction programs that were just as accurate as those using all four operators but were considerably easier to understand. Thus all subsequent work was carried out using just the + function.

3.3 Fitness Evaluation

A feature extraction algorithm is considered useful if the feature values result in high classification accuracy. This will occur if the feature values computed for each class are well separated in feature space. Thus, to evolve feature extraction algorithms, we need a way to implement the intuition that "the better the separation, the better the fitness". We have done this by computing the overlap between clusters generated by the K-means clustering algorithm. An example of this, for the case where there are two texture classes in the learning set is shown in figure 3. To get the data shown in the figure a program in the population

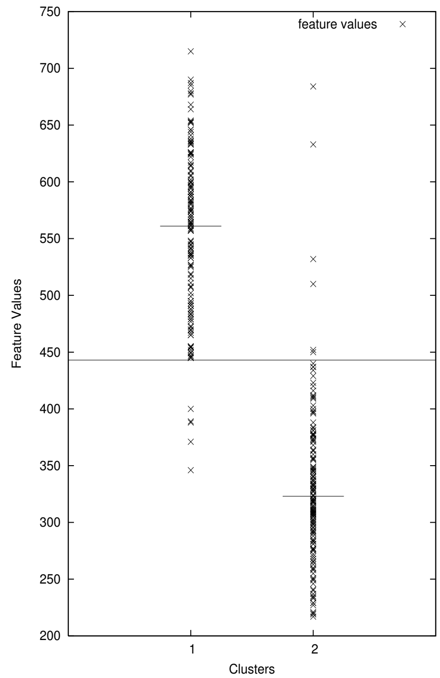

Fig. 3. Feature Space for Two Texture Classes

has been evaluated on a learning set of 160 images which consist of 80 examples of texture 1 from figure 8 and 80 examples from texture 2. The averages of the feature values for each class give cluster centroids at 561 and 323. The mid point of the two centroids is the cluster boundary, that is 443. There are 4 cluster1 feature values below the boundary and 6 cluster2 features above it, thus 10 points are incorrectly clustered. Equivalently, it can be considered that there are 10 errors.

3.4 Parameters

The RMIT-GP package [19] was modified to suit the problem. Clustering was performed using the Weka machine learning tool [20]. Default genetic programming parameters for the RMIT-GP package were used, namely a mutation rate of 0.28, a cross-over rate of 0.78 and an elitism rate of 0.02. Each run consisted of a population of 100 individuals evolved for 200 generations. The first generation of programs was generated randomly.

4 Learning from Two Classes of Textures

The goal of the experiments described in this section was to determine whether useful feature extraction algorithms could be learnt from two textures. The same

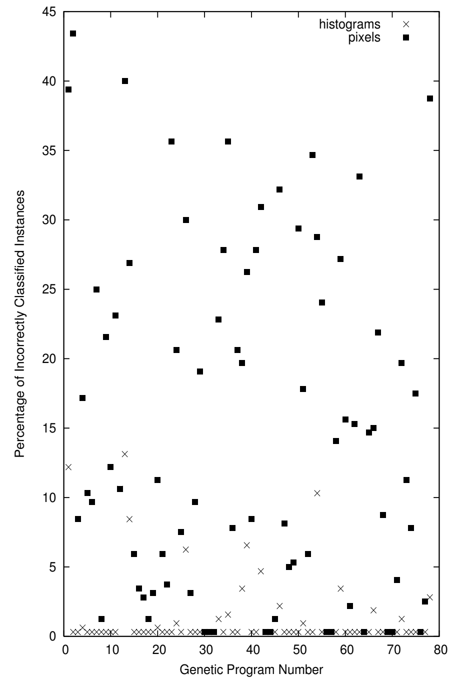

Fig. 4. Error for pixel and histogram approaches

13 textures as selected in [18] were used. These are shown in figure 8. Each of the 78 different pairs were used as the learning set in 78 different runs of the genetic programming system, giving 78 features that could be used by a classifier.

4.1 Pixel Inputs

We first investigated the use of pixels of 16×16 sub images as inputs to the evolutionary process. Our intent was to preserve spatial relationships between pixels. However, the number of incorrectly clustered instances was very high and, despite considerable experimentation with the genetic algorithm parameters, we could not get this number down to reasonable levels. Figure 4 shows a comparison of the clustering results for the pixel approach and the histogram approach for each of the 78 combinations of Brodatz textures. The percentage of incorrectly clustered instances is shown for each program. For example, the error was 39% using pixels compared with 12% using histograms for programs which use Brodatz texture classes 1 and 2. It is clear that histogram inputs give much better clustering. In figure 5, the average and best fitness values are plotted for both approaches. The top two curves are those for the pixel approach and the bottom curves for histograms. The histogram approach converged faster to a better solution than the pixel approach.

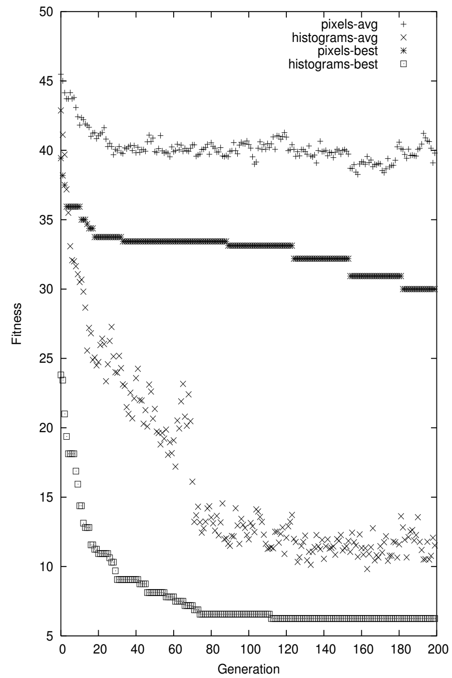

Fig. 5. Comparing runs on using pixels and histograms of image class 13 and class 2

While using pixels preserves spatial relationships, this approach has a number of disadvantages. First, the clustering is not as good. Second, the image size was limited to 16×16 pixels. Thus in our further experiments we have used the histogram representation. This has the further advantages that images of arbitrary size can be used as the histograms will always have 256 grey levels and are rotation invariant.

4.2 Histogram Inputs

To assess the quality and generality of the 78 evolved feature extraction algorithms we carried out three texture classification experiments using the standard train-a-classifier-and-test methodology. Experiment 1 was a 4 class problem involving Brodatz textures that were not in the learning set of 13. These are shown in figure 7. The training set consisted of 33 images and the test set 67. The 78 evolved feature extraction programs were applied to each image to generate a feature vector of 78 features. The feature vectors of the training set were then used to train a nearest neighbourhood classifier followed by calculation of the error rate on the feature vectors of the test set. The 78 evolved features gave a test accuracy of 100% while Haralick features gave a test accuracy of 95.5%.

Experiments 2 and 3 were carried out using the same methodology as [18], in which a large number of feature extraction algorithms were compared on a

Table 1. Performance of Various Feature Extraction Algorithms All results, except for the last 3 are from [18].

Feature Set	Brodatz	Vistex
Unser	92.6%	81.4%
Galloway	84.7%	70.4%
Laine	92.4%	75.6%
Local features	61.1%	47.1%
Fractal(1)	62.6%	54.5%
Fractal(2)	66.5%	48.5%
Laws	89.7%	79.8%
Fourier coeff.	92.7%	80.1%
Chen	93.1%	84.5%
Sun & Wee	63.9%	58.4%
Pikaz & Averbuch	79.4%	74.4%
Gabor	92.2%	75.4%
Markov	83.1%	69.6%
Dapeng	85.8%	74.6%
Amadasun	83.4%	65.6%
Mao & Jain	86.3%	73.0%
Amelung	93.0%	82.1%
Haralick	86.1%	75.5%
GP Features	*81.5%*	*74.8%*
GP Features + Haralick	88.2%	83.2%

number of texture classification problems. We did this in order to enable direct comparison with our evolved feature extraction algorithms. In experiment 2, the 13 Brodatz textures used in the learning set were used to give a 13 class problem. There were 32 images (64 ×64) per class in the training set and 64 images per class in the test set. The results are shown in the table 1. The evolved features outperform 5 of the 18 methods tested. However, this is not a true test of generality since the features are being tested on the same data they were evolved from.

In experiment 3, texture images from the Vistex set were used. In this problem there are 15 classes of textures (figure 9), each presented as a 64 ×64 image. There are 32 images per class in the training set and 64 images per class in the test set. The results are shown in the last column of table 1. Using the features from the evolved programs (GP Features) gave an accuracy of 74.8% which is very competitive with the Haralick accuracy of 75.5% and better than 8 of the 18 methods tested. The learning time is about 7 hours on a Xeon 2.8 Ghz computer.

5 Analysis of the Evolved Algorithms

Since + is the only function, all of the evolved algorithms are sums of the number of pixels at certain grey levels. For example, the feature extraction program evolved from class1 and class2 Brodatz textures is X109 + 2*X116 + 2*X117

+ X126 + 2*X132+ X133 + 2*143 + X151 + X206 + X238 + 3*X242 + X254, where X*nnn* represents the value of the histogram at grey level *nnn*. If we examine the histograms of two images, shown in figure 6, we can see the program has make use of the points between grey level 100 and grey level 150 and those above grey level 200 where class1 is significantly different from class 2.

5.1 New Texture Regularities

A major question about the evolved feature extraction algorithms is whether any previously unknown texture regularities have been discovered. In general this is a very difficult question to answer, however, if the accuracy achieved by using Haralick and GP features together is significantly higher than the accuracy achieved by using Haralick features alone, then a reasonable case can be made that some new texture regularities, not captured by the Haralick approach, have been discovered. When the 78 GP features were added to the Haralick features, the accuracy on the Brodatz problem improved from 86.1% to 88.2% and the accuracy on the Vistex problem improved from 75.5% to 83.2%. We can conclude that some new regularities have been discovered. However, we have not been able to determine the nature of these regularities.

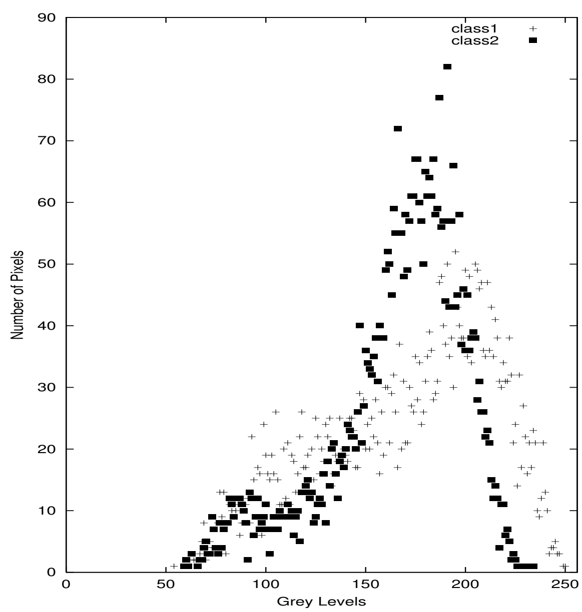

Fig. 6. Histograms of class1 and class2 Brodatz textures

6 Conclusions and Future Work

We have established that evolutionary search with genetic programming can be used to generate image texture feature extraction algorithms that are competitive with human developed algorithms on a difficult texture classification problem involving the Vistex library. Histogram inputs, rather than pixel values, should be used as inputs to the algorithms. The algorithms have captured some texture regularities, but it is very difficult to determine what they are.

Since the learning set contained only examples from the Brodatz set, but training and testing of the classifier using the evolved algorithms was performed on the Vistex set, there is some evidence that the feature extraction algorithms are general. However, more work needs to be done with other texture classification problems to verify this. We also plan to repeat the experiments with different selections from the Brodatz library and more than just pairs of textures in the learning sets.

The performance of the current feature extraction algorithms is limited by the fact that there is no spatial information in the histograms. We need to look at alternative representations that keep the number of inputs to a maximum of 256 but which capture more of the spatial information in a picture.

Acknowledgments

This work was funded by Australian Research Council SPIRT grant number C00107351 to RMIT University and Carlton and United Breweries. We acknowledge the support of Peter Rogers of Carlton and United Breweries. The evolutionary runs were performed on the Linux cluster hosted by the Victorian Partnership for Advanced Computing.

References

1. Koza JR, Keane MA, Streeter MJ, Mydlowec W, Yu J, Lanza G "Genetic Programming IV Routine Human-Competitive Machine Intelligence" Kluwer, 2003, 1-10
2. Tuceryan, M. and Jain, A.K."Texture Analysis" in Handbook of Pattern Recognition and Image processing, World Scientific, 1993, Chapter 2, 235-276
3. Rignot E, Kwok R, "Extraction of Textural Features in SAR images: Statistical Model and Sensitivity" in Proceedings of International Geoscience and Remote Sensing Symposium, Washing DC , 1990, 1979-1982
4. Jain AK, Farrokhnia, Alman DH, "Texture Analysis of Automotive Finishes" in Proceedings of SME Machine Vision Applications Conference, Detroit, 1990, 1-16
5. Chen CC, Daponte JS, Fox MD, "Fractal Feature Analysis and Classification in Medical Imaging" IEEE Transactions on Medical Imaging, 1989, 8, 133-142
6. Jain AK, Bhattacharjee SK, Chen Y "On Texture in Document Images" in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Champaign, Il, 1992
7. Tackkett WA , "Genetic Programming for Feature Discovery and Image Discrimination", in Proceedings of the 5th International Conference on genetic Algorithms, ICGA-93, University of Illinois at Urbana-Champaign, 17-21 July 1993, 303-309

8. Poli R, "Genetic Programming for Image Analysis" in Genetic Programming 1996: Proceedings of the First Annual Conference, MIT Press, 1996 363-368
9. Agnelli D, Bollini A, Lombardi L, "Image Classification: An Evolutionary Approach", in Pattern Recognition Letters 2002 ,Volume 23, 303-309.
10. Ross BJ, Fueten F and Yashkir DY, "Automatic Mineral Identification using Genetic Programming", in Technical Report CS-99-04, Brock University, December 1999
11. Yingqiang Lin and Bir Bhanu, "Learning Features for Object Recognition", GEC2003, LNCS2724, 2003, 2227-2239
12. Koza J R, "Simultaneous Discovery of Detectors and A Way of Using The Detectors Via Genetic Programming" in 1993 IEEE International Conference on Neural Networks, San Francesco, Piscataway, NJ: IEEE 1993. Volume 3, 1794 - 1801.
13. Andre D, "Automatically Defined Features: The Simultaneous Evolution of 2-dimensional Feature Detectors and An Algorithm for Using Them" in Advances in Genetic Programming, 1993 Chapter 23, 477-494
14. Song A, Loveard T and Ciesielski V, T̈owards Genetic Programming for Texture Classification;, in AI 2001 Advances in Artificial Intelligence, Proceedings of the 14th Australian Joint Conference on Artificial Intelligence. Adelaide, December 2001, Springer-Verlag, Lecture Notes in Artificial Intelligence 2256, 461-472.
15. Martin C M, "Genetic Programming for Real World Robot Vision", in Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, Sept-Oct 2002, 62-72.
16. Harris C and Buxton B, "Evolving Edge Detectors with Genetic Programming" in Genetic Programming 1996: Proceedings of the First Annual Conference, Stanford University, CA, USA, 28-31 July 1996, MIT Press, 309-315
17. Haralick M., Shanmugam K., and Distein I., "Texture Features for Image Classification", IEEE Transactions on Systems, Man, and Cybernetics, SMC-3(6):610-621
18. Wagner T , "Texture Analysis" in Handbook of Computer Vision and Applications, 1999, Volume 2, Chapter 12, 276-308
19. Dylan Mawhinney, RMIT-GP version 1.3.1, the RMIT University, 2002.
20. Len Trigg, Mark Hall and Eibe Frank, Weka Knowledge Explorer version 3-3-4, The University of Waikato 1999

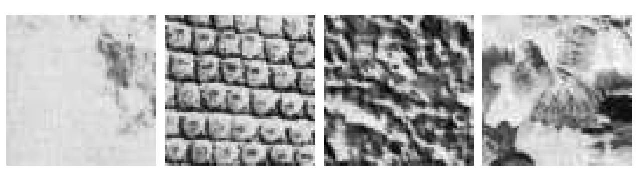

Fig. 7. Four Brodatz textures used for testing

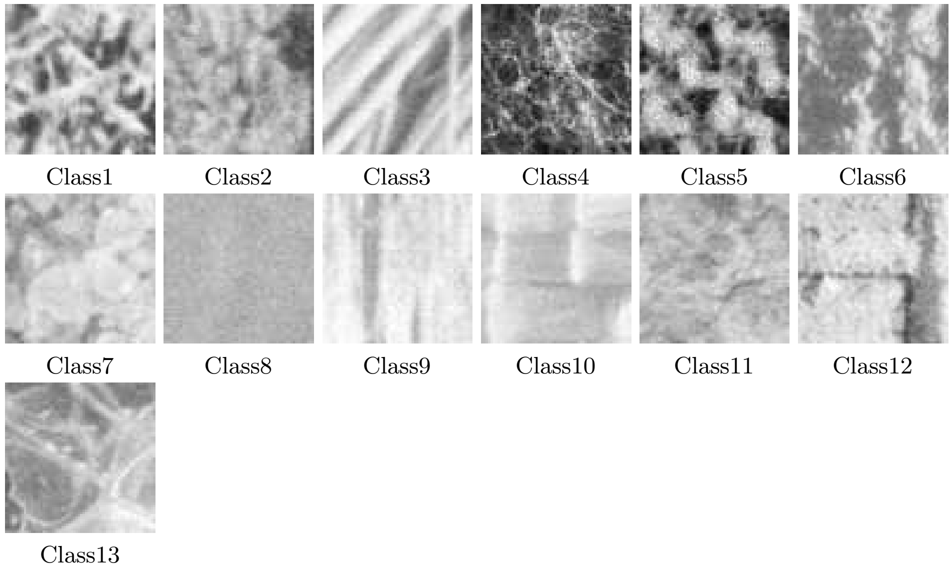

Fig. 8. Brodatz Texture Images

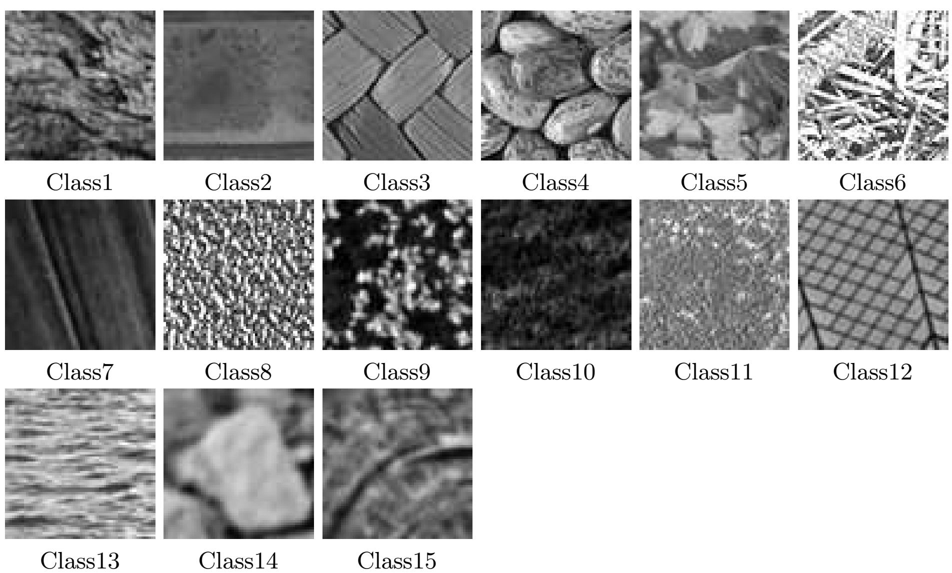

Fig. 9. Vistex Texture Images