Texture analysis using neural networks and mode filters

D. Greenhill E.R. Davies

Machine Vision Group, Dept. of Physics, Royal Holloway, University of London,
Egham Hill, Egham, Surrey, TW20 0EX, UK

Abstract

This paper describes experiments in which the Laws' approach to texture analysis is augmented by artificial neural network classifiers followed by mode filters. Analysis of image classification data confirmed that averaging filters should be carefully chosen to obtain the best balance between remanent texture noise and blurring of class boundaries, smaller mask sizes being required for segmentation vis-à-vis classification. Preliminary image normalisation was found to be generally advisable, unless the training data is fully representative of the test data, while mode filters are valuable for consolidating the output images, e.g. by eliminating patches of misclassification. Indeed, mode filters seem to constitute an under-exploited technique in image analysis as a whole.

1 Introduction

Much work has been done in the area of texture analysis [1]. However, it is a complex statistical topic and there is still new ground to be broken to understand fully how to perform texture analysis on any new set of image data. In particular, it seems important to attempt to set up texture analysis algorithms systematically, if possible by methods incorporating as much training as possible.

In this work an artificial neural network was used to classify textures. Neural networks have the advantage over other classifiers that although training times can be long, both the amount of storage and the time required to test on a particular texture are small. The outputs can be treated as giving the probability of a feature vector belonging to a particular class. However, a significant number of parameters need to be preset into neural networks, or adjustments made by human operators, so that one could end up little better off than if conventional pattern analysis tools were used. Other work on texture classification using neural networks includes [2,3]. Du Buf et al. [4] describe methods to determine the accuracy of texture segmentations.

We took as our starting point the Laws' approach to texture analysis [5,6] and Pietikäinen et al. [7], and after applying the necessary energy averaging filters, used trainable neural networks to interpret the outputs (Figure 1). We also tried the effect of smoothing the final output images using mode filters, since the output images are often characterised by small noisy patches which are wrongly classified: ultimately, the reason for this lies in the statistical nature of the textures.

In this paper, we start by describing the basic techniques used. Then we proceed to present our results on cloth textures taken from the Brodatz album [8], the results being summarised by a series of fairly complex graphs; the latter are then discussed and examined in detail, and some fairly general results are obtained, many of which are not especially geared to the use of neural networks – i.e. they

BMVC 1993 doi:10.5244/C.7.51

exhibit a degree of generality which we believe will be of wide interest.

2 The basic approach

The Laws' approach to texture analysis was chosen for a number of reasons. These include ease of implementation, generality, reversibility of the basic transforms, and capability for giving low error rates. Implementation is straightforward because the Laws' convolution masks can be applied conveniently in software or in dedicated hardware. It is also the application of convolutions that gives the method generality. In addition, for a 3 × 3 window, there are nine Laws' masks, and these merely provide a linear transformation of the input intensities, from which the latter could if necessary be reproduced. Thus there is no loss of image information until after this initial stage of low-level processing. Since they can be computed very quickly, the measures can be calculated on a pixelwise basis. With slower methods, such as co-occurrence techniques, it may be necessary to calculate a single value for each region and then interpolate values for individual pixels – a process which is likely to lead to less accurate results. We used 3 × 3 Laws' masks in all the work reported in this paper. More complex textures, or textures with larger texel sizes, would be likely to require larger mask sizes.

The next stage of processing is to average the energies (absolute values) of the mask outputs over a reasonable area, so that the highly statistical nature of the texture is suppressed as far as possible. Naturally, this process is also likely to blur the boundaries between various textured regions, so over-large smoothing masks should not be used. In fact, there is a tradeoff between suppressing texture noise and eliminating image detail and resolution, a best balance being judged either subjectively from the processed images or numerically by analysing the percentages of areas that are misclassified (this aspect will be demonstrated by the results of Section 3). There is no reason why these parameters could not be determined automatically, given a suitable set of training data.

The third stage of processing is to apply a neural network classifier to judge from the available local energy measures which is which of the textures (four in our experiments) that are presented to the classifier. In our work we used the standard back-propagation algorithm [9]. Neurons in the back-propagation network are connected together in layers. The inputs to the net were the nine energy measures. These fed into the nine nodes in the first hidden layer (Figure 1). Each of these nodes can be thought of as having the responsibility for dealing with one of the inputs. The output layer consisted of four nodes (because we had four texture classes). They examined the outputs of all the hidden layer nodes. During training, the target output was formed by setting to *high* the output node which represented the class of the training example, the others being kept *low*. During testing, the class of the input data was determined from the node with the highest output value.

The neural network was trained for about 10 million presentations. With the update coefficient η set to 0.20 and the momentum term α set to 0.01, a reasonable success rate was achievable with relatively few presentations, but a considerably larger number of presentations was used to ensure that lack of training was not responsible for unnecessary classification errors during testing: although overtraining can limit the generalisation properties of a network, there was no evidence of this in our experiments. Overall, the effect of the network is to place the features in order of importance and give them the appropriate weighting. In this respect the end result is like that obtained using principal components analysis [10], where features are ordered in terms of their correlation with each other.

The training data consisted of a set of five images. The first four were 256×256 pixel images of four Brodatz textures. From these a fifth composite texture image was generated. The testing image was a second composite image generated from different samples of the same four textures. There was therefore four times as much training data as testing data. The composite texture image was useful because it forced the classifier to learn which texture was which in regions of ambiguity close to texture boundaries. Preliminary experiments showed that, without it, the classifier did not learn so effectively how to segment at texture edges. After training had taken place, comparisons were made between the two composite texture images, and the test results of Section 3 do not make use of training data from single texture images.

As indicated above, we found that the outputs from the neural network were not perfect because of the statistical nature of the textures. We therefore employed mode filters in an effort to suppress misclassified patches in the output images (Figure 1). The mode filters used in these experiments ranged from 1×1 pixels (representing no mode filtering) to 17×17 pixels. In this case, implementation of mode filters devolved into counting up the numbers of pixels belonging to each class, and choosing the class which had accumulated the largest number of votes. Note that in the examples shown, the classes of the textures are represented by intensities, although the actual intensities used have no particular significance.

Further information was also required on the effects of normalising the input images to eliminate variations in background intensity and contrast. This was carried out by taking the basic input textures and pre-normalising them, before forming composite images for testing the texture analysis algorithms. Thus these results will not be applicable to normal composite raw images, but they are intended to provide useful data on the value of attempting to do normalisation. The reason for this is twofold: first, if we have a texture analysis technique and try it on images that clearly have significant intensity variations, then there is the possibility that the technique will 'cheat' and analyse the intensities instead of the texture; on the other hand, if we have a good texture analysis method, it may actually be muddled by the raw intensities it receives and perform worse than otherwise. We wanted to determine the reality of the situation.

The images were normalised by finding their global mean intensity I_M. A raster scan was then carried out using a large 32×32 pixel mask. The mean intensity I_R of the smaller region within the mask was found. The intensity of the current pixel was then incremented by $I_M - I_R$ so as to reduce any underlying intensity variations. The textures were then histogram equalised using the procedure described in Gonzalez and Wintz [11] p. 119. This expanded the range of intensities to fill the entire scale and flattened the intensity histogram to reduce the first-order statistics.

3 Experimental data

Our input image data are shown in Figures 2 and 3, the output image data are shown in Figures 4–7, and our experimental results are summarised in the graphs of Figure 8. There are four sets of graphs, covering tests of (1) whole images and parts of the images near edges, and (2) normalised and non-normalised images. Within the graphs, the various curves cover (a) tests on the training data and the test data, (b) use of texture energy averaging masks of four sizes in the range 5×5 to 17×17 pixels, while the variations within each curve represent the effects of final mode filters of sizes 1 × 1 to 17 × 17 (1 × 1 representing no mode filtering).

We now analyse these graphical results. There are a number of features which should be noted:

1. Test data are generally worse than training data, though at the high end of the mode filter sizes there are notable exceptions to this.
2. Small texture energy smoothing masks perform especially poorly (though this is partly offset by the effects of the largest mode filter masks).
3. The optimum size of smoothing masks for the textures is about 13 × 13.
4. Use of raw texture images gives better results than use of pure (normalised) textures.
5. Use of raw texture images also gives worse results than use of pure textures, if the parameters are badly chosen, and particularly when examining the results rigorously on test data.
6. Results in the vicinity of edges are worse than for the bulk of a textured region.
7. There is a suggestion that edge results are optimised at slightly smaller averaging filter sizes than for the full images.
8. If anything, the edge results are optimised at slightly smaller mode filter sizes than for the full images.
9. There is some sign that classification near edges is poorer for large size mode filters.
10. There is more need for carefully chosen parameters when edge information is to be measured.
11. The optimum size of mode filters for the textures used is around 11 × 11, i.e. about the same as the optimum for the previously applied energy smoothing filter.

We now comment individually on these results:

1. The test images used for these experiments give a rather spurious [1] result for the 5 × 5 test images for large mode filter masks, the test images apparently being rather more straightforward to classify, at least for this purpose (notice that the result is not replicated outside the bounds of the expected statistical variation for larger size smoothing masks). The lesson is perhaps that if system parameters are poorly chosen, then training and/or test results may arbitrarily be worsened. On the other hand the optimum will tend to be clear.
2. This should be of little surprise considering the statistical nature of the textures being analysed.
3. The same comment as for item 2 applies here (though a priori, 13 × 13 is a fairly large optimum mask size).
4. This can be explained by the fact that the raw images retain more of the original image information, though the effect is not large – at best a 0.3% improvement.
5. The poor performance resulting when using the raw images if the system is not optimally set up does tally with it being muddled by strong irrelevant (or negatively correlated) intensity information. This indicates that it will in general be best to revert to using pure (normalised) textures if there is any risk that the system is not properly set up. This applies especially if the training data is not representative of what will happen during testing (note especially the variation between the outputs for the training and testing data in the case of raw images).

[1] That this result is spurious has been confirmed by more recent experiments not described in detail here.

6. The poorer results in the vicinity of edges are to be expected as the averaging and mode filter masks will receive contradictory data in these regions.

7. It might be expected that smaller averaging filter sizes would be beneficial near edges, for the same reason as for item 6.

8. Smaller mode filter sizes would also lead to better judgement of edge positions and hence less overall error in this case.

9. This result could be misleading (see item 1), but would also not be unexpected (see item 8).

10. There is rather more scatter in the results for edges than for the bulk of the texture regions, and in view of the delicacy of the judgements of the edge positions, it is perhaps not surprising that the system parameters would have to be more carefully set up for classification in the region of edges.

11. At first it is not surprising that the mode filter sizes should be similar to those for the averaging filters. However, one might alternatively suspect that mode filter sizes should be able to compensate for poor setting of averaging filter sizes, and vice versa (i.e. the sum of the two sizes should optimally be held constant). That this does not appear to happen must be due to the difference in nature of the two types of filter, and in particular the fact that the mode filter is highly non-linear, whereas the averaging filter is linear and results in severe blurring if over-applied. Hence there is a fairly clear optimum for the averaging filter, but only a minimum effective size for the mode filter (though there is a suggestion that above this size it causes a slowly varying deterioration in performance).

There are some further observations and explanations which arise on examination of the classified images shown in Figures 4–7:

12. Examination of Figure 4 shows immediately why use of a mode filter is useful – because of the noise generated by the statistical nature of the texture. While this problem is much reduced when larger averaging masks are used, mode filtering is still found to be useful in cutting down classification errors – see Figure 7.

13. A further interesting effect occurs with mode filtering: when three different textures are in contact at a given corner of the pattern of texture squares (this always happens in these images when the same texture appears in diagonally adjacent squares), the mode filter tends to generate connectivity between them; this is because, in the region of the corner, that texture is necessarily in the majority, and will cause a link whose width is governed by the size of the mode filter mask. Note, however, that this effect is to a large extent an artefact of this data set: in general, no more than three regions would be expected to meet at any one location in an image.

14. Figure 6 shows an effect where the mode filter cannot substantially improve performance: where the raw rather than normalised images are used, the classification of two squares with the same texture near the bottom left[†] of the image is wrong over quite large regions. This is because the intensity levels in the test and training images do not match sufficiently: this is confirmed by the fact that the same two squares are well classified in the corresponding training images (Figure 5). In this context, note that Figure 6 corresponds to easily the worst cases shown in the graphs of Figure 8.

15. Figure 7 also shows several places where substantial errors are made by the classifier, the deduced boundaries not following the outlines of the square regions within each image. However, the square regions are too small for details of the herringbone texture* to be discerned accurately, with the result that both the eye and the algorithm become muddled at this point; in addition, there is a genuine defect in one of the textures in the lower part‡ of this image, so it is not surprising that a misclassification is recorded here. These points reflect the fact that texture analysis is a complex task, so purely numerical results such as "97% success rate" must be treated with due caution.

4 Conclusion

This work has used an artificial neural network as a convenient tool for implementing a texture classifier. Various tests have been made of the whole texture analysis system using this tool. In general these tests vary parameters other than those of the neural network, and arrive at useful results on how to design texture analysis systems, the most important lessons being the following:

A. Averaging filters should be carefully chosen to minimise classification error, the result being a tradeoff between eliminating texture noise and blurring the output data.

B. Mode filters should be applied to consolidate the classifications already obtained and eliminate patches of misclassification; for this purpose at least a certain minimum size of mask is needed, this size being similar to the optimum for the averaging filter, though the penalty for going above the minimum size is very small (except at points where many texture regions meet).

C. Image normalisation is in general advisable, unless it can be guaranteed that the training data is fully representative of the test data *and* that system parameters have properly been optimised: if these two conditions have been met, then a small gain in performance could result from *not* normalising the data.

D. Classification near edges is more critical and is likely to require somewhat smaller mask sizes than for large-scale texture classification. We are, however, surprised that our data shows such a small effect of this nature.

While A and D provide confirmation of previous work in the area, B and C lead to more specific recommendations: in particular, we suggest that mode filters constitute a presently an under-exploited technique in image analysis as a whole, and our work should be of value in expanding their use from image enhancement [12] to texture analysis; in the latter case mode filters have the role of making the most of the data provided by the earlier purely texture-processing procedures.

Acknowledgements

E.R. Davies is grateful to the ACME Directorate of the SERC for financial support of the research. D. Greenhill is grateful to the SERC for a Research Studentship.

References

1. Van Gool, L., Dewaele, P. and Oosterlinck, A. (1985) "Texture analysis anno 1983", Comput. Vision Graph. Image Process., 29, pp. 336–357
2. Daponte, J.S. and Sherman, P. (1991) "Classification of ultrasonic image texture by statistical discriminant-analysis and neural networks", Computerized Medical Imaging and Graphics, 15, pp. 3–9

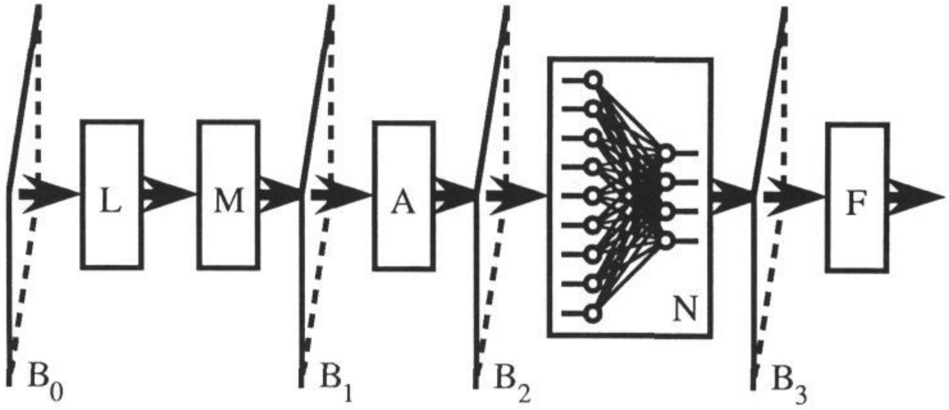


Figure 1: Architecture of complete texture classifier

This figure shows the complete texture analysis system. B_0 is the input image buffer, and B_1–B_3 are further image buffers used for storing temporary results (the operations following them involve window operations). L is the Laws' mask processor employing nine 3×3 convolution masks. M is a set of nine absolute magnitude operators, used to estimate the local texture energy. A is a set of nine masks of adjustable size, used to average the texture energy locally. N is the artificial neural network classifier, and has one hidden layer of nine nodes and an output layer of four nodes. F is an output mode filter of adjustable size, which is used to consolidate the output pixel classification.

3. Manjunath, B.S., Simchony, T. and Chellappa, R. (1990) "Stochastic and deterministic networks for texture segmentation", IEEE Trans. Acoustics, Speech, Signal Process., 38, pp. 1039–1049

4. Du Buf, J.M.H., Kardan, M. and Spann, M. (1990) "Texture feature performance for image segmentation", Pattern Recogn., 23, pp. 291–309

5. Laws, K.I. (1979) "Texture energy measures", Proc. Image Understanding Workshop, Nov., pp. 47–51

6. Laws, K.I. (1980) "Textured Image Segmentation", PhD thesis, Univ. of Southern California, Los Angeles

7. Pietikäinen, M., Rosenfeld, A. and Davis, L.S. (1983) "Experiments with texture classification using averages of local pattern matches", IEEE Trans. Systems Man Cybern., 13, no. 3, pp. 421–426

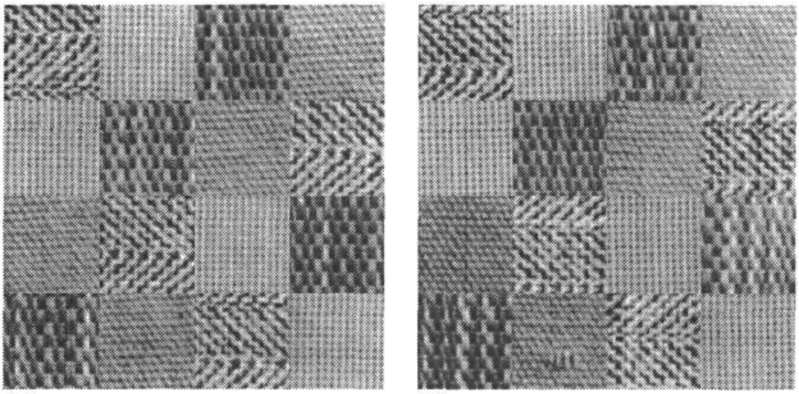
8. Brodatz, P. (1966) Textures: a photographic album for artists and designers. Dover Publications

9. Rumelhart, D.E., McClelland, J.L. and the PDP Research Group (1986) Parallel Distributed Processing. The MIT Press

10. Ade, F. (1983) "Application of principal component analysis to the inspection of industrial goods", Proc. SPIE Int. Technical Conf., Geneva, 397, pp. 216–223

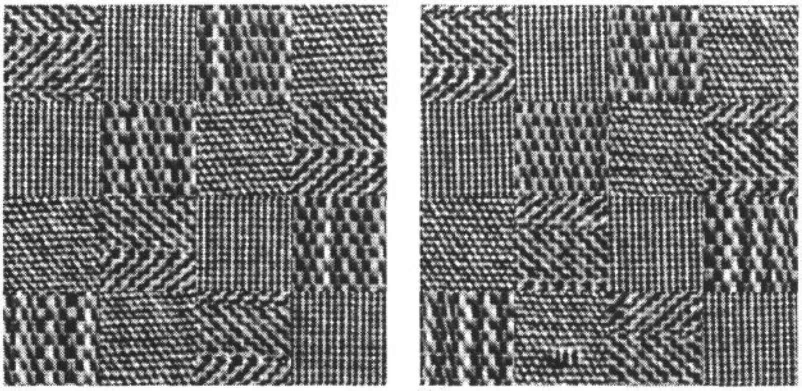
11. Gonzalez, R. and Wintz, P. (1979) Digital Image Processing, Addison Wesley

12. Davies, E.R. (1988) "On the noise suppression and image enhancement characteristics of the median, truncated median and mode filters", Pattern Recogn. Lett., 7, no. 2, pp. 87–97



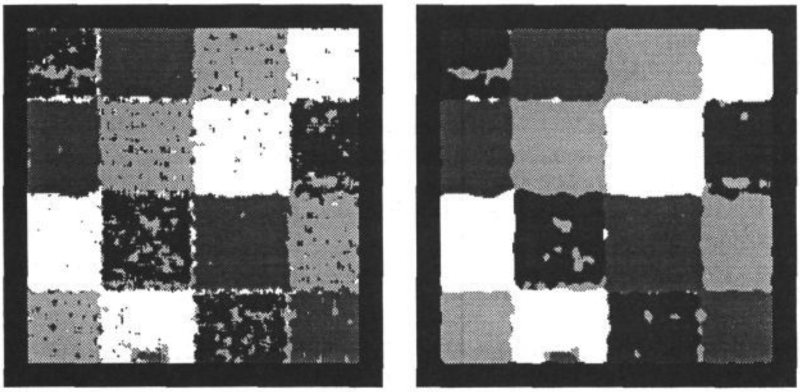
(a) (b)

Figure 2: Training and testing images: raw texture



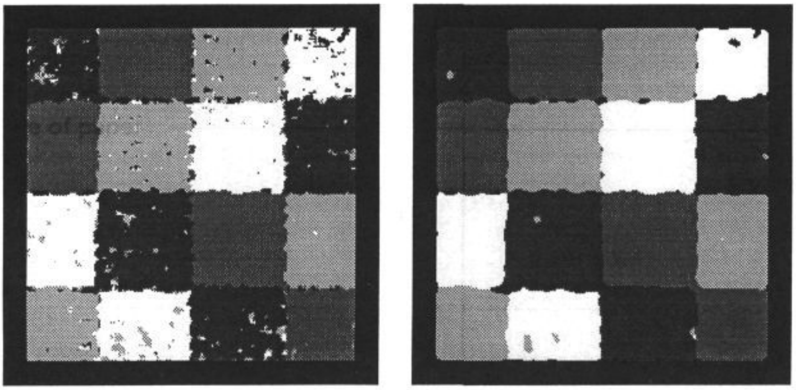
(a) (b)

Figure 3: Training and testing images: pure texture



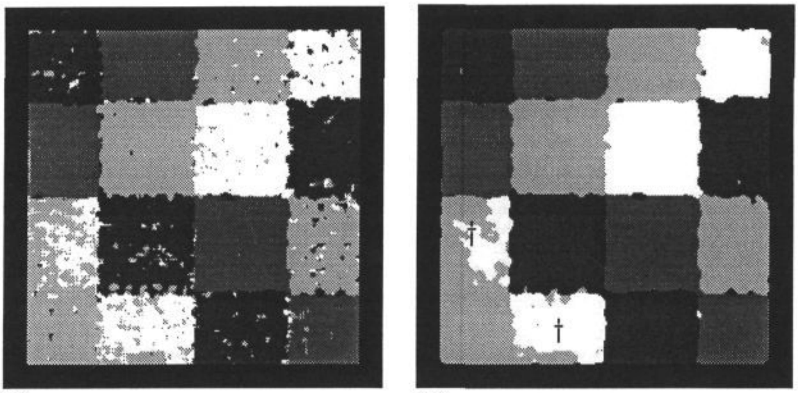
(a) (b)

Figure 4: Testing images: pure texture, 5 x 5 mean mask, (a) 1 x 1 and (b) 7 x 7 mode mask



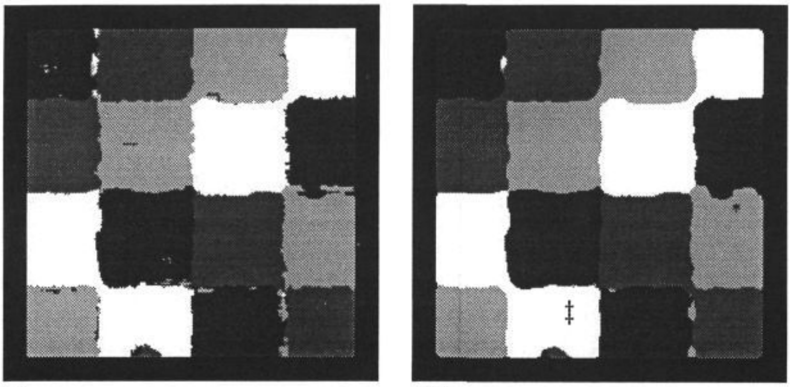
(a) (b)

Figure 5: Training images: raw texture, 5 x 5 mean mask, (a) 1 x 1 and (b) 7 x 7 mode mask



(a) (b)

Figure 6: Testing images: raw texture, 5 x 5 mean mask, (a) 1 x 1 and (b) 7 x 7 mode mask



(a) (b)

Figure 7: Testing images: pure texture, 13 x 13 mean mask, (a) 1 x 1 and (b) 7 x 7 mode mask

†*‡ See items 14, 15, 15 respectively in Section 3.

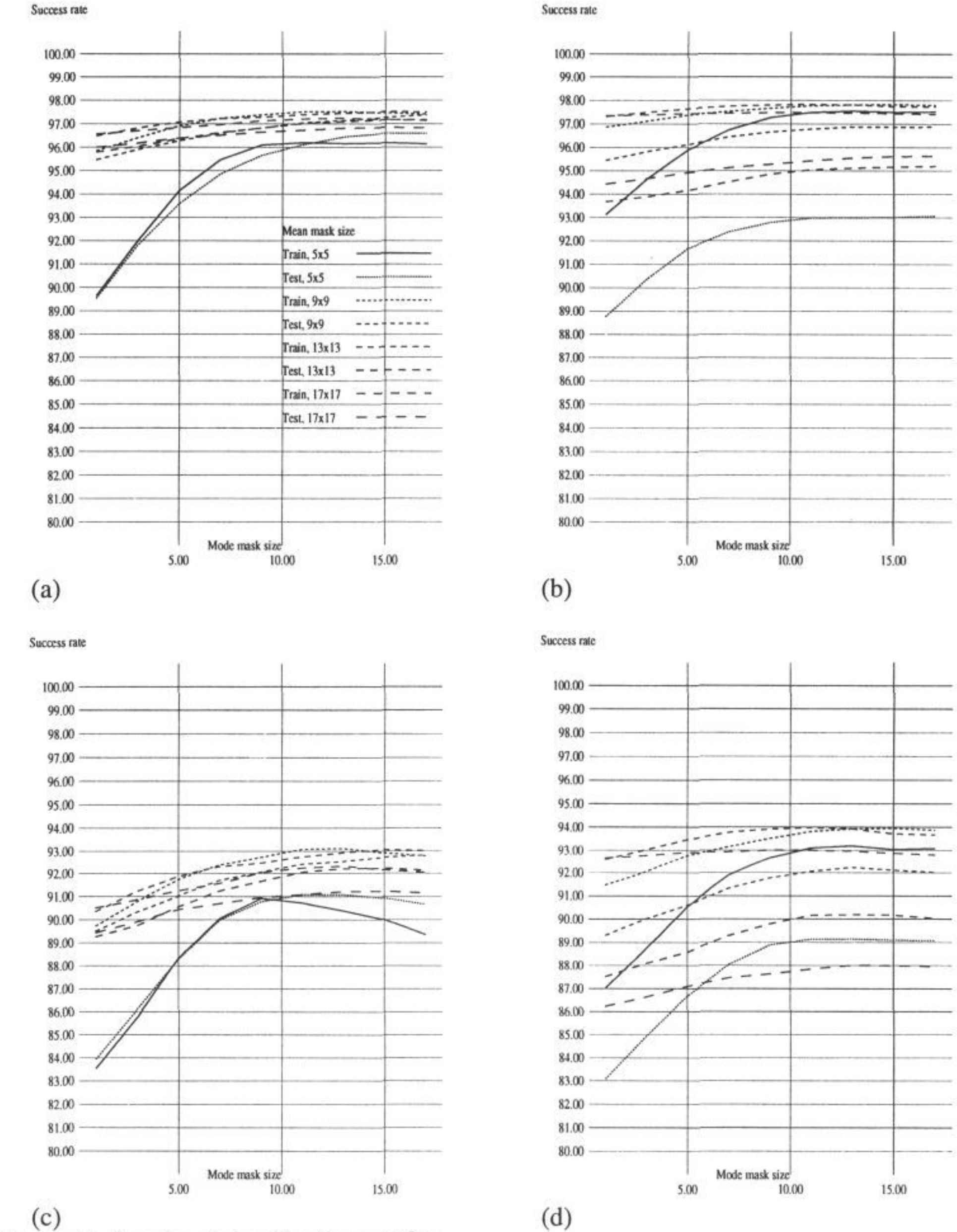


Figure 8: Graphs of classification results

These graphs show the success rates for the texture classification experiments described in the text. (a) and (b) show the respective results for 'pure' and 'raw' textures with the full images, while (c) and (d) show the corresponding results for the parts of the images near texture edge boundaries. The horizontal axis variation is the size of the mode filter; the various individual curves represent the training and testing images and the various energy averaging filter sizes (see key).