SUPERVISED TEXTURE CLASSIFICATION - SELECTION OF MOMENT LAGS

Vladimiros Antoniades and Asoke K. Nandi

Signal Processing Division, Department of Electronic and Electrical Engineering
University of Strathclyde, 204 George Street, Glasgow G1 1XW, U.K.
e-mail: vladimir@spd.eee.strath.ac.uk, asoke@eee.strath.ac.uk

ABSTRACT

This papers deals with supervised texture classification. The extracted features are the image second and third order moments. The number of possible moment lags for 2-D signals increases rapidly with the order of the moment even for small lag neighbourhoods. The paper focuses on the selection of moment lags that optimise classification performance. Lag selection also serves another purpose: it waives us from the trouble of calculating a large number of moments every time a new sample is to be classified. Lag selection is performed by a full stepwise feature selection method using four different feature evaluation measures. The selected moments are driven to four classifiers and comparative classification results are obtained.

1 INTRODUCTION

Texture classification is an important task in various image processing problems, ranging from large scale satellite images to microscopic images used in medical applications. Classification is usually performed on a feature vector extracted from the images. Different kinds of features have been proposed in the literature for this task, including cooccurrence matrices, Gabor filtering, autoregressive models and fractals. In this work, we use the image second and third order moments as features for classification. A few relevant approaches are reported in the literature ([1],[2],[3],[4]). Second order statistics have been generally used for 'clean' signal classification whereas third order statistics perform favourably on signals corrupted by noise. In this paper, we do not examine images corrupted by noise. Instead, we focus our attention on both second and third order moment lag selection. The issue has been touched by [1] and [3]. Our approach is more thorough and gives favourable results.

2 FEATURE EXTRACTION

We use as features two sets of the image moments. The second and third order moments of a 2-D stationary stochastic signal $f(\vec{x})$ are respectively defined as:

$$m_2^f(\vec{x}_1) = < f(\vec{x}) f(\vec{x} + \vec{x}_1) >$$

$$m_3^f(\vec{x}_1, \vec{x}_2) = < f(\vec{x}) f(\vec{x} + \vec{x}_1) f(\vec{x} + \vec{x}_2) >$$

where $\vec{x}_i = (x_i{}^x, x_i{}^y)$, $i = 1,2$ are the 2-D lags and $<\cdot>$ denotes the ensemble average operator. Under the assumption of ergodicity, the above formulae hold for space averages too. For zero mean random fields, moments and cumulants up to the third order are identical. In the following analysis we will use both terms with the same meaning.

With some similarity to [1], we choose as our first set of features all second order moments with lags $\vec{x}_1$, so that $-4 \le x_1{}^x, x_1{}^y \le 4$. Taking into account that permutation of lags does not change the value of the moments of any order, we reduce to 41 second order moments.

It is well known that third order cumulants are insensitive to additive gaussian (or any other symmetrical) noise. However, noise insensitivity is not satisfied by moments with more than one pixels at the same position. In this work, we will not use any additive noise on our images, that is, we will only process 'clean' images, but we still impose the noise insensitivity restrictions so that our features can be used in the future when we deal with images corrupted by noise.

We use as a second feature set all third order moments with lags $-3 \le x_1{}^x, x_1{}^y, x_2{}^x, x_2{}^y \le 3$. If we remove redundancies arising from symmetries in the relative positions of the pixels and we also impose the noise insensitivity restrictions, we reduce to a set of 720 third order moments.

Third order cumulants can be also estimated through the bispectrum. This approach reduces the computational time enormously but on the other hand needs a huge memory to store the resulting complete 4-D third order correlation function. There are ways to circumvent this difficulty on the price of worsening the estimation accuracy, but we will not present the analysis here due to space limitations.

Within each estimation block (section 5), second order moments are normalised by the variance to get the autocorrelation coefficient. After normalisation, the moment with zero lags in both image directions becomes one so it does not offer any information and is discarded from the set. We therefore render to 40 second order moments.

Third order moments are similarly normalised by the third order moment with zero lags, $m_3^f((0,0),(0,0))$.

3 FEATURE SELECTION

The next issue to be dealt with is the choice of moment lags for optimal classification results ([1], [3]). Because the total number of estimated moments is too large, it is necessary to reduce it to a smaller number by choosing the most discriminating moments from the entire set. In any case, more features do not necessarily imply better classification results.

Before starting feature selection, a measure for feature evaluation must be defined. Feature evaluation is performed in terms of discriminating performance. We use four such measures: the percentage of classification error associated with a linear discriminant classifier (*pce*), the average between-class distance (*abd*), the minimum between-class distance (*mbd*) and the ratio of the between-class variance over the within-class variance (*bwv*). The first of these measures is associated with a particular classifier while the other three depend only on the statistical properties of the data.

We now come to feature selection. In order to find the optimal set of features, one has to check the performance of all possible subsets of the initial set. In practice, even for relatively small number of features, the number of possible combinations is enormous and renders this approach impossible. This being the case, sub-optimal approaches are sought.

In this work, two different approaches are employed. The first is to evaluate the discriminating performance of each feature separately and use the best m features as the final set driven to the classifier, allowing m to take different values.

The second and more powerful approach is to implement a full stepwise feature selection procedure. This procedure starts with choosing the best feature (with respect to discriminating performance) and adds at each step that feature from the rest of the set whose combination with the already chosen ones gives the best results. In order to avoid mistakes caused by selection order, each time a feature is selected the performance of the set is checked by rejecting one feature at a time.

At the beginning, both feature selection approaches will be tested to show the superiority of the second one. After that, only the full stepwise selection approach will be used for further processing.

The top plot of figure 1 shows the results for the two feature selection approaches (applied to BD9 of figure 2). The feature evaluation measure is *pce* and the features are second order moments. The solid line corresponds to the full stepwise method while the dashed line corresponds to features selected individually. We can clearly observe the superiority of the stepwise method. We can also observe the smoothness in the stepwise method in contrast to the abrupt changes appearing in the other method. Of course, for the first feature and for the whole set, both methods give identical results. The important difference between the two methods is in the way they progress. This brings us to the question: at which stage should we stop feature selection? A closer look at the figure (bottom plot) shows that *pce* starts fluctuating when it drops enough to be close to its minimum value. We therefore choose to stop feature selection at the first selected feature that does not minimise *pce* any more. On the figure, this means to stop at the twelfth selected feature.

A similar procedure is followed for the *abd* and *mbd* measures. However, this case is a bit different. Both a*bd* and *mbd* are distances in the pattern space and as such they can never decrease with feature selection. This becomes clearer in the following: in the original space where all features are employed, *abd* and *mbd* have some certain values. When only some features are selected, *abd* and *mbd* are the projections of the original distances in the selected feature subspace. The larger this subspace becomes, the larger (or equal) the projected distances become. This means that these distances never decrease with feature selection, although their increase rate decreases. This being the case, we can not apply the criterion used in the *pce* case to stop feature selection. Since there are no abrupt changes in the *abd* and *mbd* curve shape, we have to stop feature selection when an arbitrarily chosen number of features have been selected. We set this number to 20.

Furthermore, *abd* has the following property: for each selected feature, its value is added up to its previous value. Thus, *abd* for a set of features equals to the sum of the *abd* of each individual feature. This implies that for this measure the stepwise feature selection method reduces to the best individually selected features method. However, this is not the case for *mbd*. As *mbd* is the minimum distance among all pairs of classes, *mbd* can belong to a different pair of classes for different sets of features.

For the fourth measure, *bwv*, features are firstly scaled so that they have unit within-class variance. Afterwards, the ratio *bwv* is formed. Since the within-class variance is kept constant, increase of *bwv* means increase of the between-class variance. b*wv* is additive like *abd* which means that the stepwise feature selection method turns to selection of individually best features for this measure too. As for *abd* and *mbd*, we stop feature selection for *bwv* at the 20th selected feature.

To check the robustness of the features selected by the four measures, we employ different sets of data. We first select features on each of these sets and then apply them to the rest. For each set of data, features selected by *pce* on this set give a larger *pce* value than features selected on different data. Both a*bd* and *bwv* show a better performance; they select features with the same comparative performance on all sets of data. Finally, *mbd* gives totally inconsistent results. Nevertheless, the final appreciation of each measure will be only done after classification.

4 CLASSIFICATION

For feature classification, we use four classifiers. Each of them is briefly described in the following.

4.1 Minimum distance classifier

The minimum distance classifier assigns a new pattern to that class the centre of which lies closer to the pattern. Its mathematical formulation is:

$$Md_i(\vec{x}) = \sum_{k=1}^{d} (x_k - m_{i,k})^2$$

where $\vec{m}_i$ is the mean vector for class *i*. Pattern $\vec{x}$ is classified to the class that gives the minimum value of $Md_i(\vec{x})$.

4.2 Uncorrelated gaussian classifier

The uncorrelated gaussian classifier is a special case of the gaussian classifier. The gaussian classifier models each class by a gaussian distribution using the class mean vector and covariance matrix. The uncorrelated gaussian classifier makes the further assumption that all features are uncorrelated which, for normal distributions, means also independent. This implies that only each feature's mean value and variance are entered into the classifier, avoiding this way the computational complexity of large covariance matrices. Its formulation is:

$$Ug_i(\vec{x}) = \frac{1}{(2p)^{d/2}\prod_{k=1}^{d} s_{i,k}} \exp\{-\frac{1}{2}\sum_{k=1}^{d}\left(\frac{x_k - m_{i,k}}{s_{i,k}}\right)^2\}$$

where $m_{i,k}$ and $s_{i,k}$ are the mean and variance of feature *k* for class *i*. Pattern $\vec{x}$ is assigned to the class that maximises $Ug_i(\vec{x})$.

4.3 Linear discriminant classifier

The linear discriminant classifier uses as a measure for classification a linear combination of the pattern's components, in which the linear coefficients and constants are calculated for each class separately. That is,

$$Ld_i(\vec{x}) = \sum_{k=1}^{d} C_{i,k} x_k + E_i$$

with $C_{i,k} = \sum_{h=1}^{d} S_{k,h}^{-1} m_{i,h}$ and $E_i = -0.5\sum_{k=1}^{d} C_{i,k} m_{i,k}$

$\vec{m}_i$ is the mean for class *i* and $\vec{S}$ is the class scatter matrix which is assumed to be the same for all classes. Pattern $\vec{x}$ is assigned to the class that maximises $Ld_i(\vec{x})$.

4.4 First neighbour (1-NN) classifier

The first neighbour classifier is a non-parametric classifier. It classifies a new pattern to the class which has the closest sample to the pattern. Instead of only using the first neighbour, more neighbours can be taken into account. In this case, the pattern is classified to the class that has the majority of samples among the specified number of nearest neighbours. However, there are theoretical arguments, which are in accordance with our tests, that the classification error is minimum for the first neighbour classifier. For this reason, we only take into account first neighbours.

5 CLASSIFICATION RESULTS

In our experiments, we used 6 textured images from the Brodatz album: BD9 (grass), BD16 (herringbone weave), BD19 (woollen cloth), BD24 (pressed calf leather), BD29 (beach sand) and BD84 (raffia) (figure 2). Each of these images consists of 512×512 pixels having grey tone values in the range 0-255.

Moments were estimated on 32×32 non-overlapping image estimation blocks. Within each estimation block the image was zero-meaned. Second order moments were estimated for image blocks covering the whole image range while third order moments were only estimated on the first image quarters consisting of 256×256 pixels.

Classification results are shown on figures 3 and 4. In both figures, the measure on the *y*-axis is the percentage of correct classification. On figure 3, we can observe eight groups of lines, each pair of which corresponds to a classifier. The names of the classifiers appear on the *x*-axis. From each pair of groups corresponding to each classifier, the left one is for second order moments and the right one for third order moments. Within each group, we can discern four lines. These lines correspond to the four feature evaluation measures (*pce*, *abd*, *mbd* and *bwv* from the left to the right). The circle at the middle of the line is the average percentage of correct classification on the six textures while the width of the line shows the spread.

We can clearly observe the superiority of second order moments over third order moments. This is in accordance with the results reported in the literature. The linear discriminant classifier outperforms the other three classifiers by far on both sets of moments. The gaussian uncorrelated classifier shows a slightly better performance than the minimum distance classifier which is expected since they both lie on the same principle but the gaussian one takes into account class variance in addition to the mean. The first neighbour classifier, which is of a totally different nature from the rest three ones, shows a better performance than the gaussian and the minimum distance classifiers.

The behaviour of the four feature evaluation measures is not consistent. Depending on the case, the relative performance of the four measures changes. The highest percentage of correct classification is achieved by the linear discriminant classifier applied on second order moments with features selected by the *pce* measure. This is reasonable since in this case features have been actually selected to minimise the percentage of false classification.

Figure 4 shows a similar picture. In this case, only second order moments are employed. Each image is split into four quarters and features are selected on the first

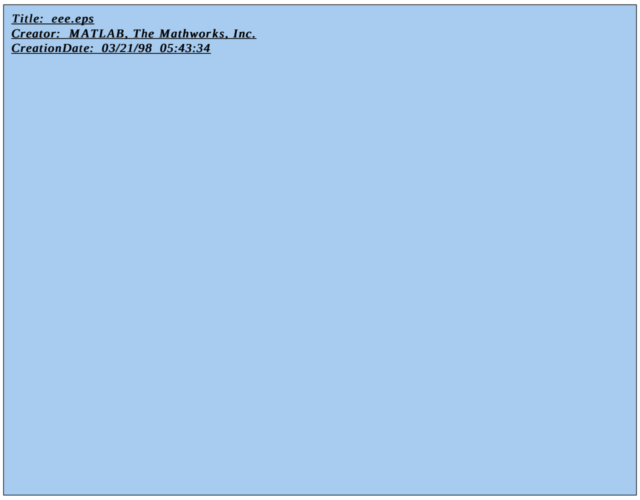

Figure 1

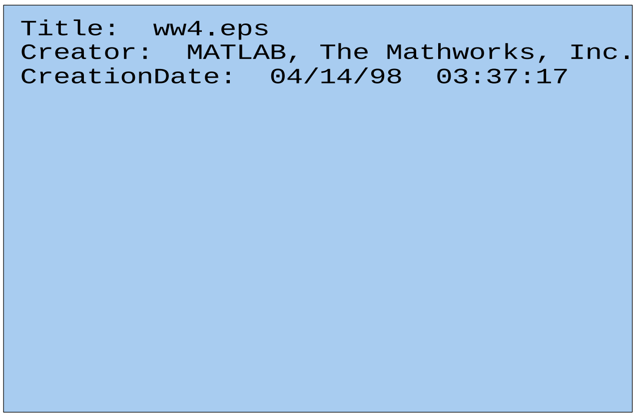

Figure 2

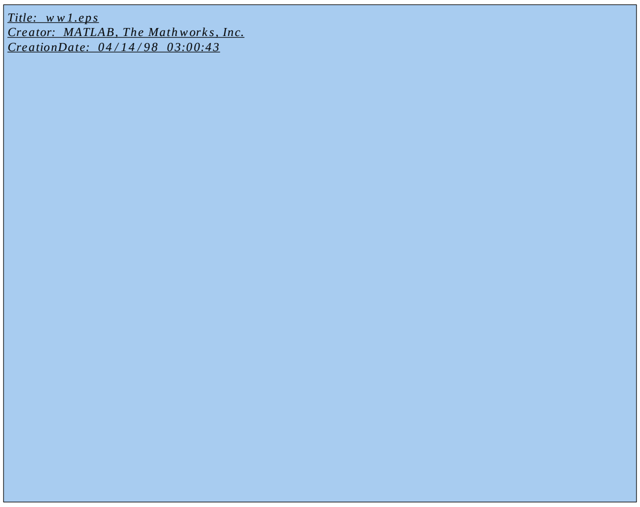

Figure 3

quarter. Then, the selected features are used for classification on all image quarters. The arrangement of the figure is similar to that of figure 3. For each classifier there are four groups of lines, each corresponding to a feature evaluation measure. Within each group, there are four lines corresponding to the four image quarters. The aim is to investigate the performance of the selected features on new sets of data.

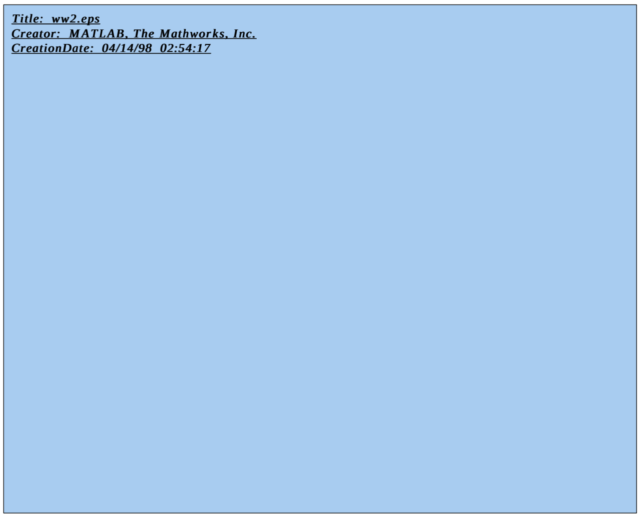

Figure 4

For the linear discriminant classifier, the classification performance is similar for all image quarters. The other classifiers show a more variable picture. The significant variation of the results does not allow us to make any rigid conclusion about the robustness of the features.

6 CONCLUSIONS

In this paper, classification has been performed on six Brodatz images. Features were both second and third order moments. A large number of moments were firstly estimated but only a few of them were kept for further processing. Our tests showed that a small loss in classification performance can be achieved by appropriate feature selection. Classification was performed by four classifiers. Considerably better classification results were achieved by the linear discriminant classifier. The robustness of the selected features was also checked by employment of different sets of data, but no clear conclusions can be drawn.

REFERENCES

[1] A.Makovec and G.Ramponi, “Supervised Discrimination of Noisy Textures using Third-Order moments”, Fourth COST 229 WG.1 and 2 Workshop on Adaptive Methods and Emerging Techniques for Signal Processing and Communications, Ljubljana, Slovenia, April 5-7, 1994.

[2] A.Makovec and G.Ramponi, “High-Order Moments for Texture Segmentation in Presence of Noise” Workshop on High Order Statistics in Signal Processing, Edinburgh, UK, September 1994.

[3] C.Ottonello and S.Pagnan, “Improvements of a Third-order Cumulant Classifier by Lag Selection”, Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, Greece, 20-22 June 1995

[4] A.Delopoulos, A.Tirakis and S.Kollias, “Invariant Image Recognition Using Triple-Correlations”, Signal rocessing VI: Thoeries and Applications, Elsevier Science Publishers B.V., 1992