Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Improved classification of crystallization images using data fusion and multiple classifiers

Samarasena Buchala and Julie C. Wilson*

Departments of Mathematics and Chemistry, University of York, York YO10 5YW, England

Correspondence e-mail: julie@ysbl.york.ac.uk

Received 18 March 2008
Accepted 13 May 2008

Identifying the conditions that will produce diffraction-quality crystals can require very many crystallization experiments. The use of robots has increased the number of experiments performed in most laboratories, while in structural genomics centres tens of thousands of experiments can be produced every day. Reliable automated evaluation of these experiments is becoming increasingly important. A more robust classification is achieved by combining different methods of feature extraction with the use of multiple classifiers.

1. Introduction

Macromolecular structure determination by X-ray crystallography requires numerous experiments to establish the conditions that will produce diffraction-quality crystals. Despite efforts to model protein-solution thermodynamics (Neal *et al.*, 1998), there is currently no *a priori* method to determine the optimum crystallization strategy for a particular protein. An exhaustive search of all combinations of reagents and experimental parameters is impossible and various screens have been designed to reduce the parameter space for crystallization conditions (see Brzozowski & Walton, 2001, and references therein). The initial results can provide information for further experiments, but the process is still highly empirical and a time-consuming and monotonous stage of the crystallization process is the repeated inspection of the experiments. The introduction of robots performing many more experiments only exacerbates the situation, particularly in structural genomics centres. Systems for image capture and storage have been developed and automatic classification of the images is becoming increasingly important.

Several research groups have published methods for the automated analysis of crystallization images, applying a variety of feature-extraction methods to obtain meaningful but compact representations for classification. The Hough transform, first used by Zuk & Ward (1991) to identify crystals, has been used by various authors to recognize geometric characteristics. Spraggon *et al.* (2002) obtained variables from straight lines detected with the Hough transform and textural features from correlations between grey levels at various distances and directions. Bern *et al.* (2004) used a curve-tracking algorithm to detect further features, while Cumbaa *et al.* (2003) generated statistical variables using the Radon transform and a Laplacian operator.

Images may also be transformed before quantifiable characteristics are obtained. Wavelet transforms effectively decompose an image into different levels of detail and have been applied extensively in image analysis. Watts *et al.* (2008)

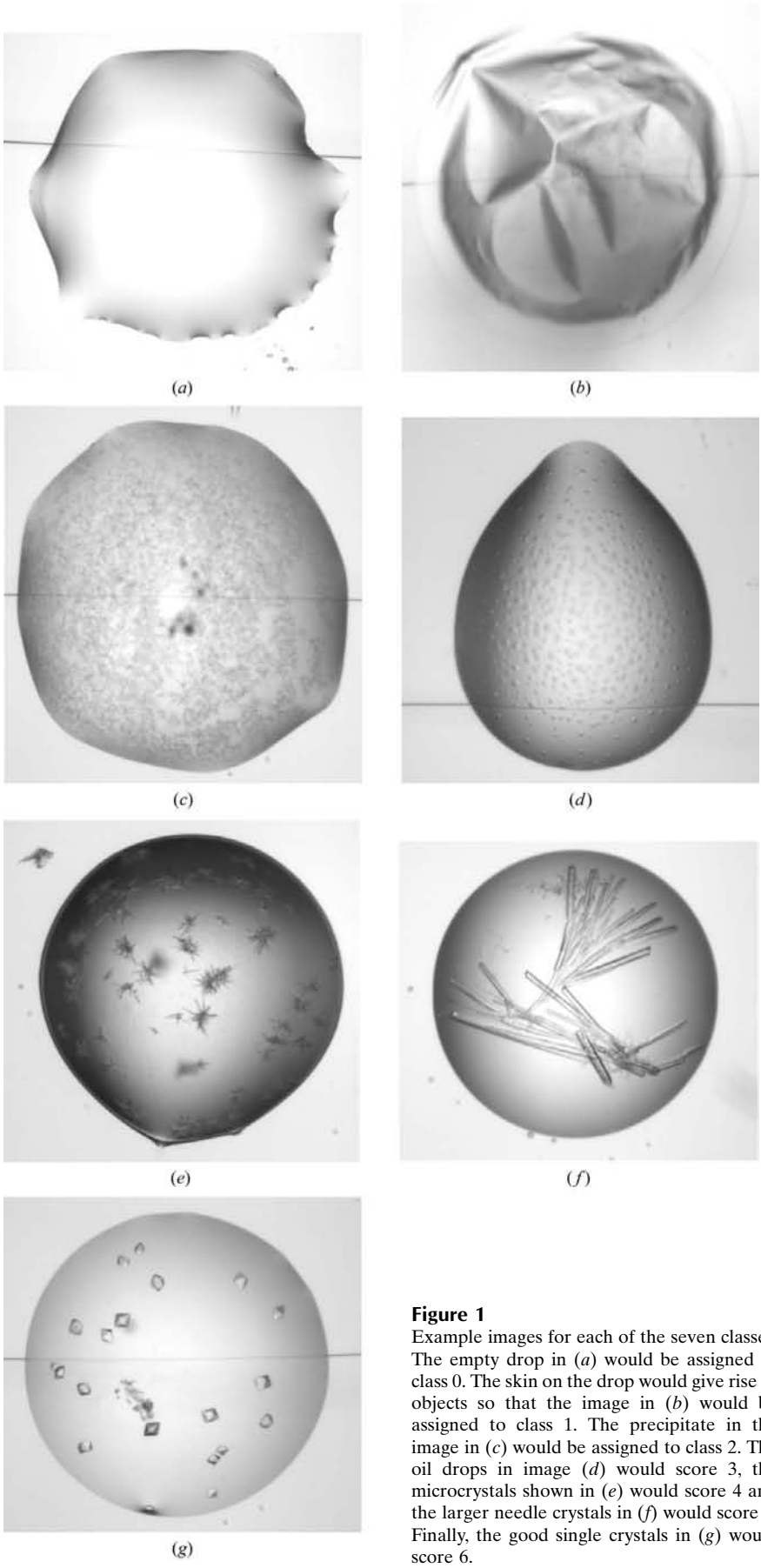

Figure 1
Example images for each of the seven classes. The empty drop in (*a*) would be assigned to class 0. The skin on the drop would give rise to objects so that the image in (*b*) would be assigned to class 1. The precipitate in the image in (*c*) would be assigned to class 2. The oil drops in image (*d*) would score 3, the microcrystals shown in (*e*) would score 4 and the larger needle crystals in (*f*) would score 5. Finally, the good single crystals in (*g*) would score 6.

used statistical measures from different levels of the transform to classify the contents of the crystallization drop as a whole, while Pan *et al.* (2006) used features based on texture and the Gabor wavelet decomposition to classify overlapping sub-images within the crystallization drop. In another approach, a prototype image-acquisition system is presented in Jurisica *et al.* (2001) which uses the Fourier transform. The Fourier transform can identify periodic and directional structure and has been used widely to classify textures. Both Bern *et al.* (2004) and Walker *et al.* (2007) have made use of Fourier analysis in the classification of crystallization images.

However features are extracted, the analysis must be restricted to the crystallization drop, so that all methods require identification of the drop boundary. In Wilson (2002), individual objects within the drop are then located and evaluated separately. In this case features are extracted from each object rather than the crystallization drop as a whole. This spatial domain method has been compared with texture-based methods using both wavelet transforms in Watts *et al.* (2008) and Fourier transforms in Walker *et al.* (2007). It was found that combining complementary methods could improve classification results.

Various statistical classifiers and machine-learning algorithms have been used to classify crystallization images using the extracted features. Self-organizing maps were used by Spraggon *et al.* (2002) and Wilson (2004), while Bern *et al.* (2004) used a decision-tree classifier. Pan *et al.* (2006) used support-vector machines (SVM) for the classification of crystallization images and linear discriminant analysis (LDA) has also been used (Cumbaa *et al.*, 2003; Cumbaa & Jurasica, 2005). Although accuracy rates are given, direct comparison of the results in the separate studies is not possible as different test images were used in each case. Furthermore, the number of classes used differs significantly. However, Kawabata *et al.* (2006) compared the performance of SVM and LDA classifiers on a data set of 300 images. Using the leave-one-out method, they reported 88.7% accuracy by SVM compared with 76.3% accuracy by LDA for a binary

classification of the images. Zhu *et al.* (2004) also compared the results from two different classifiers. Binary classification of 520 images was carried out using an SVM classifier and a C5.0 classifier using tenfold cross-validation and it was found that the C5.0 classifier gave better results. Although internal cross-validation was used during training in these comparisons, the results on an independent test set were not given so it is not possible to assess how much the different classifiers were over-fitting the data.

Here, we provide a systematic evaluation of different classifiers using data from the object-based method (Wilson, 2002) and the wavelet-based method (Watts *et al.*, 2008). Methods to combine the data from these complementary techniques using multiple-classifier systems are investigated.

2. Image data and class system

The Oxford Protein Production Facility (OPPF) at the University of Oxford supplied the images used in this study. Crystallization experiments were performed in 96-well Greiner plates (microtitre format) and the images were taken using an automated Oasis 1700 imaging system (Veeco, Cambridge, England). Native images are 1024 × 1024 × 8 bit bitmap (BMP) images (∼1 Mbyte in size), corresponding to a pixel width of about 3 µm.

The number of classes used to evaluate crystallization experiments varies between authors, but many favour a binary system simply indicating the presence or absence of crystals (Cumbaa *et al.*, 2003; Zhu *et al.*, 2004; Kawabata *et al.*, 2006; Pan *et al.*, 2006). Whilst the identification of crystals is always the primary aim, a two-class system gives no information for subsequent trials. In the absence of crystals, other phenomena can indicate conditions that are close to those required and can be refined in optimization protocols to obtain diffraction-quality crystals (Bergfors, 2002). Comprehensive molecular characterization and prior experience allows a more systematic approach to crystallization and a number of laboratories have set up in-house databases in order to develop crystallization strategies (Hennessy *et al.*, 2000). The collection of information on both successful and failed experiments offers the potential for crystallization parameter prediction using data-mining and machine-learning algorithms (Rupp & Wang, 2004; Cumbaa & Jurisica, 2005). An image-analysis system that can classify different experimental results as well as identifying the presence of crystals will provide valuable information for the development of automated screening procedures.

In the image-analysis system *ALICE* (*AnaLysis of Images from Crystallization Experiments*; Wilson, 2002; Watts *et al.*, 2008) being developed in York, the main aim is to sort the images and drastically reduce the number of images to be inspected by eye. We use a seven-class system to score the images, allowing them to be examined in order of merit. As soon as some high-scoring conditions are confirmed, no further images need be considered. The scores range from 0 for an empty drop to 6 for drops containing good crystals. Example images from each of the seven classes are shown in Fig. 1 and examples of typical experimental results associated with each class are given in Table 1. The table also shows how the scores are pooled to give just three classes for the purposes of reporting the results.

Table 1
Examples of experimental outcomes for each of the seven classes together with the class it relates to in the reduced class system (*i.e.* pooled results).

Class, seven-class system	Result	Class, three-class system
0	Empty drop	0
1	Denatured protein, skin, dirt, foreign bodies such as fibres	
2	Amorphous precipitate	1
3	Oil drops, phase separation, crystalline precipitate	
4	Microcrystals, sea urchins	2
5	Crystal clusters, needles	
6	Single crystals	

The training and test sets consist of images obtained from the Oxford Protein Production Facility. Directories consisting of 96 bitmap images, each corresponding to a crystallization tray reported by crystallographers to contain some favourable experimental results, were copied from the OPPF. The images in each directory were inspected by eye and sorted into the seven classes described above, with the classification of each image agreed on by three individuals. No special requirements were placed on the images and they were simply assigned to classes until the desired number (400) had been reached for any particular class. As soon as a complete set of 400 images was available for some class, further images from that class were ignored. The first 250 images from each of the seven classes were used to train the classification algorithms and the other 150 images from each class formed an independent test set. If the crystallization drop cannot be located (because, for example, the drop edge is indistinct), *ALICE* does no further processing and simply outputs 'No mask found'. This was the case for some images in the training and test sets and explains why the results are reported for <150 images for some classes (see Table 6*a*).

3. Feature extraction

ALICE combines techniques for feature extraction to exploit different sources of information. Wavelet transforms, which effectively decompose an image into different levels of detail, are used to extract features from the crystallization drop as a whole. This is achieved by modelling the distribution of wavelet coefficients in each sub-image. The model parameters together with statistical measures provide variables that can be used for classification. As well as the first-order statistics determined from each detail sub-image, second-order statistics are calculated from joint probability distributions. The decay of the wavelet coefficients across the levels of the transform can be used to characterize different types of edge. Sharp changes such as crystal edges give rise to large wavelet coefficients across all scales, whereas smoother changes in

greyscale arising from shadows, for example, will produce wavelet coefficients that change gradually with subsequent levels of the transform. This information can be extracted by considering the correlation between corresponding wavelet coefficients on different levels of the transform. Full details of the variables calculated from the wavelet-transformed images are given elsewhere (Watts *et al.*, 2008). The feature vectors consisting of these variables calculated over a training set can be used in learning algorithms to associate particular values with an image class.

In a complementary approach, individual objects are identified within the crystallization drop. Objects are defined as connected sets of pixels above a threshold determined by the intensity statistics and each object is evaluated separately. Boundary-related variables include measures of curvature and the length of straight sections. Ordered patterns in the gradient direction anywhere within objects, in straight lines or blocks, indicate the presence of regular objects and various shape descriptors and statistical measures provide information about other types of object (Wilson, 2002). In this case, the feature vectors relate to individual objects and the classification must be converted into an image score. This is achieved by producing a new feature vector in which the variables are the percentage of the total number of objects assigned to each class and the percentage of the total number of pixels in each class. This vector of length 14 is used in a second-level classification to provide a score for the image.

4. Classifiers

Classifiers with inherently different mechanisms for class separation were chosen to fully exploit differences in the data. Different classifier methodologies have different classification rates and the sets of mis-classifications do not necessarily overlap. It has been shown that the combination of several classifiers can provide more robust classification (see, for example, Dietterich, 2000) and that even poor classifiers and poor feature sets can contain information that will improve the performance of classifier ensembles (Duin & Tax, 2000). We found that the best individual classifiers for our data (both object-based and wavelet features) were support-vector machines with both a linear (SVM_linear) and a radial basis-function kernel (SVM_RBF) and that the combination of even just these two related classifiers gave improved results. SVMs are supervised learning methods, that are also known as maximum margin classifiers as they simultaneously maximize the geometric margin between classes whilst minimizing the classification error. Learning-vector quantization (LVQ) and self-organizing maps (SOMs) on the other hand are special cases of artificial neural networks in which the weights of the network are changed gradually in order to classify the training data correctly. Both methods performed well as individual classifiers. The classification rates obtained with linear discriminant analysis (LDA) and naive Bayes individually were lower. We found that while LDA could be successfully combined with other classifiers, the naive Bayes classification actually reduced combined classification rates. Decision trees, such as the C5.0 classifier, that use if–then rules to separate the classes were found to expand too much during training so as to be impractical for prediction. Restrictions on tree size gave rules that could be used for class prediction, but the results were poor.

These classifiers are all supervised learning techniques and must be trained on a set of input feature vectors $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$ of known class c_i for $i = 1, \ldots, C$ say. Supervised classifiers learn to predict the output class for new input vectors having seen the N training examples. The distinction is in the way the different classifiers create rules to associate the input training data with the output class label. Brief descriptions of the classification mechanisms for the best individual classifiers are given in the following sections and the different separation boundaries obtained are illustrated in Fig. 2. Whilst LDA and SVMs with a linear kernel obviously both have linear separation boundaries, SOMs, LVQs and SVMs with a radial basis kernel all allow the separation boundary to be nonlinear.

4.1. Support-vector machines

SVM classifiers were originally developed to differentiate between just two classes, $c_i \in \{-1, +1\}$, $i = 1, 2$. They do this by finding the optimal separation hyperplane, *i.e.* the hyperplane with maximal margin of separation between classes and minimum classification errors. For linearly separable classes the hyperplane is calculated in the original input space, but nonlinearly separable classes can be dealt with by applying a nonlinear transformation of the input space to a higher dimensional feature space in which the classes are linearly separable. As the nonlinear mappings or kernels allow computations to be performed in the input space, SVMs are not computationally expensive.

In the linearly separable case, a hyperplane is defined by

$$H = \sum_{i=1}^{k} \omega_i \mathbf{x}_i + b, \tag{1}$$

where k is the length of the feature vector and the parameters b and $\boldsymbol{\omega}_i = (\omega_{i1}, \omega_{i2}, \ldots, \omega_{ik})$ for $i = 1, \ldots, C$ (the number of classes) are determined during training so that the decision boundary is given by $H = 0$. Thus, an input vector $\mathbf{x}$ is assigned to class $c_1 = -1$ if $H < 0$ and to $c_2 = +1$ if $H > 0$. In order to generalize to unseen data, the distance between the training samples and the decision boundary should be maximized. The training samples closest to the decision boundary are called support vectors and the margin is defined as the width that the boundary could be increased by before touching a support vector. The linear SVM classifier finds the decision boundary that maximizes this margin (Cortes & Vapnik, 1995). It can be shown that SVM learning involves finding the multipliers α_i that maximize the Lagrangian

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j c_i c_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \alpha_i \geq 0 \text{ and } \sum_{i=0}^{N} \alpha_i c_i = 0 \tag{2}$$

for classes c_i, $i = 1, \ldots, N$ and input feature vectors $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$, where $\mathbf{x}_i^T$ denotes the transpose of $\mathbf{x}_i$.

Nonlinear SVM classifiers can be created by replacing the dot products in (1) with a nonlinear kernel function. The kernel function implicitly maps the example data points into a higher dimensional feature space and takes the inner product in that feature space. This allows the maximum-margin hyperplane to be fitted in the transformed feature space and can be achieved by applying different nonlinear mappings such as polynomial, radial basis, sigmoidal or spline functions. Although the classifier is a hyperplane in the high-dimensional space it can be non-linear in the original input space.

Separation of the feature vectors by a hyperplane works when there are only two classes. Several approaches have been suggested to deal with more than two classes, the most popular being 'one against many', in which each class is separated in turn from all other merged classes, and 'one against one', requiring $C(C-1)/2$ models where C is the number of classes.

SVMs have been shown to give comparable or better results than neural networks and other statistical models on many problems in computer vision, pattern recognition and data mining (Meyer *et al.*, 2003).

4.2. Learning-vector quantization and self-organizing maps

When used for classification, learning-vector quantization (LVQ) aims to represent the feature vectors in the training set by a smaller number of prototype vectors, each with an associated class. These prototype vectors, also known as codebook vectors, have a higher between-class variation and lower within-class variation than the original input data. LVQs apply naturally to multi-class systems with new feature vectors classified according to the class of the closest (in terms of Euclidean distance) codebook vector.

LVQ is an artificial neural network with a layer of input neurons and a layer of output neurons. Unlike other neural networks, the weights can be readily interpreted as prototype vectors $\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_{ik})$ representing typical data in the same input space. In general, several prototypes will be used to represent each class. The weights of the network are changed in order to classify the training data correctly in a competitive learning approach. In each cycle of an iterative training procedure, each vector in the training set is presented to the network and the winning codebook vector identified as having weights closest to the input vector. The weights of the winning vector are then updated, as shown in (3), depending on whether the assigned class c in the output layer is same as the actual class of the input vector $\mathbf{x}_i$, *i.e.*

$$w_j(\text{new}) = \begin{cases} w_j(\text{old}) + \alpha[x_{ij} - w_j(\text{old})] & \text{if } c = c_i \\ w_j(\text{old}) - \alpha[x_{ij} - w_j(\text{old})] & \text{if } c \neq c_i \end{cases}, \qquad (3)$$

where the learning rate α is gradually decreased over a number of cycles.

The original LVQ algorithm was the precursor of self-organizing maps (SOMs; Kohonen, 1987). The aim of the SOM is to cluster the prototype vectors to produce a two-dimensional map that preserves the topology of the original data. During training, physically close neurons, or nodes in the map, learn to recognize similar input patterns. This is achieved by updating the weights not only of the winning neuron but also those within a specified neighbourhood. SOMs are commonly used for unsupervised learning, *i.e.* without using class information, to identify patterns in the data. However, they can also be used for classification by assigning the class of

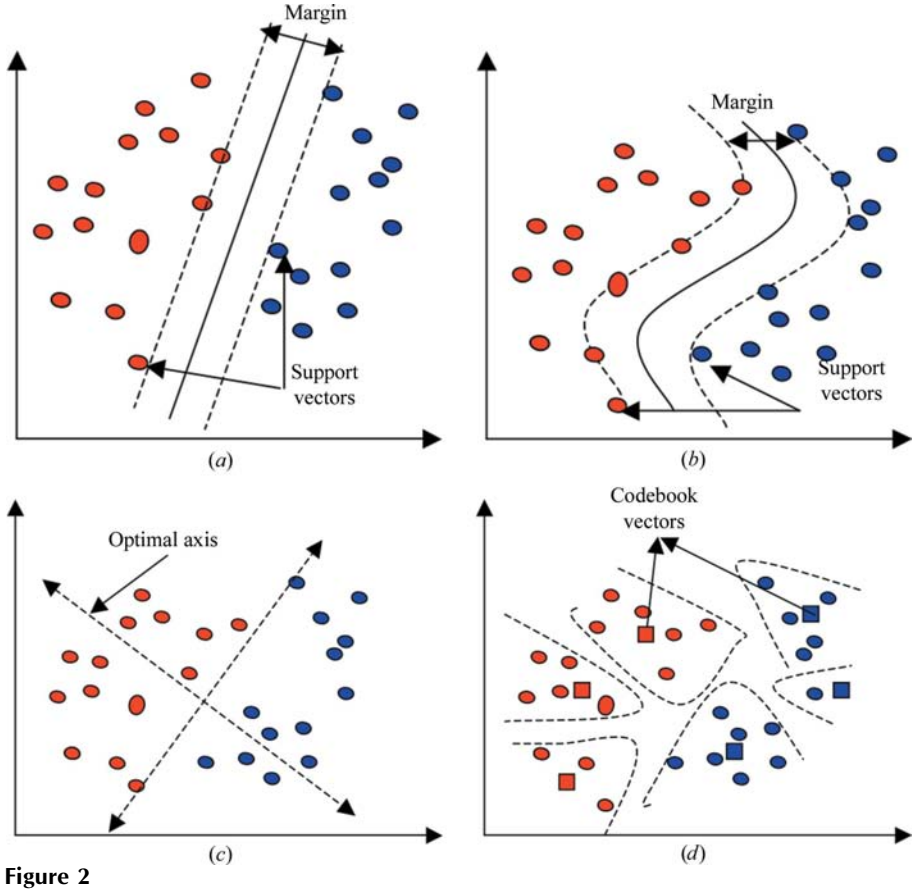

Figure 2
The dotted lines in (*a*) and (*b*) pass through the support vectors defining the separation hyperplane (solid line) for the linear and nonlinear cases, respectively. Support-vector machines (SVMs) find support vectors which define the hyperplane that maximizes the margin of separation (the distance between the dotted lines) whilst minimizing classification errors. The optimal axes of separation onto which the data are projected using linear discriminant analysis (LDA) are illustrated in (*c*). The resultant data have better between-class separation and lower within-class spread. Codebook vectors used to represent subclasses of the input data using learning-vector quantization (LVQ) are illustrated in (*d*), showing how samples are assigned to the nearest subclass before being given the class that the subclass belongs to.

the closest training vector to each node after the final map has been created.

4.3. Linear discriminant analysis

The objective of linear discriminant analysis (LDA) is to find the linear combination of feature variables that maximizes the difference between classes. A measure of class separation is given by Fisher's criterion, the ratio of between-class variance to within-class variance,

$$J = \frac{\mathbf{S}_\mathrm{B}}{\mathbf{S}_\mathrm{W}} = \mathbf{S}_\mathrm{w}^{-1}\mathbf{S}_\mathrm{B}, \tag{4}$$

where $\mathbf{S}_\mathrm{B}$ is the between-class covariance matrix given by

$$\mathbf{S}_\mathrm{B} = \frac{1}{(C-1)}\sum_{j=1}^{C} n_j(\overline{x}_j - \overline{x})(\overline{x}_j - \overline{x})^T \tag{5}$$

and $\mathbf{S}_\mathrm{W}$ is the within-class covariance matrix given by

$$\mathbf{S}_\mathrm{W} = \frac{1}{(N-C)}\sum_{j=1}^{C}\sum_{x\in c_j}(\overline{x}_j - \overline{x})(\overline{x}_j - \overline{x})^T, \tag{6}$$

where $\overline{x}_j$ and $\overline{x}$ are the mean of the jth class and the global mean, respectively, C is the number of classes, c_j the jth class and n_j is the number of samples in class j.

LDA applies a linear transformation, D, to the data in order to maximize the separation between classes. The optimal transformation, D^OPT, can be obtained by solving the following optimization problem (Duda *et al.*, 2000),

$$D^\mathrm{OPT} = \arg\max_D\left[\mathrm{trace}\left(\frac{D^T S_\mathrm{B} D}{D^T S_\mathrm{W} D}\right)\right], \tag{7}$$

where $\arg\max_x f(x)$ means the value of x for which $f(x)$ has the maximum value and trace is the sum of the elements on the main diagonal.

LDA is used extensively for classification and has been successful in many applications.

5. Combining classifiers

Previous studies have shown that combining classifiers can improve overall classification performance (Kittler *et al.*, 1998; Al-Ani & Deriche, 2002; Lu *et al.*, 2003). Methods have included the combination of different classifiers on the same feature set and the use of different feature sets, and it has been shown that the best performance is achieved by combining both different feature sets and different classifiers (Duin & Tax, 2000).

A classifier combination is useful if the individual classifiers are largely independent and utilize different methodologies to take advantage of the data characteristics. A linear classifier, such as LDA for example, exploits different features in a data set to a nonlinear neural network classifier. Classification independency can also be achieved by using different training sets. Re-sampling techniques such as bootstrapping and rotation can be used to artificially create multiple training sets from the same data. Different features in the images can also be exploited to provide multiple training sets. For example, Jain *et al.* (2000) used Fourier transforms and principal component analysis among other techniques to extract different features from the same image set. We found that the use of artificially created training sets from the same data gave little improvement in the results but that the combination of the object-based method with a texture-based approach using Fourier- or wavelet-transformed images significantly increased correct classification rates (Walker *et al.*, 2007; Watts *et al.*, 2008).

Several methods for combining individual classifier decisions have been proposed. Here, we consider methods that do not require further training. These include simple averaging over the class outputs and the use of order statistics such as the minimum, maximum or median of the class outputs. Majority voting assigns the class that is selected by the majority of classifiers and the probability sum rule involves addition of the probabilities generated by different classifiers for each class to give a final class with the highest probability.

An added complication is the fact that the classified objects must be combined in some way in order to give an image score. The results of the object classification can be used to provide variables in a second-level classification. This offers the possibility of using multiple classifiers both to categorize individual objects and then to classify the image. However, the high speed at which images are acquired places considerable constraints on the image processing and classification success has to be balanced against computational efficiency. This was also the reason that the Fourier-based method was not included in the classification scheme. Whilst using multiple classifiers does greatly improve the results for individual object classification, a second-level multiple classifier system to convert the object scores into an image score did not give significantly better classification rates than a single classifier. We therefore chose the best single classifier, an SVM classifier with an RBF kernel, for this stage. The flowchart in Fig. 3 shows the image-classification system used. Multiple classifiers are used to determine an image class from the statistical variables obtained after wavelet analysis as well as for the classification of individual objects. However, a single classifier is used to provide the image score from the object classification before all image scores are combined to give the final class for the image.

6. Results

A full confusion matrix comparing the true image class with the predicted class has 49 entries for a seven-class system, making comparisons difficult. In order to compare classification performances, we merged the results to give just three classes. However, it should be emphasized that the results were obtained using a seven-class system, which we believe gives greater sorting ability. In addition, we reduced the confusion matrices to a single number for easier comparison of the different classification methods. Table 2(*a*) shows a confusion matrix for the classification of crystallization images by different crystallographers. The rows correspond to the mean image scores and the columns to the classes chosen by

the crystallographers, so that the diagonal entries give the percentage of exact matches. The further away from the diagonal the greater the disagreement, with images classified lower than the mean score above the diagonal and images classified higher below the diagonal. The results are for a different set of images to those used for either training or testing *ALICE*. It can be seen that there are different numbers of images in each class, with empty drops (class 0) and those containing precipitate (class 2) considerably outnumbering other classes. This set of images consists of ten crystallization plates imaged consecutively at the OPPF in Oxford with a further ∼300 images added to increase the number of more interesting outcomes. The fact that the classes are so unbalanced reflects the real situation but makes the image set unsuitable for training. Each row in the table corresponds to the 'true' class, given here by the mean of the image scores over 16 crystallographers. The columns correspond to the 'predicted' class, *i.e.* the class chosen by the different crystallographers. It can be seen that class 6 (single crystal) images have high agreement rates as might be expected, with 84.7% total agreement and another 13.7% being classified as class 5 (crystal cluster). Unsurprisingly, class 0 (empty drops) also cause little difference of opinion, with 91.6% exact agreement. Other classes show more variation, with images classified by the mean image score as 3, 4 and 5 being assigned to every class possible. However, the table shows that most mis-classifications or, more accurately, 'differently classified' images are assigned to an adjacent class. This is not so apparent when the classes are merged to give just three classes as in Table 2(*b*) and it is clear that reducing the number of classes can only be detrimental to classification.

Whilst images assigned to a neighbouring class need not be considered incorrect (as two different crystallographers may disagree on the 'correct' class), crystals classed as an empty drop, for example, most certainly are and should be penalized far more. The overall classification rate (CR), or percentage of exact classifications, does not allow for this and therefore gives little information about the classification. The classification rate for the data in Table 2(*a*), *i.e.* the agreement rate between crystallographers, is only 62.9%, which does not reflect the distribution in the table. We therefore define a continuous classification rate (CCR), which take into account how bad any mis-classifications are.

The CCR is calculated from the confusion matrix as

$$Z = \frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{n}\sum_{j=0}^{N-1} w_{ij}C_{i,j}, \tag{8}$$

where

$$n = \max_{k\in(0,N-1)} |i-k|$$

and $w_{ij} = (n - |i - j|)$. Here, N is the total number of classes and $C_{i,j}$ is the percentage of class i images classified as class j.

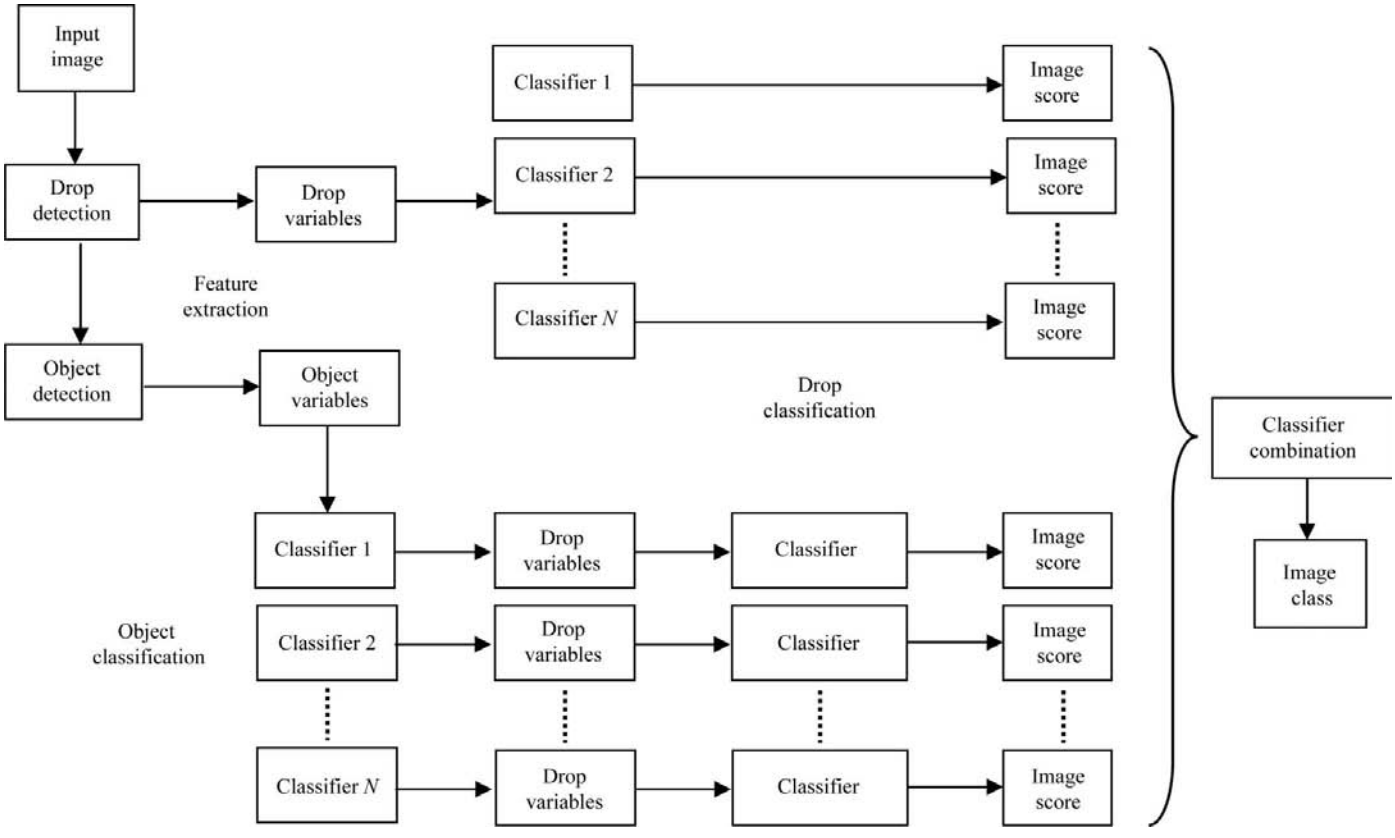

Figure 3
Flowchart of image-classification system.

Table 2
Agreement rates between crystallographers on the class of an image.

The rows correspond to the mean image scores and the columns to the classes chosen by the 16 crystallographers. The results are given as percentages of the total number of images in each class (according to the mean score), with the entries on the main diagonal showing the percentage of exact matches with the mean score. The full confusion matrix is shown in (*a*) and the reduced class system in (*b*). Although the results shown relate to a different set of images to that used for testing *ALICE*, the overlap between classes is demonstrated and must be taken into account when assessing the accuracy of automated classification.

(*a*) Full confusion matrix.

		Predicted class						
Mean class	No. of images	6	5	4	3	2	1	0
6	43	84.7	13.7	1.2	0.0	0.0	0.4	0.0
5	109	11.2	68.5	17.7	1.5	0.1	0.3	0.4
4	71	3.3	29.0	50.4	12.6	1.8	1.3	1.1
3	99	0.6	2.0	18.9	46.7	23.5	6.5	1.4
2	428	0.0	0.1	1.9	20.6	59.3	14.4	3.5
1	185	0.1	0.0	0.8	8.5	18.1	39.5	32.5
0	358	0.0	0.0	0.1	0.9	1.8	5.4	91.6

(*b*) Reduced class system.

		Predicted class		
Real class	No. of images	2	1	0
2	223	93.2	5.3	1.2
1	527	11.8	75.1	12.9
0	543	0.5	14.7	84.5

The CCR for the data in Table 2(*a*) is 89.0, which gives a better indication of the actual results.

Training data sets obtained from individual objects within the crystallization drop (object data) and from statistical analysis of wavelet-transformed images (wavelet data) were used to train five different classifiers: learning-vector quantization (LVQ), a self-organizing map (SOM), linear discriminant analysis (LDA) and support-vector machines using both a linear kernel (SVM_linear) and radial basis functions (SVM_RBF). For each classifier, the object-classification results were used to provide 14 variables: the number of pixels in each object class as a percentage of the total number of pixels in the drop and the number of objects in each class as a percentage of the total number of objects. These variables were used in a second-level classification using an SVM_RBF classifier to provide an image score. Classification rates for the individual classifiers are given in Table 3 and, for the object data, show the results after the object scores have been combined to give the image class. Although the CCRs for some of the classifiers look very similar, the distribution of classifications can be quite different. For example, the CCRs for object-based classification using SOM and linear SVM classifiers are 80.1 and 80.5, respectively, but the confusion tables show the differences in the classification (see the reduced classification tables in Table 4). This independence between classifiers allows combination schemes to improve the results.

The classifier combination schemes tested used majority voting, the sum rule and taking the maximum, median and mean output class. Majority voting assigns the class that is selected by the majority of classifiers. In the event of a tie, we took the maximum class to minimize the probability of missing crystals. As long as there are not too many, false positives are not as serious as false negatives, but we found that simply taking the maximum class from all classifiers did cause a lot of empty drops to be classified as crystals. Although the crystals in the image shown in Fig. 4(*b*) are picked up as objects and classified correctly, the statistical analysis of the wavelet data is based on the whole drop, which is mostly empty. This not only leads to such images being classified incorrectly as empty drops using texture-based methods, but also associates empty drop-like variables with crystals during training. Using the maximum class allows the false positives this creates to dictate the final class. This can be avoided by using the mean or median class. The median is less sensitive to outliers, which explains why better results were obtained using the median class as the final output. The sum rule is more complicated and involves the generation of class probabilities from each classifier rather than a single output class. These probabilities are then summed and the class with the highest probability sum chosen. As some classifiers naturally output a single predicted class rather than a probability for each class, the probability sum method can be more difficult and crucially more computationally expensive to implement. Therefore, as the results for the first scheme were so much worse than the median or majority-voting method, it was not implemented for other schemes.

Table 3
Classification rates for individual classifiers for both the object data set and the wavelet data set.

After classification of objects within an image, the results from each classifier were combined to give an image score using an SVM_RBF classifier. Thus, the classification rates (CR) and continuous classification rates (CCR) in the table relate to the classification of images rather than individual objects. Therefore, all rates can be compared directly.

	CR (%)	CCR
Objects, LVQ	58.0	81.4
Objects, LDA	35.3	71.2
Objects, SOM	54.3	80.1
Objects, SVM_linear	57.1	80.5
Objects, SVM_RBF	57.9	82.4
Wavelets, LVQ	51.6	79.1
Wavelets, SOM	48.9	77.8
Wavelets, SVM_linear	52.5	80.3
Wavelets, SVM_RBF	51.5	80.0

The five combination schemes were tested using all possible subsets of the potential classifiers. It was found that most classifiers added to the classification ability, although LDA actually reduced classification rates. As LDA produces so many false positives (see Table 4*c*), this classifier is most unfavourable when the maximum class is used but in any case does not improve the results. Another classifier that does not perform particularly well individually is the SOM with wavelet data (CCR 77.8; see Table 4*d*). However, comparison of scheme 1 with scheme 4 in Table 5 shows that some improvement is achieved by including this classifier.

Table 4
Reduced classification tables for individual classifiers.

(*a*) The results of classification using the object data set (after combination to provide an image score) for the SOM.

		Predicted class		
Real class	No. of images	2	1	0
2	449	70.8	19.0	10.2
1	299	12.0	82.0	6.0
0	294	8.1	12.5	79.5

(*b*) The results of classification using the object data set (after combination to provide an image score) for the linear SVM.

		Predicted class		
Real class	No. of images	2	1	0
2	449	78.1	12.7	9.1
1	299	13.0	77.3	9.7
0	294	12.2	4.4	83.4

(*c*) The results of classification using the object data set (after combination to provide an image score) for LDA.

		Predicted class		
Real class	No. of images	2	1	0
2	449	55.9	33.9	10.2
1	299	36.0	55.0	9.0
0	294	17.0	15.3	67.7

(*d*) The results of SOM classification using the wavelet data set.

		Predicted class		
Real class	No. of images	2	1	0
2	449	72.3	18.7	8.9
1	299	12.4	72.2	15.4
0	294	13.9	16.9	69.2

Table 5
Comparison of combination schemes.

Scheme 1 combines the four classifiers SOM, LVQ, SVM_RBF and SVM_linear on both object data and wavelet data, giving eight classifiers in total. Scheme 2 is the same as scheme 1 with the addition of LDA on the object data. Scheme 3 is the same as scheme 2 but without the SOM classifier on the wavelet data. Scheme 4 is the same as scheme 1 but without the SOM classifier on the wavelet data. The probability sum method was difficult.

Combination method	Scheme 1		Scheme 2		Scheme 3		Scheme 4	
	CR (%)	CCR	CR (%)	CCR	CR (%)	CCR	CR (%)	CCR
Median	60.3	87.0	60.3	85.4	60.3	85.4	59.0	84.7
Mean	50.7	84.7	48.4	84.2	48.4	84.2	51.5	84.6
Majority vote	61.8	84.3	61.5	84.0	61.5	84.0	60.5	83.6
Probability sum	49.4	78.8	—	—	—	—	—	—
Maximum	44.5	74.0	35.4	68.1	36.9	69.1	46.6	75.2

Table 6
Classification rates obtained using the median class from eight classifiers: SOM, LVQ, SVM_RBF and SVM_linear with each of the object and wavelet data sets.

The rows correspond to the image scores assigned visually and the columns to the classes predicted so that the diagonal entries show exact matches. The results are given as percentages of the total number of images in each class.

(*a*) Full confusion matrix.

		Predicted class						
Real class	No. of images	6	5	4	3	2	1	0
6	150	56.0	18.7	12.7	8.0	0.7	1.3	2.7
5	150	27.3	31.3	34.7	4.7	0.7	0.0	1.3
4	149	14.8	22.8	32.9	19.5	6.0	0.7	3.4
3	150	4.0	4.0	10.0	50.7	23.3	4.7	3.3
2	148	0.0	0.0	0.0	4.7	94.6	0.7	0.0
1	148	3.4	4.1	7.4	8.1	10.8	64.9	1.4
0	146	1.4	1.4	0.0	1.4	0.0	2.7	93.2

(*b*) Reduced class system.

		Predicted class		
Real class	No. of images	2	1	0
2	449	83.7	13.2	3.1
1	299	9.0	86.7	4.3
0	294	8.8	10.1	81.1

The combination method using the median class consistently performs better than any other classification scheme, with the best results obtained using four classifiers, SOM, LVQ, SVM_RBF and SVM_linear, on both object data and wavelet data, so that eight classifiers are combined in total. The full confusion matrix and reduced table for this classification scheme are given in Table 6 for comparison with Table 2. However, it should be pointed out that the results are obtained from different data sets, as the unequal classes in the data set used to obtain the results in Table 2 make it unsuitable for either training or testing classification techniques.

Fig. 4(*a*) shows an image that was mis-classified as an empty drop by all classifiers using the wavelet data. Although the drop contains crystals, most of the drop is clear, causing problems when using statistical measures. However, the image was classified correctly using the object-based method. In contrast, the image in Fig. 4(*b*) has crystals throughout the drop and was classified correctly by all classifiers with both data sets. The image in Fig. 4(*c*) was assigned to various classes (ranging from 1 to 5) using object data but was classified correctly by all classifiers using the wavelet data. Thus, the two methods are complementary and, as these examples show, classification rates can be improved by their combination. Furthermore, the use of several classifiers can provide a more robust classification.

7. Conclusions

When comparing the results from different classifiers and different classifier combination schemes, it is important to consider the natural overlap between classes. This reflects the continuous nature of the outcomes of crystallization experiments and will be the case however many discrete classes are used. The lack of agreement between crystallographers on the 'true' class of an image shows that mis-classification into adjacent classes need not necessarily be considered incorrect. We have defined a continuous classification rate, CCR, which provides a number (rather than a percentage) between 0 and

100 that takes into account how far from the real class the predicted class is and therefore gives a better representation of the classification results.

We found that the single best classifier for object classification and for the combination of object scores into an image score was the SVM classifier with an RBF (*i.e.* nonlinear) kernel. This classifier was also very successful with statistical data extracted from the drop as a whole, although linear classification (again using SVM) actually performed slightly better in this case. We found that even classifiers that did not give particularly good results individually, such as the SOM, could improve the results in a classifier ensemble. The combination of different classifiers gave significantly better results for both object data and drop data, but the use of multiple classifiers to provide an image score from individual object scores was not deemed worthwhile.

The best overall results were obtained by combining the two feature sets (object data and drop data) using four classifiers: two SVM classifiers, one with a linear kernel and one with a nonlinear kernel, and two neural networks, learning-vector quantization and a self-organizing map. Taking the median of all eight classifier outputs gave the optimal combination scheme, although majority voting gave very similar results.

Funding for SB was provided by BIOXHIT [Biocrystallography (X) on a Highly Integrated Technology Platform for European Structural Genomics] under the 6th Framework Programme of the European Commission (LSHG-CT-2003-503420).

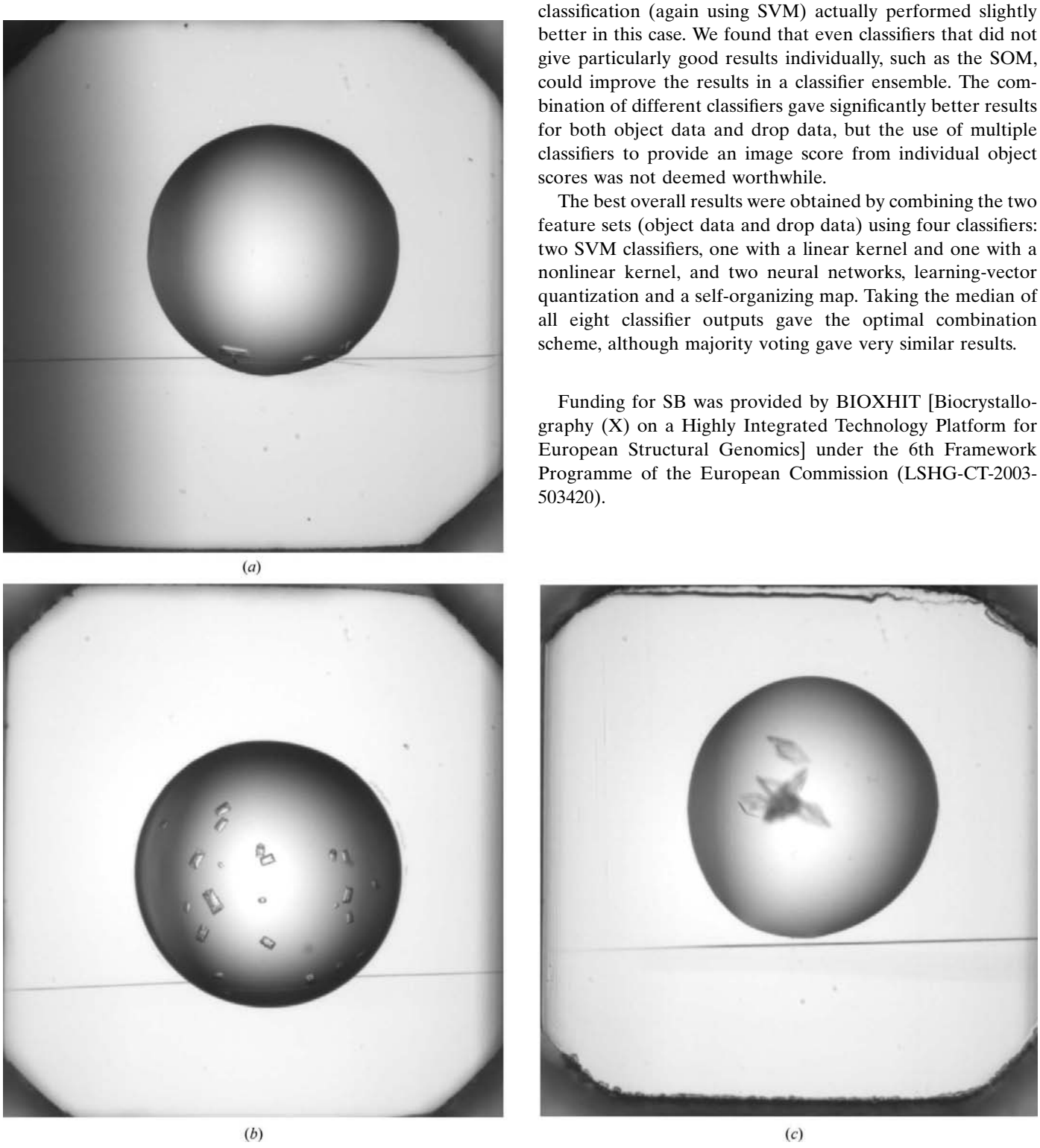

Figure 4
The image in (*a*) was mis-classified as an empty drop by all classifiers using the wavelet data but was classified correctly using object data, whereas the image in (*b*) was classified correctly as containing crystals by all classifiers with both data sets. The image in (*c*) was assigned to various classes (ranging from 1 to 5) using object data, but was classified correctly by all classifiers using the wavelet data.

References

Al-Ani, A. & Deriche, M. (2002). *J. Artif. Intell.* **17**, 333–361.

Bergfors, T. (2002). Editor. *Protein Crystallization: Techniques, Strategies and Tips. A Laboratory Manual.* La Jolla: International University Line.

Bern, M., Goldberg, D., Stevens, R. C. & Kuhn, P. (2004). *J. Appl. Cryst.* **37**, 279–287.

Brzozowski, A. M. & Walton, J. (2001). *J. Appl. Cryst.* **34**, 97–101.

Cortes, C. & Vapnik, V. (1995). *Mach. Learn.* **20**, 273–297.

Cumbaa, C. & Jurisica, I. (2005). *J. Struct. Funct. Genomics*, **6**, 195–202.

Cumbaa, C. A., Lauricella, A., Fehrman, N., Veatch, C., Collins, R., Luft, J. R., DeTitta, G. & Jurisica, I. (2003). *Acta Cryst.* D**59**, 1619–1627.

Dietterich, T. G. (2000). *Proceedings of the First International Workshop on Multiple Classifier Systems*, edited by J. Kittler & F. Roli, pp. 1–15. London: Springer-Verlag.

Duda, R., Hart, P. & Stork, D. (2000). *Pattern Classification.* New York: Wiley.

Duin, P. W. R. & Tax, D. M. J. (2000). *Proceedings of the First International Workshop on Multiple Classifier Systems*, edited by J. Kittler & F. Roli, pp. 16–29. London: Springer-Verlag.

Hennessy, D., Buchanan, B., Subramanian, D., Wilkosz, P. A. & Rosenberg, J. M. (2000). *Acta Cryst.* D**56**, 817–827.

Jain, A. K., Duin, P. W. R. & Mao, J. (2000). *IEEE Trans. Patt. Anal. Mach. Intell.* **22**, 4–37.

Jurisica, I., Rogers, P., Glasgow, J. I., Fortier, S., Luft, J. R., Wolfley, J. R., Bianca, M. A., Weeks, D. R. & DeTitta, G. T. (2001). *IBM Syst. J.* **40**, 248–264.

Kawabata, K., Takahashi, M., Saitoh, K., Asama, H., Mishima, T., Sugahara, M. & Miyano, M. (2006). *Acta Cryst.* D**62**, 239–245.

Kittler, J., Hatef, M., Duin, R. P. W. & Matas, J. (1998). *IEEE Trans. Patt. Anal. Mach. Intell.* **20**, 226–239.

Kohonen, T. (1987). *Self-Organization and Associative Memory.* Berlin: Springer.

Lu, X., Wang, Y. & Jain, A. K. (2003). *2003 International Conference on Multimedia and Expo*, Vol. 3, pp. 13–16. Washington DC: IEEE Computer Society.

Meyer, D., Leisch, F. & Hornik, K. (2003). *Neurocomputing*, **55**, 169–186.

Neal, B. L., Asthagiri, D. & Lenhoff, A. M. (1998). *Biophys. J.* **75**, 2469–2477.

Pan, S., Shavit, G., Penas-Centeno, M., Xu, D.-H., Shapiro, L., Ladner, R., Riskin, E., Hol, W. & Meldrum, D. (2006). *Acta Cryst.* D**62**, 271–279.

Rupp, B. & Wang, J. (2004). *Methods*, **34**, 390–407.

Spraggon, G., Lesley, S. A., Kreusch, A. & Priestle, J. P. (2002). *Acta Cryst.* D**58**, 1915–1923.

Walker, C. G., Foadi, J. & Wilson, J. (2007). *J. Appl. Cryst.* **40**, 418–426.

Watts, D., Cowtan, K. & Wilson, J. (2008). *J. Appl. Cryst.* **41**, 8–17.

Wilson, J. (2002). *Acta Cryst.* D**58**, 1907–1914.

Wilson, J. (2004). *Crystallogr. Rev.* **10**, 73–84.

Zhu, X., Sun, S., Cheng, S. E. & Bern, M. (2004). *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, Vol. 3, pp. 1628–1631. Piscataway: IEEE.

Zuk, W. M. & Ward, K. B. (1991). *J. Cryst. Growth*, **110**, 148–155.