Final DMP Report

Mayra Vavrica
San Diego State University
vavrica@rohan.sdsu.edu

August 18, 2006

Abstract

Crystallographers synthesize proteins to be used in pharmacological processes. Synthesizing crystals requires experiments in which different precipitates are mixed and stored under varying temperature and pressure conditions. Robot microscopes are used to take pictures of the completed experiment. Using an existing automated protein crystallization classifier as benchmark, we explored the effects of extracting different texture features and using a different classifying algorithm. In section 2 we give a brief overview of the Pan et al method for feature extraction. In section 3 we discuss the features we extracted. Section 4 talks about the classifiers used, and section 5 explores our results.

1 Introduction

Viruses such as Malaria are like keys that fit into our body's key holes. If the keyholes are blocked, then Malaria will not be successful in attaching to vital systems. In a sense, protein crystallography aim at artificially creating key-hole plugs. The idea is that if a crystal's structure is a better fit for the receptor keyhole in our body than Malaria, it will effectively block Malaria from reaching its intended destination. This process is not just used for Malaria, but in many other pharmacological processes that save many lives.

Protein crystallization is the process of growing crystals by mixing precipitate at various temperatures, pressures, and reaction times. At the end of the reaction time a robot microscope takes a digital picture of the result obtained from that experiment. The digital pictures must then be classified as either crystal or not crystal [8]. As we can see from figures 1,2, and 3, even expert human eyes have can have a hard time identifying crystals from no-crystals. Figure one is a very good batch of micro crystals ready to be harvested and taken to a laser line for structure identification. Figure 2 shows what looks like microcrystals but in fact are impurities, bubbles and precipitate. Figure 3 shows an image of crystals mixed within nebulous precipitate residue.

Pan et al [11] discovered that classifying crystals in 'textured' areas is the biggest challenge that a classification system much surmount. We explore extracting textured features based on the co-occurrence matrix which have been found to outperform Gabor features in recognizing objects in 'noisy' (textured) backgrounds [12]. Once we had developed our features, we found a way to differentiate 'texture' from 'non-textured' areas with 96% accuracy.

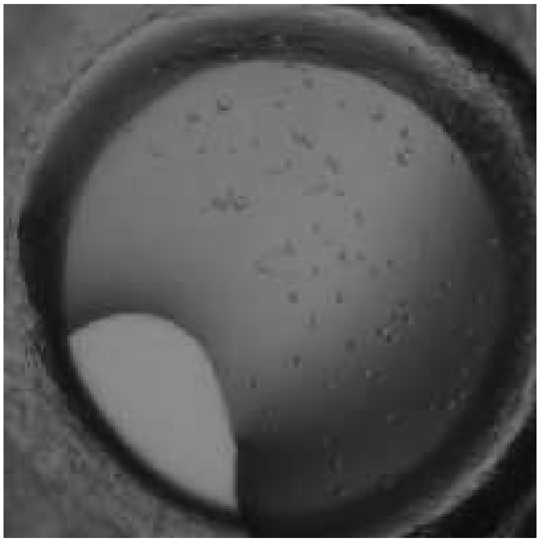

Figure 1: Crystals in 'untextured' background–precipitate.

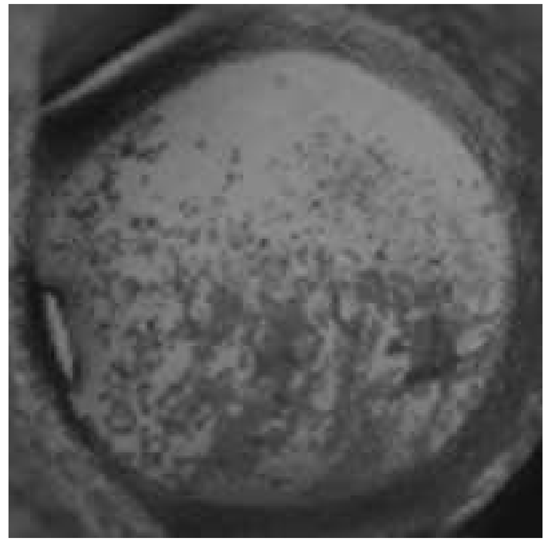

Figure 2: Texture–precipitate, bubbles, inpurities, only. No Crystals.

2 Background

Previous work carried out by Pan et al [11] extracted Blobworld texture features combined with Gabor wavelet decomposition results in a Support Vector Machine (SVM) classifier.

2.1 Blobworld Texture Features

Blobworld is a system for image retrieval based on finding coherent image regions which roughly correspond to objects. Each image is segmented into regions called "blobs" with associated texture descriptions [4].

Because computing texture features at the wrong scale would lead to confusion, Edge Polarity was chosen as one of the Blob World texture measures. Polarity is a measure of the extent to which the gradient vectors in a certain neighborhood all point in the same direction. This has a stabilizing effect since the polarity is calculated at every pixel of the image at various scales, later smoothed with a Gaussian filter [7].

Anisotropy was chosen as a texture measure because it would reflect different properties according to the direction of measurement, and would hence, make it sensitive to both textured and untextured areas in the image blocks [7].

The last feature chosen was the Texture Contrast which defines the homogeneity of the pixels within the local region. There should be Crystal high Contrast insidide crystal regions, signaling a homogenous area.

2.2 Gabor Wavelet Decomposition

Gabor wavelets are complex exponential signals modulated by Gaussians. Two properties were taken into account when choosing Gabor

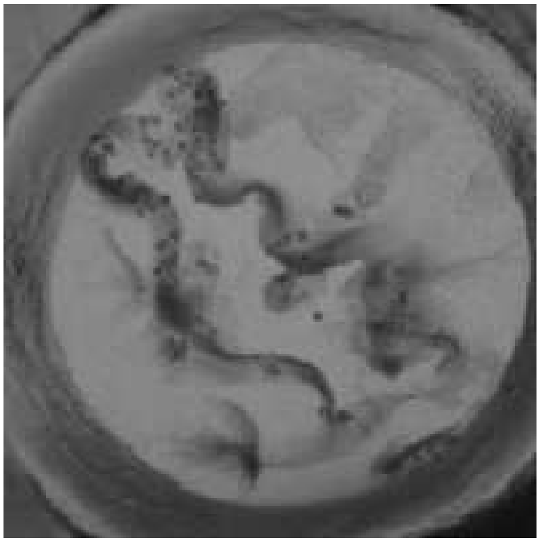

Figure 3: Crystals in a 'textured' background.

wavelets: good edge localization and the absence of image dependent parameter tuning [6]. The only Gabor feature used is the *response*, computed by assigning each pixel (x, y) an 8-component vector $H(x, y)$ *response* as in equation 1.

$$response(x, y) = 0.5 + \frac{arctan(\frac{(|H(x,y)| - |\overline{H}|)}{|\tilde{H}|})}{\pi} \tag{1}$$

2.3 Support Vector Machines

Support Vector Machines (SVM) are binary classifiers. Given a training data set of n subjects, an SVM aims at finding a maximum-margin hyperplane or optimal hyperplane, obtained by mapping a feature vector into a very high-dimensional space and cutting it in half[3]. Often, points in a hyperplane are not readily linearly separable. In those cases a kernel function is used to map the hyperspace, or kernel space, into that is linearly separable. Figure (1) shows a hyperspace that we separated using a kernel function[3].

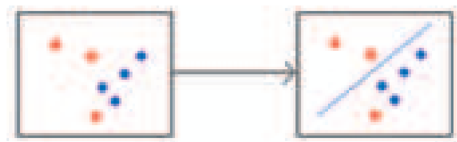

Figure 4: Given a set of points representing inputs, SVMs find an optimal separation to classify inputs as belonging to one of two classes.

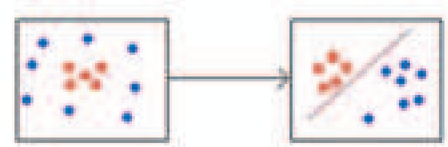

Figure 5: Kernel functions map points not linearly separable into a linearly separable arrangement before the hyperplane is divided. Each point in the hyperplane represents an input image as represented by a feature vector.

After segmenting the image into 40 by 40 pixel windows with a 20 pixel vertical and horizontal overlap, a numeric feature vector was computed for each image block being analyzed, which we then used to train the SVM classifier. Images are classified as containing crystals if one of its blocks is classified as positive by the SVM[11].

3 Feature Extraction

The theory we based our work in revolved around the idea of extracting texture measures from an input image in order to classify the image or portions of interest within the image as textured or not textured. Crystals, we believe are found in low or no texture areas. Furthermore, the space inside the crystals should be untextured. To reach our goal, we would have e to combine textural feature extraction with an edge detecting algorithm whose output would be used to identify candidate crystal areas through a method of finding connected segments and their associated convex hulls.

3.1 Co-Occurrence Based Features

A *Co-occurrence Matrix* is a two-dimensional array **C** whose elements represent the joint probability that two image pixels that are neighbors in the direction θ at a distance d have gray values equal to i and j [14]. The co-occurrence matrix C_d is calculated by:

$$C_d[i,j] = |\{[r,c]|I[r,c] = i and I[r+dr, c+dc] = j\}|, \quad (2)$$

where **(dr,dc)** is a displacement vector, and **I** is the image being analyzed. The contributions of the matrices in different directions (textural orientations) can be summed up in a unique matrix [13]:

$$P_d(i,j) = \frac{1}{||S||}\sum_{\theta\in S} C_{\theta,d}(i,j), \quad (3)$$

where $S = \{0, 45, 90, 135\}$. The resultant matrix $P_d(i,j)$ exploits the local distribution of intensity values of the input image in a similar fashion as the Polarity of the Blob World texture does. The difference is that the co-occurrence matrix bypasses the problem of relying on gradient derived computations.

Just as the Anisotropy was normalized before, we normalize our co-occurrence matrix so that the values of the matrix lie between 0 and 1, allowing them to be thought of as probabilities in a larger matrix. The difference is that unlike with Anisotropy of the blobworld features, employing the co-occurrence matrix we can have more parameters to feed to our classifying algorithm[14]. The *normalized* gray-tone co-occurrence matrix N_d is obtained by

$$N_d[i,j] = \frac{P_d[i,j]}{\sum_i \sum_j P_d[i,j]} \quad (4)$$

From the normalized co-occurrence matrix we can calculate five numeric features that can be used to represent the texture more compactly and allows us comparing two textures.

The first of the texture measures that I extracted was the *Energy*[14]. *Energy*, also called Angular Second Moment, is a measure of the uniformity of an image. Texture inside crystals should be homogeneous, which would correlate to large values for the energy measure since a homogeneous image contains very few dominant gray tone transitions.

$$Energy = \sum_i \sum_j N_d^2[i,j] \quad (5)$$

Entropy measure the disorder of an image[5]. *Entropy* is very large when the image is not very texturally uniform, Hence, *Entropy* inside crystals is expected to be very low, since that would signal no drastic gray tone transitions.

$$Entropy = -\sum_i \sum_j N_d[i,j] log_2 N_d[i,j] \quad (6)$$

Contrast is a weighted texture measure, the squared term being the weighted part. Weights increase away from the diagonal, since diagonal values show no contrast. *Contrast* indicates the level of local variability or smoothness and is larger for images with quickly varying intensities, such as edges of crystals or fuzzy textured images. Algebraically, it is represented by[14]:

$$Contrast = \sum_i \sum_j (i-j)^2 N_d[i,j] \qquad (7)$$

Homogeneity will be large for images with constant or near constant patches. It is the inverse of the contrast since the weights decrease away from the diagonal. Crystal images should have constant patches (representing crystals or crystal portions), hence we are interested in images with high homogeneity values. Homogeneity is obtained with the following formula[14][5],

$$Homogeneity = \sum_i \sum_j \frac{N_d[i,j]}{1+|i-j|} \qquad (8)$$

The *Correlation* measures the linear dependency among neighboring pixels. It gives a measure of abrupt transitions. Values at which correlation values decline suddenly may be taken as one definition of the size of definable objects within an image, such as the edge of a crystal or the presence of several objects in an image. The correlation is a weighted statistical measure obtained by[14]:

$$Correlation = \frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j) N_d[i,j]}{\sigma_i \sigma_j} \qquad (9)$$

where

$$\mu_i = \sum_{i=1}^{N_g} i \sum_{j=1}^{N_g} P(i,j), \qquad (10)$$

$$\mu_j = \sum_{j=1}^{N_g} j \sum_{i=1}^{N_g} P(i,j), \qquad (11)$$

$$\sigma_i^2 = \sum_{i=1}^{N_g} (i-\mu_i)^2 \sum_{j=1}^{N_g} P(i,j), \qquad (12)$$

$$\sigma_j^2 = \sum_{j=1}^{N_g} (j-\mu_j)^2 \sum_{i=1}^{N_g} P(i,j) \qquad (13)$$

3.2 Edge Detection Based Features

Edge detection was carried out using a Canny edge detection filter because Canny is especially good when applied to images that naturally have sharp straight lines[]Shapiro, such as is the case with crystals. The Canny operator works in a multi-step process. First, it convolves the image using a Gaussian smoothing mask of scale sigma. This gets rid of background noise; the larger the value of sigma the more information is ignored. In our experiment, a sigma of 2 was optimal, since a larger value got rid of too much information, and a smaller value introduced too much superfluous information. Table 1 shows the accuracy obtained with different sigma values[14].

After the smoothing phase is completed, the gradient magnitude and direction are computed. Edges give rise to ridges in the gradient magnitude image. The algorithm tracks along the top of these ridges and sets to zero all pixels that are not actually in the ridge top so as to give a thin line output, a process known as *non-maximal suppression*. Lastly, in a process that combines thresholding and hysteresis, the algorithm uses an upper threshold to find the start of a line, it then traces the edge's path through the image pixel by pixel, making an edge while the pixel values stay above a low threshold[14].

Three parameters affect the performance of the Canny operator, the width of the Gaussian used in the smoothing phase, and the upper and lower thresholds used by the tracker[14]. Early trials suggested an upper threshold of 0.38 obtained by testing systematically improvements in edge detection. Ultimately we chose to utilize an automatic threshold detecting function whose results were slightly improvements on our manual observation.

Classifier	Sigma	Detected Crystals
Decision Tree	1	36.4%
Neural Network	1	52.7%
Decision Tree	2	41.8%
Neural Network	2	47.3%
Decision Tree	3	38.1%
Neural Network	3	51.8%

Table 1. Correctly classified crystals using different Gaussian masks.

Once the edges have been identified, the connected segments are found and the convex hulls computed. The convex hulls are enclosing convex polygons that enclose a number of connected segments. If a single segment or segments in a straight line are found in the image, we cannot create any polygon around them. Since the classification takes place n small windows, it is possible that a portion of a crystal edge is found in our block only. Though we cannot compute an associated area for this instance, we can catalogue the crystal as containing edges. Figure 2 shows a Canny filter applied to a crystal image.

4 Classifiers

An off the shelf WEKA Machine Learning Java Classifier Suit was used. Neural network algorithms were compared to Meta Classifiers for effectiveness. The Waikato Environment for Knowledge Analysis (WEKA) is a suit of Java libraries that implement many learning and data mining algorithms[9]. It comes with tools for pre-processing data, feeding it to learning schemes, and analyzing the resulting classifiers and their performance. In our study the neural networks classifier consistently outperformed the decision tree classifiers tested.

4.1 Neural Networks

Neural Networks are non-linear modeling techniques that have proved effective in the classification of noisy data, in particular, unlike with Meta classifiers, over-fitting problems are greatly reduced. Over fitting is when the classifier performs better on the trained images than on an unknown set of imaages, while a second classifier performs better with an unknown set of data than with the training data. In a sense, it occurs when the classifier has memorized all the answers to its training questions but has not learned how to figure the problems themselves[10].

Neural networks can be applied to either supervised or supervised classification. The basic unit of a Neural Network is a perceptron. A perceptron is an attempt to model animal neurons. The way perceptrons work is by giving weights to different inputs, optimized through training, and setting a crossholding value, again optimized through training. If the combination of inputs and their weights is above the threshold, then the perceptron responds with a yes, otherwise, it returns a no[10]. Figure 1 shows a perceptron schematic.

Multi Layer Networks consist of inputs plus one or more *hidden layers* which are simultaneously the output of the input and the input of the next layer. Figure 2 shows the basic schematic of a multi Layer Neural Network with one hidden layer[10].

The primary neural network we employed was the supervised non-linear Multi Layer Percep-

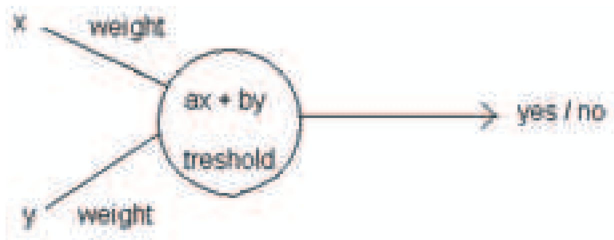

Figure 6: Perceptron.

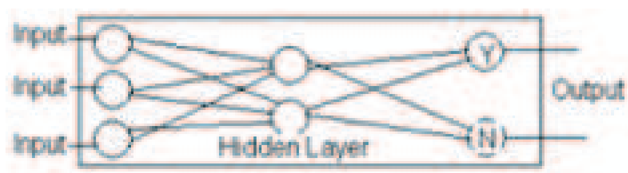

Figure 7: Neural Network.

tion, ideal for a binary class such as ours, using a cross validation method with one hidden layer and Back propagation for active learning. Back propagation is a method of training whereby the answer to a problem is given to the classifier and it learns by working backward to the previous layer to get to the input which is also known. Ideally it learns the optimal weights to solve problems correctly. The number of nodes in the hidden layer corresponds to the number of features being used in classification[10]. The results of the classification are evaluated via a confusion matrix. The confusion matrix tells us the number of instances classified correctly and incorrectly for each class.

4.2 Meta Classifiers

Meta Classifiers are classifiers that use the output of other classifiers as their input[15]. The Meta classifier we used was Decorate with bagging. Diverse Ensemble Creation by Oppositional Re-labeling of Artificial Training Examples (Decorate) builds an ensemble of different classifiers that provide effective diverse committees. The classifiers learn both on the sample data as well as artificially created data. Artificial data is generated by picking random points that appear within the Gaussian distribution of the training data. The probability of these points is computed and used as one f the attributes of the learning process.

Decorate with querying by bagging has been shown to be optimal in building committees for active learning[1]. Bagging is a method for generating multiple versions of a predictor and using those in turn to generate aggregate predictors[2]. The aggregation does a plurality vote when predicting a class. Each of the classifiers is given a number of votes depending on their accuracy, the votes are added and compared against a threshold, if the votes are above the threshold, then the outcome is positive (image classified as crystal). In our experiments over fitting made the meta classifier consistently under perform the neural network classifiers.

5 Results and Discussion

Feature extraction is one part in a larger system that classifies images as crystals or non-crystals. Feature extraction alone does not yield the 2.9

However, we did discover that we can eliminate images with low contrast values. Such images correlate to empty images. Table 2 shows the number of images properly discriminated with different contrast values. This is akin to the elimination of 'bad moves' in an A.I. chess player. A.I. chess players build decision trees several layers deep. To optimally allocate memory resources, algorithms that choose not to expand branches of trees that are 'obvious bad moves' are in place. Such moves are not just moves that end in losing the game, but moves that humans automatically discard such as losing a higher ranked piece in exchange for a lower ranked piece. Analogously, a human classifying images automatically discards empty images, setting a low threshold for the contrast correlates to discarding empty images and has proven in our work an early yet effective way to minimize the number of images that must be classified. Finding the appropriate thresholds for 'obvious' no crystal images is an important step in optimizing a full classifier system. Hence more resources can be allocated to the task of discerning crystals in heavily textured areas, a task which remains difficult.

6 Acknowledgments

We'd like to thank the Distributed Mentor Program and Professor Linda Shapiro for an enriching experience. Also our thanks to Marta Penas-Centeno, Tyler Robison, Pradeep Shenoy, and J C for insightful discussions and moral support.

References

[1]

[2] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):23–140, 1996.

[3] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[4] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24:1026–1038, 2002.

[5] Davis A. Clausi and M. Ed jernigan. A fast method to determine teh co-occurrence texture features. *IEEE Transactions on Geoscience and Remote Sensing*, 36(1):298–300, 1998.

[6] J.H. Van Deemter and J.M.H. Du Buf. Simultaneous detection of lines and edges using compound gabor filters. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(4):757–777, 2000.

[7] Jan-Mark Geusebroek, Arnold W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. In *ECCV*, pages 99–112, 2002.

[8] W. G. J. Hol. Protein crystallography and computer graphics - toward rational drug design. *Angewandte Chemie (Int. Ed.)*, 25:767–778, 1986.

[9] ICONIP/ANZIIS/ANNES'99. *Weka: Practical machine learning tools and techniques with java implementations*, June 1999.

[10] Gardner M.W. and Dorling S.R. Artificial neural networks (the multilayer perceptron)- a review of applications in the atmospheric sciences - applications for wind retrieval from spaceborne scatterometer data. *Atmospheric Environment*, 32(14):2627–2636, 1998.

[11] Shen Pan, Gidon Shavit, Marta Penas-Centeno, Dong-Hui Xu, Linda Shapiro, Richard Ladner, Eve Riskin, Wim Hol, and

Deirdre Meldrum. Automated classification of protein crystallization images using support vector machines with scale-invariant texture and gabor features. *Biological Crystallography*, 62(2):271–279, 2006.

[12] Mari Partio, Bogdan Cramariuc, Moncef Gabbouj, and Ari Visa. Rock texture retrieval using gray level co-occurrence matrix. In *5th NOrdic Signal Processing Symposim Proceedings*, pages 308–311. The NOrwegian Signal Processing Society, October 2002.

[13] L.A. Ruiz, A Fdez Sarria, and J.A. Recio. Texture feature extraction for classification of remote sensing data using wavelet decomposition: A comparative study. In *XXth ISPSR Congress Proceedings*, pages 1109–1114. International Society for Photogrammetry and Remote Sensing, July 2004.

[14] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, Upper Saddle River, New Jersey, 2001.

[15] Ljupco Todorovski and Saso Dzeroski. Combining classifiers with meta decision trees. *Machine Learning*, 50(3):223–249, 2004.