Automated Classification of Protein Crystallization Images Using Support Vector Machines and BlobWorld Textures

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

Abstract

Protein crystallography labs are performing an increasing number of experiments to obtain crystals of good diffraction quality. Better automation has enabled researchers to prepare and run more experiments in a shorter time. However, the problem of identifying which experiments are successful remains difficult. In fact, most of this work is still being done manually by humans. Automating this task is therefore an important goal. As part of a project to develop a new and automated high-throughput, capillary-based protein crystallography instrument, we have been developing a new image classification subsystem to greatly reduce the number of images that require human viewing. This system must have low rates of false negatives (missed crystals), possibly at the cost of raising the number of false positives. The image classification system employs a support vector machine (SVM) learning algorithm to classify the blocks of each image. A new algorithm to find the area within the image that contains the drop is employed. The SVM uses numerical features, based on texture and the Gabor wavelet decompostion, that are calculated for each block. If a block within an image is classified as containing a crystal, then the entire image is classified as having a crystal. In a study of 375 images, with 87 containing crystals, a false negative rate of less than 4%, with a false positive rate of about 40% is consistently achieved.

1 Introduction

Protein crystallography provides access to structural information with atomic resolution for macromolecules such as DNA, RNA, saccharides and proteins. This structural information helps biologists to understand the function of the macromolecular systems investigated. Understanding structure-function relationships can then also help projects aiming at blocking or improving the properties of biomacromolecules. One example is “Structure-Guided Drug Design” (e.g. Hol [9]; Veeranpandian [20]; Guillemont *et al* [10]) where drug designers use detailed knowledge of the shape of key cavities and binding modes of ligands to improve affinities and pharmacological properties of “leads”, i.e. of potential pharmaceuticals. In other areas knowledge of e.g. the insulin structure has led to research to obtain variants of human insulin with improved pharmacokinetic properties (See e.g. Vajo *et al* [19])

Crystallographers go through several essential experimental steps. During the protein crystallization stage, proteins are injected into many different solutions in hopes of crystal growth. Then the crystallographers identify the successful experiments that yield mountable crystals. The final stage is the computation of protein structures from X-Ray diffraction patterns of protein crystals.

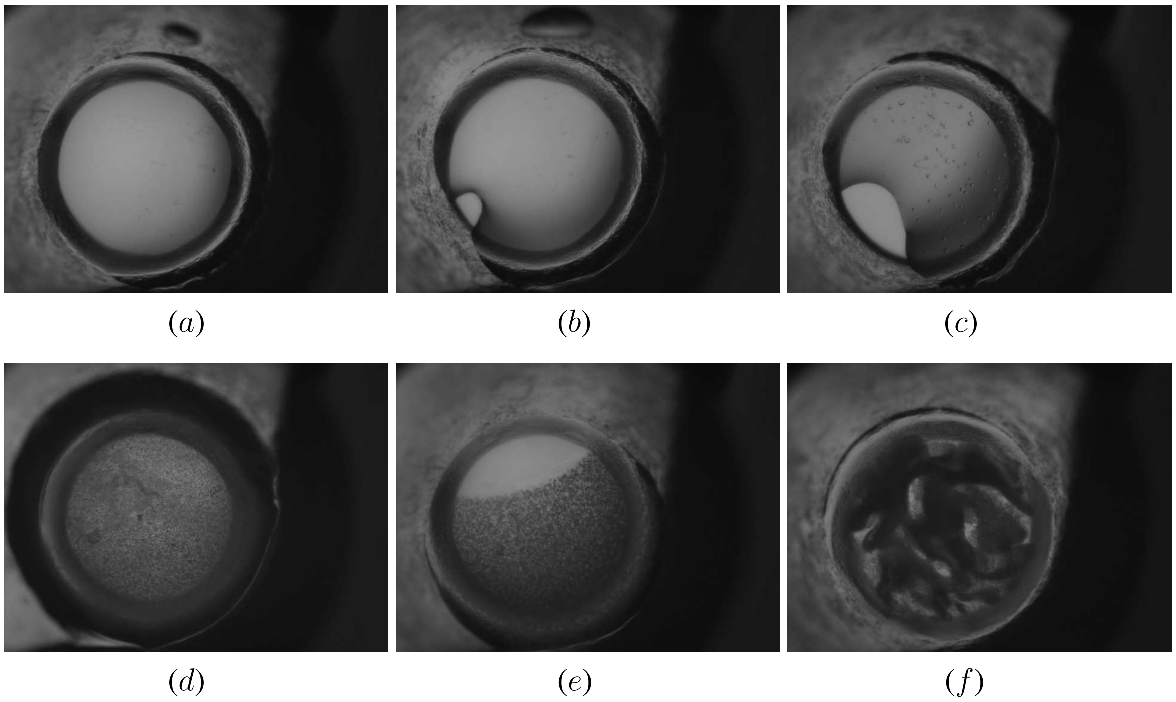


Figure 1: Examples of images of crystallization trials taken by a SGPP RoboDesign microscope camera. (a) Clear, (b) Clear with an air bubble, (c) Crystal with an air bubble, (d) Precipitate, (e) Precipitate with an air bubble and (f) Complex precipitate (or denatured protein).

Currently, during the stage of identification of successful experiments, human crystallographers need to manually examine images taken by microscopic cameras. With the increasing number of crystal growth experiments, the total is up to millions of images per year. The process is both time consuming (hence expensive) and prone to human error. Automating this task is therefore an important goal. As part of the capillary-based protein crystallography instrument project, we have developed a new image classification subsystem to greatly reduce the number of images that require human viewing.

Figure 1 shows some example images of protein crystallization trials taken by a Structural Genomics of Pathogenic Protozoa (SGPP) RoboDesign microscope camera (RoboDesign International Incorporated, Carlsbad, California). As one can observe, great variability is present in the images on which we need to perform classification. Different drop shapes, crystal shapes and precipitate clouds in the images make automatic classification a very challenging machine-learning problem.

2 Related Literature

Research on this problem is still in its early stages. Only a few papers have appeared on the subject, and standard image collections and benchmarks do not yet exist. In early work, Zuk and Ward [25] used the Hough transform [16] to identify straight edges of crystals but did not attempt to classify precipitates.

Wilson [23] used the Sobel edge detector to locate drop boundaries and detected objects (connected component of edge pixels) with high circularity. The classification of objects inside the drop boundary was based on features computed from edge pixels. Their results had 86% of true crystals classified as crystals, while 77% of precipitates were classified as precipitates. Jurisica *et al.* [11] used a custom-built image acquisition and image processing system to analyze protein crystallization experiments under oil. They detected drop boundaries by fitting a conic curve and classified images based on spectral analysis.

Spraggon *et al.* [17] applied the Canny edge detector and circle fitting on the largest connected component of edge pixels to drop boundary detection. Their features included geometric features related to straight lines (detected by the Hough transform) and texture features based on correlations between intensities at various distances and directions. For classification, they used a self-organizing neural network, achieving 25% false negative and false positive rates.

Cumbaa *et al.* [4] described a custom-built system which uses a probabilistic graphical model for drop boundary detection and for classification. Features are based on correlation filters and Radon transform, which also detects straight lines [22]. The classifier achieves a balanced error rate of 15% for binary classification of crystal-positive and crystal-negative. Bern *et al.* [1] used edge detection followed by dynamic-programming curve tracking to determine the drop boundary. They used a decision-tree classifier with hand-crafted thresholds operating on features derived from the Hough transform and from curve tracking. The classification achieves a false negative rate of 12% and a false positive rate of 14%.

THE best result so far is reported in [24] by Zhu, using geometric features characterizing local pixel gradients and texture features derived from the gray-level coocurrence matrix. For classification, a support vector machine (SVM) is compared with an automatic decision-tree classifier, achieving a false positive rate of 14.6% and a false negative rate of 9.6%.

Results reported above are not directly comparable with the 40% and 4% reported here because of the different wells used to perform protein crystallization and the different sets of both training and testing images.

3 System Overview

The system we have built accomplishes several important goals. First of all, it handles the required throughput (thousands of images per day). Secondly, it correctly rejects the majority of the non-crystal images, maintaining a false positive rate of less than 40%. This means that three out of five images without crystals do not need to be manually examined. Since the majority of images do not contain crystals, this is a significant time savings. Most importantly, the system consistently identifies images with crystals correctly with less than 4% error. Finally, it can be integrated into the overall system pipeline illustrated in Figure 2 and easily used by crystallographers with no special computer skills.

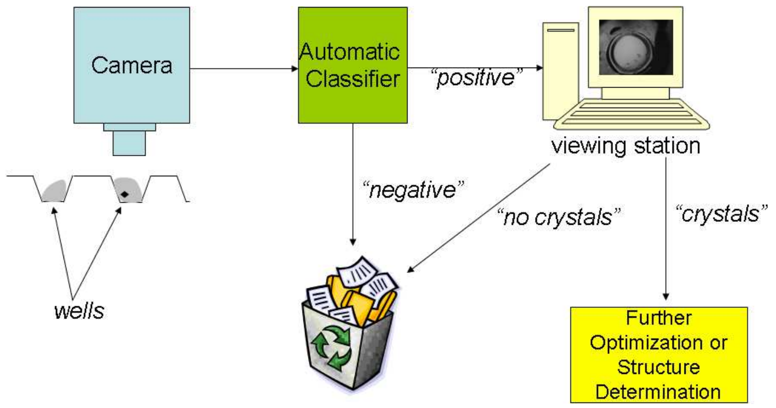


Figure 2: System Overview.

Figure 2 shows the role our imaging subsystem plays in the overall crystallography system. It receives images taken by a RoboDesign microscopic camera of plates containing results from completed experiments. The classifier then labels the images it receives as positive (containing crystals), or negative (containing no crystals). Images classified as containing no crystals are discarded. Images classified as containing crystals will be transferred to a human viewing station for further confirmation. Based on results from the human viewing station, successful experiments are identified, and crystallographers can proceed to the next stage.

Our image analysis system consists of five stages. During the first stage, the system preprocesses the raw image obtained directly from the microscopic camera. In the second stage, it performs image segmentation to locate the area of interest and crop the image. In the third stage, it performs feature extraction to obtain features that will serve as numerical input to the classifier. Then, the system divides the image into small overlapping blocks of size 40x40 pixels in steps of 20 pixels. The size of the blocks was determined experimentally by the average size of crystals on images. It was selected to be large enough to encompass entire crystals, but small enough to not present a huge computational burden. Sliding blocks are used to provide a finer coverage of the image. Because we classify the entire image as containing a crystal if any of the overlapping blocks are classified positive, we bias the classifier to detecting crystals when they occur.

In the fourth stage, for each block, the system computes the numerical representations of features extracted in the third stage. In the fifth and final stage, the system utilizes a trained SVM to classify each block by its numerical feature vector. An image is classified as containing crystals if at least one block in the image is classified positive by the SVM.

4 Image Segmentation

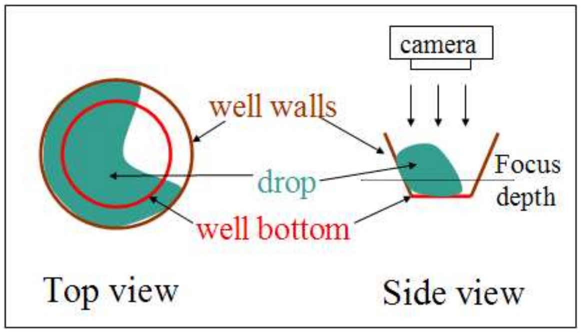


Figure 3: The well on a plate

In the images taken by the microscopic camera, only the regions associated with the solution drop inside the well are of interest to the classification step. Several problems arise in the classification step if one tries to classify the entire image. First of all, random noise outside of the region of interest is very likely to be classified as crystals. Secondly, boundary edges of the well and of the occasional air bubble inside the drop often show up as high contrast textures or significant segments in feature extraction, which could potentially mislead the classifier. Finally, computation time is wasted on processing information that should not matter in the first place. Therefore, finding and segmenting the solution drop area precisely is an important task.

Unfortunately, the images can be very noisy due to clouds of precipitants and other artifacts on the plate outside of the well. The solution drop is sometimes difficult to locate because drop shrinkage due to evaporation often results in bubbles. Furthermore, there are many variations in image characteristics, such as well position and size, general lighting and focus depth.

To locate the well, we use an algorithm that finds the best approximation circle in the image. Then we remove the regions containing bubbles from the area of interest.

4.1 Locating The Well

We first examined the possibility of using the classic circle-finding algorithm based on the Hough transform [16]. The algorithm produces accurate well-detection results for clear images. However, it fails to work well on images where the well boundary is obscured by precipitants inside the well on images with excessive noise. For those images, the algorithm often yields a circle that is off from the center of the well. Therefore, we designed a new algorithm that better suits our application.

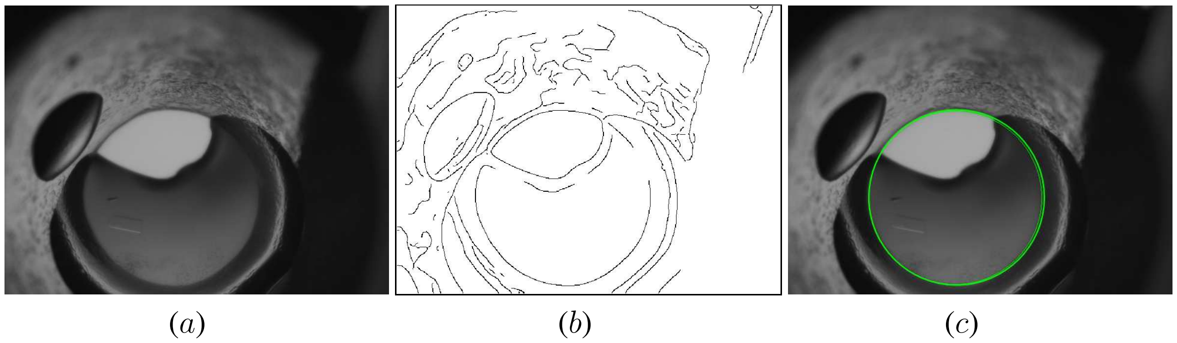


Figure 4: Illustration of the algorithm for locating the well. (a) Original image, (b) Canny edge detected image, (c) Final image with located well.

The well-detection algorithm is illustrated by example in the images of Figure 4. The algorithm first preprocesses the image using a median filter to reduce noise. It then applies the Canny edge detector to the image to obtain an edge image. Afterwards, it fits a circle to the edge pixels in the image using a novel algorithm that takes advantage of the fact that concentric circles of edge pixels are often observed in the images.

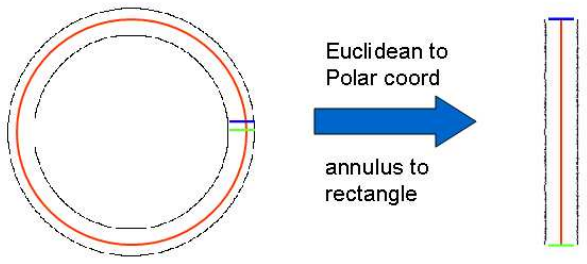


Figure 5: Euclidean (x,y)-Polar (r, θ) coordinate transformation of an annulus.

The key idea of the new circle-fitting algorithm is illustrated by Figure 5: a Euclidean (x,y)-Polar (r,θ) coordinate transformation about the center of a circle turns the circle into a straight line in the new coordinate system. This idea can be easily used to determine the correctness of the center of the circle determined to be the best approximation of the well. For each possible center, we transform an annulus about it to a rectangle. The rectangle with the straightest lines is selected and the corresponding center is the best candidate for the center of the well. To determine the straightness of the lines in the transformed rectangular image, we compute the variance of the means of pixel intensities of all columns in the transformed image. The higher the variance, the more contrast between the columns, therefore the straighter the lines in the image.

One problem introduced by this algorithm is the selection of the size of the annulus on which we perform the transformation. This is closely related to the question of how much information is needed for each possible center for the algorithm to make the best decision. Due to random noise in the images, some images work better with a larger annulus, while others work better with a smaller one. The solution to this problem is to record for each possible center only the highest variance computed among a range of annulus sizes.

Since the wells are located within a certain range on all the images, we only need to do the coordinate transformation and variance computation for a small number of center candidates. The center candidate with the highest computed variance is determined to be the center of the best approximation circle for the well. After the correct center is located, the radius of the circle must be determined. This can be accomplished by finding the radius that makes the circle overlap with the most edge pixels, because it is often the case that well boundaries contain the largest number of adjacent edges in an image.

The results produced by this algorithm are excellent. We tested the algorithm on 514 images of various noise levels and drop shapes. Out of those 514 images, the algorithm produced a slightly off-center, circular fit to the well for only 5 images, a success rate of 99%.

4.2 Locating And Removing Bubbles

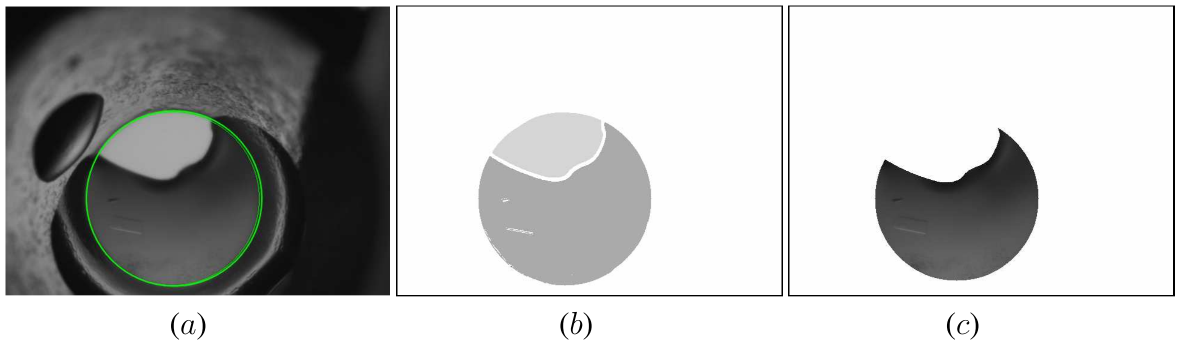


Figure 6: Illustration of the algorithm for locating the bubbles. (a) Original image with well located, (b) Bubble regions found inside the well, where each region appears as a different shade of gray, and edges are in white, (c) Final region of interest.

Figure 6 shows the process of finding the bubbles inside an image. The algorithm first segments the well into regions with gradual changes in pixel intensities by growing regions along the boundary of the well. The region growing is done by a classic flood algorithm that recursively grows a region pixel by pixel until the image gradient at a pixel exceeds a certain predefined threshold in all directions. The recursion is implemented using a stack for performance. The algorithm then checks to see if regions contain holes (produced by edge pixels) and removes those regions from the list of potential bubbles. Finally, the algorithm thresholds the list of bubbles by pixel intensity statistics and sizes (bubbles are generally not very large). The remaining regions are then marked as bubbles.

Problems arise in the determination of the thresholds. Currently, the thresholds are set manually through experimenting with the images. Some bubbles in precipitate images do not have well defined boundaries, and the algorithm fails to produce bubbles that are visually identifiable. The final product of the well and bubble locating routines is a mask of the region of interest.

5 Feature Extraction

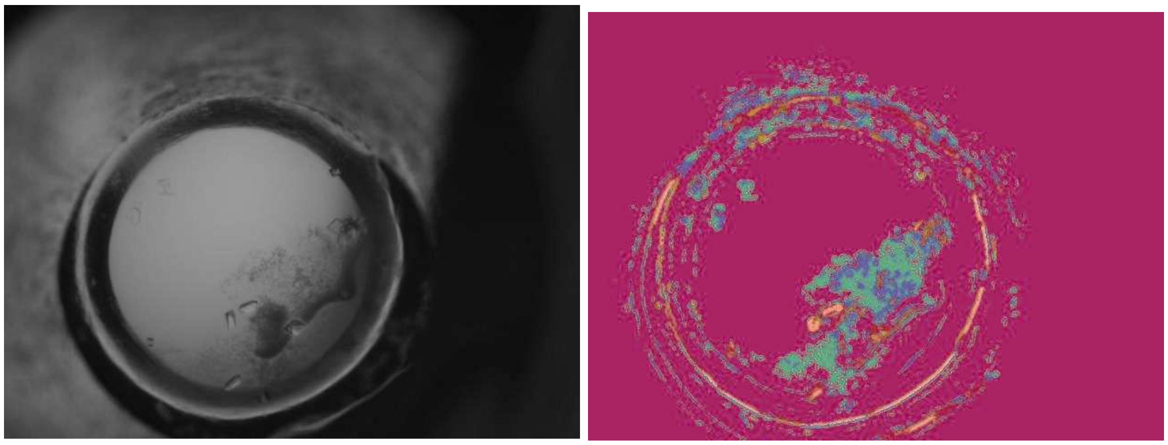
Feature extraction is the most important stage in the system, because for the SVM classifier to classify well, the features must distinguish well between crystals and non-crystals. After much consideration and experimenting, we selected the following features:

- Pixel intensity statistics
- BlobWorld Texture Features (UC Berkeley) [2]
- Results from Gabor wavelet decomposition

Pixel intensity statistics for a block are the mean and standard deviation of the intensity values in that block. In the following sections, the texture and wavelet features are explained in detail.

5.1 BlobWorld Texture Features

Although texture is a well-studied property, most work extracts texture features either on a global scale or on a uniform texture scale across the whole image. Since texture is a local neighborhood property, texture features computed at the wrong scale would lead to confusion. In order to find an appropriate scale for each pixel we adapted the method of scale selection based on edge polarity stabilization which is derived from gradient of gray level intensity [2]. Polarity is a measure of the extent to which the gradient vectors in a certain neighborhood all point in the same direction. The proper scale is selected based on the derivative of the polarity with respect to scale. Polarity is calculated at every pixel in the image for various scales, and each polarity image is smoothed using a Gaussian filter. The best scale is selected based on the difference between values of polarity at successive scales [2].



(a) (b)

Figure 7: Example of BlobWorld texture. (a) Original image, (b) Texture image with polarity, anisotropy and contrast values as its color. It is possible to see the different texture of the crystals. One example would be the crystal located near the center of the well showing up as having a different color from its neighborhood in the texture image.

After a scale is selected for each pixel, three texture descriptors, which are derived from the gradient of gray level intensity, are assigned to that pixel. The three texture descriptors are polarity at that scale, anisotropy and normalized texture contrast. Anisotropy measures the degree of local orientation. Texture contrast defines the homogeneity of the pixels within the local region [2].

5.2 Gabor wavelet decomposition

Gabor wavelets [8] are complex exponential signals modulated by Gaussians and are widely used in computer vision for: edge detection [12, 21], noise filtering [7, 3], image compression [5, 14], texture analysis and synthesis [18, 15], etc. Specifically, Gabor wavelets are suitable for edge detection due to two important properties: good edge localization [6] and the absence of image-dependent parameter tuning.

In this work, Gabor wavelets have been used for the extraction of the low-level directional primitives present in an image. Precisely, a bank of 8 filters centered at frequency $\frac{1}{4}$ and 8 orientations ($\frac{k\pi}{8}, k = 0 \ldots 7$) has been used. The Gabor decomposition through the bank of wavelets maps the input image into 8 resulting images that have been integrated into a unique result, called the *response*. To compute the *response*, each pixel (x, y) is assigned an 8-component vector $H(x, y)$ composed of its Gabor decomposition results. Then, the *response* is computed through equation 1.

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

where $|H(x, y)|$ is the modulus of $H(x, y)$, $|\bar{H}|$ is the modulus of the mean of $H(x, y)$ over the whole image and $|\tilde{H}|$ is the square deviation of the mean of $H(x, y)$ over the whole image. The *response* is bounded by the range $[0, 1]$; high *response* values indicate the presence of a directional primitive in the pixel. The *response* is the only Gabor feature currently employed in our work.

5.3 Blocking and Feature Vectors

After feature extraction, we need some way of summarizing the features into numerical values which are the required input to the SVM classifier. The way we accomplish this is to first divide the region of interest into overlapping blocks of size 40x40 pixels in steps of 20 pixels. Then we summarize for each block a vector of numerical values from the output of the feature extraction process. These overlapping blocks are the key to our low false negative rates. Any crystal that is missed by one block is very likely to be captured by a nearby overlapping block. The feature vector computed for a particular block determines the classification of the block. We have a total of 18 feature values for each block (see Table 1). All feature values above are computed on a per block basis.

	Pixel Intensity
F1	Mean of pixel intensities
F2	Standard deviation of pixel intensities
	Texture
F3	Maximum of polarity
F4	Minimum of polarity
F5	Mean of polarity
F6	Standard deviation of polarity
F7	Maximum of anisotropy
F8	Minimum of anisotropy
F9	Mean of anisotropy
F10	Standard deviation of anisotropy
F11	Maximum of contrast
F12	Minimum of contrast
F13	Mean of contrast
F14	Standard deviation of contrast
	Gabor wavelet decomposition
F15	Maximum of Gabor response
F16	Minimum of Gabor response
F17	Mean of Gabor response
F18	Standard deviation of Gabor response

Table 1: Features

6 Classification Using Support Vector Machines

Each block feature vector is classified using a pre-trained support vector machine. SVMs are binary classifiers. They first map the feature vector into a very high-dimensional space, via a non-linear kernel function. They then cut the space in half with a linear hyperplane. This hyperplane is chosen to maximize the margin of separation between positive and negative samples in a training set. The image is classified as containing crystals if any one of its blocks is classified positive by the SVM. Figure 8 is an overview of the entire process of training and classification. Both training and classification use very similar processes.

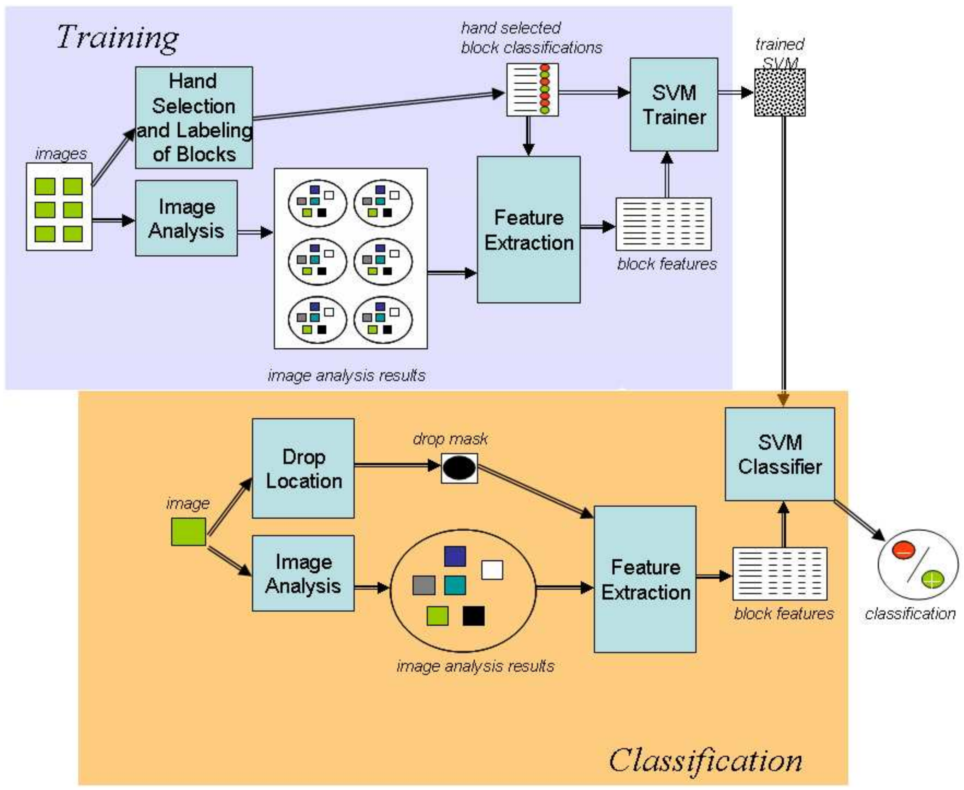


Figure 8: Training / Classification Pipeline.

6.1 Training Support Vector Machine

For training, we manually label (positive/negative) blocks in selected training images to generate a list of labeled (classified) image blocks; images whose blocks appear in that list are processed; a vector of numeric features is computed for each block on the list, and the generated list of labeled vectors is used to train the SVM classifier. The classifier learns to differentiate between positive and negative samples.

6.2 Classification

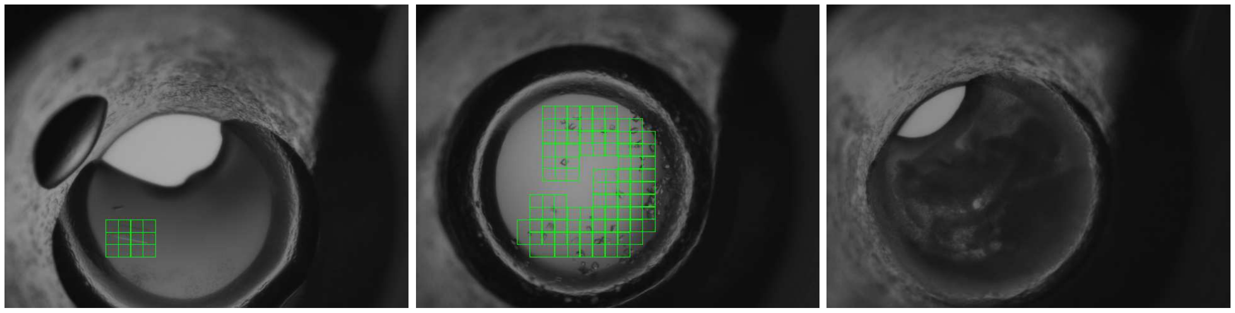

(a) (b) (c)

Figure 9: Classification examples. A green block indicates that block is classified positive. (a) A crystal is correctly classified, (b) Multiple crystals are correctly classified, (c) Precipitates are correctly classified negative.

Classification of new images proceeds similarly: the image is processed as before, and feature vectors are computed for all blocks in the region of interest in the image. These vectors are then classified using the previously-trained SVM. Figure 9 shows some classification examples.

7 Results

Experiments were performed over a data set of 375 images annotated as positive or negative by hand. This dataset consists of 138 clear images, 150 precipitate images and 87 crystal images. These images were obtained by a SGPP RoboDesign microscope camera. An average of 4 to 5 blocks were manually selected and labeled from each of the images in the data set. The selected blocks in a positive image contain crystals and are labeled as positive. On the other hand, the selected blocks in a negative image contain non-crystals (either clear or precipitate) and are labeled as negative.

A total of four different experiments of varying sizes of the training set were performed over the data set. In each of the experiments, the full data set was divided randomly into two disjoint data sets - one training set and one test set. The percentages of the training set over the test set in these experiments are 50%, 60%, 70% and 80%, respectively. We initially tried experiments with 90% of the data in the training set, but the remaining 10% were not sufficient for a test test.

In each experiment, 10 separate trials were performed, and the false negative rate and false positive rate were averaged over the 10 trials. In each trial, the feature vectors of the manually selected blocks in each image in the training set serve as the input to train a classifier. The trained classifier then classifies all the images in the test set for that trial. The resulting false negative rate and false positive rate for each trial were computed based on the classification results over whole images, rather than the blocks in the images. The results of all four experiments are shown in Table 2.

Training Set	50%	60%	70%	80%
Average Training Block #	640	772	897	923
False Negative	3.86%	3.75%	3.42%	2.94%
False Positive	42.50%	41.19%	40.58%	37.68%

Table 2: Results from four experiments of varying sizes of the training set.

It is easy to see that the false negative rate and the false positive rate consistently got better as the size of the training set increased. The best overall results, a false negative rate of 2.94% and a false positive rate of 37.68%, were obtained from the experiment with 80% of the data set as the training set. This series of experiments was performed in hope of finding out the best number of training images or blocks that are needed when one is required to use the system in practice against thousands of images. Although the data set over which the experiments were performed is too small to accomplish this goal, a trend of diminishing returns is clearly visible from the results.

Finally, Figures 10 and 11 shows some common misclassifications in these experiments. These misclassifications were due to such factors as obscured crystals, crystals mixed with precipitate, small crystal-shaped bubbles, precipitate at the edge of a bubble, precipitate containing edges, and impurities that look like crystals.

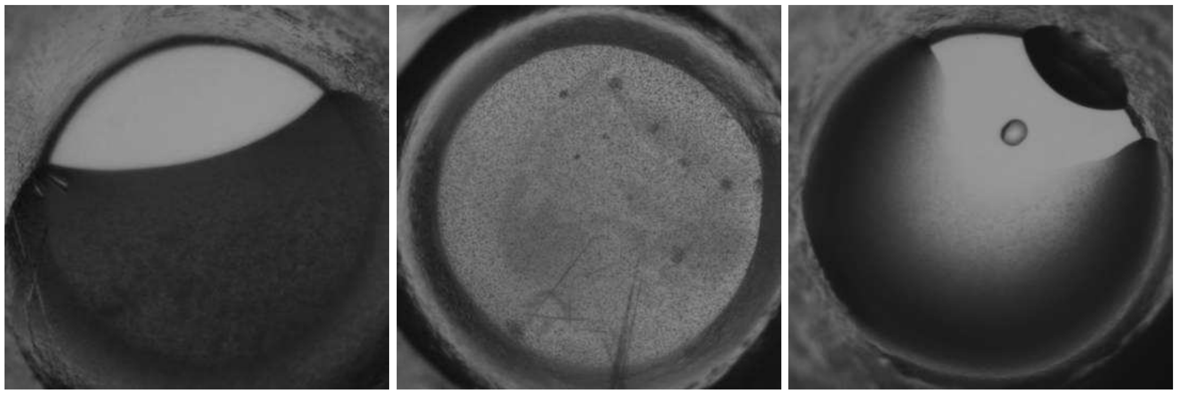


Figure 10: Some mistaken classifications, shown as details of larger images. (a) Obscure crystal (at 9 o'clock) classified as non-crystal. (b) Precipitate and crystal mixture classified as non-crystal. (c) Bubble classified as crystal.

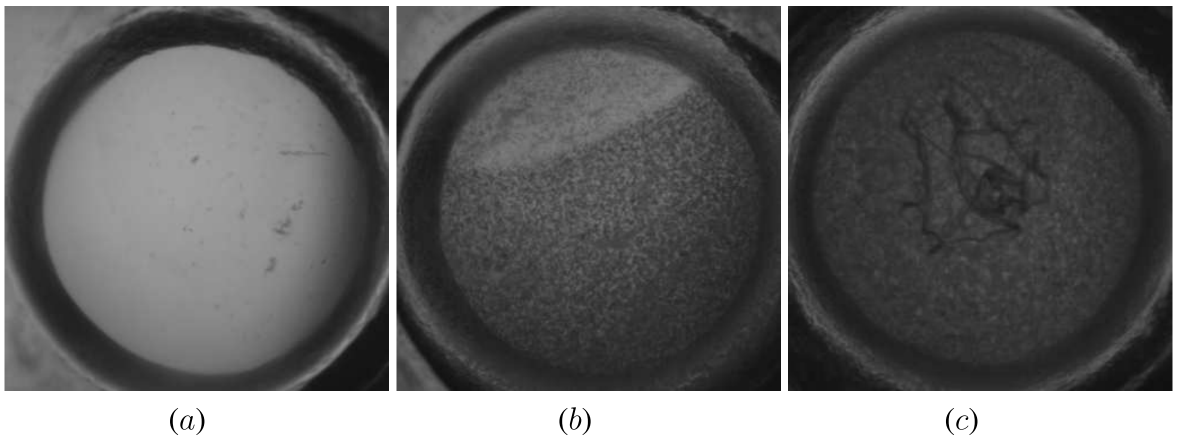


Figure 11: More mistaken classifications, shown as details of larger images. (a) Light impurity classified as crystal. (b) Precipitate at the edge of the bubble classified as crystal. (c) Edgy precipitate classified as crystal.

8 Conclusion and Discussion

The results presented here show that the system has an extremely low false negative rate of 2.9% while maintaining a tolerable false positive rate of 37.7%. The false negative rate is low partly because the system is classifying overlapping blocks of interest in the images so that none of the details of an image is missed. However, this also contributes to the fact that it has a relatively high false positive rate. This shows that the features described here are not sufficient to completely distinguish crystal blocks from non-crystal blocks. There is a need for the inclusion of new features that take advantage of subtle cues such as the presence of corners and parallel segments that humans use to classify images. Attempts to include features of perceptual groupings [13] that capture high-level structures composed of line segments were not successful due to the difficulties involved in summarizing such features numerically. Better summarizing schemes need to be devised in order to best incorporate features from these perceptual groupings into the system. In addition to the inclusion of better features into the system, there is the possibility of developing a better classifier that classifies crystal and non-crystal images into subclasses to help the optimization step of the protein crystallization experiments. Furthermore, optimizations in performance are under consideration since currently the system takes approximately 30 seconds to process an image on a Pentium IV machine. Finally, the most intriguing direction for future work to take is to design a system that helps recognize the trends in crystallization conditions, basing its decisions on the classification results, so that protein crystallization experiments could be done much more efficiently.

The authors thank Larry DeSoto, Stewart Turley, Mark Robien, Oleksandr Kalyuzhniy, Mark Holl, Shawn McGuire and members of SGPP for useful assistance and data. This work was funded in part by the National Institutes of Health NIGMS, “High-Throughput, Capillary-Based Protein

Crystallography" (grant number NIH 5 R01 GM 068878-02).

References

[1] M. Bern, D. Goldberg, R. C. Stevens, and P. Kuhn. Automatic classification of protein crystallization images using a curve-tracking algorithm. *Journal of Applied Crystallography*, 37:279–287.

[2] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1026–38, 2002.

[3] G. Cristbal and R. Navarro. Space and frequency variant image enhancement based on a Gabor representation. *Pattern Recognition Letters*, 15(3):273–277, 1994.

[4] C. A. Cumbaa, A. Lauricella, N. Fehrman, C. Veatch, R. Collins, J. Luft, G. DeTitta, and I. Jurisica. Automatic classification of sub-microlitre protein-crystallization trials in 1536-well plate. *Acta Crystallographica*, 59:1619–27.

[5] J. G. Daugman. Six formal properties of two-dimensional anisotropic visual filters: structural principles and frequency/orientation selectivity". *IEEE Tran. on Systems, Man and Cybernetics*, 13(5):882–887, 1983.

[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] D. Donoho. De-noising by soft-thresholding. *IEEE Trans. on Information Theory*, 41(3):613–627, 1995.

[8] D. Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers*, 93:429–457, 1946.

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

[10] Guillemont J, Pasquier E, Palandjian P, Vernier D, Gaurrand S, Lewi PJ, Heeres J, de Jonge MR, Koymans LM, Daeyaert FF, Vinkers MH, Arnold E, Das K, Pauwels R, Andries K, de Bethune MP, Bettens E, Hertogs K, Wigerinck P, Timmerman P, and Janssen PA. Synthesis of novel diarylpyrimidine analogues and their antiviral activity against human immunodeficiency virus type 1. *J Med Chem*, 48:2072–2079, 2005.

[11] I. Jurisica. Intelligent decision support for protein crystal growth. *IBM Systems Journal*, 2001.

[12] R. Mehrotra, K. R. Namaduri, and N. Ranganathan. Gabor filter-based edge detection. *Pattern recognition*, 25(12):1479–1494, 1992.

[13] M. Penas. *Perceptual Grouping Techniques for the Detection of Objects in Digital Images*. Department of Computer Sciene, University of A Corunha, Spain, 2005.

[14] M. Porat and Y. Y. Zeevi. The generalized Gabor scheme of image representation in biological and machine vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(4):452–467, 1988.

[15] M. Porat and Y.Y. Zeevi. Localized texture processing in vision: Analysis and synthesis in the gaborian space. *IEEE Trans. on Pattern Analysis & Machine Intelligence PAMI*, 10:452–468, 1988.

[16] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[17] G. Spraggon, A. Kreusch, S. A. Lesley, and J. P. Priestle. Computational analysis of crystallization trials. *Acta Crystallographica*, 58:1915–23.

[18] M. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55:71–82, 1986.

[19] Z. Vajo, J. Fawcett, and W.C. Duckworth. Recombinant dna in the treatment of diabetes: Insulin analogs. *Endocrine Reviews*, 22:706–717, 2001.

[20] P. Veerapandian. *Structure-Based Drug Design*. Marcel Dekker, 1997.

[21] Z. Wang and M. R. M Jenkin. Using complex Gabor filters to detect and localize edges and bars. *Advances in Machine Vision: Strategies and Applications*, pages 151–170, 1992.

[22] E. Weisstein. Radon transform. *MathWorld-A Wolfram Web Resource*, 1999-2005.

[23] J. Wilson. Towards the automated evaluation of crystallization trials. *Acta Crystallographica*, 58:1907–14.

[24] X. Zhu, S. Sun, S. E. Cheng, and M. Bern. Classification of protein crystallization imagery. 26th Annual International Conference of IEEE Engineering in Medicine and Biology Society,(EMBS-04), 2004.

[25] W. M. Zuk and K. B. Ward. Methods of analysis of protein crystal images. *Journal of Crystal Growth*, 110, 1991.