Classification of Protein Crystallization Trial Images using Geometric Features

M. Sigdel[1], **M. S. Sigdel**[1], **İ. Dinç**[1], **S. Dinç**[1], **M. L. Pusey**[2], **and R. S. Aygün**[1]
[1]Computer Science Department, The University of Alabama in Huntsville, Huntsville, Alabama, United States
[2]iXpressGenes, Inc., 601 Genome Way, Huntsville, Alabama, United States

Abstract—*In this paper, we describe our method for classification of protein crystallization trial images using geometric features. The objective is to automatically categorize a protein crystal according to the presence of protein crystal types in the images. We consider only the images consisting of protein crystals for the classification. The images are classified into 4 categories- needles, small crystals, large crystals and other crystals. Image classification consists of two main steps - image feature extraction and applying decision tree classifier. Our feature extraction includes application of canny edge detection, extraction of edge related features from the edge image, and extraction of blob related features from multiple thresholding techniques. We performed our experiments on 212 expert labeled images and tested our results using 10-fold cross validation. Our results indicate that the proposed classification technique produces a reasonable classification performance. The overall accuracy of the classification is 75%.*

Keywords: crystallization, edge detection, blob features

1. Introduction

Protein crystallization is the process for formation of protein crystals. Protein crystallization is a rare process and requires thousands of trials for successful crystallization [1]. The objective of crystallization trials is to determine suitable conditions for protein crystallization and produce protein crystals suitable for X-ray diffraction.

High throughput systems have been developed in recent years trying to identify the best conditions to crystallize proteins [1]. Imaging techniques are used to monitor the progress of crystallization. The crystallization trials are scanned periodically to determine the state change or the possibility of forming crystals. With large number of images being captured, it is necessary to have a reliable classification system to distinguish the crystallization states each image belongs to. The fundamental aim is to discard the unsuccessful trials, identify the successful trials, and possibly identify the trials which could be optimized.

Many research studies have been done to distinguish the protein images as non-crystal (does not contain crystal) or crystal (has crystal). For example, Cumba et al. (2003)[2], Cumba et al. (2005) [3], Berry et al. (2006) [4], Pan et al. (2006) [5] and Po and Laine (2008) [6] have described the classification of crystallization trials into non-crystal or crystal categories. In our previous work [7], we described classification of crystallization images into three categories (non-crystals, likely-leads and crystals). Saitoh et al. (2006) [8] proposed crystallization trials classification into five categories (clear drop, creamy precipitate, granulated precipitate, amorphous state precipitate, and crystal). Spraggon et al. (2002) [9] have described classification of the crystallization imagery into 6 different categories (experimental mistake, clear drop, homogeneous precipitant, inhomogeneous precipitant, microcrystals, and crystals). Likewise, Cumba et al. (2010) [10] classified into 6 basic categories (phase separation, precipitate, skin effect, crystal, junk, and unsure).

Not all protein crystals are suitable for X-ray diffraction. The main interest for crystallographers is the formation of large 3D crystals. Other crystal structures are also important as the crystallization conditions can be optimized to get better crystals. Therefore, it is necessary to have a reliable system that distinguishes between different types of crystals according to the shapes and sizes. In the previous studies, classification of the different types of crystals has not been the main focus.

Various classification techniques have been proposed for the classification of protein crystallization trials. Classification algorithms such as support vector machines (SVMs), decision trees, neural networks, boosting, and random forest have been used [7]. Alternatively, combination of multiple classifiers has also been studied in the literature [8]. The recent study by Hung et al. (2014) [11] have proposed protein crystallization image classification using elastic net.

In terms of the feature extraction, a variety of image processing techniques have been proposed. Research studies Cumba et al. (2003) [2], Saitoh et al. (2004)[12] and Zhu et al. (2004) [13] used a combination of geometric and texture features as the input to their classifier. Saitoh et al. (2006) [8] used global texture features as well as features from local parts in the image and features from differential images. Cumba et al. (2010) [10] extracted several features such as basic statistics, energy, Euler numbers, Radon-Laplacian features, Sobel-edge features, microcrystal features, and GLCM features to obtain a large feature vector. Increasing the number of features may not necessarily improve the accuracy. Moreover, it may slow down the classification process.

This study describes our technique for protein crystallization image classification. Our focus is on classifying crystallization trial images according to the types of protein crystals present in the images. Our feature extraction includes edge related features from canny edge image and extracting blob related features from multiple thresholding techniques. The images are classified into 4 categories- needle crystals, small crystals, large crystals and other crystals. Image classification consists of two main steps - image feature extraction and applying decision tree for the classification. We are able to achieve a reasonable classification performance.

This paper is arranged as follows. The following section describes the image categories for the classification problem considered in this paper. Section 3 provides the image processing and feature extraction steps used in our research. Experimental results and discussion are provided in Section 4. The last section concludes the paper with future work.

2. Image Categories

The simplest classification of the crystallization trials distinguishes between the non-crystals (trial images not containing crystals) and crystals (images having crystals). In this study, we are interested in developing a system to classify different crystal types. We consider four image categories (Needle crystals, Small crystals, Large crystals, Other Crystals) for protein crystallization images consisting crystals. Description of each of these categories is provided next.

Needle Crystals - Needle crystals have pointed edges and look like needles. These crystals can appear alone or as a cluster in the images. The overlapping of multiple needle crystals on top of each other makes it difficult to get the correct crystal structure for these images. Fig. 1[a-c] show some sample images under this category.

Small Crystals - This category contains small sized crystals. These crystals can have 2-dimensional or 3-dimensional shapes. These crystals can also appear alone or as a cluster in the images. Because of their small size, it is difficult to visualize the geometric shapes expected in crystals. Besides, the crystals may be blurred because of focusing problems. Fig. 1[d-f] provide some sample images under this category.

Large Crystals - This category includes images with large crystals with quadrangle (2-dimensional or 3-dimensional) shapes. Depending on the orientation of protein crystals in the solution, more than one surface may be visible in some images. Fig. 1[g-i] show some sample images under this category.

Other Crystals - The images in this category may be combination of needles, plates, and other types of crystals. We can observe high intensity regions without proper geometric shapes expected in a crystal. This can be due to focusing problems. Some representative images are shown in Fig 1[j-l].

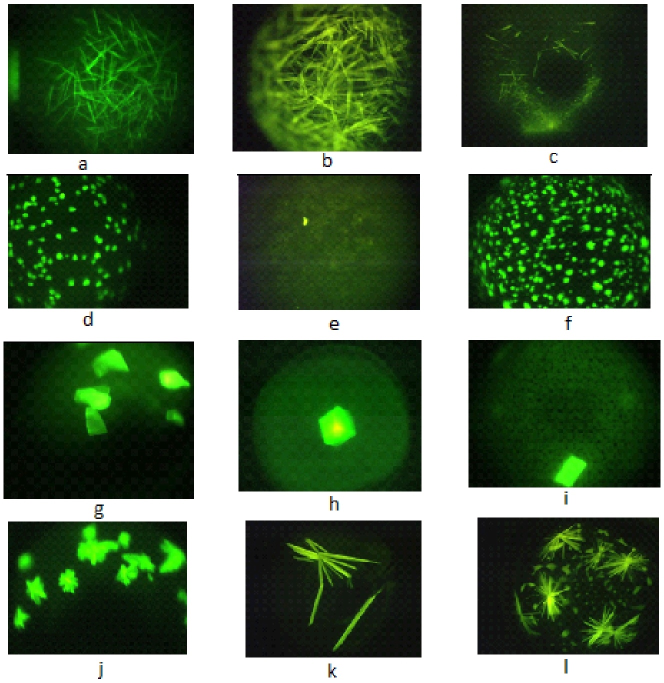

Fig. 1: Sample protein crystallization images: [a-c] Needle Crystals [d-f] Small Crystals [g-i] Large Crystals [j-l] Other Crystals

3. Feature Extraction

The images of crystallization trials are collected using CrystalX2 software from iXpressGenes Inc. Protein solutions are trace fluorescently labeled and the images are collected with green light as the excitation source. As such, the crystals are expected to be highlighted (high intensity) in the image. This can simplify further image processing as the desired objects (crystals) become distinct.

The distinguishing characteristics of protein crystals are the presence of straight lines and quadrangular shapes. Therefore, we focus on extracting geometric features of the objects (or regions) in the image. Fig. 2 shows the components for image pre-processing and feature extraction of our system. Firstly, we down-sample the image and generate binary images using two thresholding techniques. Next, we apply image segmentation and extract features related to the blobs from these binary images. Similarly, we apply canny edge detection and link the edges to get separated segments (graphs) in the image. We then find features related to the segments and the edges. Details of our image processing and feature extraction technique is provided next.

3.1 Image downsampling

A high resolution image may keep unnecessary details and increases the computation time significantly. Therefore, we down-sample the images before further processing. In our

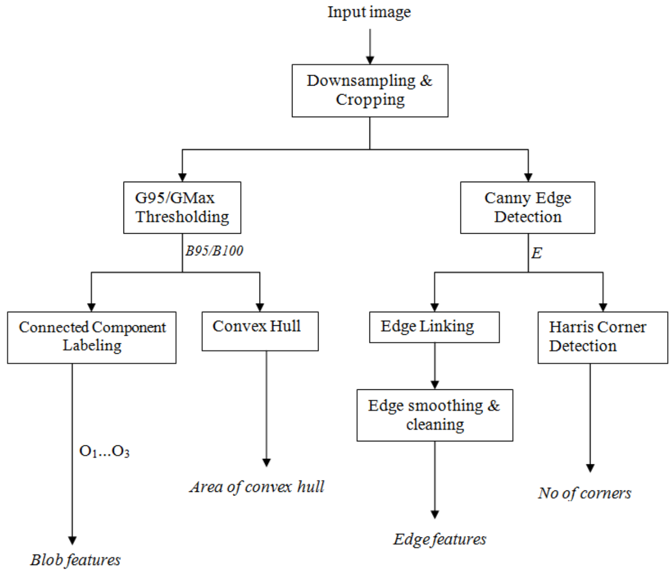

Fig. 2: Component diagram for image processing and feature extraction

experiments, the original size of the images is 2560x1920 pixels. We reduce the image size by 8-fold to get 320x240 sized image. Our analysis shows that the down-sampled images contain sufficient detail for feature extraction.

3.2 Image binarization

Image binarization is a technique for separating foreground and background regions in an image. For the protein images consisting of crystals, the crystal regions are expected to be represented as the foreground in the binary image. Images vary depending on crystallization techniques and imaging devices. This makes it difficult to use a fixed threshold for binarization. Therefore, dynamic thresholding methods are preferred. Different thresholding techniques provide good results for different images. Hence, extracting features from multiple thresholding techniques can be helpful. We apply two percentile based thresholding methods. The implementation and results for each of these techniques are described next.

1) 95^{th} *Percentile of Green (G95)* - When green light is used as the excitation source for fluorescence based acquisition, the intensity of the green pixel component is observed to be higher than the red and blue components in the crystal regions [7]. We utilize this feature for image binarization. First, threshold intensity τ_{g95} is computed as the 95^{th} percentile intensity of the green component in all pixels. This means that the number pixels in the image with the green component intensity below this intensity constitute around 95% of the pixels. Also, a minimum gray level intensity condition (τ_{min} = 40) is applied. All pixels with gray level intensity greater than τ_{min} and having green pixel component greater than τ_{g95} constitute the foreground region while the remaining pixels constitute the background region.
2) *Max green threshold (GMax)* - This technique is similar to the 95^{th} percentile green intensity threshold described earlier. In this method, maximum intensity of green component (τ_{gmax}) is used as the threshold intensity for green component. All pixels with gray level intensity greater than τ_{min} and having green pixel component greater than τ_{gmax} constitute the foreground region while the remaining pixels constitute the background region. The foreground (object) region in the binary image from this method is usually smaller than the foreground region from G95 threshold.

Fig. 3 shows some sample thresholded images using the two methods. From the original and binary images in Fig. 3, we can observe that a single technique may not yield good results for all images. For the images (i) and (ii), the binary images with G95 provide better representation of the crystal objects. However, for image (iii), the result from GMax threshold provides better representation of the crystals.

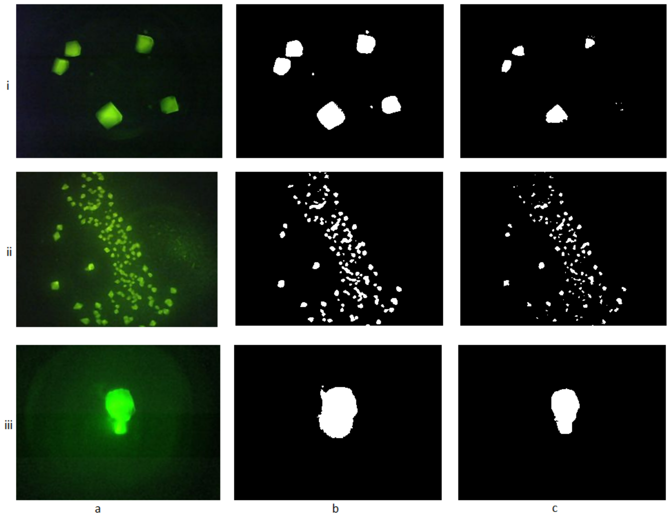

Fig. 3: Figure showing results of two image binarization techniques on crystallization trial images a) Original images b) G95 thresholded images c) GMax thresholded images

3.3 Image segmentation

After we generate the binary image, we apply connected component labeling to segment the regions (crystals). The binary image could be obtained from any of the thresholding methods. Let O be the set of the blobs in a binary image B, and B consists of n number of blobs. The blobs are ordered from the largest to the smallest such that area(O_i)

$\geq$ area(O_{i+1}). Each blob O_i is enclosed by a minimum boundary rectangle (MBR) having width (w_i) and height (h_i). In our implementation, we define the minimum size of the blob to be 25 pixels.

We include the number of blobs in the binary image as one of the image feature. Likewise, for the 3 largest blobs (O_1, O_2 and O_3), we extract the following features and append it to our feature vector.

1) *Blob area* - This is the area of the minimum bounding rectangle (MBR) enclosing the blob. In other words, it is simply the number of pixels in the blob image.
2) *Blob perimeter* - This is calculated as the sum of distance between each adjoining pair of pixels around the border of a blob.
3) *Blob filled area* - This is calculated as the number of white pixels in the blob.
4) *Blob eccentricity* - This measure corresponds to the ratio of the length of the MBR to the the width of the MBR. Eccentricity value lies between 0 and 1 where 0 is obtained when the blob is a circle and 1 is obtained when the blob corresponds to a line segment.

If a binary image contains less than 3 blobs, the value 0 is used for each of these features. It should be noted that the blobs may not necessarily represent crystals in an image. For such cases, the blob features may not be particularly useful for the classifier.

3.4 Convex hull area

In binary images, convex hull is the smallest set of points that forms a polygon shape, which contains the entire objects under consideration [14]. Convex hull points of an object indicates us the smallest number of enclosing object points which can be useful to detect boundaries of the object. We use area of convex hull as another image feature. This feature is useful to determine how the crystals are spread in the image.

3.5 Canny edge detection

Canny edge detection algorithm [15] is one of the most reliable algorithms for edge detection. The algorithm consists of four major steps. Firstly, Gaussian smoothing is done to reduce noise in the image. After Gaussian smoothing, intensity gradient of the image is calculated in different directions. Edge detection operators like Robers, Perwitt, Sobel are used to find the first derivative in the horizontal direction (G_y) and the vertical direction (G_x). Then edge gradient and direction are determined as follows:

$$g = \sqrt{{G_x}^2 + {G_y}^2} \tag{1}$$

$$\theta = G_y / G_x \tag{2}$$

After finding the edge gradient and direction, the edges which do not have local maximum are suppressed and classified as weak edges. Likewise, edges with local maximum are classified as strong edges. If a weak edge is in the neighbor of a strong edge, then it is reclassified as strong edge. The strong edges and the reclassified weak edges form the complete edge image. The result of applying canny edge detector on three images is shown in Fig. 4. Our results show that for most cases, the shapes of crystals are kept intact in the resulting edge image.

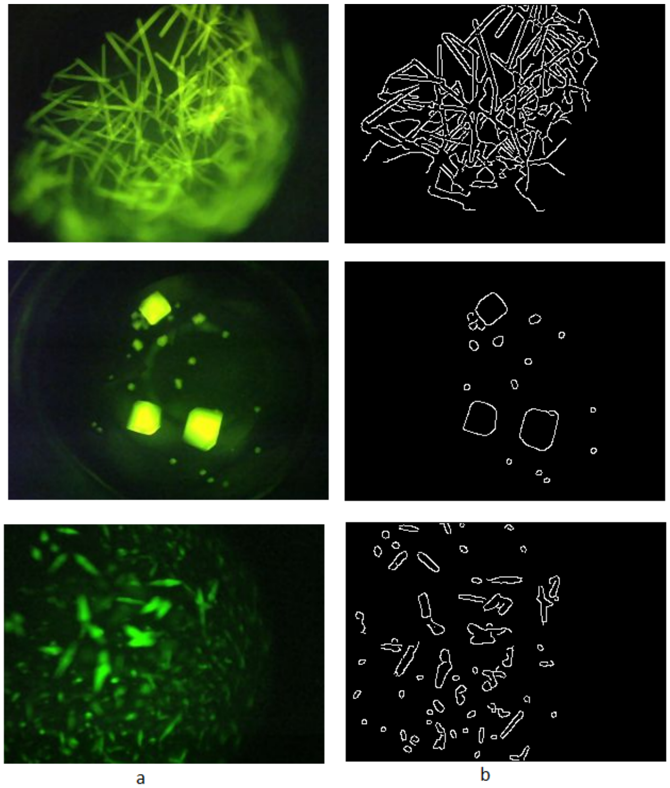

Fig. 4: Applying canny edge detection for 3 images a) Original image b) Canny edge image

3.6 Edge linking

An edge image can contain many edges which may or may not be part of the crystals. To analyze the shape and other edge related features, we link the edges to form graphs or segments. We used the MATLAB procedure by Kovesi [16] to perform this operation. The input to this step is a binary edge image. Firstly, isolated pixels are removed from the input edge image. Next, the information of start and end points of the edges, endings and junctions are determined. From every end point, we track points along an edge until an end point or junction is encountered, and label the image pixels.

The result of edge linking is shown in Fig. 5c and Fig. 6c. The corresponding edge images are provided in Fig. 5b and Fig. 6b respectively.

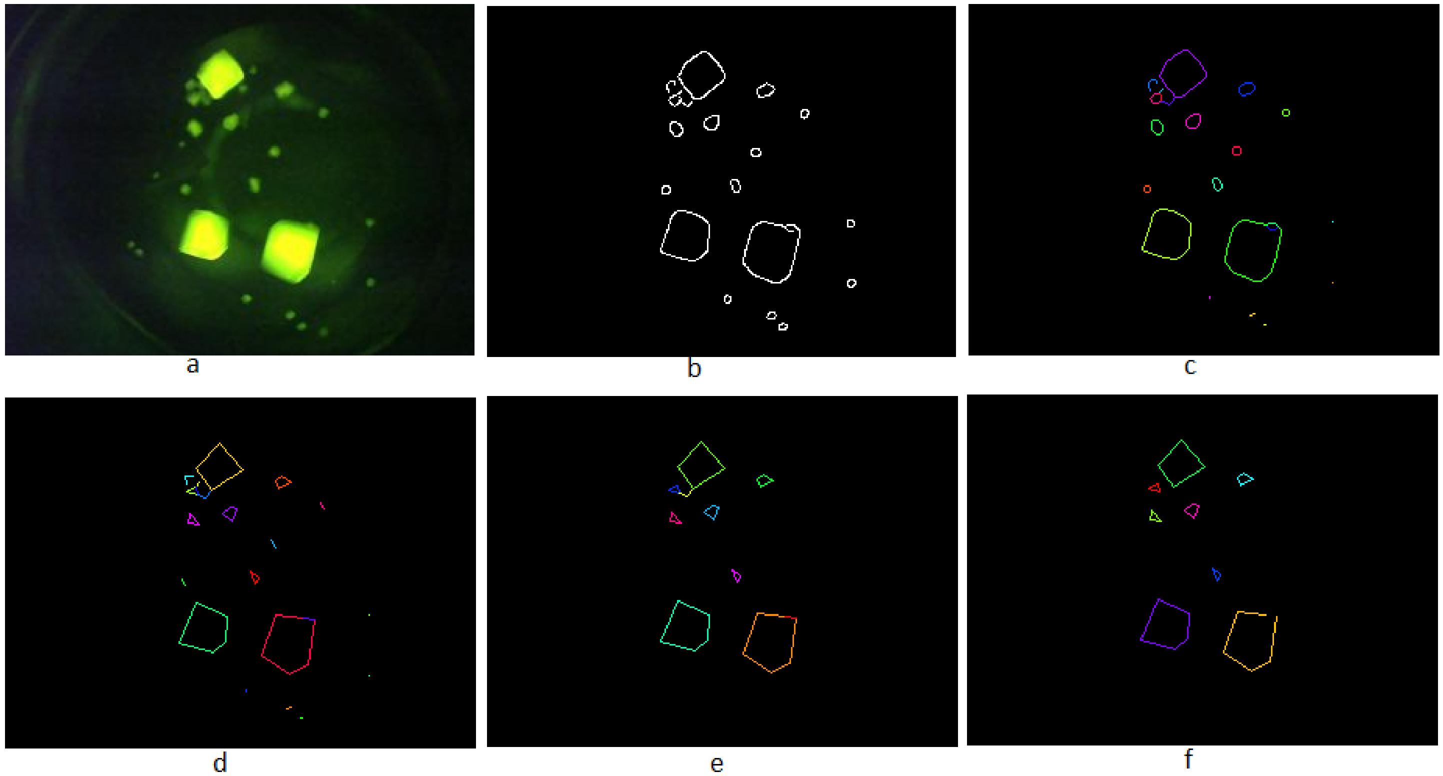

Fig. 5: Figure showing edge detection and edge feature extraction a) Original image b) Canny edge image c) Edge linking d) Line fitting e) Edge cleaning f) Image with cyclic graphs or edges forming line normals

3.7 Line fitting and edge cleaning

Due to problem with focusing, many edges could be formed. To reduce the number of edges and to link the edges together, line fitting is done. In this step, edges within certain deviation from a line are connected to form a single edge. The result from line fitting is shown in Fig. 5d and Fig. 6d. Here, the margin of 3 pixels is used as the maximum allowable deviation. From the figures, we can observe that after line fitting, the number of edges is reduced and the shapes resemble to that of exact shapes of the crystals. However, although desirable, this may not be achieved in all images.

Likewise, isolated edges and edges that are shorter than a minimum length are removed. The result from removing the uninterested edges is shown in Fig. 5e and Fig. 6e. Thus obtained list of edges is used to extract the following edge related features.

1) *Length of edges* - We determine the length of each edge using Euclidean distance measure. For an edge with the edge points (x_1, y_1) and (x_2, y_2), the length (l) of the edge is computed using equation (3).

$$l = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3}$$

2) *Angle between the edges* - We determine the slope of each line and use it to compute the angle between connected lines. If two adjacent lines are almost perpendicular to each other, that provides a hint for the object to be small crystal or large crystal.
3) *Line normals* - Two lines are said to form line normals if the angle between the lines is 90 degrees. For each connected edge segment, we determine if two edges are perpendicular with each other. We consider two lines to be normals if the angle between the lines θ lies between 60 and 90 i.e., $60 \leq \theta \leq 120$.
4) *Cyclic graphs* - We check the edge link list and determine if the edges form a cycle. This is a useful feature to distinguish between needle crystals and other crystals.

Fig. 5f and Fig. 6f provide the edge linked image with only the edge segments that are cyclic or have line normals.

3.8 Harris corner detection

Corner points are considered as one of the uniquely recognizable features in an image. A corner is the intersection of two edges where the variation in both x and y gradient vector directions is very high. Harris corner detection [17] exploits this idea and it basically measures the change in intensity of a pixel (x, y) for a displacement of a search window in all directions. We apply Harris corner detection

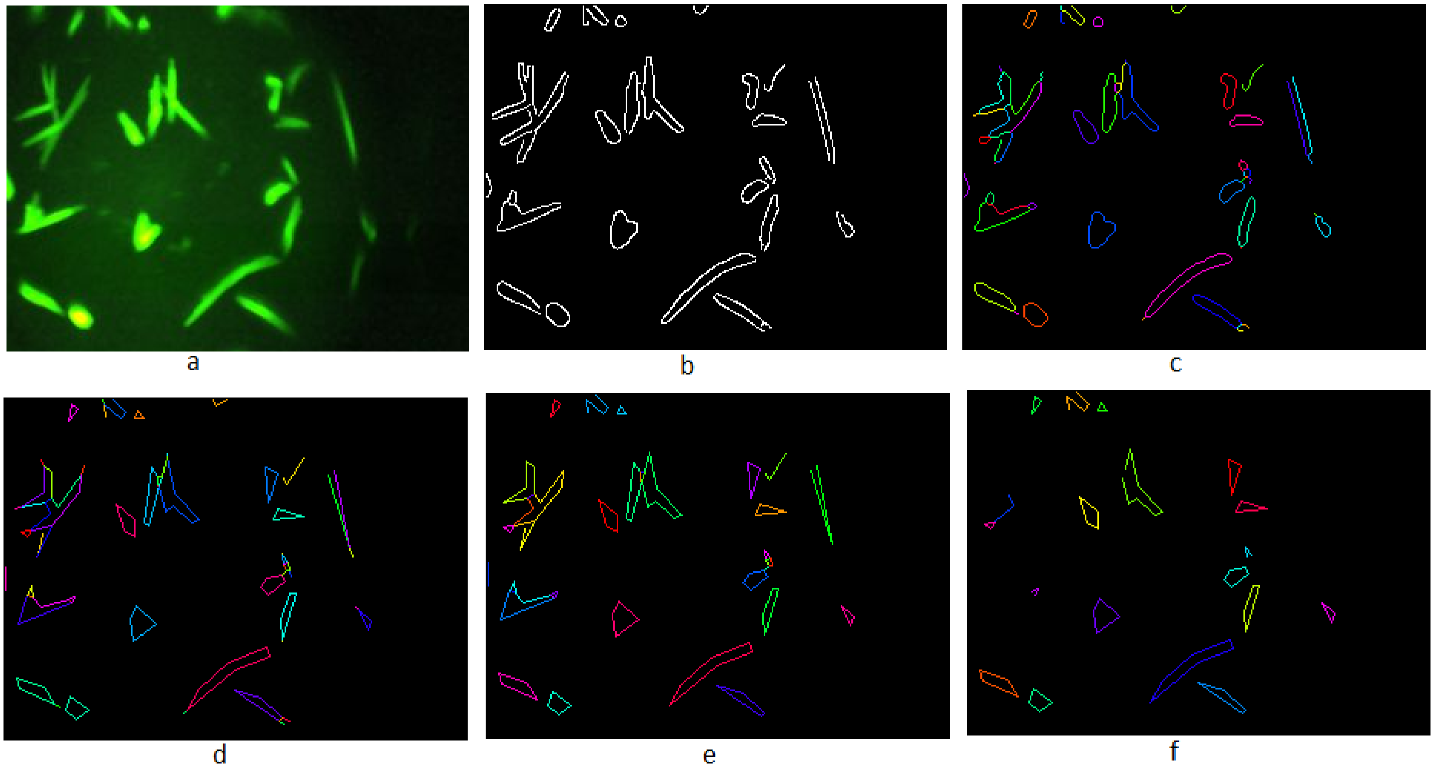

Fig. 6: Figure showing edge detection and edge feature extraction a) Original image b) Canny edge image c) Edge linking d) Line fitting e) Edge cleaning f) Image with cyclic graphs or edges forming line normals

and count the number of corners as the image feature.

3.9 List of features

For each image, we apply 2 dynamic image thresholding methods. Connected component labeling is done on the thresholded images and corresponding blob features are extracted. From each binary image, we extract 3*4 + 2 = 14 blob features. Likewise, we apply canny edge detection and extract 11 edge and corner features. Therefore, we extract a total of 2*14 + 11 = 39 features per image. Below is the list of all the extracted features.

1) Blob features
    a) Area of the 3 largest blobs
    b) Perimeter of the 3 largest blobs
    c) Filled area of the 3 largest blobs
    d) Eccentricity of the 3 largest blobs
    e) No of blobs
    f) Area of convex hull
2) Edge features
    a) No of segments (graphs)
    b) No of 1 edge graphs
    c) No of 2 edge graphs
    d) Has cyclic graph (0 or 1)
    e) Has line normals (0 or 1)
    f) No of cyclic graphs
    g) No of graphs with line normals
    h) Average length of edge in all segments
    i) Sum of lengths of all edges
    j) Maximum length of an edge
    k) No of Harris corner points

4. Experimental Results

Our experimental dataset consists of 212 expert labeled images. The images are hand-labeled by an expert into 4 different categories - Needle Crystals (NC), Small Crystals (SC), Large Crystals (LC) and Other Crystals (OC). These are represented in the proportion 24%, 20%, 35% and 21% respectively. Each image is processed as described in the earlier section and 39-dimension feature vector is obtained by extracting the blob, edge and corner features. We use decision tree as the classifier and evaluate the performance using 10-fold cross validation. Table 1 provides the resulting confusion matrix. We are able to achieve an accuracy of 75% [(38+36+58+26)/212] on average for a four-class classification problem.

Among the 4 classes, we can observe that the system distinguishes the small crystals and needle crystals with high accuracy. Distinction between large crystals and other crystals is the most problematic.

From our discussion with the expert, small and large crystals are the most important crystals in terms of their

Table 1: Confusion Matrix

Actual Class	Observed Class			
	OC	NC	SC	LC
OC	**26**	4	4	10
NC	6	**38**	5	2
SC	1	3	**36**	3
LC	10	2	4	**58**

usability for the diffraction process. Therefore, it is critical not to misclassify the images in these categories into the other two categories. From Table 1, we can observe that our system misses 4 Small Crystals (1 image grouped as other crystals and 3 images grouped as needles). Likewise, our system classifies 10 Large crystals as Other Crystals and 2 Large Crystals as Needles. In overall, our system misses 16 critical images. Thus, the rate of miss of critical crystals of our system is around 8% [16/212]. This is a promising achievement for crystal subclassification of crystal categories.

5. Conclusion and Future Work

In this paper, we described a method for classifying different types of protein crystals in protein crystallization trial images. We extracted features related to edge and the shape characteristics of high intensity regions (blobs). We applied decision tree to develop the classification model and tested our experiments using 10-fold cross-validation. Our results indicate that the proposed classification technique produces a reasonable classification performance.

Crystallographers can not fully rely on the system as the classification accuracy is not very high. Hence, we need to improve the accuracy. The performance of our system depends on the accuracy of image binarization. In some images, the thresholded images do not capture the shapes of crystals correctly. Therefore, the features extracted from blobs may not necessarily represent crystals. Because of this, the features extracted from those blobs are not useful. To solve this problem, we plan to investigate different thresholding techniques. Our initial study shows that using the best thresholded image for feature extraction improves the classification performance.

We also plan to investigate hierarchical classification to obtain the decision model for the classification problem.

6. Acknowledgement

This research was supported by National Institutes of Health (GM090453) grant.

References

[1] M. L. Pusey, Z.-J. Liu, W. Tempel, J. Praissman, D. Lin, B.-C. Wang, J. A. Gavira, and J. D. Ng, "Life in the fast lane for protein crystallization and x-ray crystallography," *Progress in Biophysics and Molecular Biology*, vol. 88, no. 3, pp. 359 – 386, 2005.

[2] 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 plates," *Acta Crystallographica Section D: Biological Crystallography*, vol. 59, no. 9, pp. 1619–1627, 2003.

[3] C. Cumbaa and I. Jurisica, "Automatic classification and pattern discovery in high-throughput protein crystallization trials," *Journal of structural and functional genomics*, vol. 6, no. 2-3, pp. 195–202, 2005.

[4] I. M. Berry, O. Dym, R. Esnouf, K. Harlos, R. Meged, A. Perrakis, J. Sussman, T. Walter, J. Wilson, and A. Messerschmidt, "Spine high-throughput crystallization, crystal imaging and recognition techniques: current state, performance analysis, new technologies and future aspects," *Acta Crystallographica Section D: Biological Crystallography*, vol. 62, no. 10, pp. 1137–1149, 2006.

[5] S. Pan, G. Shavit, M. Penas-Centeno, D.-H. Xu, L. Shapiro, R. Ladner, E. Riskin, W. Hol, and D. Meldrum, "Automated classification of protein crystallization images using support vector machines with scale-invariant texture and gabor features," *Acta Crystallographica Section D: Biological Crystallography*, vol. 62, no. 3, pp. 271–279, 2006.

[6] M. J. Po and A. F. Laine, "Leveraging genetic algorithm and neural network in automated protein crystal recognition," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE, 2008, pp. 1926–1929.

[7] M. Sigdel, M. L. Pusey, and R. S. Aygun, "Real-time protein crystallization image acquisition and classification system," *Crystal Growth Design*, vol. 13, no. 7, pp. 2728–2736, 2013.

[8] K. Saitoh, K. Kawabata, and H. Asama, "Design of classifier to automate the evaluation of protein crystallization states," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1800–1805.

[9] G. Spraggon, S. A. Lesley, A. Kreusch, and J. P. Priestle, "Computational analysis of crystallization trials," *Acta Crystallographica Section D: Biological Crystallography*, vol. 58, no. 11, pp. 1915–1923, 2002.

[10] C. A. Cumbaa and I. Jurisica, "Protein crystallization analysis on the world community grid," *J Struct Funct Genomics*, vol. 11, no. 1, pp. 61–9.

[11] J. Hung, J. Collins, M. Weldetsion, O. Newland, E. Chiang, S. Guerrero, and K. Okada, "Protein crystallization image classification with elastic net," in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2014.

[12] K. Saitoh, K. Kawabata, S. Kunimitsu, H. Asama, and T. Mishima, "Evaluation of protein crystallization states based on texture information," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 2725–2730.

[13] X. Zhu, S. Sun, and M. Bern, "Classification of protein crystallization imagery," in *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, vol. 1. IEEE, 2004, pp. 1628–1631.

[14] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[15] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[16] P. D. Kovesi, "MATLAB and Octave functions for computer vision and image processing," CETSEEentre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[17] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.