Super-thresholding: Supervised Thresholding of Protein Crystal Images

İmren Dinç*,Semih Dinç*, Madhav Sigdel*, Madhu Sigdel*, Marc L Pusey†, Ramazan S Aygün*
*DataMedia Research Lab, Computer Science Department,
University of Alabama in Huntsville,
Huntsville, Alabama 35899, United States
†iXpressGenes, Inc., 601 Genome Way, Huntsville, Alabama 35806, United States
*{id0002,sd0016,ms0023,mss0025,aygunr}@uah.edu †marc.pusey@ixpressgenes.com

***Abstract*—In general, a single thresholding technique is developed or enhanced to separate foreground objects from background for a domain of images. This idea may not generate satisfactory results for all images in a dataset, since different images may require different types of thresholding methods for proper binarization or segmentation. To overcome this limitation, in this study, we propose a novel approach called "super-thresholding" that utilizes a supervised classifier to decide an appropriate thresholding method for a specific image. This method provides a generic framework that allows selection of the best thresholding method among different thresholding techniques that are beneficial for the problem domain. A classifier model is built using features extracted *priori* from the original image only or *posteriori* by analyzing the outputs of thresholding methods and the original image. This model is applied to identify the thresholding method for new images of the domain. We performed our method on protein crystallization images, and then we compared our results with 6 thresholding techniques. Numerical results are provided using 4 different correctness measurements. Super-thresholding outperforms the best single thresholding method around 10%, and it gives the best performance for protein crystallization dataset in our experiments.**

***Index Terms*—Supervised Thresholding, Image Binarization, Protein Crystallization.**

I. Introduction

One of the widely used region-based segmentation approaches is image thresholding [1]. In image thresholding, a grayscale image is converted into a black-and-white image, and it is commonly used in many real time systems such as surveillance systems, medical images, biological images, etc. It reduces the computational load of the systems, since a pixel can be represented with one bit in binary images. In addition, it is fast, easy to implement, and generates acceptable results for many applications. However, there is not an optimal solution that works for all cases, and choosing an invalid threshold value may yield incorrect binary image leading incorrect segmentation. There is a great deal of thresholding techniques proposed in the literature for different domains and types of images. This paper does not intend to mention all these methods. Sezgin et al. [2] provide detailed comparison of the thresholding techniques.

It is desirable that the thresholding method is *sound* and *complete*. In this context, soundness indicates that the output of the method for a sample image should be acceptable and good, while completeness indicates that the method should generate acceptable results for all images in the domain. A method that has a lower precision on some images but generates acceptable results for all images may be preferred to a method that generates precise results for the majority but fails even for a few images in the set. Therefore, we should not only check the accuracy of thresholding, but we should also consider whether the method is complete or generates less number of improper thresholded images.

A. Protein Crystallization Images

Protein crystallization is a critical approach to understand the functionality and the structure of a particular protein [3]. The images of protein solutions are acquired and it is very important to detect well-shaped crystals (see Section III-A) since they provide important information about the structure.

Since the shapes of crystals are important for determining the usability of crystals for further analysis, proper segmentation is critical. Moreover, image segmentation and thresholding may help to determine the phase of a protein image in automated systems. We studied thresholding techniques that have been proposed mostly in the last decade for protein crystallization imagery. Usually, crystal images are expected to have distinguishable features such as high intensity, sharp clear edges, and proper geometric shapes. However, in some cases, these features may not be dominant due to focusing or reflection problems even if there is a protein crystal in the image [4]. Therefore, a single type of thresholding technique may not provide an informative binary image for classifying images. Moreover, binary images may lose some important information or it may keep some unnecessary information leading to incorrect classification. For example, incorrect thresholding method may not detect a blurred

crystal in an image. In our previous work [5], we used three thresholding techniques (Otsu's threshold, 90^{th} percentile green intensity threshold, and max green intensity threshold) together to classify protein crystallization images not to lose any informative feature. All these binary images were used regardless whether they were proper or not. However, when we include features of these three binary images, we may also include unnecessary features that may yield incorrect classification for some of the samples.

We have also tried each of these thresholding techniques one at a time and we have noticed that there is at least one thresholding technique that works for a sample image in general. However, there is no single consistent technique that works for all images. This leads to the idea to construct a system that selects proper thresholding method for a specific sample. In this way, our system may not be bound to limitations of a single thresholding technique.

In summary, protein crystallization images is a challenging problem domain for thresholding due to following reasons:

1) No single thresholding technique works for all images in our protein crystallization image dataset,
2) Since images are collected from different phases of protein crystal growth, crystals may have varying sizes, shapes, and intensities,
3) The sizes and the number of crystals may vary,
4) Images may be captured under different illuminations, and
5) Since crystals may have 3D shapes or they may appear at different depths from the camera, some crystals may be blurred or out of focus.

B. Our Approach

In this paper, we explore whether thresholding can benefit from supervised learning algorithms, and we propose a supervised thresholding methodology that selects the best thresholding technique for a particular image using a classifier. Since our method uses supervised learning, we call our method as "super-thresholding." Super-thresholding has two different feature extraction approaches to select the thresholding method: priori and posteriori. In priori feature extraction approach, features are extracted from original images only. In posteriori feature extraction approach, we first apply different thresholding methods to original images. Then, we map the thresholded image to the original image to extract some features from foreground, background and borders of the regions. Once the features are ready, we train the classifier by these features to select the best thresholding method. Our technique tries to select the most informative and reliable thresholding method for each protein crystal image. This approach provides a generic framework for a set of thresholding techniques that are suitable for the domain. In this paper, our method has been compared to 6 different global and local thresholding methods.

This research uses protein crystallization image dataset provided by iXpressGenes, Inc. Protein images are categorized into three main groups (noncrystals, likely-leads, and crystals). Each group has its own specific characteristics that need to be considered independently. In this paper, we focus on only "crystals" and propose a solution to select the best thresholding technique for each crystal image.

The contributions of our work can be briefly listed as follows:

1) We show that supervised learning methods can be used for thresholding images and introduce our super-thresholding methodology,
2) We apply super-thresholding on protein crystallization images and compare with other thresholding algorithms,
3) We compare both priori and posteriori feature extraction approach of super-thresholding on protein crystallization images, and
4) We show that super-thresholding can be used for developing sound (generates acceptable result for an image) and complete (generates acceptable results for all images) thresholding frameworks.

The rest of the paper is presented as follows. Section 2 provides information about the thresholding background and related work. Our dataset and image binarization techniques that are used in the experiments are described in Section 3. Super-thresholding method is explained in depth in Section 4. Experimental results are provided in Section 5. Finally, our paper is concluded with the last section.

II. Related Work

There has been significant research on image thresholding and segmentation techniques. The thresholding techniques can be roughly categorized as local thresholding and global thresholding. In global thresholding, a single threshold is used for all pixels in the image. In local thresholding, the threshold value may change based on the local spatial properties around a pixel.

Global thresholding generally depends on maximizing variances [6], [7] or entropy [8], [9], [10] between the classes and minimizing the error within the classes. In addition, it does not use spatial information in an image [11]. Generally, the global thresholding techniques benefit from the histogram peaks of the intensities of the image. If there are two distinctive peaks in the histogram of the intensities, finding the optimal threshold value turns out to be straightforward. However, there are some cases where we cannot obtain two separate peaks in the histogram. In such cases, thresholding by iterative partitioning might be a good solution [12], [13].

Unlike global thresholding, local thresholding uses spatial features of a neighborhood in an image [14], [15], [16], [17], [18]. Although local thresholding techniques look more

generic and superior to global thresholding, tuning parameters, partitioning the image, and the time complexity are some issues to be considered [17]. First, parameters of non-automated local thresholding techniques are required to be set by the user for images taken under different conditions. Second issue about local thresholding is that it may classify background pixel as object pixel for poorly illuminated images, even though there is no object in the sub-image. For example, Niblack (1985) calculates the threshold value of each pixel using the mean and standard deviation of its rectangular neighborhood [14]. One of the disadvantages of this method is that tuning the size of a neighborhood is not automated. Since small window size amplifies the noise and inappropriate window size yields incorrect binarization image, it is important to set a proper window size for an image [17].

One of the common research areas where local thresholding is widely used is document binarization [19], [15], [20], [21]. It is possible to get improper results when we employ document binarization techniques on medical or biological images, since there could be some assumptions about background color (e.g., white background) in documents. However, they could be used in different domains after some pre-processing.

Image segmentation methods may also provide promising results for different datasets [22], [23], [24], [25]. However, these methods generally take more time compared to thresholding. In this paper, we have also studied one of the popular image segmentation methods called Pylon [25], which uses a segmentation tree based on pPb edge detector proposed by Arbelaez et al. [26]. The major problem about Pylon is that the generation of segmentation tree takes significant time, which is not applicable to real-time systems. For an image having size 320x240, generating segmentation tree takes around 70 seconds, and Pylon takes around 6 seconds on a 2.5Ghz Quad Core 128 GB RAM server. Figure 1 shows results of Pylon on a protein image, and the last column shows the results for our super-thresholding. As can be seen in Figure 1-d, Pylon has merged two separate crystal objects into a single object, and it has classified small crystal regions around the large crystals as background regions, which could be very critical for crystallographers. Since this method does not fit well for real-time systems, we do not include this method in our experiments.

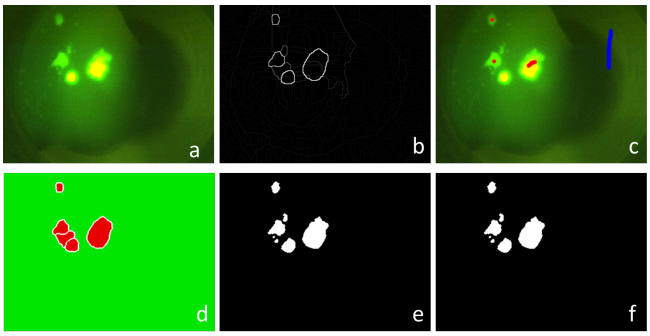

Figure 1. Sample output of Pylon on a protein image: a) original image, b) segmentation tree (Arbelaez et al.) [26], c) user seeds (brushes), d) Pylon result, e) super-thresholding priori approach, and f) super-thresholding posteriori approach

III. Background

In this section, we firstly provide brief information about protein crystal images. Later, we explain the thresholding techniques used in our experiments.

A. Dataset

In this paper, we focused on binarization of the crystal images, which contains 3 types of the crystal objects: 2D plates, small 3D crystals, and large 3D crystals. We believe explaining protein images would help reader understand the problem domain. However, note that the phase information (or category) of the images is not used in our system in any way.

1) 2D Plates: 2D plate images have quadrangular shapes, and they may have any size in the images. If the objects are out of focus, this makes binarization of these images challenging. For some specific cases, it is hard to detect or observe edges of those objects due to noise, poor illumination, and focusing problems. Figure 2 a-c shows some sample images for this category.

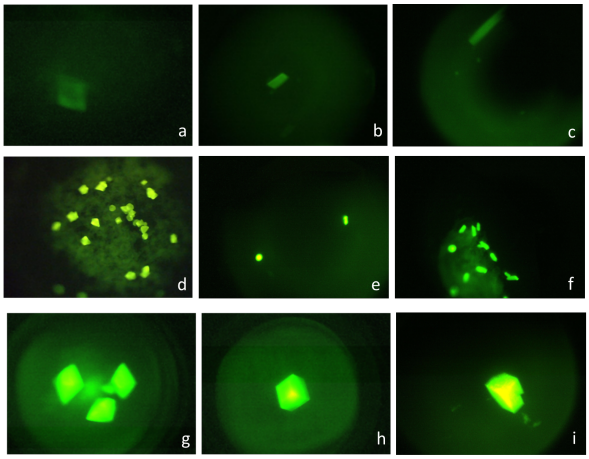

Figure 2. (a-c) 2D plates, (d-f) small 3D, (g-i) large 3D crystal samples.

2) Small 3D Crystals: The images in this category generally contain many small objects that are distributed throughout the image. The binarization issue about this category is that since there are many small objects in the image, it is possible to miss some of the out-of-focus objects if chosen threshold value is not appropriate for them. Edge sensitive thresholding methods also fail, because it is hard to detect

edges of those objects due to small size. Figure 2 d-f shows some sample images for this category.

3) Large 3D Crystals: High intense background can be observed in these images due to light reflection, which causes most of the binarization methods to fail. These bright background regions can be incorrectly classified as object regions in the binary images. This situation may yield improper binary images for this category. Figure 2 g-i shows some sample images for this category.

B. Image Binarization Methods

In this section, we give brief explanations of the three thresholding methods that are used in super-thresholding. Although we have tested more than three methods in our preliminary experiments, other methods (i.e., thresholding using component tree (Silva, 2011) [27], image segmentation using double local thresholding (Chuang, 2011) [18], edge sensitive thresholding [17], thresholding based on iterative partitioning [13], Otsu's thresholding [6], and Pylon [25]) neither generate proper binary images for protein images nor improve super-thresholding accuracy. Therefore, these methods were not included in our experiments for super-thresholding method.

1) 97^{th} Percentile Green Intensity Threshold ($g97$): When the 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 [5]. 97^{th} percentile green intensity threshold utilizes this feature for image binarization. First, the threshold intensity (τ_{g97}) is computed as the 97^{th} percentile intensity of the green component in all pixels. This means that the pixels in the image with the green component intensity below this intensity constitute around 97% of the pixels. Also, a minimum gray level intensity condition ($t_{min} = 40$) is applied. All pixels with gray level intensity greater than t_{min} and having green pixel component greater than ($g97$) constitute the foreground region while the rest constitute the background region [5]. The main reason that both $g97$ and $g100$ generate proper binary images for this domain is that the images are captured under green light. Figure 3 (d-f) shows some of the result binary images for this method for the original images in Figure 3 (a-c).

2) Maximum Green Intensity Threshold ($g100$): Maximum green intensity threshold is similar to the 97^{th} percentile green intensity threshold described earlier. In this method, the maximum intensity of green component (τ_{g100}) is used as the threshold intensity for green component. All pixels with gray level intensity greater than t_{min} and having green pixel component equal to (τ_{g100}) constitute the foreground region. Figure 3 (g-i) shows some of the result binary images for this method.

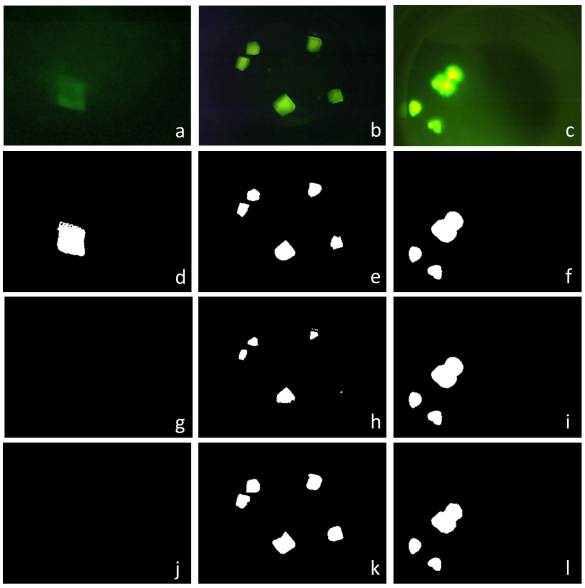

Figure 3. Binarization results of different techniques: (a-c) original images,(d-f) $g97$, (g-i) $g100$, and (j-l) $R-Howe$.

3) Document Binarization using Laplacian Energy ($R-Howe$): This is an automated document binarization method using Laplacian energy [28], [21]. This technique tries to minimize the global energy function which depends on the Laplacian of the image as well as edge discontinuities information using Canny edge operator. Since this technique was proposed for document binarization, it is hard to get proper results without any pre-post-processing on the image. Before we apply this method to our dataset, we negate our samples, since our images have black background. When we binarize our negative image, we observe a frame effect at the border of the image. We remove those artifacts from binary images. Interestingly, this method produced proper binary images for 56% of our images. Figure 3 (j-l) shows some of the resulting binary images for this method. Since we reverse (or negate) the image and apply preprocessing, we will refer this adapted method as ($R-Howe$) in the rest of the paper.

IV. Proposed Method: Super-thresholding

Binarization techniques are usually constructed based on some assumptions which may or may not be suitable for every image on a dataset. Almost every thresholding technique fails under some specific circumstances, and usually there is a better alternative to that technique in the literature [2]. It is observed that some techniques may generate better results for some images while others do a better job for other images. Our main goal was to exploit the powerful features of different binarization methods and use them whenever they perform well.

A. Building the Training Set

Since super-thresholding uses supervised classifiers before image binarization, we should generate a training set for building a model. After running available thresholding techniques, the labeling can be done manually with assistance of domain experts for all images in the dataset. For instance, if one protein image is binarized more accurately with the $R-Howe$'s method we labeled that image as "1;" if the best method is $g97$, we labeled the image as "2." If the best method is $g100$, we labeled the image as "3." Such a training set is satisfactory to build the model. In addition, we have manually identified the correct regions of foreground to generate the actual ground-truth binarized images. We use these ground-truth images to quantify how effective the thresholding algorithms are. Since the ground-truth images are available, the labels of images are generated automatically using the correctness measurement provided in Section IV-C.

B. The Framework of Super-thresholding

Once the images are labeled, we examined the features of the images and analyzed if there is a relationship between some features of the image and the thresholding techniques. After trying some basic features such as mean, standard deviation of intensity, autocorrelation of the images, we noticed that some of these features can be informative to establish the relations between protein images and thresholding techniques. For instance, in our previous study, we concluded that if the standard deviation of the image is less than 12.86, $g90$ thresholding method usually generates the best results. Similarly, if the standard deviation is more than 40.22, Otsu's method produces the most promising results [29].

Presence of a relation between image features and thresholding methods encouraged us to automate the detection of this relation. Thus, we decided to employ supervised classifiers (Bayesian classifier (BYS), Decision Tree ($ID3$), Random Forest (RF), and Artificial Neural Network classifiers (ANN)) in order to construct a training model [30]. Since the classification process is sensitive to the factors such as data type or distribution, we examined 4 classifiers having different characteristics. If we categorize the methods, Bayesian is a probability based classifier, Random Forest is an ensemble classifier, Decision Tree is a rule based classifier, and finally Neural Networks is a powerful classifier particularly for non-linearly distributed data. We intend to determine the one that offers the best classification results for our dataset.

Super-thresholding can binarize fast compared to complex segmentation methods. Figure 4 provides a general overview of super-thresholding. As shown in the figure, super-thresholding consists of four main stages: preprocessing stage, training stage, testing stage, and binarization stage. In the preprocessing stage, the dataset is labeled by an expert. Later, the dataset is divided into training and test sets. In the training stage, a classifier model is built using the features extracted from images. Feature extraction is done by two approaches called "priori" and "posteriori". Either of these approaches can be used in the feature extraction stage based on the preference. Classification model is trained based on the features coming from the preferred approach. Table I presents the features used in this paper for both approaches.

1) Priori Approach: In the priori approach, the features are extracted from original images only. Any type of feature extracted such as the mean intensity, standard deviation, etc. from the original image can be included in this approach. This approach is relatively fast for feature extraction, since no information is extracted from the output binary images.

2) Posteriori Approach: The posteriori approach requires running all thresholding methods to extract features. When all thresholded images are generated, they are mapped to the original images. Then foreground, background, inner and outer pixels of the object regions are detected (see Figure 5). Later, a set of statistical features are extracted from these regions to feed classifiers (see FS_5 in Table I). This approach is less efficient than the priori approach due to the necessity of all binary images for feature extraction, however, it can easily be parallelized, since each thresholding method can be run independently.

The main idea behind the posteriori approach is that inner and outer boundary regions can be used as an indicator whether a thresholded image is an accurate binary image or not. Normally, we expect a significant intensity change between inside and outside of the objects. Therefore,we both dilate and erode image using 5x5 structuring element to obtain information around the boundary pixels of the foreground as in Eq. 1 and 2:

$$F_{out} = I\_Bin \oplus S = \bigcup_{s \in S} I\_Bin_s \tag{1}$$

$$F_{in} = I\_Bin \ominus S = \bigcap_{s \in S} I\_Bin_{-s} \tag{2}$$

where I_Bin is the input binary image and S is the structuring element. Figure 5-f shows the total region that we focus around boundary.

Once we extract features from the dataset using either the priori or posteriori approach, we are able to generate classifier model in training stage. We use the same features for classifying test images to determine the best thresholding method. In order to evaluate the correctness of binary images, we compared the results with ground-truth binary images generated by our research group. Our evaluation with respect to the ground-truth binary images is explained in Section V.

Super-thresholding offers a generic solution to any image binarization problem. It provides a framework that does not

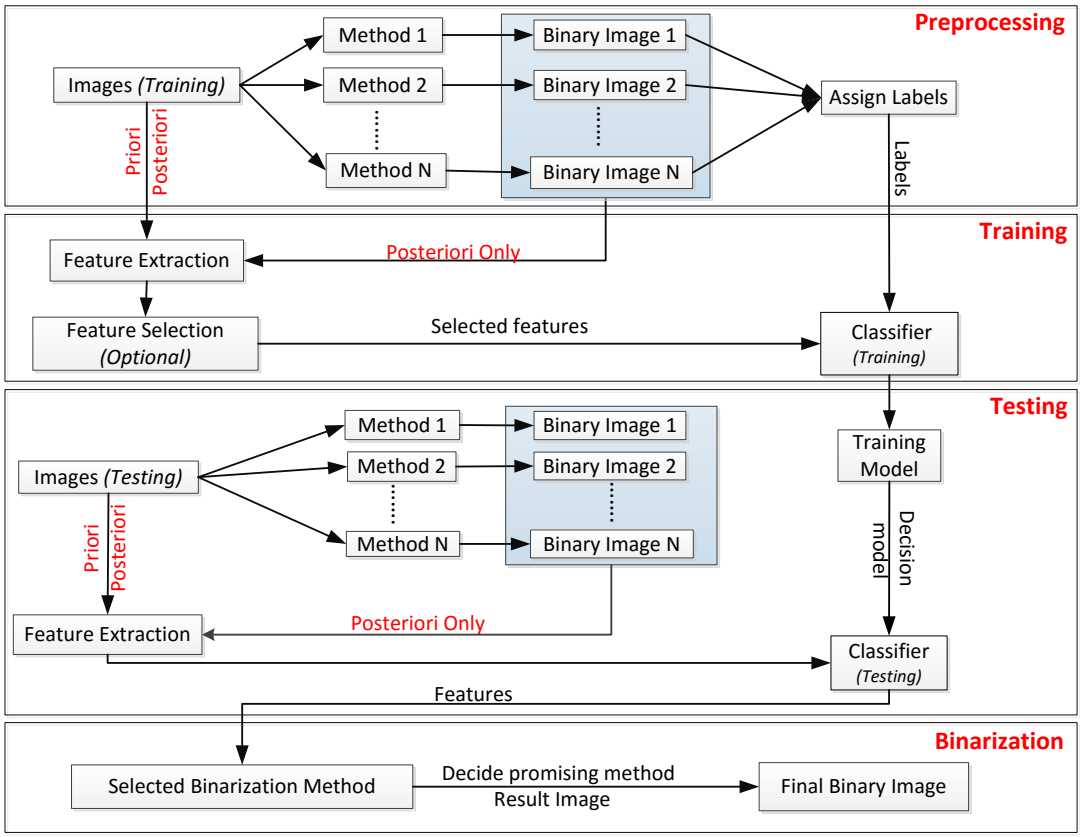

Figure 4. The framework of Super-thresholding.

depend on a specific binarization technique. It can easily be modified to a new domain by changing the chosen binarization techniques and re-training the system. Different sets of features can also be included in the system if they are more informative in that domain. All these characteristics make super-thresholding a flexible and practical framework in image binarization area.

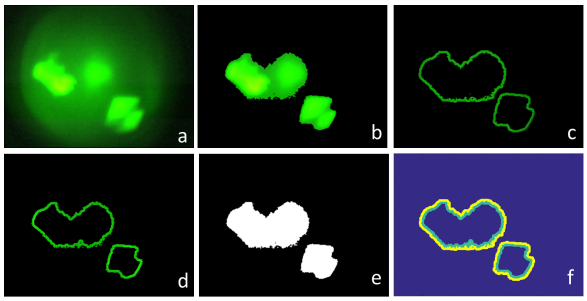

Figure 5. Posteriori feature extraction: a) original image, b) foreground image, c) outer pixels, d) inner pixels, e) thresholded image, and f) inner and outer boundaries of foreground (b)

Alternatively, the binary image results could be combined or fused using a weighted sum for the final decision. However, in our experiments we noticed that this idea did not yield satisfactory binary images, since in many cases, only one method provides the correct result while all other methods fail (see Figure 7). Moreover, the way of assigning weights to each method is not obvious and it may cause biased decision towards higher weighted method even though it may fail.

Since super-thresholding is independent from the problem domain, we preferred not to mention the application details of super-thresholding to our protein crystallization problem so far. We think it will make more sense to the reader when the method and problem is separated. In the following sections, we first explain how we measure the performance of thresholding methods and how we applied our method for protein crystal image domain.

C. Correctness Measurement

It is usually a subjective task to evaluate the results of the binarization process. Since a simple visual comparison of each binary image would not provide objective and dependable results, in this study, we decided to generate reference (ground-truth) binary images of all protein images in our dataset. We have manually extracted the protein instances using an image editing software [31] that has the capability of auto selection of objects on the image. Once

the rough object region is selected by the software, domain experts manually edit the borders for fine level corrections.

Once the reference images are ready, it is possible to calculate the correctness of any binary image by comparing with the reference image. We take an output binary image (generated by a binarization method) and the corresponding reference binary image, then measure the similarity between two images using "weighted sum" of the images. Suppose the pixels of protein instances (foreground) are represented by "1," and the background area is represented by "0" in a binary image. When we multiply reference binary image by 2 and sum with the output binary image, we have the sum image, which can represent all the pixels on the image as correctly classified or misclassified. Following equation shows this idea:

$$I_S = 2 \times I_R + I_O \tag{3}$$

where I_S, I_R, and I_O are the sum image, reference binary image, and the output binary image, respectively. The sum image includes 4 regions. We can easily refer these regions as True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). If the value of pixel p_{ij} on the sum image is "3," it is a TP where both output image and reference image have foreground pixel. If the pixel value is "2," it is a FN. Similarly, if the pixel value is "1," it is a FP. Finally, if the pixel value is "0," it is a TN. Figure 6 presents a sample sum image and its 4 regions.

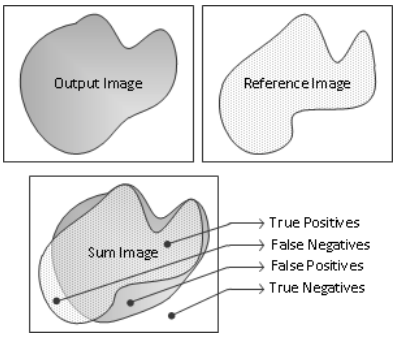

Figure 6. Sum image.

We use TP, TN, FN, and TN to measure the correctness of an output binary image. In the literature, there are several measures that offer correctness measures from different perspectives. It is often a significant factor to select a proper measure that is more relevant to the characteristics of the problem. For example, the classical accuracy measure may not be a proper measure for our study. Because in a typical protein binary image, there are usually very few number of foreground pixels compared to the background pixels. In other words, TN pixels can easily suppress the accuracy even if there are no TP pixels. In order to avoid bias towards a specific measurement method, we tried 4 well-known measures: Accuracy (ACC), F-Score ($F1$) [32], Matthews's correlation coefficient (MCC) [33], and Jaccard similarity ($JACC$) [34]. Using a variety of correctness measures, we aim to provide reliable results.

D. Application of Super-thresholding for Protein Crystal Image Binarization

Super-thresholding is a good solution if there are several binarization techniques, where none of them can produce accurate results for all images, but there are some techniques that generate acceptable binary image for a portion of dataset. Thus, protein image binarization problem is a convenient application area for our method. In this problem, there are thresholding methods that generate good results for some images but not for all.

In our previous work [29], we used only three thresholding techniques and one classifier (Decision Tree) using only one statistical feature. In this study, we extend the number of techniques and features to see whether new methods and features can improve the accuracy of the results. We generated 4 different feature sets (FS) to test the performance of the priori approach and 1 feature set for the posteriori approach.

Table I shows brief descriptions and formulas of the features where $I_{Gray}, I_{Green}, F, B, F_{in}$, and F_{out} represent gray level image, green channel of original image, foreground image, background image, inner boundary image, and outer boundary image, respectively. i, j, and k represent indices of the corresponding set or image. In addition, G represents the set of connected graphs of the canny edge image, and l_i represents the length of the i^{th} line in the set of lines, L, extracted from the edge image. In the beginning, we extracted 17 histogram features [35] and 12 edge features [36] in our dataset. These features were tested and they generated satisfactory results in our earlier studies [36] and [5]. However, to reduce the number of features we applied 2 feature selection methods in priori approach experiments. We used Random Forest feature selection in the first 3 feature sets. The first feature set (FS_1) contains a subset of histogram and edge features. It has 1 edge feature, 4 texture features and 1 histogram feature. For FS_2 and FS_3, we selected 5 of the histogram features and 6 of the edge features, respectively. In FS_4, we selected 6 of 29 combined features using minimal-redundancy-maximal-relevance criterion (mRMR) feature selection method [37]. Finally, we extracted 6 statistical features for each binary image in FS_5 using posteriori approach.

V. Experiments & Results

In our experiments, 3 different thresholding methods ($g97$; $g100$; $R - Howe$) and 4 classifiers (bayesian classifier (BYS), decision tree ($ID3$), random forest (RF), and artificial neural networks (ANN)) are evaluated to binarize protein crystal images. We have run our experiments using

Table I
DEFINITIONS OF FEATURES FOR PRIORI AND POSTERIORI APPROACH.

	Feature Name	Description	Formulas
FS_1	$H(X)$[2]	Measures of vertical (given formula) and horizontal auto-correlation of gray level co-occurrence matrix	$H(X) = -\sum p(x_i) log p(x_i)$
	$\sigma\left(I_{Gray}\right)$	Standard deviation of the gray level image	$\sigma\left(I_{Gray}\right) = \sqrt{\frac{\sum_{(i,j)\in I_{Gray}} \left(I_{Gray}\left(i,j\right) - \mu\left(I_{Gray}\right)\right)^2}{\left\|I_{Gray}\right\| - 1}}$
	r_k[2]	Measure of horizontal and vertical auto-correlation of gray level co-occurrence matrix	$r_k = \frac{\sum_{(i,j)}(I_{Gray}(i,j)-\mu(I_{Gray}))(I_{Gray}(i-k,j)-\mu(I_{Gray}))}{\sum_{(i,j)}(I_{Gray}(i,j)-\mu(I_{Gray}))^2}$
	$\hat{L}$	Sum of all edge lengths in the canny edge image	$\hat{L} = \sum_{i\in L} l_i$
FS_2	r_k	Measure of horizontal auto-correlation of gray level co-occurrence matrix	$r_k = \frac{\sum_{(i,j)}(I_{Gray}(i,j)-\mu(I_{Gray}))(I_{Gray}(i-k,j)-\mu(I_{Gray}))}{\sum_{(i,j)}(I_{Gray}(i,j)-\mu(I_{Gray}))^2}$
	$\mu\left(I_{Gray}\right)$	Average intensity level of the grayscale image	$\mu\left(I_{Gray}\right) = \frac{\sum_{(i,j)\in I_{Gray}} I_{Gray}\left(i,j\right)}{\left\|I_{Gray}\right\|}$
	$\sigma\left(I_{Gray}\right)$	Standard deviation of the gray level image	$\sigma\left(I_{Gray}\right) = \sqrt{\frac{\sum_{(i,j)\in I_{Gray}} \left(I_{Gray}\left(i,j\right) - \mu\left(I_{Gray}\right)\right)^2}{\left\|I_{Gray}\right\| - 1}}$
	k	Measure of peakedness of the histogram of the gray level intensity of the image	$k = \frac{\sum_{(i,j)\in I_{Gray}} \left(I_{Gray}\left(i,j\right) - \mu\left(I_{Gray}\right)\right)^4}{\left(\left\|I_{Gray}\right\| - 1\right)\left(\sigma\left(I_{Gray}\right)\right)^4}$
	$H(X)$	Measure of horizontal spatial disorder or spatial randomness of gray level co-occurrence matrix	$H(X) = -\sum p(x_i) log p(x_i)$
FS_3	$\|G\|$	Number of connected edges (lines) in the edge image	$\|G\|$
	$\tilde{G}$	Number of graphs with perpendicular edges in the canny edge image	$\tilde{G} = \sum \perp(G_k)$ where $\perp(G_k) = \begin{cases} 1 & \exists l_i \in L_k \text{ and } \exists l_j \in L_k \text{ and } 70 \le \alpha(l_i, l_j) \le 90 \\ 0 & otherwise \end{cases}$
	$\mu(\mathbf{L})$	Average length of all edges in the canny edge image	$\mu(L) = \frac{\sum_{i\in\mathbf{L}} l_i}{\|\mathbf{L}\|}$
	$\hat{L}$	Sum of all edge lengths in the canny edge image	$\hat{L} = \sum_{i\in\mathbf{L}} l_i$
	$\bar{\bar{G}}$	Sum of all edge lengths in the graphs with no perpendicular edges	$\bar{\bar{G}} = \sum_{i\in L_k} l_i$ where $\perp G_k = 0$
	$max(\mathbf{L})$	Length of the longest edge in the canny edge image	$\max_{1\le i\le\|\mathbf{L}\|}(l_i)$
FS_4	$H(X)$[2]	Measures of vertical (given formula) and horizontal autocorrelation of gray level co-occurrence matrix	$H(X) = -\sum p(x_i) log p(x_i)$
	k	Measure of peakedness of the histogram of the gray level intensity of the image	$k = \frac{\sum_{(i,j)\in I_{Gray}} \left(I_{Gray}\left(i,j\right) - \mu\left(I_{Gray}\right)\right)^4}{\left(\left\|I_{Gray}\right\| - 1\right)\left(\sigma\left(I_{Gray}\right)\right)^4}$
	l_o	1 if $\eta_p > 0$, 0 otherwise	$l_o = \exists l_i \in L_k$ and $\exists l_j \in L_k$ and $70 \le \alpha(l_i, l_j) \le 90$
	η_c	Number of graphs whose edges form a cycle	$\eta_c = \|G_i\|$, where G_i is cyclic graph
	η_{hc}	Number of Harris corners	[38]
FS_5		**For each binary image, following features are extracted:**	
	$\mu(F)$	Mean intensity of foreground region	$\mu(F) = \frac{\sum_{(i,j)\in F} I_{Green}(i,j)}{\|F\|}$
	$\sigma(F)$	Standard deviation of foreground region	$\sigma(F) = \sqrt{\frac{\sum_{(i,j)\in F}\left(I_{Green}(i,j) - \mu(F)\right)^2}{\|F\| - 1}}$
	$\mu(B)$	Mean intensity of background region	$\mu(B) = \frac{\sum_{(i,j)\in B} I_{Green}(i,j)}{\|B\|}$
	$\sigma(B)$	Standard deviation of background region	$\sigma(B) = \sqrt{\frac{\sum_{(i,j)\in B}\left(I_{Green}(i,j) - \mu(B)\right)^2}{\|B\| - 1}}$
	$\mu(F_{in})$	Mean intensity of inner pixels of the foreground region	$\mu(F_{in}) = \frac{\sum_{(i,j)\in Fin} I_{Green}(i,j)}{\|F_{in}\|}$
	$\mu(F_{out})$	Mean intensity of inner pixels of the foreground region	$\mu(F_{out}) = \frac{\sum_{(i,j)\in Fout} I_{Green}(i,j)}{\|F_{out}\|}$

MATLAB 2014b on a 16GB 3.4GHz Quad-Core CPU (excluding pylon experiments). For random forest classifier, we used the source code [1] that is published by Jaiantilal et al. We set the number of trees for random forest classifier as 500, and square root of the total number of features is selected as the number of candidate features at one node of a decision tree [39]. In addition, we use MATLAB built-in neural network toolbox with two layers. The hidden layer has $n-1$ nodes where n is number of features in the dataset. Super-thresholding technique is compared with some other thresholding methods ($g97$; $g100$; $R-Howe$; Chuang, 2011; Silva, 2011; and Otsu's method [6]).

A. Protein Crystal Dataset

Our dataset consists of 170 protein crystal images of size 320 $\times$ 240, and all images have been captured by using Crystal X2 of iXpressGenes, Inc. We labeled the dataset with 3 different thresholding techniques such that 29% of them were labeled as $g100$, 15% of them were labeled as $g97$, and 56% of them were labeled as $R-Howe$. In addition, as given in the introduction, none of these methods has completeness ratio of 100%. Our calculations show that the completeness ratio of $R-Howe$'s method, $g97$, and $g100$ are calculated as 83%, 40%, and 70%, respectively. In order to evaluate the size of the training set, we train our model with 25%, 50%, and 75% of the data, respectively. The remaining are reserved for testing.

B. Results

In our previous study [29], we had a relatively small dataset and only 3 thresholding methods ($g90$, $g100$, and $Otsu$) were available. When we extend the dataset and supply more thresholding methods to the system, we obtained the best results using 3 methods ($g97$; $g100$; $R-Howe$), and we removed the methods that do not contribute to the overall performance. We generate 5 different feature sets to evaluate the performances of priori and posteriori approaches on super-thresholding. The first four feature sets (i.e., FS_1, FS_2, $FS3$, and FS_4) in Table I were used to test the priori approach. FS_5 was used to evaluate the posteriori approach. Visual results for 3 sample images are given in Figure 7, which clearly shows the superiority of super-thresholding over other methods.

In order to evaluate the performance of the methods, we performed a comprehensive experimental setup. We tested the super-thresholding for 3 different training set sizes, 4 correctness measures, and 5 feature sets. For each case, we repeated our experiments 5 times to avoid biased results. Table III shows the $mean$ values of different correctness measures. According to the table, super-thresholding gives the best results using Bayesian classifier on feature set FS_5 (posteriori approach) regardless of the training set size. Our super-thresholding achieved ACC=0.99, $F1$=0.86, MCC=0.87, and $JACC$=0.77 on the average (highlighted bold in the table). These results are also the best results in overall experiments. Although the results of each training set size seem to be very similar, they cannot be directly comparable, since varying test set sizes also affect the results. Thus, it is more reasonable to look at the improvement over the best thresholding method for each training set size. The improvements over the best method ($R-Howe$) are $86.2\% - 81.0\% = 5.2\%$, $86.2\% - 78.6\% = 7.6\%$, and $85.5\% - 75.1\% = 10.4\%$ using the $F1$ measure for training sizes of 25%, 50%, and 75%, respectively. Nonetheless, the experiments show that 25% training set size could achieve very satisfactory results.

According to the results, the posteriori approach gives higher accuracy than the priori approach. The priori approach yields best results using FS_1 set. The $F1$ measures using Bayesian and random forest classifiers for FS_1 are calculated as 0.811 and 0.805, respectively. Considering the feature extraction efficiency of the priori approach, these results are also significant for real time systems. Employing only histogram (FS_2) or edge (FS_3) features does not improve the performance significantly. Similarly, FS_4, which is generated from both histogram and edge features using mRMR, did not improve performance as well. However, FS_4 provides very close to or slightly higher than $R-Howe$ method. In order to compare super-thresholding with our previous study "DT-Binarize" [29], we repeated the experiments for 3 different training sizes. The results also show that super-thresholding following the posteriori approach outperforms DT-Binarize around 5-6% in terms of $F1$ measure. These results show that including new features, thresholding methods, and classifiers improves the binarization accuracy.

Classficiation Accuracy. Considering only the classification accuracy might be misleading in our problem. In Table II, we provide a sample confusion matrix of the best experiment discussed above (Bayesian classifier on FS_5 using 75% training data). According to this table, the classification accuracy of the experiment is 83.3%. However, the classification accuracy is not a major indicator in this problem, since the actual labels of images are considered based on only the highest $F1$ measure. For example, for an image I, assume that $F1$ measures are $F1_{g97}$=0.865, $F1_{g100}$=0.678, and $F1_{R-Howe}$=0.854. Based on this information, actual class label of the image I will be $g97$. However, if the system selects $R-Howe$ method for that image, it is also acceptable in terms of thresholding. Thus, this table may not be a proper performance indicator. Giving higher weight to a thresholding method may not improve the accuracy as well since there are cases where one method is the only one that generates the correct binarized image.

Soundness and Completeness. Another important issue

[1] https://code.google.com/p/randomforest-matlab/

Table II
SAMPLE CONFUSION MATRIX OF THE EXPERIMENT USING FS_5 AND BAYESIAN CLASSIFIER.

		Actual		
		$G100$	$G97$	$R-Howe$
Predicted	$G100$	9	1	2
	$G97$	1	6	0
	$R-Howe$	2	1	20

about the binarization of protein crystal images is the soundness and completeness. It is very likely to generate improper binary images due to illumination or reflection problems. For some cases, binary images may have minor problems, which are acceptable for this problem domain unless it affects the performance of the system that will use these results. However, it is possible to have complete black or white images for some of the binarization methods if the image has a blurred or a very bright large sized object. This causes the system to miss those crystals in the analysis, which cannot be acceptable. We also evaluate the binarization methods in this aspect. According to our results, super-thresholding gave the best accuracy, and it also did not generate any unacceptable results for our dataset with Bayesian classifier on feature set FS_5 and Bayesian classifier on feature set FS_1 as long as the problematic images (mentioned in Section V-A), which all thresholding methods failed were not in the test set. Using Bayesian classifier, super-thresholding generally generated the best results in our experiments. Moreover, super-thresholding for these sets has generated unacceptable binary images for only 4% of the dataset (when problematic images are included in the test set), while $R-Howe$'s method generated improper binary images for 21% of the dataset. As we stated before, generating proper binary images is as important as the overall accuracy.

Performance Upper Bound Analysis. The performance of classification to select the best technique depends on the success of the binarization methods that are selected for the problem domain. This means that there is a practical limit of the performance of super-thresholding. In other words, if none of the selected methods are able to generate a proper binary image for a specific image, super-thresholding does not produce accurate binary image, as well. Figure 7 shows sample cases where each method fails. We computed the upper bound by selecting the best 3 thresholding methods for each image and compared with our results. In Table III, the last row shows the upper bound for each correctness measure. Correctness measures of the upper bound are calculated using the best binarization method for all images. Results of super-thresholding are within 97.3% ($0.765 \div 0.786$) of the upper bound for Bayesian classifier using 75% of training data with respect to the Jaccard coefficient.

Time Analysis. We have also evaluated the run-time performance of super-thresholding on a 3.40GHz Intel i7 Quad Core 16GB RAM system using 320x240 images. In the Table IV, we provided the timings of feature extraction, classification, and binarization for an image in milliseconds. According to the table, the feature sets having more edge features take more time than the others (i.e., FS_3 consists of only edge features). Once the classifier model is built, an image can be binarized in 133 milliseconds using BYS on FS_2 (the priori approach), and in 385 milliseconds using BYS on FS_5 (the posteriori approach), and these timings are feasible for our system.

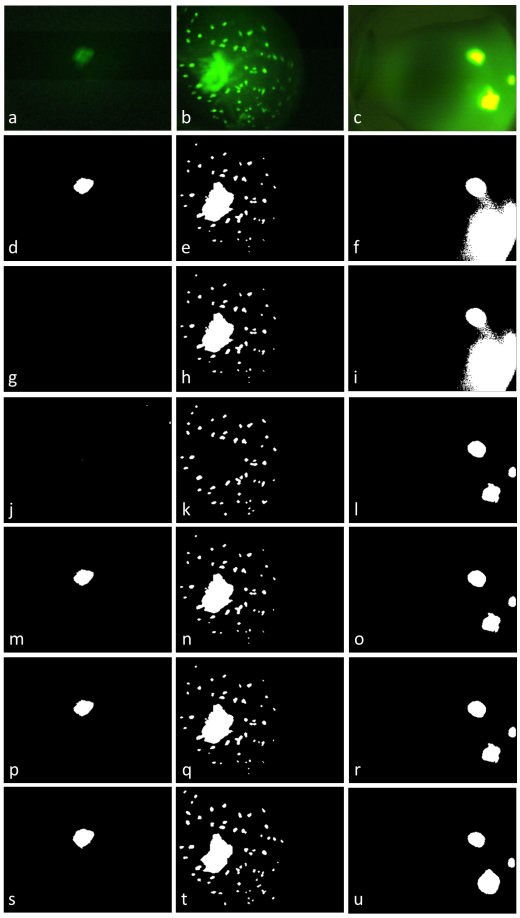

Figure 7. Results of super-thresholding: (a-c) original images; (d-f) $g97$, (g-i) $g100$, (j-l) $R-Howe$, (m-o) super-thresholding priori, (p-r) super-thresholding posteriori, and (s-u) ground truth images.

C. Discussion

Comparison of Performance Measures. In this problem domain, the accuracy (ACC) is not a distinctive measure since the number of true negatives is significantly more than the number of true positives. Thus, we consider $F1$, MCC, and $JACC$ measures more significant than accuracy to measure correctness, because our focus is crystal regions of the images. Among these coefficients, Jaccard coefficient has a simple and easily interpretable value. It is equivalent to the

Table III
CORRECTNESS MEASURE RESULTS OF THE EXPERIMENTS FOR EACH FEATURE SET AND CLASSIFIER

Training Size	25%				50%				75%			
	ACC	$F1$	MCC	$JACC$	ACC	$F1$	MCC	$JACC$	ACC	$F1$	MCC	$JACC$
$g100$	0.980	0.718	0.741	0.615	0.977	0.700	0.726	0.599	0.971	0.663	0.691	0.563
$g97$	0.981	0.771	0.789	0.661	0.979	0.761	0.781	0.652	0.972	0.727	0.752	0.614
$Otsu$	0.899	0.634	0.663	0.550	0.900	0.634	0.663	0.551	0.880	0.589	0.623	0.508
$R-Howe$	0.985	0.810	0.815	0.725	0.984	0.786	0.792	0.701	0.985	0.751	0.756	0.671
Silva, 2011	0.973	0.630	0.660	0.495	0.973	0.620	0.652	0.486	0.971	0.596	0.629	0.465
Chuang, 2011	0.968	0.697	0.717	0.564	0.968	0.690	0.710	0.559	0.969	0.669	0.691	0.534
Dinc et al. [29]	0.975	0.818	0.828	0.725	0.980	0.811	0.821	0.720	0.973	0.790	0.801	0.701
BYS, FS_1	0.985	0.825	0.832	0.735	0.985	0.832	0.838	0.740	0.986	0.811	0.817	0.720
$ID3, FS_1$	0.984	0.824	0.833	0.732	0.984	0.815	0.823	0.725	0.985	0.798	0.806	0.709
RF, FS_1	0.985	0.829	0.836	0.739	0.984	0.823	0.829	0.733	0.986	0.805	0.811	0.718
ANN, FS_1	0.981	0.766	0.786	0.662	0.978	0.726	0.749	0.625	0.972	0.706	0.730	0.606
BYS, FS_2	0.985	0.824	0.830	0.736	0.985	0.833	0.838	0.743	0.986	0.812	0.817	0.723
$ID3, FS_2$	0.985	0.820	0.827	0.727	0.982	0.793	0.801	0.704	0.983	0.762	0.769	0.677
RF, FS_2	0.985	0.833	0.839	0.744	0.984	0.823	0.829	0.736	0.985	0.774	0.778	0.691
ANN, FS_2	0.981	0.781	0.797	0.679	0.979	0.757	0.774	0.659	0.972	0.709	0.732	0.609
BYS, FS_3	0.985	0.802	0.812	0.712	0.983	0.796	0.805	0.708	0.984	0.768	0.778	0.677
$ID3, FS_3$	0.982	0.786	0.797	0.695	0.981	0.766	0.778	0.674	0.983	0.737	0.747	0.650
RF, FS_3	0.984	0.799	0.808	0.711	0.982	0.770	0.780	0.682	0.984	0.736	0.746	0.652
ANN, FS_3	0.982	0.741	0.761	0.638	0.978	0.714	0.737	0.612	0.972	0.688	0.713	0.588
BYS, FS_4	0.985	0.811	0.818	0.723	0.984	0.803	0.808	0.718	0.985	0.765	0.770	0.684
$ID3, FS_4$	0.984	0.808	0.815	0.717	0.987	0.800	0.808	0.711	0.984	0.767	0.775	0.678
RF, FS_4	0.985	0.815	0.821	0.729	0.984	0.800	0.806	0.714	0.985	0.750	0.757	0.669
ANN, FS_4	0.981	0.750	0.768	0.650	0.978	0.729	0.748	0.630	0.975	0.688	0.708	0.595
BYS, FS_5	**0.992**	**0.862**	**0.867**	**0.774**	**0.992**	**0.862**	**0.866**	**0.774**	**0.991**	**0.855**	**0.859**	**0.765**
$ID3, FS_5$	0.987	0.833	0.841	0.744	0.989	0.840	0.845	0.751	0.985	0.807	0.812	0.721
RF, FS_4	0.988	0.845	0.850	0.758	0.985	0.842	0.847	0.756	0.986	0.824	0.828	0.740
ANN, FS_5	0.981	0.768	0.786	0.664	0.977	0.772	0.790	0.671	0.973	0.743	0.765	0.639
Max-Limit	0.993	0.888	0.890	0.809	0.993	0.884	0.886	0.804	0.992	0.870	0.873	0.786

ratio of common areas to the union of regions in both images (input and ground truth). For example, 0.5 as Jaccard coefficient indicates that the common (or overlapping) regions is half of the union of regions with respect to the ground-truth. The method $g100$ has an average Jaccard coefficient around 0.563. This is actually a low value; however, we still cannot discard it as it gave the best result in 29% of our experiments.

Performance of Thresholding Methods. In terms of thresholding methods, we would like to have a proper thresholding method for each image. That is the major criteria for selecting thresholding methods. If we compare $g97$ and $g100$ methods, $g100$ works better when the foreground is separated better than the background. In protein crystallization images, we expect protein crystal regions to have the highest intensity. Whenever the protein crystal regions have higher intensity than other regions, $g100$ works fine. Large 3D crystals are usually distinguishable in terms of intensity and have higher intensity than other regions. $g100$ works best for images containing large 3D crystals. Since crystals float in a solution, the depth of crystals from the microscope may differ. Only crystals at the depth-of-field appear in focus. Other crystals may be blurred and may have lesser intensity than crystals in focus. In those cases, $g100$ may not provide good binarization. Whenever the foreground intensity is not high, the sizes of crystals are smaller, and crystals appear at different depths, the $g97$ method is likely perform better than the $g100$ method. R-Howe's method has three components: minimizing global energy for labeling pixels, use of Laplacian to distinguish ink from the background, and use of edge detection to handle discontinuities. The edges are critical factors on separation of crystals. The straight boundaries of crystal regions are one of the important indicators for a crystal. For regions with clear boundaries, $R-Howe$ generally provides better results. If the intensity is lower or image is blurred, $g97$ may be preferred. The advantage of $g100$ is that it can easily remove the background since any pixel with low intensity is considered as the background.

Performance of Classifiers. The Bayesian classifier works slightly better than decision tree and random forest classifiers. The artificial neural networks performed worst among

Table IV
TIMINGS OF FEATURE EXTRACTION, CLASSIFICATION, AND BINARIZATION METHODS[1].

Category	Method	Time per image (milliseconds)
Binarization	$g100$	110.500
	$g97$	108.900
	Otsu	12.400
	$R-Howe$	130.000
	Silva, 2011	25.000
	Chuang, 2011	83.000
Training	BYS	25.67
	$ID3$	11.313
	RF	100.3488
	ANN	2596.005
Testing	BYS	0.097
	$ID3$	0.006
	RF	0.051
	ANN	0.005
Feature Extraction	FS_1	48.800
	FS_2	3.190
	FS_3	399.900
	FS_4	443.800
	FS_5	35.700

[1] The total running time of an experiment is calculated by adding the times of feature extraction, testing, and binarization stages. For example, in priori approach, if the selected method is $R-Howe$ using BYS on FS_2, the total time of binarization for an image will be $130+0.097+3.190=133.287$ milliseconds. However, in posteriori approach, the total time of the binarization will be $110.5+108.9+130+0.097+35.7=385.197$ milliseconds using BYS on FS_5. The training timings are calculated using 75% of the dataset as training. Please note that in posteriori approach we extract features using the output of all thresholding methods.

them. The biggest challenge for building the decision model is the thresholding methods performing almost the same for some images. While labeling we choose the best one (with the highest $F1$ measure) even though it may be slightly better than the second best one. Neural networks could not learn this difference as others and make mistakes on similar ones. Bayesian classifier is resilient to noise and less affected by thresholding methods having similar performance for an image. The decision tree is also affected by similar performing thresholding methods. Random forest performs slightly better than the decision tree but its performance is lower than Bayesian classifier for the posteriori approach. Random forest may overfit the training data, and hence its performance may be lower for the test data.

Performance of Feature Sets. The feature sets for FS_1, FS_2, FS_3, and FS_4 are used for the priori approach. FS_3 contains mostly edge related features and performed worst among these feature sets. Relying only edge related features is not satisfactory for this domain. FS_1 and FS_2 containing texture-related features perform similarly due to the similarity between feature sets. FS_2 slightly outperforms FS_1. Note that FS_2 has histogram related feature and does not have edge related features. This difference between FS_1 and FS_2 has a positive impact on the accuracy for FS_2. FS_4 was generated using mRMR feature selection method. Although FS_4 performs better than FS_3, it does not perform as well as FS_1 and FS_2. It looks like features based on intensity statistics is important for the accuracy. The feature set for the posteriori approach performs best among all feature sets. Although FS_5 relies on intensity features, it performed better than any other feature set. Comparison of pixels in the foreground and background as well as the comparing pixels at the boundaries of regions are better features for analyzing the performance of thresholding methods.

VI. CONCLUSION & FUTURE WORK

This paper provides a new generic framework to combine different kinds of thresholding techniques using a supervised classifier. A classifier model is constructed using some image features such as autocorrelation, standard deviation, etc. of the protein crystallization images that are labeled by the experts. The labels (or classes) of images correspond to a binarization method which is proper for the image. We select a binarization method for a given test image using the same classifier, and we apply the selected method to the protein crystallization image to generate binary image.

In this paper, we include 3 different thresholding techniques to our classifiers, and we compared our method with 7 different thresholding techniques (provided in the first 7 rows of Table III) in order to evaluate performance of the super-thresholding. Knowing the performance of individual thresholding techniques is helpful to understand how much improvement can be made with the supervised approaches. We concluded several results at the end of this study:

1) Single thresholding techniques may not be enough for some of the datasets that have poorly illuminated, noisy and unfocused images,
2) Using the posteriori approach, super-thresholding provided the best performance for Bayesian classifier on FS_5 with $F1$=0.86, MCC=0.87, and $JACC$=0.77 on the average for our dataset. These results are very close to the upper bound.
3) Super-thresholding can be considered the best in terms of soundness and completeness since it generated more proper binary images for protein crystal images than any other method,
4) Super-thresholding improved accuracy around 10%, and 6% compared to the best single thresholding method for Bayesian classifier using FS_5 and FS_1 with 75% of training data, respectively,
5) $R-Howe$'s thresholding technique which is proposed for the document binarization shows the best performance among the other thresholding techniques, but

it generated improper binary images for 21% of the dataset on the average,

6) The success of super-thresholding depends on the success of the thresholding techniques which are selected for the problem domain, and its success also depends on performance of the classifier,
7) Super-thresholding did not produce satisfactory results on neural network classifier,
8) Using only edge or histogram features did not improve the accuracy,
9) Since super-thresholding produces single binary image using only a few simple features of the images for the priori approach, it is feasible for most of real-time classification systems, and
10) The posteriori feature extraction approach of super-thresholding can be easily parallelized, since each thresholded image can be generated independently.

We evaluated the performance of our approach with 4 different accuracy measures to have more reliable results. For most cases, our method outperformed other single thresholding methods. Moreover, super-thresholding reaches 97.3% $(0.765 \div 0.786)$ of the upper bound with respect to Jaccard coefficient.

It is difficult to generalize or verify the soundness and completeness based on the algorithmic approaches involved in developing the thresholding methods. Expert opinion is usually needed to determine the correctness (or soundness) of a thresholding method. When thresholding techniques are used in automatic analysis systems, incorrect thresholding may lead improper decision making. Therefore, completeness is a critical factor in our domain. Another issue is regarding to the choice of the best thresholding method. When building the training set, a number of methods generated good results for a specific image. In those cases, we again selected the best one using the ground-truth images although the second-best is as good as the first one. This significant similarity between methods for some images make the training difficult. This is the reason why we have not reached the optimal model. In future work, we plan to apply super-thresholding on other domains. Moreover, additional features can be extracted by comparing the output binary images and the original images. As future work, these features can be used to build a more advanced model to build a classifier and check how much they improve the accuracy. In the future, deep learning can be applied to binarization of images when there are enough ground-truth binary images available.

VII. Acknowledgement

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

References

[1] R. Gonzalez and R. Woods, *Digital Image Processing*. Pearson/Prentice Hall, 2008.

[2] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–168, 2004.

[3] A. McPherson and J. A. Gavira, "Introduction to protein crystallization," *Acta Crystallographica Section F: Structural Biology Communications*, vol. 70, no. 1, pp. 2–20, 2014.

[4] M. Sigdel, M. Sigdel, S. Dinc, I. Dinc, M. Pusey, and R. Aygun, "Focusall: Focal stacking of microscopic images using modified harris corner response measure," *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

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

[6] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[7] J. Zhang and J. Hu, "Image segmentation based on 2d otsu method with histogram analysis," in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 6, Dec 2008, pp. 105–108.

[8] M. P. de Albuquerque, I. Esquef, A. G. Mello, and M. P. de Albuquerque, "Image thresholding using tsallis entropy," *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1059 – 1065, 2004.

[9] J. Kapur, P. Sahoo, and A. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 273 – 285, 1985.

[10] G. Johannsen and J. Bille, "A threshold selection method using information measures," in *ICPR*, vol. 82, 1982, pp. 140–143.

[11] W. Oh and W. Lindquist, "Image thresholding by indicator kriging," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 7, pp. 590–602, Jul 1999.

[12] S. Shaikh, A. Maiti, and N. Chaki, "Image binarization using iterative partitioning: A global thresholding approach," in *Recent Trends in Information Systems (ReTIS), 2011 International Conference on*, Dec 2011, pp. 281–286.

[13] S. H. Shaikh, A. K. Maiti, and N. Chaki, "A new image binarization method using iterative partitioning," *Machine vision and applications*, vol. 24, no. 2, pp. 337–350, 2013.

[14] W. Niblack, *An introduction to digital image processing*. Strandberg Publishing Company, 1985.

[15] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern recognition*, vol. 33, no. 2, pp. 225–236, 2000.

[16] S. G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," *Image Processing, IEEE Transactions on*, vol. 9, no. 9, pp. 1522–1531, 2000.

[17] N. Ray and B. Saha, "Edge sensitive variational image thresholding," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 6, Sept 2007, pp. VI – 37–VI – 40.

[18] M.-C. Chuang, J.-N. Hwang, K. Williams, and R. Towler, "Automatic fish segmentation via double local thresholding for trawl-based underwater camera systems," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, Sept 2011, pp. 3145–3148.

[19] J. Sauvola, T. Seppänen, S. Haapakoski, and M. Pietikäinen, "Adaptive document binarization," in *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, vol. 1. IEEE, 1997, pp. 147–152.

[20] B. Su, S. Lu, and C. L. Tan, "Combination of document image binarization techniques," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 22–26.

[21] N. R. Howe, "Document binarization with automatic parameter tuning," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 3, pp. 247–258, 2013.

[22] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[23] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.

[24] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[25] V. Lempitsky, A. Vedaldi, and A. Zisserman, "Pylon model for semantic segmentation," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 1485–1493.
[26] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 898–916, 2011.
[27] A. Silva, "Region-based thresholding using component tree," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, Sept 2011, pp. 1445–1448.
[28] N. Howe, "A laplacian energy for document binarization," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, Sept 2011, pp. 6–10.
[29] I. Dinç, S. Dinç, M. Sigdel, M. S. Sigdel, M. L. Pusey, and R. S. Aygün, "Dt-binarize: A hybrid binarization method using decision tree for protein crystallization images," in *Proceedings of The 2014 International Conference on Image Processing, Computer Vision & Pattern Recognition*, ser. IPCV'14, 2014, pp. 304–311.
[30] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
[31] J. L. Starks and A. Fehl, *Adobe Photoshop CS6: Comprehensive*, 1st ed. Boston, MA, United States: Course Technology Press, 2012.
[32] Y. Sasaki, "The truth of the f-measure," *Teach Tutor mater*, pp. 1–5, 2007.
[33] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.
[34] G. Chowdhury, *Introduction to modern information retrieval*. Facet publishing, 2010.
[35] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, no. 6, pp. 610–621, 1973.
[36] M. Sigdel, M. S. Sigdel, İ. Dinç, S. Dinç, R. S. Aygün, and M. L. Pusey, "Chapter 27 - automatic classification of protein crystal images," in *In Emerging Trends in Image Processing, Computer Vision and Pattern Recognition*. Morgan Kaufmann, 2015, pp. 421–432.
[37] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1226–1238, 2005.
[38] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Citeseer, 1988, p. 50.
[39] 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.

İmren Dinç is a Ph.D. student at Computer Science Department, University of Alabama in Huntsville, and working as a GRA since 2013. She received her B.S. degree in computer engineering from Dokuz Eylul University, Turkey in 2012 with first rank in the department and an honor degree. She is currently studying experimental design for protein crystallization experiments and visualization. Her research areas are data mining, image processing, and pattern recognition.

Semih Dinç is a Ph.D. student at Computer Science Department in University of Alabama in Huntsville since 2012. He received his B.S. degree in Computer Science at Dokuz Eylul University in 2004, and his M.S. degree in Control Engineering at Yildiz Technical University in 2012. His research areas are computer vision, image processing, and pattern recognition. He is recently studying vision based trajectory tracking system for mobile robots.

Madhav Sigdel received the Bachelor's Degree in Computer Engineering from Pulchowk Campus, Kathmandu, Nepal in 2008, and the M.S. and Ph.D. degree in Computer Science from the University of Alabama in Huntsville in 2012 and 2015, respectively. His Ph.D. thesis is related to real-time protein crystallization image analysis. His research interests include data mining, computer vision, multimedia systems, and information visualization.

Madhu S. Sigdel received his Bachelor's Degree in Computer Engineering from Institute of Engineering, Kathmandu, Nepal in 2011, and a M.S. degree in Computer Science from the University of Alabama in Huntsville in 2014. His M.S. thesis is related to focal stacking and autofocusing of microscopic images. His research interests include image processing, computer vision and data mining.

Marc L. Pusey is a research scientist working at iXpressGenes, Inc., Huntsville Alabama. Dr. Pusey received his Ph.D. in Biochemistry from the University of Miami, Fl., then did postdoctoral research at the University of Minnesota. He moved to Huntsville in 1985, and worked at NASA/MSFC for the next 23 years in the new field of protein crystal growth. After retirement from NASA, he obtained his current position at iXpressGenes with the focus of his research being improved methods for protein crystal screening, crystal detection, and crystallization condition identification using visible fluorescence methods.

Ramazan S Aygün: received the B.S. degree in computer engineering from Bilkent University, Ankara, Turkey in 1996, the M.S. degree from Middle East Technical University, Ankara in 1998, and the Ph.D. degree in computer science and engineering from State University of New York at Buffalo in 2003. He is currently an Associate Professor in Computer Science Department, University of Alabama in Huntsville. His research interests include protein crystallization image analysis, data mining, image and video processing, spatio-temporal indexing and querying, multimedia databases, semantic computing, multimedia networking, and multimedia synchronization.