Region of Interest Identification in Breast MRI Images

Jason Forshaw
jasonlforshaw@gmail.com

Ryan Volz
rvolz@stanford.edu

***Abstract*—Identifying the region of interest in a breast MRI image is a tedious manual process that would be much more suited for a computer, but it is necessary for performing useful post-processing on the image for breast cancer research and treatment. We devise such an approach following the concept that an MRI image can be sectioned into a number of contiguous regions based on pixel intensity clusters and edge detection. We then train an SVM algorithm to identify the specific sections that belong in the region of interest. Although our approach is not yet ready to replace hand-drawn regions of interest, the technique shows promise for eventually being able to do so.**

I. Introduction

In breast cancer research, magnetic resonance imaging (MRI) is being explored as a potential tool for monitoring tumor size and assessing response to chemotherapy. Currently, researchers and clinicians often manually interpret MRI images, but as MRI gains wider adoption in the clinical realm, the need for fast and robust automated image processing techniques is becoming clear. In order to be able to process the raw images and extract the required data, it is first necessary to identify the region of interest (ROI) of the image. Roughly, the ROI should include all fibroglandular (breast) and fat tissue but exclude other portions of the image such as the skin, chest wall, and image artifacts. As of now, identification of the ROI is done manually by drawing a single closed contour that encompasses the breast tissue and fat. The purpose of this work is to apply machine learning techniques in order to automate this process, in particular classifying one section of an MRI image as the ROI.

As MRI advances, the number of 2-D images collected during an exam increases; within our data set, the number of images per scan has increased over time from 44 to 152 images. With hundreds of images per MRI scan and multiple scans necessary per patient during the course of treatment, considerable time must be invested into identifying the ROI by hand. Another benefit of this project is standardization of the ROI. With the current manual process, different research groups and even different people within the same group may identify the ROI of a given image differently. This is not from lack of agreement on what should be included in the ROI, but rather it is from different interpretations of the how the contour should "best" be drawn.

The data we have used for this research, which includes 692 breast MRI images from 11 MRI scans and the accompanying hand-labeled ROI, was provided by Dr. Nola Hylton and the Breast MRI Research Program at the University of California, San Francisco (UCSF). The images have a resolution of 512x512 pixels and are in grayscale format with pixel intensity represented by a 16-bit integer, while the labeled ROIs have been provided as MATLAB data files consisting of a sequence of points which are intended to be drawn as the interpolation points of a closed Bezier curve.

II. Approach

In our view, the most critical step in applying machine learning to this problem is the identification of appropriate features. The "raw" data in each image basically consists of two attributes for each pixel: intensity and location. If we were to treat each image and its accompanying per-pixel binary ROI labeling as a single element of our training set, which might seem like a natural thing to do at first, the feature vector would have $512 * 512 = 2^{18}$ elements each taking on one of 2^{16} values while the number of possible labels would be $2^{2^{18}}$. Clearly another approach, one which extracts from this raw data only the "important" information, is necessary.

A. Drawing a ROI

Our first step in trying to determine how to make this data more amenable to machine learning was to think about what information we, as humans, look for when drawing the ROI. At a high level, the first thing we do when we look at an image is to pick out the approximate region we are interested in by identifying the breast portion of the image. We are looking for a distorted semi-circle shape, and the information crucial to accomplishing this task is rather coarse. The majority of the breast tissue is composed of approximately uniform intensity gray pixels and is surrounded on one side by black pixels representing empty space and on the other side by a more varied mixture of grays and whites representing the chest cavity (see Figure 1 for an example). Basically all the information that is necessary comes from identifying sets of pixels with approximately the same intensity.

Once the approximate region is identified, the information needed to actually draw a region of interest changes from coarse sets of pixel intensity to specific intensities for individual pixels. By taking the coarse view initially, we have effectively already decided that the pixels in the center of the approximate region are in the ROI. So if we want to draw a closed boundary separating the ROI from the rest of the image, we are necessarily interested in finding edges in the breast region. We do this by looking for sharp gradients in

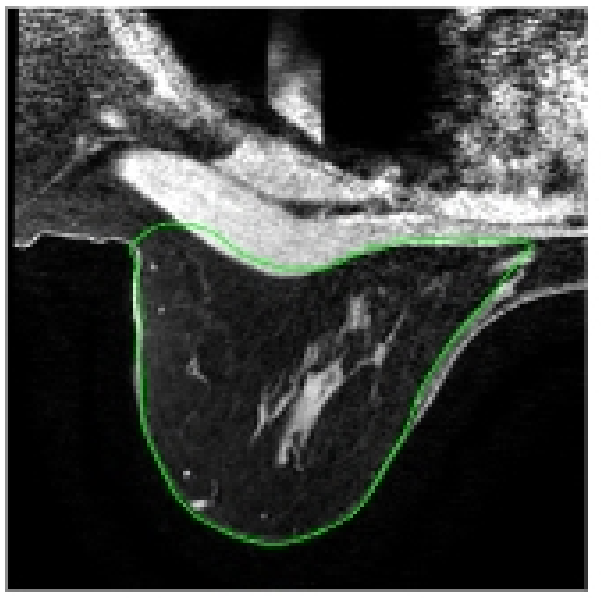

Fig. 1. MRI image showing the hand-drawn ROI boundary in green. Because the boundary is marked as a set of points and drawn as a Bezier spline curve, it includes portions of the skin (lighter pixels, left edge of boundary) and chest wall (lighter pixels, top of boundary) that would not normally be in the ROI.

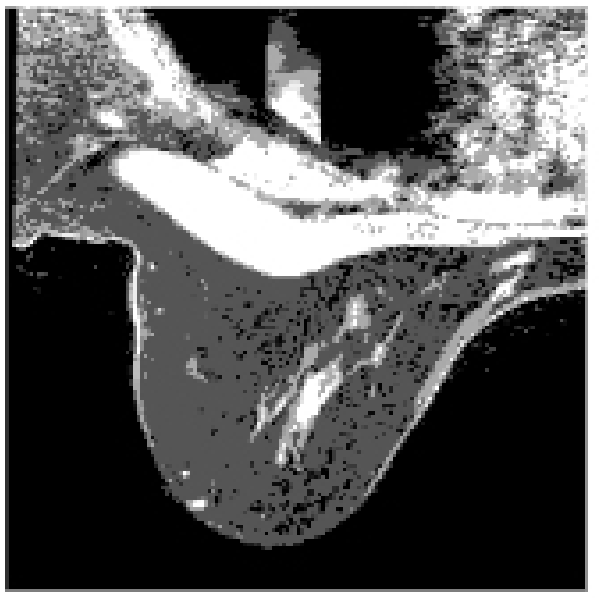

Fig. 2. MRI image clustered into four regions, shown in black, dark gray, light gray, and white.

pixel intensity. Once we have identified those, drawing the ROI is a simple matter of following the edges that lie near the boundary of the breast region and interpolating between them in places where no edges exist.

B. Data Reduction

With this breakdown of the steps for manually drawing the ROI, we see that only a small fraction of the pixel information is necessary in order to accomplish the task. Specifically, instead of classifying individual pixels as being in the ROI or not, we could section the image based on approximate pixel intensity and edge detection, then classify the resulting regions. Clustering the pixels into a small number of groups based on intensity would naturally divide the image into contiguous regions of pixels belonging to the same cluster. Edge detection on the original image could then be used to further split those regions at any boundaries that the clustering may have missed. This way, the boundary for the ROI would be likely to coincide with the boundaries between some of the regions. Such an approach extracts the important information–edges that could potentially make up the ROI boundary–from the raw image data. It is much easier to assign features to a few hundred regions and train or classify based on them than it would be to do the same with each of the 2^{18} individual pixels.

III. Procedure

The procedure we used to automatically identify the ROI of a given image follows from the concept of sectioning the image into regions. Each image in our data set includes both the left and right breast, and though the eventual goal would be to identify the ROI for each breast, for development of this procedure we decided to work with only half of the image, corresponding to either the left or right breast and depending on the availability of hand-drawn ROI data. Extending the procedure to include the entire image would be straightforward.

A. Clustering

The first step is to cluster the pixels by intensity, assign them a new value based on the cluster they belong to, and store the resulting image. We employed k-means clustering using MATLAB's *kmeans* command to group the pixels into four clusters, roughly corresponding to black, dark gray, light gray, and white in the original image.

To find the cluster centroids, we first shrink the image so that it is 128 pixels in height, then run k-means clustering on the smaller image with cluster centroids initialized to be evenly spaced over the range of pixel intensities. The reduced size allows the k-means clustering to converge much more quickly, and the resulting centroids are then used to initialize clustering on the full image. In both cases, we used the "cityblock" (L1-norm) metric, which in MATLAB for 1-D data places the cluster centroids at the median value of the clusters, rather than the mean. We found that with this metric, the resulting images more closely resembled the originals. Note that the initial clusters are not randomized in any way, which means they might not converge to the global optimum. We chose to do this so that the results for each image would be exactly reproducible every time the procedure is run, and in any case the resulting centroids work well.

Once the clusters have been obtained, we create a new image that condenses the intensity range down to the number of clusters. An example of the resulting image, which can be compared with the original of Figure 1, is shown in Figure 2.

B. Edge Detection

The second step toward sectioning the image involves finding edges. To do this, we use the Canny edge detection algorithm, available in MATLAB's Image Processing Toolbox. Under the default threshold settings, use of this algorithm via the *edge* command easily identifies all of the major edges, which are of most interest to us, plus many minor edges that are potentially helpful. One can see in Figure 3 that for our example image, the algorithm finds the boundaries between the breast tissue and skin and between the breast tissue and

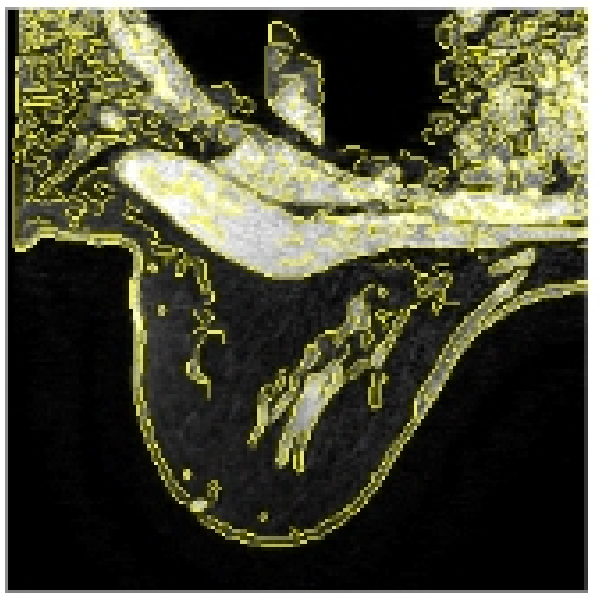

Fig. 3. MRI image with edges shown in yellow.

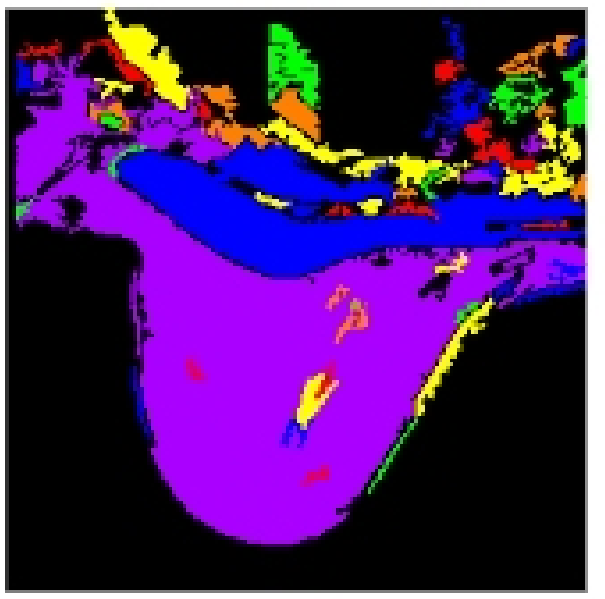

Fig. 4. MRI image divided into regions according to clustering and edge detection. Regions are shown in different colors.

chest wall, which are precisely the edges we need for creating the ROI boundary.

C. Region Identification

With the clusters and edges, we are now ready to section the image into contiguous regions. Of most use for this task is MATLAB's *regionprops* command, which can take a black and white image, identify regions of connected pixels, and calculate a number of properties pertaining to those regions such as area and centroid location. Thus, for each cluster, we create a black and white image with pixels belonging to the cluster shown in white. Then to ensure that no regions cross any edges in the original image, we force all pixels belonging to the edges to be black, regardless of their cluster. In order to avoid processing many small regions that will be inconsequential for identifying the ROI, we discard any groups of connected white pixels which have an area below a certain threshold (30 pixels). We do this separately for each cluster and combine the results into a new clustered image with black now representing pixels that do not belong to any sizable region. Representing the fact that the final ROI will be contiguous without any holes, we fill in any black sections of the resulting image that are completely surrounded by clustered pixels.

Once again extracting black and white images identifying pixels in each cluster, we then run the *regionprops* command specifying a pixel connectivity of four, which means that a pixel will only be considered part of a connected region if it touches that region on any of its four sides and not just at its corners. Collecting the regions for each cluster gives the regions for the entire image. The pixels belonging to each region, the region's area, and its centroid are all stored for later use. Finally, in order to reduce the number of regions that the supervised learning algorithm must work with, we discard regions that we assume are extremely unlikely to end up in the ROI. Namely, we discard any regions with an area greater than one third of the entire image area and any regions with their centroid lying in the top third of the image. This has the effect of discarding the large region of black surrounding the main image and many of the regions located well within the chest cavity. This step is not strictly necessary, but it saves computation time and does not hurt accuracy. The result of this processing for our example image is shown in Figure 4.

D. Region Attributes

Now that the image data has been organized into at most a few hundred regions, it is feasible to perform training and classification on those regions. The question then becomes: what attributes can we assign to these regions so that the training and classification works well? Obviously the location of the region is important, as is information about its intensity. For location information, we include in the feature vector the region's geometric centroid given as an (x,y) coordinate pair from the upper left corner of the image. For intensity information, we include the number of the cluster to which the region's pixels belong. To allow for classification of images of different sizes and perhaps a different number of clusters, we also normalize these values by the image dimensions and number of clusters, respectively. Other information could be included in the feature vector, but we found that including other data that was readily available, such as region area, did not improve classification accuracy.

E. Region Labeling

In the case when we have a ROI to train on, it is necessary to label the regions according to their inclusion in the ROI. Starting with the ROI boundary spline as shown in Figure 1, we first identify all of the pixels that it contains. Then for each region that we've identified, we label it in the affirmative if it meets either of the following requirements: at least a fraction of its pixels (0.5) are in the ROI, or at least a minimum number of its pixels (30) are in the ROI and those pixels account for at least a certain fraction (0.05) of the region's pixels. This way, any regions that contain a significant portion of the ROI's pixels are included, while regions that may contain some ROI pixels but otherwise lie substantially outside the ROI are not included. We place more emphasis on the former because there are frequent instances when portions of the ROI boundary correspond to no discernible edge in the image. This