Learning to detect lesion boundaries in breast ultrasound images

Pavel Kisilev*, Ella Barkan*, Greg Shakhnarovich** and Asaf Tzadok*

*IBM Research - Haifa, Israel, **TTI-Chicago

Abstract. This paper presents a novel method for automatic lesion detection in breast ultrasound images; the method performs multi-stage learning of lesion-specific boundaries represented by a bag of robust features. The proposed method can be seen as an edge pruning procedure that leaves only object-specific edges and filters out the rest. It can be combined with segmentation algorithms that rely on edge information. We show an example of such combination with one of the state-of-art segmentation algorithms; our method yields improved segmentation results. The proposed method is tested on a set of 400 breast ultrasound images, with the goal to automatically detect lesion boundaries. However, we believe that our method can be used by radiologists as an assistance tool during examination routine, in which case it may help to better localize lesions and document the findings. The performance of our method is compared to a state-of-art object boundary classification algorithm; we show that our method outperforms it in different tests.

Keywords: Automatic lesion detection, breast ultrasound image, boundary probabilities, bag of features, cascade classification

1 Introduction

Object detection and segmentation is an active area of research in computer vision, and has various applications in the medical imaging domain. Many object segmentation methods rely on object boundary detection as the first step of an algorithm. One of the most common methods for this first step of edge detection is the Canny edge detector [1], which usually provides consistent results. On the other hand, one of the the current state-of-art in the edge detection is '*gPb*-owt-ucm', the probability of boundaries contour detection and segmentation method of [2] which is trained to detect edges in natural images, and then to combine the edges into segments.

Object boundaries in natural images are usually well defined, and, therefore, objects can be clearly distinguished from their background. In contrast, in medical images, objects such as lesions or even organs do not always have fully closed contours with clear boundaries. The reasons for that include limitations of acquisition methods and devices (for example, having low resolution and/or low dynamic range), occlusions of other tissues and organs, and others. In these cases, standard segmentation methods may produce segments that

have no physical meaning. In particular, such methods merge neighboring areas if they are 'similar' enough, or if there are no sufficiently strong boundaries detected between the areas. In these cases a better strategy would be to detect most probable object boundaries, by learning their typical properties from available examples, and then merge them into closed segments by some ad-hoc approach. We therefore present a new such approach to lesion-specific boundary detection; it is based on multi-stage cascade learning. We apply our approach to the problem of automatic detection of lesions in breast ultrasound images.

The paper is organized as follows. In the next section we give a detailed description of the proposed method. Then, we present various experiments where we compare the performance of our method with state-of-art methods, and conclude with the summary.

2 The proposed method

In the next subsection we give a high-level overview of the proposed approach, and then provide more details in the following subsections.

2.1 Overview of the method

The proposed method of learning-based localization of objects can be summarized as follows. First, an edge detection procedure is applied to an image. Edge detection can be performed by any known approach, for example, Canny edge detector, or more sophisticated methods that consider not only intensity changes but also other image characteristic changes, such as color and texture as in [2]. The edges obtained in the first step are used as an initial boundary guess. Next, boundary descriptors are calculated for individual pixels on the edges (an extension to the case of group of pixels is described below). Given a set of *training* images with object boundary ground truth available (e.g., manually created in advance), we collect positive and negative examples of edge pixel descriptors. Based on these examples, we train a cascade classifier and use the learned models to classify boundaries with a score that reflects the likelihood of a particular boundary or its section to belong to an object in question. An example of the output of our method in the case of lesion detection from breast ultrasound (US) is shown in Figure 1, right. Finally, we may want to aggregate the resulting positively classified boundaries' sections into closed segments or areas. As in the case of initial edge detection, the last step can utilize one of the known contour aggregation methods such as ObjCut [3], or Ultrametric contour map of [4], which we use in our experiments.

The contribution of our method is twofold: 1) we propose a new description of lesion boundaries represented by a feature set which is robust to various changes in object appearance as a result of variability in characteristics of image acquisition devices, and 2) we present a new cascade-type classifier which is trained to detect lesion-specific boundary sections and to merge these boundary sections at each consecutive classification stage.

2.2 Feature channels and initial edge detection

Our goal is to train the system to detect object specific boundaries. We start with description of the boundary feature channels. We generate additional channels from a given image as follows. Suppose the image contains only intensity channel, such as in ultrasound or X-ray. If an image has three color channels, such as in colonoscopy, we can use them in a similar way. Using the intensity channel, the 1^{st} feature channel, we calculate the following additional channels that describe various textural characteristics and local directional coherence features. The first additional, that is the 2^{nd} feature channel, is the texton-based texture map as used in [2]. The next three channels are based on the local entropy of the pixel neighborhood [5], calculated at three scales (window sizes of 7x7, 15x15, and 35x35). These channels characterize well textures at various resolutions and thus encode multiscale texture properties. The 6^{th} feature channel is the phase congruency from [6]. This channel provides fine edge information that, in our case, represent a complementary description of textures in an image, and encode local directional characteristics.

The first step in our method is an edge detection; it needs to find most of the object boundaries, including the weak ones, even in expense of many noisy detections (which will be filtered out by our learning-based method). We experimented with two different approaches to perform the initial edge detection. In the first approach, we used the state-of-art probability of boundaries algorithm from [2]. The second approach is quite simple: we applied Canny edge detector to each one of the six channels described above, and took a pixel-wise maximum of the six resulting edge maps from those six channels. Notice that the first method is trained to detect edges in natural images. We found empirically that the second approach was better suited to medical images, as it found consistently more weak edges than the first approach[1].

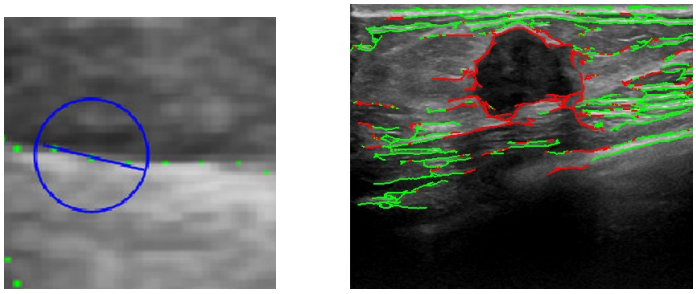

Fig. 1. Left: an example of sampled points on boundaries from the 1^{st} stage of classifier, and of a disk ROI around one such point. Right: final edge classification for the whole image obtained at the 8^{th} cascade classification stage. Red edges classified as object-related, green edges classified as non-object related.

[1] In general, it is possible to train the method in [2] on medical images; we leave it for the future research

2.3 Boundary descriptors

We now turn to the definition of boundary descriptors. The boundary map from the edge detection step contains separate *edge sections* (or, edge segments). For each such section, we take every m^{th} pixel with coordinates (x,y) on the corresponding boundary section. We then place a circular disc of radius R at (x,y), and split it into two halves according to the major boundary direction (see illustration in Figure 1, left). We found that sampling each 20^{th} pixel on boundaries and a disk of $R=20$ yields good results. Given the two halves of the disk for a given pixel (x,y), for each one of the six information channels we calculate two histograms of channel values obtained from both halves of the disk. We write it compactly as a list of histogram bins with non-zero probability, i.e.

$$\{(z_i, p_i^{k,ch})\}_{i=1}^{n}$$

where z_i is a bin-center, $p_i^{k,ch}$ is the corresponding probability mass for that bin for the k^{th} probability distribution ($k = 1, 2$), corresponding to the k^{th} half, $ch = 1...6$ is the channel number, and n is the number of such bins. Note that we have fixed both the number of bins and the bin-centers across the two distributions. This standard technique is quite prevalent in object recognition, for example in the Bag of Visual Words technique (see e.g. [7]).

Further, for each one of the six channels, we calculate the D-Field pairwise descriptor which was recently proposed in [8]. In particular, for bin i of the first distribution D-Field δ_i is defined by

$$\delta_i = \sum_j f_{ij}(z_j - z_i)$$

where f_{ij} is the *flow* matrix between two distributions obtained as a by-product of computation of the Earth Mover's Distance (EMD) [9]. Intuitively, the D-Field descriptor captures – for each bin – where its probability mass moves.

In general terms, the D-Field descriptor captures the relationship or transformation between two probability distributions of two halves of the disk. One of its advantages argued in [8], is its robustness to various changes in image appearance that may occur as a result of different acquisition devices having very different characteristics such as dynamic range, resolution, and others.

Finally, the bag of boundary features for a pixel on the boundary with coordinates (x,y) and a corresponding disk around it comprises of: 1) two histograms (for each half of the disk) per each of the six channels, 2) D-Field descriptors per each channel, 3) the EMD distance between the two distributions (for each half of the disk) per each channel, and 4) the coordinates (x,y). Overall, it yields $(6 \times 3 \times n) + 8$ where n is the number of bins in the histograms.

2.4 Cascade classifier of boundary features

In this section, we describe the structure of the proposed cascade classifier[2] As we mentioned above, we first identify separate edge sections where each pixel

[2] We describe main ideas, omitting some technical details because of the lack of space.

(or, alternatively, each m-th pixel) is a center of the local neighborhood that has a disk shape. After we calculated a descriptor for each one of the pixel areas, we train the 1^{st} classification stage.

Training. Given a set of training images with marked inner and outer object boundaries (the ground truth), we label pixels on the boundary as positive examples if they are situated in between those two contours, and as negative ones otherwise. A training set of 300 images yields quite a large amount of training samples - around 1.5 million - with the number of positives about 20 times smaller than negatives. Since the number of features is much smaller than the number of samples, it is more convenient and efficient to use a linear SVM classifier. We divide the full set of examples into two subsets; one is used in the odd classification stages, and the other one is used in the even stages. We start with training the 1^{st} classification stage on the 'odd' subset as follows: we use weighted cost that penalizes the errors in positive examples about 10 times higher than those in negative ones. This asymmetric cost assures that nearly all positive samples are correctly identified, in expense of large number of errors in the negative class. This is, however, exactly what we are up for, since we propagate the results from this 1^{st} stage to the next, 2^{nd} classification stage. In particular, we now use the model obtained from the 1^{st} classification stage to classify the examples in the 'even' subset. We then use all the positively classified samples to train the 2^{nd} stage model. This time, however, instead of taking separate edge pixel areas, we merge neighboring pairs of pixel areas whose centers are 20 pixels apart on the same edge section. In fact, we calculate the same descriptors described above, but using a neighborhood combined of two disks around the two closed-by pixels (centers of disks), and a 'corridor' between them. We train the 2^{nd} stage model on the 'even' subset, and use the obtained model to classify the examples from the 'odd' subset. Positively classified examples from the 2^{nd} stage are used then to train the 3^{rd} classification stage; at this stage, we use pairs of pairs of pixels. This hierarchical process continues till there are no boundary sections remaining that can be merged and classified. At each training stage we optimize the SVM parameters using 10-fold cross validation to prevent possible overfitting.

Testing. Having trained the hierarchy of models, the testing (that is, object boundary detection) is straight forward. As before, the 1^{st} step is the edge detection applied to an inquiry image. We then start from classifying individual edge pixels using the 1^{st} stage SVM model. Next, positively classified neighboring pixel areas are merged, and the pairs are classified using the 2^{nd} stage model, and so force.

3 Experiments

The system performance was tested on 400 ultrasound images collected from 4 different acquisition devices, with about 100 images from each device. Ground truth was created manually by a radiologist who marked inner and outer lesion contours. The two contours required to count for the uncertainty in cases of lesions without clear boundaries, and for the variability in radiologists' markings.

During the training stage we label edge sections found between inner and outer contours as positive examples, and all others as negatives.

We performed two types of tests evaluating the accuracy of the boundary detection. In the first, 'random subsampling' test, we divided the whole set randomly into training and testing subsets with roughly 300 and 100 images respectively. We then performed 50 such complete training and testing cycles for each such set, and averaged the performance of our system over 50 independent runs. This experiment provides figures of performance of the method when we train and test the system with images from the same device or a set of devices. This test verifies the robustness and adaptivity of the method. In the second, 'blind' test, we separated training and testing sets of images according to the acquisition devices. We then used images from three different devices for training and images from the other one for testing in a leave-one-out manner, and averaged the results over the 4 runs. It simulates a situation wherein the algorithm needs to perform in an environment with various devices whose images are not available during the training stage. This test verifies the ability of our method to generalize well to previously unseen images.

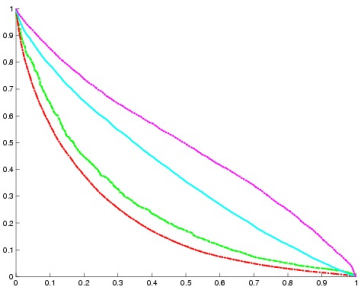

Fig. 2. False negative versus false positive errors: averaged ROC curves of lesion/non-lesion edge classification. Red and Green: our method in 'random subsampling' and in 'blind' tests respectively; Blue and Magenta: method of [10] in 'random subsampling' and 'blind' tests respectively.

Although there is extensive literature on the breast ultrasound (e.g., a survey in [11]), it is difficult to compare different methods, since algorithms and datasets are usually proprietary, and not available for use. We therefore compare the performance of our method with general segmentation and boundary detection methods. We test our method against object boundary classification method from [10]. This method was designed for natural images, but this is the only other approach that we are aware of, that learns object-specific *boundaries* of general objects (versus algorithms trained to detect specific objects like cars, faces etc., and use object specific features). In [10], the authors use a patch centered on the edge, rotated so that its x-axis is aligned with the edge tangent; color channel values from the patch are used as the local edge descriptor. Since our images have one intensity channel, we added additional channel of texture, to have a more 'fair' comparison. We trained a linear SVM classifier as done in [10], for each set of training images (described above), and applied the trained model for detection of

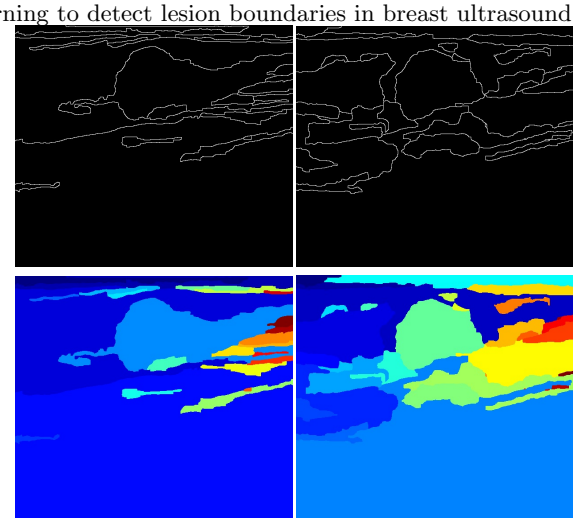

Fig. 3. Contours (upper) and segments (bottom): based on boundaries from [2] (left), and on the proposed method (right).

lesion boundaries in the testing set. We judge the performance of the competing methods by comparing their ROC curves (false negative versus false positive errors) obtained from classification of edges into lesion/non-lesion classes. The summary of these experiments is given in Figure 2; the ROC curves are averaged over either 50 or 4 runs, according to the test performed (as explained above). Obviously, simple descriptors from [10] cannot provide good detection of lesion boundaries since similar patches may appear anywhere in an image. In the case of random subsampling experiment, the classifier based on [10] succeeded to learn and separate 2 classes, though with high error; in the case of 'blind' test, it fails completely. In contrast, our method learns well from examples: in the random subsampling test, it yields 27% mean equal error rate; in the 'blind' test, our method shows its robustness and yield 33% mean equal error rate.

In another experiment, we compared our method to the state-of-art '*gPb*-owt-ucm' edge detection and segmentation method of [2]. We replaced the boundary detection method, known as the probability of boundaries 'gPb' algorithm, in '*gPb*-owt-ucm' method, with our boundary detection method, and used the contour based segmentation of 'owt-ucm' algorithm to combine detected boundaries into closed contours forming segments. Figure 3 shows an example of such comparison; the *gPb*-owt-ucm fails to localize correctly the lesion, while our method outlines nicely the important region, using the estimated probabilities of lesion boundaries.

4 Conclusions

In this paper we presented a novel method for automatic lesion detection in breast ultrasound images. The method is based on multi-stage learning of object-specific boundaries represented by a bag of robust features. The proposed method

can be combined with segmentation algorithms that rely on object edge information. We show an example of such combination of our method with one of the state-of-art contour based segmentation algorithms. Our method improves upon the results of the original algorithm.

The proposed method was tested on breast ultrasound images, with the goal to automatically detect suspicious lesion boundaries. However, we believe that the proposed method can be used by radiologists as an assistance tool during the examination routine. In this case, our method may help to better localize lesions and document the findings. Furthermore, since the method is general, we are planning to apply it to the problem of automatic lesion detection in mammogram images, and, possibly, in other modalities.

References

1. Canny, J.: A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on (6) (1986) 679–698
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(5) (2011) 898–916
3. Kumar, M.P., Ton, P., Zisserman, A.: Obj cut. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Volume 1., IEEE (2005) 18–25
4. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 2294–2301
5. Awate, S., Tasdizen, T., Whitaker, R.: Unsupervised texture segmentation with nonparametric neighborhood statistics. Computer Vision–ECCV 2006 (2006) 494–507
6. Kovesi, P.: Phase congruency: A low-level image invariant. Psychological research **64**(2) (2000) 136–148
7. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, Ieee (2007) 1–8
8. Kisilev, P., Freedman, D., Wallach, E., Tzadok, A., Naveh, Y.: Dflow and dfield: New features for capturing object and image relationships. In: Pattern Recognition (ICPR), 2012 21st International Conference on, IEEE (2012) 3590–3593
9. Rubner, Y., Tomasi, C., Guibas, L.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision **40**(2) (2000) 99–121
10. Prasad, M., Zisserman, A., Fitzgibbon, A., Kumar, M., Torr, P.: Learning class-specific edges for object detection and segmentation. Computer Vision, Graphics and Image Processing (2006) 94–105
11. Nover, A.B., Jagtap, S., Anjum, W., Yegingil, H., Shih, W., Shih, W., Brooks, A.D.: Modern breast cancer detection: A technological review. International Journal of Biomedical Imaging (2009)