Coarse-to-Fine Foraminifera Image Segmentation Using Markov Random Fields

***Abstract*—Foraminifera are single-celled marine organisms, which are usually less than 1 mm in diameter. One of the most common tasks associated with foraminifera is the species identification of thousands of foraminifera contained in rock or ocean sediment samples, which can be a tedious manual procedure. Thus an automatic visual identification system is desirable. Some of the primary criteria for foraminifera species identification come from the characteristics of the shell itself, and features like segmentation of chambers and apertures. This project describes solving one such problem using a sophisticated approach.**

***Index Terms*—Markov Random Fields, Image Segmentation, Foraminifera**

I. Introduction

One of the most common tasks associated with foraminifera is the species identification of thousands of foraminifera contained in rock or ocean sediment samples, which can be a tedious manual procedure. As discussed in [1], 3D and appearance-based features can be used to produce an edge probability map, which can be further refined to obtain a segmentation approach. The aim is to appropriately segment the images using the edge probability maps provided. We adopt a coarse-to-fine strategy to extract the vague edges in the images for segmentation using a relatively small training set. A coarse edge probability map is first predicted by properly fused features extracted from original input images and further refined by the features extracted from this coarse prediction. Finally segmentation is obtained by the post-processing on the refined edge map.

II. Data Set

This data set contains images from six widely used planktonic foraminifera species. We have been provided with a dataset of 320 sample images each of size 230*230 pixels. Each image has two files, one contains labeled segments and another, the probability map. Each sample has been derived from 16 images taken under different light source directions through a microscope with 30x magnification [1]. In this report, we propose a learning-based pipeline for image segmentation on a foraminifera data set.

III. Methodology

The schematic diagram for the methodology is shown in Figure 2.

- **Preprocessing:** We process all the probability maps by learning certain features from the given data set. To enable accurate edge detection, we learnt a minimum threshold i.e. 50, and a maximum threshold i.e. 230 for a pixel being an edge. Values above maximum threshold are guaranteed edges and values below the minimum threshold are guaranteed non-edges. These filtered probability maps are passed through a canny edge detector to generate an edge image which will be used for image segmentation.
- **Hidden Markov Random Fields (HMRF):** We employ the HMRF method to perform edge completion. This method uses K-means to generate initial edge prediction which are then maximized and optimized using Expectation Maximization. The Expectation Maximization iteratively tries to minimize the total posterior energy using Maximum a Posteriori estimation. This generates an edge-connected image on which we perform a post processing to get out final segments.
- **Watershed transformation:** The edge-connected image generated by HMRF model are segmented using watershed which uses 8-connected neighborhoods. The local segments generated by HMRF are transformed into bigger groups by watershed transform.

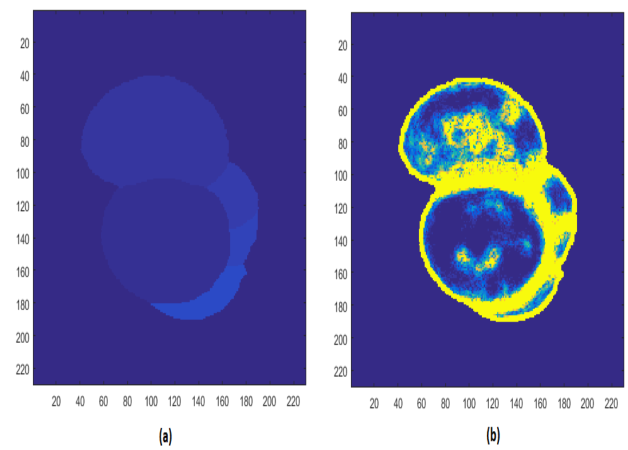

Fig. 1. Sample data image. (a) Shows the image with labeled segments. (b) Shows the probability map for the same image.

A. Preprocessing

The probability maps provided are coarse considering the edge labels. Each probability map is a 230x230 matrix where each cell represents a pixel value. The range of each pixel is [0, 255] which represent the probability of the pixel being an edge or not; 0 being least likely and 255 being most likely. Figure 3 shows a sample image with it's inner and outer

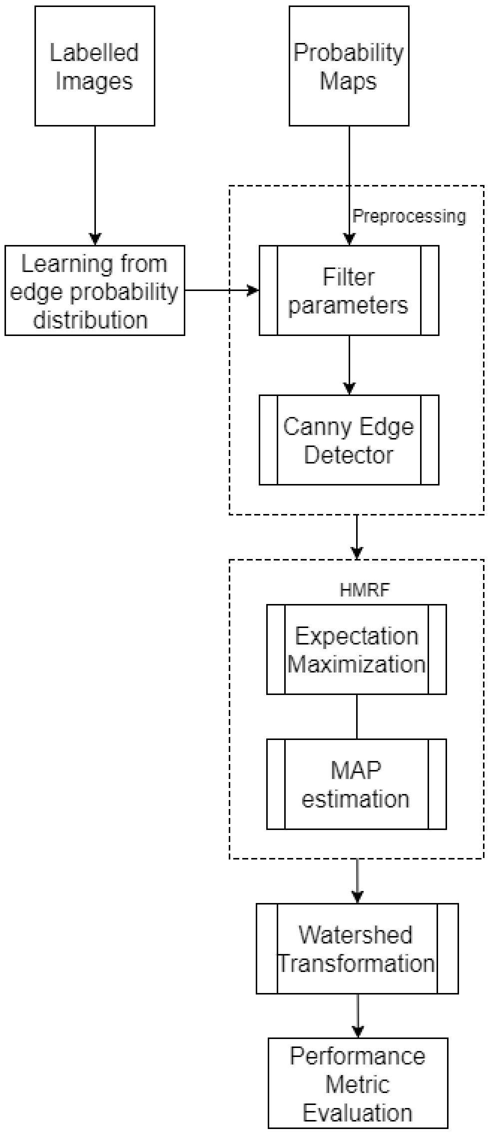

Fig. 2. Schematic flow diagram

edges. Since we have a neat distinction here, we considered making use of the provided labeled image to modify the probability map.

1) Naive approach: Initially we considered converting the image into an edge map as shown in figure 4. A classifier will then use this image to learn probability value corresponding to edges. However, since there were huge amount of zeros in the matrix compared to ones, the classifier performed poorly while predicting ones. So we decided to follow another approach.

2) Probability map filtering + Canny edge detection: Instead, of using a classifier, we tried a canny edge detector for identifying edges from the probability map. Each image has an outer and inner edge. The probability values for both of these edges are different so both of them need to be analyzed and taken into consideration.
Figure 5 and figure 6 show the discrete and cumulative prob-

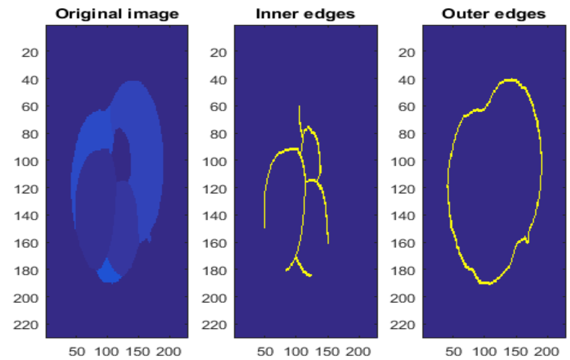

Fig. 3. Sample images with inner and outer edges

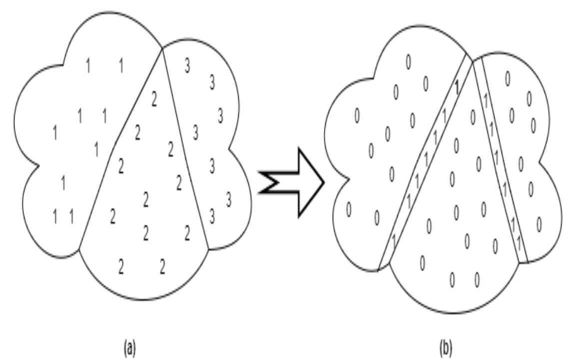

Fig. 4. Naive preprocessing of labeled images: (a) image before processing (b) image after processing

ability distribution respectively of the values of outer edges while figure 7 and figure 8 show the discrete and cumulative probability distribution of the values of inner edges. We separately analyzed the outer edges which had generated by identifying the cluster boundaries and labelling values inside the cluster as 0 and any outer edge as 1; and inner edges as any edge that is not the outer boundary. This was performed for all the images. After observing the probability counts distribution for all, a minimum and maximum threshold for an edge and non-edge was detected. Any value above 230 was modified to be 255 since they showed a high probability of being an edge. However, this was observed to give poor results after segmentation using the MRF. Similarly, upon observing inner edge probability distribution, we set a minimum threshold of 50 for a pixel to be an edge. All values less than 50 were changes to 0 owing to less likeliness of being an edge.

These processed probability maps were passed to a canny edge detector. The results were convincing since inner regions which were initially being identified as edges were now removed as demonstrated in figure 9.

B. Hidden Markov Random Field

A Markov Random Field seemed suitable for our problem as we needed to consider the spatial information of the image which is encoded through contextual constraints of

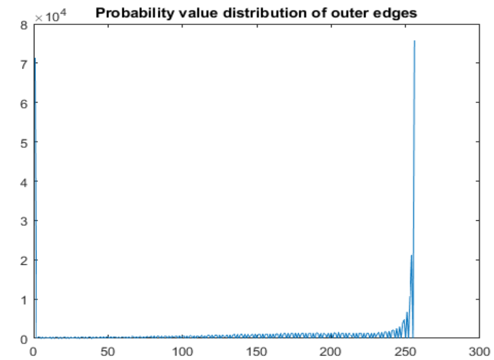

Fig. 5. Discrete frequency distribution of outer edge probabilities

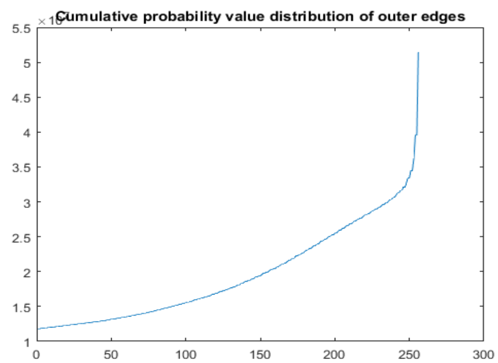

Fig. 6. Cumulative frequency distribution of outer edge probabilities

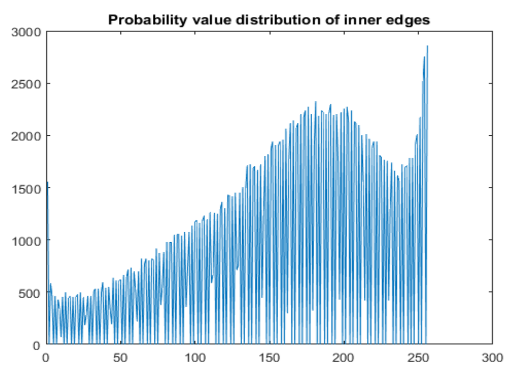

Fig. 7. Discrete frequency distribution of inner edge probabilities

neighboring pixels. For image segmentation, we had a 2-D matrix of observations or pixel values, which we needed to evaluate. Neighboring pixels are expected to have the same class labels or similar intensities. This is achieved through characterizing mutual influences among pixels using conditional MRF distributions. According to the local characteristics of MRFs, the joint probability of any pair of (X_i, Y_i), given X_i's neighborhood configuration X_{N_i}, is

$$P(y_i, x_N | x_{N_i}) = P(y_i|x_i)P(x_i|x_{N_i})$$

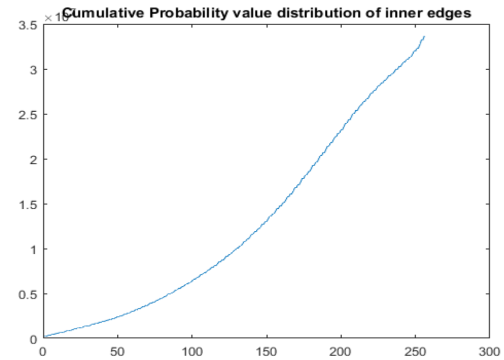

Fig. 8. Cumulative frequency distribution of inner edge probabilities

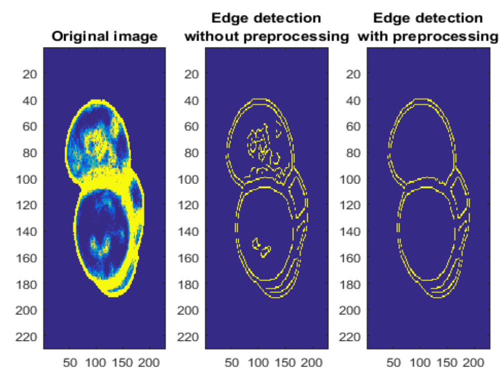

Fig. 9. Canny Edge detection

However, Hidden Markov Random Field (HMRF), which is an improved version of the Hidden Markov Models, evaluates the pixels observations and neighboring information generated by a Markov chain whose state sequence cannot be observed directly. Each observation is assumed to be a stochastic function of the state sequence. So we used an HMRF model for the same.

Given an image $y = y_1, .., y_N$ where each y_i is the intensity of a pixel, to which labels $x = x_1, .., x_N$ need to be configured where $x_i \in L$ and L is the set of all possible labels. According to the MAP criterion, we seek the labeling x^* which satisfies

$$x^* = \arg\max_x P(y|x, \Theta)P(x)$$

The prior probability $P(x)$ is a Gibbs distribution, and the joint likelihood probability is

$$P(y|x, \Theta)P(x) = \prod P(y_i|x_i, \Theta_{x_i})$$

where $P(y_i|x_i, \Theta_{x_i})$ is a Gaussian distribution with parameters $\Theta_{x_i} = (\mu_{x_i}, \sigma_{x_i})$. $\Theta = \theta_l | l \in L$ is the parameter set, which is refined after running the EM algorithm.

We implemented the HMRF-EM algorithm described in [2] which applies K-means to get an initial set of parameters for segmentation. By iterating on the Expectation Maximization algorithm to minimize the energy potentials using Maximum a Posteriori estimation, the initial parameters are optimized. These are briefly described below:

1) Expectation-Maximization (EM) Algorithm: The EM algorithm was used to estimate the parameter set $\Theta = \theta_l | l \in L$. After starting the algorithm with an initial parameter set $\Theta^{(0)}$ generated after running [5] K-Means, the EM algorithm was run to improve on the parameter set. It can be further described as follows:

- E-step: At the t^{th} iteration, we have $\Theta^{(t)}$, and we calculate the conditional expectation:

$$Q(\Theta|\Theta^{(t)}) = E[lnP(x, y|\Theta)|y, \Theta^(t)]$$

which is calculated from the set of all possible configurations of labels.

- M-step: Now $Q(\Theta|\Theta^{(t)})$ is maximized to obtain the next estimate:

$$\Theta^{(t+1)} = \arg\max_{\Theta} Q(\Theta|\Theta^{(t)})$$

Then let $\Theta^{(t+1)} \rightarrow \Theta^{(t)}$ and repeat from the E-step.

2) Maximum a Posteriori (MAP) Estimation: In the EM algorithm, we needed to solve for x^* to minimize the total posterior energy,

$$x^* = \arg\min_{x \in \chi} U(y|x, \Theta) + U(x)$$

with given y and Θ where the likelihood energy is

$$U(y|x, \Theta) = \sum_i U(y_i|x_i, \Theta)$$

The prior energy function U(x) is of the form

$$U(x) = \sum_{c \in C} V_c(x)$$

where $V_c(x)$ is the clique potential and C is the set of all possible cliques.

To minimize the total posterior energy, we used the Maximum a Posteriori probability estimation algorithm. To start with, we have an initial estimate $x^{(0)}$ from the previous iteration of the EM algorithm.

Provided $x^{(k)}$, for all $1 \leq i \leq N$, we found

$$x_i^{(k+1)} = \arg\min_{x \in \chi} U(y_i|l) + \sum_{j \in N_i} V_c(l, x_j^{(k)})$$

Repeated iterations of the above step were run until $U(y, x|\Theta) + U(x)$ converged or a maximum k was achieved.

A sample of images obtained during different steps of the HMRF steps are displayed in figure 11.

C. Watershed Transform

Watershed is a transformation defined on a grey scale image. The watershed transformation treats the image it operates upon like a topographic map, with the brightness of each point representing its height, and finds the lines that run along the tops of ridges. The figure 12 depicts the segmentation obtained from the edge-connected image of HMRF process.

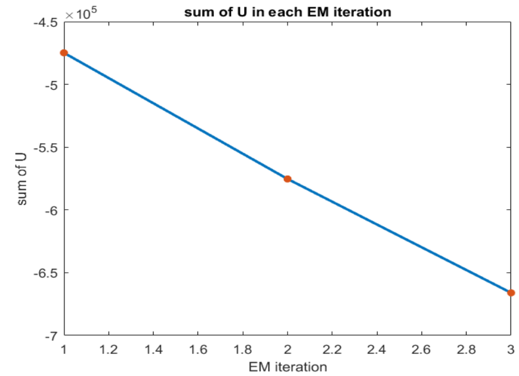

Fig. 10. Cumulative energy potential with every iteration

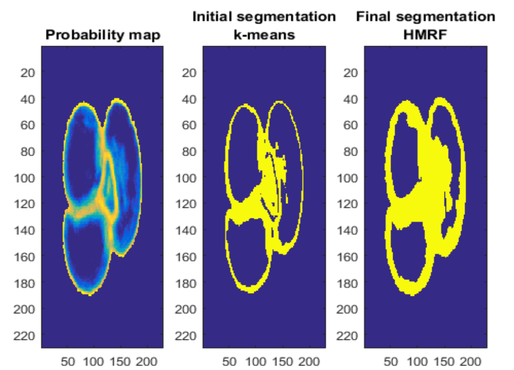

Fig. 11. Image through stages of HMRF process

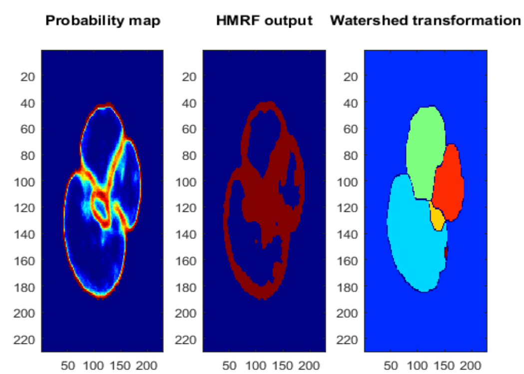

Fig. 12. Watershed transformation

IV. Parameter Selection and Tuning

A. Threshold of Canny edge detector

The edge detector used earlier in preprocessing uses a canny edge detector whose threshold value determines whether a pixel is an edge or not. All values smaller than the threshold are ignored. The graph of coverage vs threshold of canny edge detector is shown in figure 13. The value of selected threshold was the one giving the maximum coverage. It can be observed that below the value of 0.6 doesn't change much. So we selected 0.5 as the value of threshold.

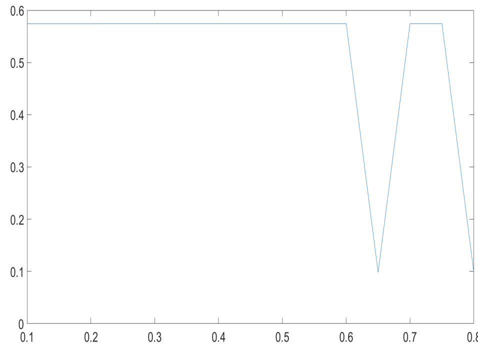

Fig. 13. Coverage vs threshold

B. Number of centroids in K-means

The number of K-mean centroids play an important role in generating the segments later on in the pipeline. Hence, we selected this parameter for tuning our model. The graph of coverage vs number of centroid is shown in figure 14 The value of k selected for K-Means was selected after observing the highest coverage on the image set. We selected k=2 for our model since as the value increases, computation cost and overfit increases.

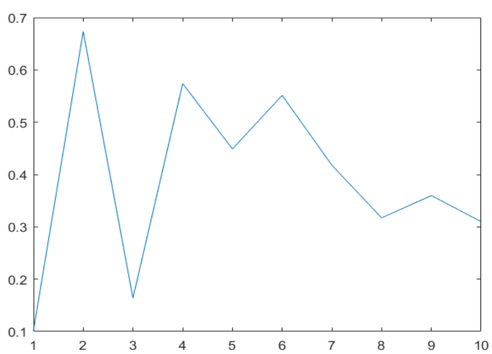

Fig. 14. Coverage vs number of K-means centroids

V. Performance Metric

To evaluate the quality of segments that were created using the segmentation algorithm, we used the overlap ratio metric provided by Dr. Lobaton to us. It defines overlap between the segments of the labeled image provided as ground truth, and the segmented image derived after processing the probability map as :

$$O(R, R') = \frac{R \cap R'}{R \cup R'}$$

The covering of the segmentation S by the segmentation S' can be defined as

$$C(S, S') = \frac{1}{N_s} \sum_{R \in S} |R| \max_{R' \in S'} O(R, R')$$

where N_S is the total number of pixels in the ground truth segmentation S excluding the background region. That is, S considers the background as a region and ignores apertures. It was observed that changing the indexes didn't affect the score significantly confirming that the overlap is independent of the label assigned to it.

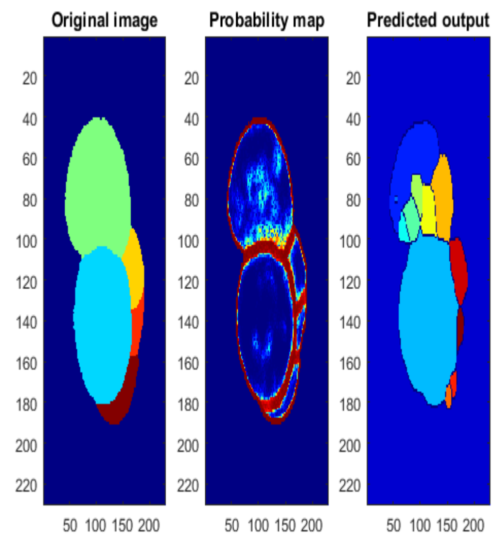

Fig. 15. Sample prediction 1

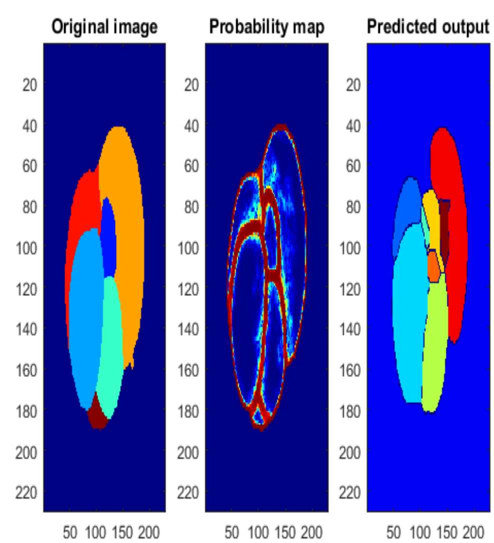

Fig. 16. Sample prediction 2

VI. Results

The overlap ratio obtained by our model is shown in the table below. Figure 18 shows the overlap ratio obtained for each of the 320 images in the data set. Sampmle images and their outputs are shown in image **??**, **??** and **??**

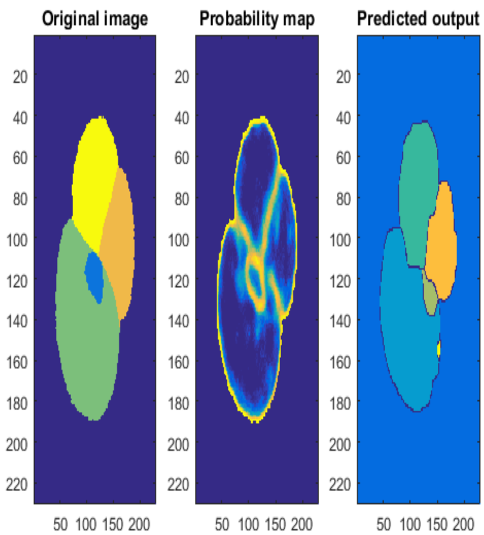

Fig. 17. Sample prediction 3

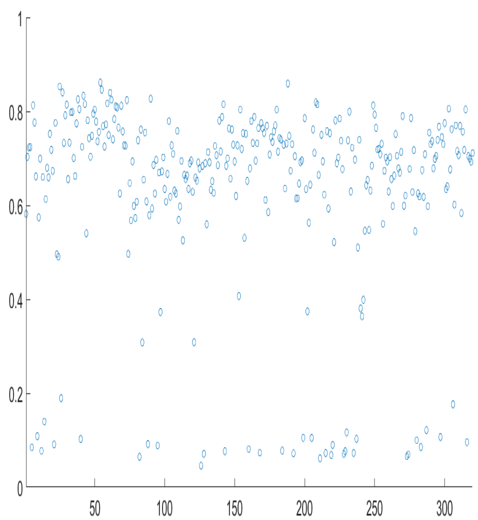

Fig. 18. Overlap Ratio of all 320 images

	Min	Max	Avg
Overlap Ratio	0.05	0.86	0.62

VII. Toolboxes

We used the HMRF-EM toolbox in Matlab.

- HMRF_EM: Hidden Markov Random Field using EM algorithm
- MRF_MAP: MAP algorithm
- gaussianBlur: To blur the image using Gaussian Kernel
- ind2ij: To convert the index to 2D image coordinates
- image_kmeans: K-Means algorithm for 2D image segmentation to provide the initial parameters to HMRF
- watershed: perform watershed transformation on the image

References

[1] Q. Ge, B. Zhong, B. Kanakiya, R. Mitra, T. Marchitto, E. Lobaton, *Coarse-to-Fine Foraminifera Image Segmentation through 3D and Deep Features*, IEEE Symp. Series on Computational Intelligence (SSCI),2017.

[2] Wang, Quan, *HMRF-EM-image: Implementation of the Hidden Markov Random Field Model and its Expectation-Maximization Algorithm* arXiv:1207 18 Dec 2012

[3] Hochbaum, Dorit S. *An efficient algorithm for image segmentation, Markov random fields and related problems*, Journal of the ACM (JACM) 48.4 (2001): 686-701.

[4] Dubuisson, M-P., and Anil K. Jain. *A modified Hausdorff distance for object matching* Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on. Vol. 1. IEEE, 1994.

[5] Zhang, Y et al. *Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm* IEEE Transactions on Medical Imaging Vol. 20, Jan 2001