COMPUTER VISION MIDTERM ASSIGNMENT

-Vineeth S Subramanyam (2221296)

I. AIM

Given a set of consecutive MRI axial cross sections, develop image processing algorithms in 2D and 3D that are capable of producing segmented masks for the following classes: Air, Skin/Scalp, Skull, CSF, Gray Matter and White Matter. Additionally carry out experiments to empirically validate the developed methods.

II. METHODOLOGY

For the tasks of both 2D and 3D segmentation, a combination of different image processing techniques were tested. The final techniques that were selected involved the use of multi class Otsu Thresholding and Image K means clustering.

Image KMeans Clustering Algorithm Summary

1. Set the number of cluster centers **k.**
2. Initialize array of **k** cluster centers as **k** random colors.
3. Initialize **k** lists to hold points from each cluster.
4. Loop till convergence, or number of iterations:
   a. Iterate through each point in the dataset
      - Compute Euclidean distance between current point to each cluster center.
      - Add the current point to the cluster list of the class with lowest euclidean distance from its center.
   b. Compute new cluster centers to be the mean of the list of each cluster points.

Multi Otsu Threshold Algorithm Summary

1. Set the number of classes to be **k.**
2. Compute all possible combinations of **k-1** thresholds, such that **t2>t1, t3>t2**, etc. (E.g. (1,2,3,4), (1,2,3,5), (1,2,3,6)….etc)
3. Iterate over each possible combinations:
   a. Compute mean, variance and weight of each class
   b. Compute between class variance given by formula

$$\sigma_B^2 = \sum_{k=1}^{K} P_k (m_k - m_G)^2, \quad \textbf{where} \quad P_k = \sum_{i \in C_k} p_i \quad \textbf{and} \quad m_k = \sum_{i \in C_k} i p_i$$

4. Select the set of thresholds that have maximum between class variance

A. 2D Segmentation Algorithm

1) Load dataset (**Brain.mat**)
2) For each (**image, label**) pair:

a) Apply **Multi class Otsu Thresholding** to segment out the outer ring(*Class 1*) , and the inner portion of the scan(*Class 3-5*).

b) Find all connected components in the image mask(**bwlabel**).

c) Sort the components by area, and select the 2 biggest components. (*Biggest component consists of Classes 3-5, and the next biggest component is the outermost ring*)

Background Class

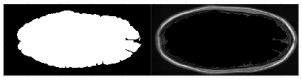
d) To extract the background class, fill (**imfill**) the outermost ring (*obtained from* ***c)***), and take the complement of the image obtained.

Classes 1-2

e) Extract the mask corresponding to only the inner portion of the brain(*obtained from step c*), and apply **imfill** to close all holes.

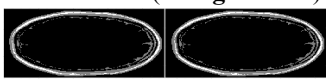
f) Find all pixels in this mask that contain a value=1.

g) Create a copy of the original image and set all the pixels from step **f)** to be 0.



h) Apply one of two methods:

- ***Thresholding Based:*** Apply **Multi class Otsu Thresholding** to segment the image mask into 3 classes (**multithresh** followed by **imquantize**).
- ***Clustering Based:*** Apply **K Means Clustering** to segment the image mask into 3 classes.(**imsegkmeans**)



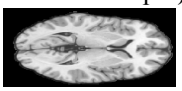
i) Extract the mask corresponding to Class 1, and take its complement. (*Additional pixels inside the ring get filled when creating the masks of classes 3-5, thus reducing error.*)

j) Extract the mask with index corresponding to Class 2 and then subtract the background, and inner matter.

Classes 3-5

k) Erode the mask containing the largest component obtained from step **c)**.

l) Create a mask from the original image containing only pixels inside the largest component after step **k)**.



m) Apply one of two methods:

Thresholding Based

- Apply Multi class Otsu Thresholding to segment the image mask into 4 classes(**multithresh+imquantize**)
- Extract the masks corresponding to Class 3,4 and 5 depending on the quantized image indexes. (*e.g. Class 3 mask obtained by using L==3)*

K means Based:

- Apply **K Means Clustering** to segment the mask into 3 classes.
- Since K Means Clustering initializes centers randomly, the class indexes differ from iteration to iteration for each label, therefore an additional check is added to compare similarity of masks to the ground truth to find the correct index.
- **For** i=3:5 (*Class 3-5 of ground truth*)
  - Create a ground truth mask corresponding to class **i** using **label** *(e.g. label==i)*.

  - **For** j=1:4 (*Indexes corresponding to clusters*)
  - Create masks corresponding to class **j** using the clustered image *(e.g. L==j)*.

- Compute Structural Similarity measure(**SSIM**) between the ground truth mask, and each clustered mask.

- **If SSIM**>0.8: *(Indicates that the correct index has been obtained)*:
  - Extract predicted mask for the corresponding **i** th ground truth class.

B. 3D Segmentation Algorithm

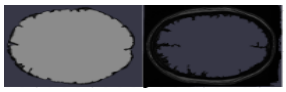
1. Load dataset (**Brain.mat**)
2. Apply **Multi class Otsu Thresholding** to segment out the outer ring(*Class 1*), and the inner portion of the scan(*Class 3-5*) using the input 3D scan.
3. Find all connected components in the 3D mask(**bwlabeln**).
4. Sort the components by area, and select the 2 biggest components. (*Biggest component consists of Classes 3-5, and the next biggest is the outermost ring*)

Background Class:

5. To extract the background class, fill the outermost ring (**imfill3**), and take the complement of the 3D mask obtained.

Classes 1-2:

6. Extract the 3D mask corresponding to only the inner portion of the brain(*obtained from step 4*), and apply **imfill3** to close all holes. Find all pixels that contain a value=1.
7. Create a copy of the original 3D input mask and set all the pixels from step **6)** to be 0.

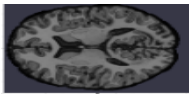


8. Apply one of two methods:
   - ***Thresholding Based:*** Apply **Multi class Otsu Thresholding** to segment the 3D mask into 3 classes (**multithresh** followed by **imquantize**).
   - ***Clustering Based:*** Apply **K Means Clustering** to segment the 3D mask into 3 classes.(**imsegkmeans3**)

9. Extract the mask corresponding to Class 1, and take its complement. (*Additional pixels inside the ring get filled when creating the masks of classes 3-5, thus reducing error.*)
0. Extract the mask with index corresponding to Class 2 by subtracting the background, and inner matter.

Classes 3-5

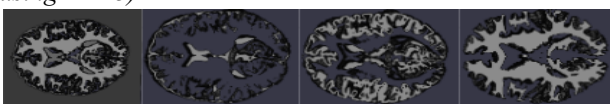
1. Erode the mask containing the largest component obtained from step **4)**.
2. Create a mask from the original image containing only pixels inside the largest component after step **9)**.



3. Apply one of two methods:

Thresholding Based

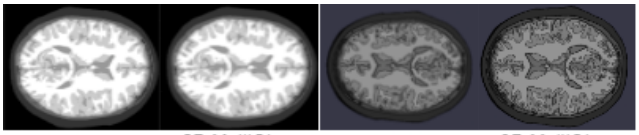
- Apply Multi class Otsu Thresholding to segment the image mask into 4 classes(**multithresh+imquantize**)
- Extract the masks corresponding to Class 3,4 and 5 depending on the quantized image indexes. (*e.g. Class 3 mask obtained by using L==3)*



K means Based:

- Apply **K Means Clustering** to segment the 3D mask into 3 classes.
- Since K Means Clustering initializes centers randomly, the class indexes differ from iteration to iteration for each label, therefore an additional check is added to compare similarity of masks to the ground truth to find the correct index.
- **For** i=3:5 (*Class 3-5 of ground truth*)
  - Create a ground truth mask corresponding to class **i** using **label** *(e.g. label==i)*.
  - **For** j=1:4 (*Indexes corresponding to clusters*)
  - Create masks corresponding to class **j** using the clustered image *(e.g. L==j)*.
  - Compute Structural Similarity measure(**SSIM**) between the ground truth mask, and each clustered mask.
  - **If SSIM**>0.8: *(Indicates that the correct index has been obtained)*:
    - Extract predicted mask for the corresponding **i** th ground truth class.

III. RESULTS



The approaches are tested and validated so as to maximize the following criteria:

- **Pixelwise Accuracy:** The segmentation approach should be capable of classifying individual pixels correctly.
- **Overall Image structure:** The segmentation approach should ensure that each mask segmented has the same overall structure as the ground truth mask. This metric disregards individual pixels, and rather emphasizes on the entire image.
- **Overlap between ground truth and segmented label:** The segmentation approach should create masks that overlap perfectly with the ground truth label. At the same time, it is important to consider that the background class dominates a majority of each class mask, and therefore the selected approach should have a high precision and recall.
- **Inference time:**The approach chosen should be capable of being used in a real time application.

The segmented results are evaluated to maximize the set of criteria as defined above using the following metrics:

- **Intersection over Union (IOU):** It is an estimate of overlap between the ground truth and predicted label.

- **Structural Similarity Measure(SSIM)**: It is an estimate of overall similarity of two images using their luminance, contrast and overall structure.(Reference)
- **Accuracy:** It represents a total of how many pixels were correctly classified.
- **F1 Score**: It is the Harmonic mean of Precision and recall. Allows in selecting a model with a balance between precision and recall.
- **Mean Score**: The final metric used to compare the algorithms is the average of these different scores.

$IOU = \frac{TP}{TP+FP+FN}$	$\text{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}.$
$Acc = \frac{TP+TN}{TP+TN+FP+FN}$	$F1 = \frac{2*(P*R)}{P+R}$

$$MeanScore = \frac{IOU + SSIM + F1 + ACC}{4}$$

TP: True Positive, FP: False Positive, TN: True Negative, FN: False Negative, P:Precision, R:Recall

K Means 2D algorithm Scores (Overall Mean = 0.94)

Metric	C0	C1	C2	C3	C4	C5
IOU	0.995	0.948	0.864	0.753	0.90	0.936
SSIM	0.988	0.956	0.959	0.886	0.863	0.907
F1 Score	0.997	0.973	0.927	0.859	0.947	0.967
ACC	0.998	0.992	0.991	0.978	0.975	0.986
Mean Score	0.995	0.967	0.936	0.869	0.92	0.949

Thresholding 2D algorithm Scores (Overall Mean = 0.938)

Metric	C0	C1	C2	C3	C4	C5
IOU	0.995	0.949	0.866	0.753	0.894	0.931
SSIM	0.988	0.956	0.960	0.886	0.857	0.900
F1 Score	0.997	0.973	0.928	0.859	0.944	0.964
ACC	0.998	0.992	0.991	0.978	0.973	0.984
Mean Score	0.994	0.968	0.936	0.869	0.917	0.945

K Means 3D algorithm Scores (Overall Mean = 0.942)

Metric	C0	C1	C2	C3	C4	C5
IOU	0.995	0.951	0.893	0.760	0.897	0.944
SSIM	0.987	0.943	0.966	0.855	0.875	0.914
F1 Score	0.997	0.974	0.943	0.864	0.945	0.971
ACC	0.998	0.992	0.993	0.979	0.974	0.987
Mean Score	0.994	0.965	0.949	0.865	0.923	0.954

Thresholding 3D algorithm Scores (Overall Mean = 0.940)

Metric	C0	C1	C2	C3	C4	C5
IOU	0.995	0.951	0.894	0.77	0.887	0.930
SSIM	0.987	0.944	0.966	0.863	0.865	0.892
F1 Score	0.997	0.975	0.944	0.870	0.940	0.963
ACC	0.998	0.992	0.993	0.980	0.972	0.984
Mean Score	0.994	0.965	0.949	0.871	0.916	0.942

Time Taken by each algorithm

Method	Time Taken (seconds)
K Means 2D	3.767
Multi Thresholding 2D	1.998
K Means 3D	4.108
Multi Otsu Thresholding 3D	2.107

IV. CONCLUSION

A. *Final Algorithm Selection*

From the tables in **III**, we can see that the performance of both the **Multi-class Otsu thresholding**, and **K-means clustering** approaches result in very similar mean scores for both 2D and 3D. The **K-means clustering** technique results in a marginally better mean overall score for both 2D and 3D with scores of **0.94** and **0.942** respectively, however are significantly slower than the thresholding approaches due to the presence of additional checks in clustering required to get the correct mask indexes. Therefore the final chosen algorithm makes use of only **Multi-class Otsu thresholding** which results in a mean score of **0.938** and **0.94** for 2D and 3D respectively.

B. *Result Analysis*

From the tables in **III**, we can infer that the performance of the algorithm suffers for **classes 2 and 3**. This is because of the slight overlap between the two class boundaries, as well as the fact that both classes have a very thin mask where minor pixel errors lead to a large error. The algorithm performs significantly well in **classes 0,1 and 5**, since these classes are easily differentiable by their pixel intensities and have more or less solid masks without complex structures. The algorithm performs moderately well on **class 4** due to its mask having a complex structure with a significant number of holes inside it.

C. *Comparison of 2D and 3D approaches*

When comparing the 3D approaches to their 2D counterparts, we see more or less an identical performance with there only being a slight improvement in the scores of the 3D algorithm in the masks of **class 2** and **3.** These minor differences can be attributed to the contour detection algorithm(**bwlabel**) using a connectivity of **8** pixels for 2D, and a connectivity of **26** for 3D, as well as certain minor differences in the implementation of the codes. Such a similarity in performance is expected since the working of the algorithms used does not differ between 2D and 3D. The main difference between 2D and 3D is in the inference time, where the 3D algorithm is slower than its 2D counterpart.

D. *Future Work and Additional Experiments*

Additionally, algorithms were tested that made use of active contours and region growing, however since they required manual selection of seed points and masks, they are not described in this report. An alternative approach would be the use of Deep Learning models for segmentation which would significantly improve the performance since they do not rely solely on image intensities.