Introduction

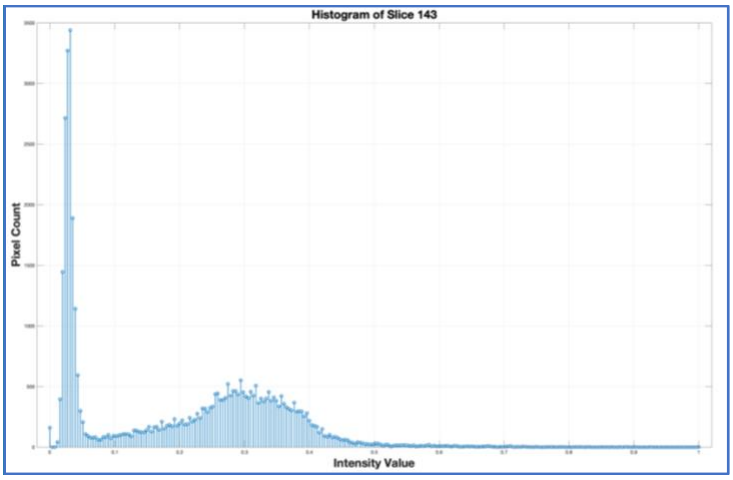
For this assignment, using a 3D MRI brain image data (*sub-13_T1w.nii.gz*), we implement tissue labeling using two approaches:

- First use one of the background removal approaches (binary classifiers such as ISODATA, Otsu, or Bayesian) discussed in the class as pre-processing step to identify foreground from background pixels and then classify foreground pixels into white matter, gray matter, csf, and no-brain tissue (Ex. scalp).
- Or use the approach discussed in the module to directly divide pixels into background, white matter, gray matter, csf, and no-brain tissue (Ex. scalp).
- Implementations should include gain factor estimation.

We need to display the original slice along all different tissues and gain factor images. And we limit the gain factor estimation display only to the brain area.

Main Work

1) We upload the NIFTI image **sub-13_T1w.nii.gz** from the folder data. Make sure that the folder exists with the file. If not, please modify the line: *vol = niftiread('data/sub-13_T1w.nii.gz')* to point to the location with the file.
2) The image is flipped to facilitate image manipulation.
3) Then we look at the histogram intensity of slice 143 which shows that it is bimodal so the OTSU's method should be the most appropriate to distinguish background from foreground pixels.



We also compute the classification threshold with Bayesian Minimum Error Thresholding algorithm. If we compare the classification of foreground pixels from background pixels**:**

- The resulting binary mask from Otsu's method, appears to segment the brain well, but it includes some background artifacts and fails to exclude portions of the scalp completely.

- Minimum Error Thresholding's threshold (T = 0.51) is higher than Otsu's threshold (T=0.18), leading to a more aggressive exclusion of background pixels.
- The binary mask from Minimum Error Thresholding, is cleaner and includes fewer background artifacts, but it potentially excludes some brain tissues near the edges, such as regions close to the scalp or the outer parts of the brain.

To effectively identify not only white and gray matter but also cerebrospinal fluid (CSF) and non-brain tissue (e.g., scalp), the preprocessed image generated using Otsu's method is preferred due to its ability to retain more detailed segmentation. In addition, the absolute "goodness" coefficient—a measure of thresholding effectiveness—is significantly higher for Otsu's method (0.8) compared to the minimal value for the "minimum error thresholding" method (0.09) (**Fig. 1**).

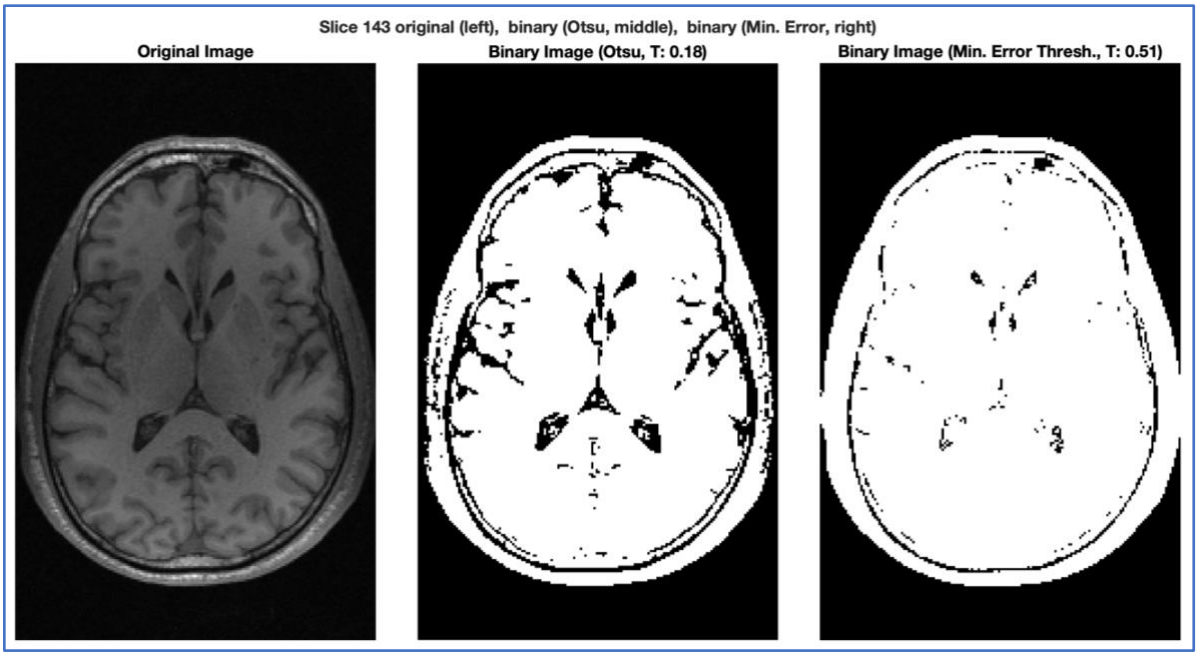


Figure 1: Automatic Thresholding using Otsu's Method (middle) and Minimum Error Thresholding(right).

Otsu's method is implemented through two key functions: **otsuThreshold** and **makeMeanTables**, which adhere to the steps outlined in **Algorithm 1**. These functions compute the optimal threshold by minimizing the intra-class variance, consistent with the original Otsu's algorithm.
Similarly, the **Minimum Error Thresholding** approach is implemented based on **Algorithm 2**. The corresponding functions, **minimumErrorThreshold** and **makeSigmaTable**, calculate the threshold by minimizing the probabilistic classification error between the foreground and background classes.

Algorithm 2 Otsu Threshold

Input: $h : [0 \to K-1] \to N$ ▷ gray scale histogram
Output: Optimal threshold value or -1 if not found
$K \leftarrow \text{size}(h)$ ▷ number of intensity levels
$(\mu_0, \mu_1, N) \leftarrow \text{MakeMeanTable}(h, K)$ ▷ table of means
$\sigma^2_{\max} \leftarrow 0$
$q_{\max} \leftarrow -1$
$n_0 \leftarrow 0$
for $q = 0$ to $K-2$ **do**
    $n_0 \leftarrow n_0 + h(q)$
    $n_1 \leftarrow N - n_0$
    if $n_0 > 0 \wedge n_1 > 0$ **then**
        $\sigma_b^2 \leftarrow \frac{1}{N^2} n_0 n_1 (\mu_0(q) - \mu_1(q))^2$
        if $\sigma_b^2 > \sigma^2_{\max}$ **then**
            $\sigma^2_{\max} \leftarrow \sigma_b^2$
            $q_{\max} \leftarrow q$
        end if
    end if
end for
return $q_{\max}$

Algorithm 3 Make Mean Tables

Input: $h : [0 \to K-1] \to N, K$ ▷ gray scale histogram
Output: μ_0, μ_1, N ▷ tables of means and total count
$n_0 \leftarrow 0, S_0 \leftarrow 0$
for $q = 0$ to $K-1$ **do**
    $n_0 \leftarrow n_0 + h(q)$
    $S_0 \leftarrow S_0 + q \cdot h(q)$
    $\mu_0(q) \leftarrow \begin{cases} S_0/n_0 & \text{if } n_0 > 0 \\ -1 & \text{otherwise} \end{cases}$
end for
$N \leftarrow n_0$
$n_1 \leftarrow 0, S_1 \leftarrow 0$
$\mu_1(K-1) \leftarrow 0$
for $q = K-2$ to 0 **do**
    $n_1 \leftarrow n_1 + h(q+1)$
    $S_1 \leftarrow S_1 + (q+1) \cdot h(q+1)$
    $\mu_1(q) \leftarrow \begin{cases} S_1/n_1 & \text{if } n_1 > 0 \\ -1 & \text{otherwise} \end{cases}$
end for
return (μ_0, μ_1, N)

Algorithm 1: Otsu's Method.

Algorithm 1 Minimum Error Thresholding

Require: $h : [0, K-1] \to \mathbb{N}$ ▷ Grayscale histogram
Output: Optimal threshold th for binary classification or -1 if not found
$K \leftarrow \text{size}(h)$
$(\sigma_0^2, \sigma_1^2, N) \leftarrow \text{MakeSigmaTable}(h, K)$
$n_0 \leftarrow 0$
$q_{\min} \leftarrow -1$
$e_{\min} \leftarrow \infty$
for $q \leftarrow 0$ to $K-2$ **do**
    $n_0 \leftarrow n_0 + h(q)$
    $n_1 \leftarrow N - n_0$
    if $n_0 > 0$ and $n_1 > 0$ **then**
        $P_0 \leftarrow \frac{n_0}{N}$
        $P_1 \leftarrow \frac{n_1}{N}$
        $e \leftarrow P_0 \cdot \ln(\sigma_0^2(q)) + P_1 \cdot \ln(\sigma_1^2(q)) - 2(P_0 \ln(P_0) + P_1 \ln(P_1))$
        if $e < e_{\min}$ **then**
            $e_{\min} \leftarrow e$
            $q_{\min} \leftarrow q$
        end if
    end if
end for
return $q_{\min}$

Algorithm 2: Minimum Error Thresholding.

Algorithm 2 MakeSigmaTable

Require: h, K

Output: Creates maps $\sigma_0^2, \sigma_1^2 : [0, K-1] \to \mathbb{R}$

$n_0 \leftarrow 0$
$A_0 \leftarrow 0$
$B_0 \leftarrow 0$
for $q \leftarrow 0$ to $K-1$ **do**
  $n_0 \leftarrow n_0 + h(q)$
  $A_0 \leftarrow A_0 + h(q) \cdot q$
  $\sigma_0^2(q) \leftarrow \begin{cases} \frac{1}{12} + \left(B_0 - \frac{A_0^2}{n_0}\right)/n_0 & \text{if } n_0 > 0 \\ 0 & \text{otherwise} \end{cases}$
end for
$N \leftarrow n_0$
$n_1 \leftarrow 0$
$A_1 \leftarrow 0$
$B_1 \leftarrow 0$
$\sigma_1^2(K-1) \leftarrow 0$
for $q \leftarrow K-2$ to 0 **do**
  $n_1 \leftarrow n_1 + h(q+1)$
  $A_1 \leftarrow A_1 + h(q+1) \cdot (q+1)$
  $B_1 \leftarrow B_1 + h(q+1) \cdot (q+1)^2$
  $\sigma_1^2(q) \leftarrow \begin{cases} \frac{1}{12} + \left(B_1 - \frac{A_1^2}{n_1}\right)/n_1 & \text{if } n_1 > 0 \\ 0 & \text{otherwise} \end{cases}$
end for
return $(\sigma_0^2, \sigma_1^2, N)$

MakeSigmaTable function.

4) We perform tissue classification as described in [1] using only the foreground pixels obtained from the binarized image generated by Otsu's method. The implementation follows these steps:

1. **Initialization**:
   Initial estimates for the centroids γk are obtained, and the gain field gj is initially set to 1 for all pixels.
2. **Indicator Function Update**:
   The indicator function z_j^* is solved by minimizing the energy function:

$$z_j^* = \arg\min_{z_j}\left(\left(y_j - g_j \sum_k z_{jk}\gamma_k\right)^2 + \beta \sum_{i \in N_j} z_j^T V z_i\right)$$

where y_j is the intensity of pixel j, β controls the smoothness of the segmentation, Nj denotes the neighborhood of pixel j, and matrix V "ensures that a pixel belonging to the same class as its neighbors is favored over other configurations":
$V = [1,\ldots,1]^T\ [1,..,1] - I$ where I is identity matrix.

3. **Centroid Update**:
   The means γj for each tissue class are updated using the current indicator function zj and gain field gj.

$$\gamma_k = \frac{\sum_j z_{jk} g_j y_j}{\sum_j z_{jk} g_j^2}$$

4. **Gain Field Update**:
   The gain field is refined by solving the following equation for the polynomial coefficients fn:

$$[z_{jk}\gamma_k P_n(j)]\,[f_n] = [z_{jk} y_j]$$

$$f = (P^T P)^{-1} P^T Y$$

$$g_j = \sum_{n=1}^{N} f_n P_n(j)$$

where Pn(j) are the basis functions representing the gain field, and fn are the coefficients to be determined.

Code implemented in functions:

- initializeCentroids
- getNeighbors
- computePolynomialBasis
- estimateIndicatorFunctionWithGain_ICM_V
- updateCentroids
- unsupervisedClassification
- preprocessAndClassify

5) When displaying the original slice, segmented image, and gain field side by side (function **showResultsWithGain**), we observe the following (**Fig. 2**):

1. Original Slice (Left Panel):

- The original slice clearly displays anatomical structures but exhibits non-uniform intensity distribution due to field inhomogeneities or imaging artifacts.
- These inhomogeneities emphasize the importance of gain field correction to normalize intensities for accurate tissue classification.

2. Tissue Segmentation (Middle Panel):

- The segmentation distinguishes brain tissue from non-brain regions (e.g., background and scalp), leveraging Otsu's thresholding to identify foreground pixels.
- While effective, segmentation may suffer from artifacts or misclassifications at boundaries or regions with subtle intensity differences due to uncorrected inhomogeneities.

3. Gain Field (Right Panel):

- The gain field reflects multiplicative correction factors applied across the image to address intensity inhomogeneities.
- Bright areas indicate regions requiring higher intensity correction (e.g., edges where intensities were underestimated).
- Dark areas represent regions closer to true intensity values with minimal correction.
- The smooth, continuous gain field demonstrates successful bias correction, enabling more reliable tissue analysis.

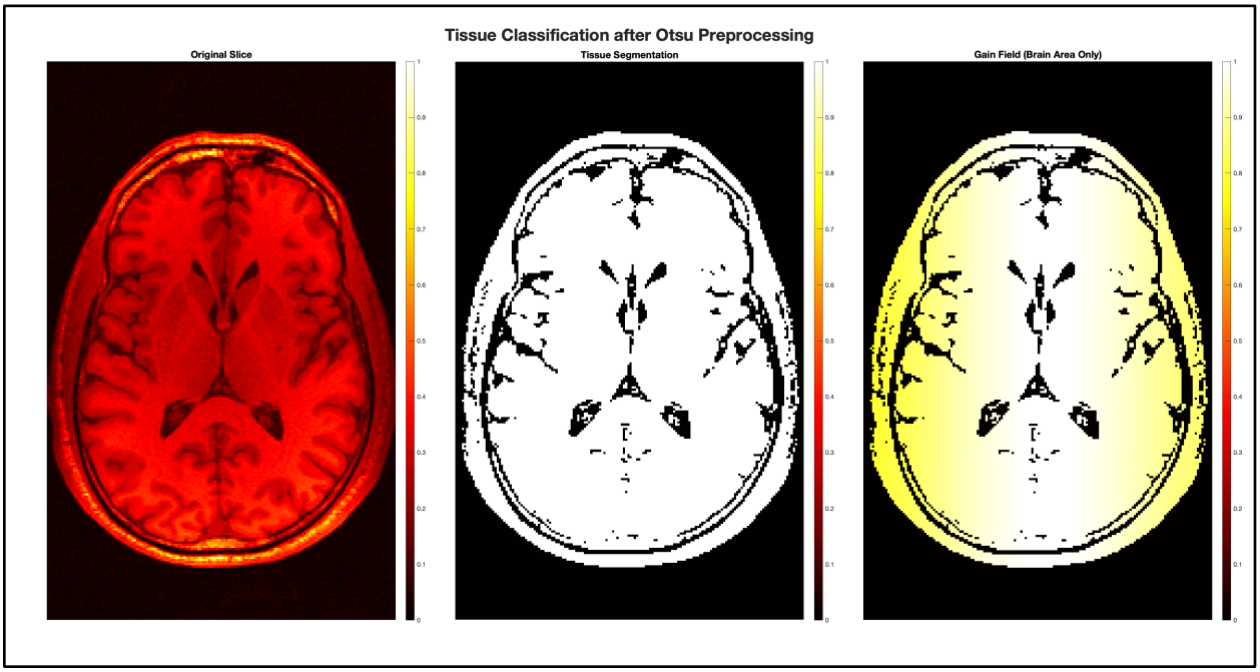


Figure 2: unsupervised tissue classification on preprocessed image using Otsu's method with gain factor impact (right).

6) We now apply the same tissue classification but directly on the original image and observe:

The direct segmentation includes both brain and non-brain tissues, resulting in additional classes (e.g., scalp, background) being represented. Regions outside the brain, such as the periphery, are segmented into different classes: the segmentation seems less refined and more affected by intensity inhomogeneity, with more noticeable artifacts at the boundaries (Fig. 3).

Key Differences:

1. **Preprocessing with Otsu's Method**:
   - Provides cleaner segmentation by isolating the brain region before classification.
   - Gain field estimation focuses only on the relevant regions (brain tissue), leading to smoother corrections and less pronounced compared to direct classification gain.
   - Algorithm convergence occurs quickly (after 3 iterations).
2. **Direct Classification**:
   - Includes both brain and non-brain regions in the classification process, which increases the complexity of segmentation.
   - Segmentation of tissue classes within the brain is less accurate due to interference from non-brain tissues.
   - Gain field correction is less targeted and less smooth.

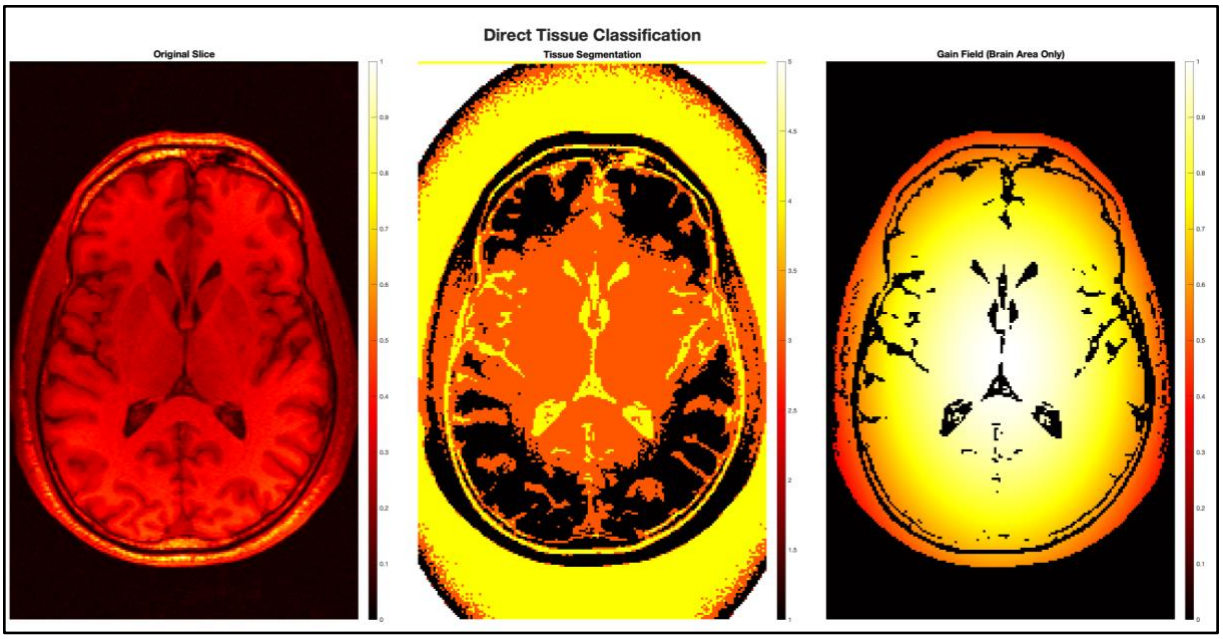


Figure 3: unsupervised tissue classification on image itself (middle) with gain factor impact (right).

Challenges

- The Geometry Preserving Anisotropic Diffusion filter is a complex filter to implement. It has a larger number of critical sections quite intricated, and careful attention to detail is essential to ensure the accuracy and effectiveness of its implementation.
- It was slow to run and needed some code optimizations to increase its processing time and overall efficiency.

Future Directions

- The different filter parameters (T,dt, σ_g , σ_s including the diffusion parameters: a_1 and a_2), could be tuned further using a more exhaustive grid search and for loss function the voxel squared error difference, or other loss functions related to image quality assessment (IQA) metrics (for ex., Peak Signal to Noise Ratio (PSNR), Mean Absolute Error (MAE), and Structural Similarity Index (SSIM)).
- **T Stopping Criterion**: The filter is iterative and runs for a fixed number of iterations (T). A more sophisticated stopping criterion, that halts the process when a convergence or condition is met, such as when the rate of change in the image error becomes minimal; could be implemented.
- **Hybrid approach**: we propose to engineer an hybrid approach with first, a deep learning model which extracts features of interests from properly annotated datasets by physicians or pathologists; then a deep learning model using a second order optimizer based on a loss which mimics the intensity gradient landscape; is used.

  Specifically:
  1. **A Deep Learning Model for Feature Extraction**: A deep learning model, such as a convolutional neural network (CNN), is trained on annotated datasets curated by experts like physicians or pathologists. This model learns to identify and extract features of interest (e.g., edges, tissue boundaries, lesions) from medical images. By leveraging expert annotations, the model gains an understanding of key structures that should be preserved during the filtering process.
  2. **A Gradient-Based Loss Function**: A secondary deep learning model is integrated with a second-order optimization approach, in which the loss function is designed to reflect the intensity gradient landscape. The optimizer utilizes second-order derivatives (e.g., Hessian-based approaches) to refine the filtering process, aligning it with the intensity gradient landscape learned from the feature extraction model.

This is the approach followed by Rodrigo Caye Daudt et al., who propose an iterative learning method to extract useful features from large datasets using a CNN to guide an anisotropic diffusion algorithm to perform edge preserving filtering [1].

Reference

[1] Unsupervised tissue classification, DL Pham, PL Bazin, Handbook of Medical Image Processing and Analysis, 209-222

[2] Robust unsupervised tissue classification in mr image (2004), Dzung L. Pham et al. in Proceedings of the IEEE International Symposium on Biomedical Imaging

[1] Rodrigo Caye Daudt, Bertrand Le Saux, Alexandre Boulch, and Yann Gousseau. 2021. Weakly supervised change detection using guided anisotropic diffusion. Mach. Learn. 112, 6 (Jun 2023), 2211–2237. https://doi.org/10.1007/s10994-021-06008-4