DS 261: Artificial Intelligence for Medical Image Analysis

Assignment 1

Due Date: 18 September, 2025 - 11.59 pm IST

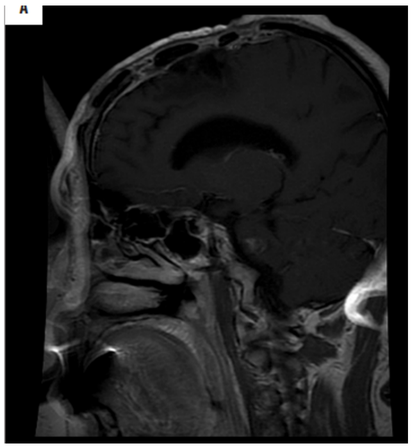
Important Notes:

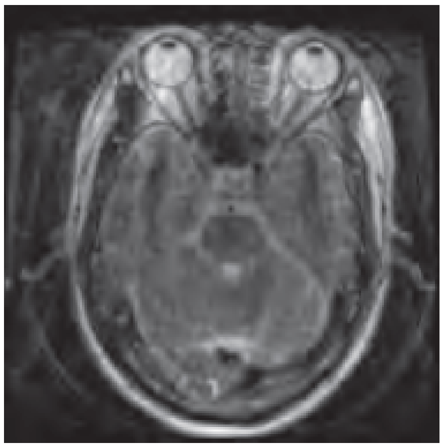
1. **Use of ChatGPT, Gemini, Perplexity or any other LLM-frameworks for assignment completion is strictly prohibited.**
2. **The assignment has to be attempted individually by each person. Group attempts are not allowed.**
3. **Students must submit the report discussing their observations and results for each question strictly in Latex in a legible manner. Failure to submit reports in Latex will be penalized.**
4. **Students are permitted to use up to 2 grace days in total across the 3 assignments, late submissions beyond that will not be graded. Hence, use your grace days judiciously.**
5. **Marks Distribution: Q1-5 Marks, Q2-15 Marks, Q3-10 Marks**

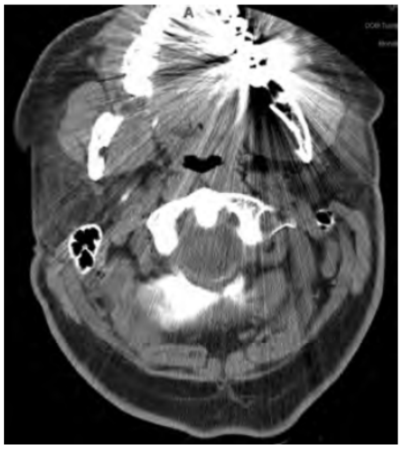
---

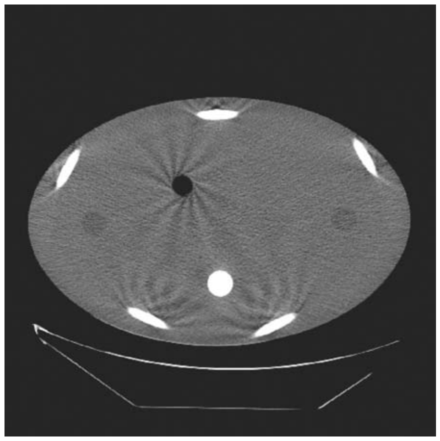
1. In each of the following images in Figure 1, (i) find the source of artifact, and (ii) list the solutions to reduce/overcome each of them. (5 marks)
2. Implement a Gaussian Mixture Model (GMM) using the Expectation–Maximization (EM) algorithm from scratch for pixel-wise segmentation of infarcts from 2D DWI slices. You cannot use library implementations of GMM (e.g., scikit-learn), only standard tools such as NumPy, SciPy, and Matplotlib for math, I/O, and visualization can be used. With the images provided in the `Sample` folder (with ground-truth masks), experiment with different numbers of mixture components k. From the resulting clusters, identify the infarct region and generate a binary infarct mask. Clearly describe the rule you used for selecting the infarct cluster and the convergence criteria you opted for GMM. You are encouraged to try preprocessing (e.g., normalization, smoothing) and postprocessing (e.g., morphological cleanup) steps to improve segmentation quality and discuss the observations in your report.

   Apply your method to the images in the `TestImages` folder. Save the binary infarct masks for the `TestImages` in a folder named `TestResults`, with filenames `Test1-mask.nii.gz`, `Test2-mask.nii.gz`, etc. Report the performance metrics such as the Dice score and lesion wise f1-score and lesion wise absolute element difference from these TestResults for each image and also report the average in a table. Your report should include the experiments carried out, reasoning for the design choices, and visual results. For all the 15 slices in the `TestImages`, generate plots showing the original DWI, all the individual cluster masks produced by GMM, and the final infarct mask. (15 marks)


(a)


(b)


(c)


(d)

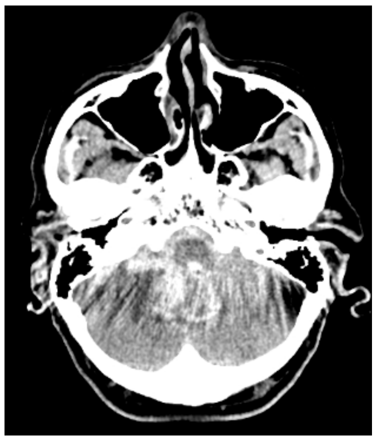

(e)

Figure 1: Artifacts

Note: Grading will be based on correctness of the EM-based GMM implementation, segmentation performance on the Test images, clarity of experiments and reasoning, figures and quality of analysis.
Submission format:
`Assignment1_{name}_{last_5_digits_SRNO}/Q2/gmm_infarcts.ipynb` – should contain code, with all cell outputs.
`Assignment1_{name}_{last_5_digits_SRNO}/Q2/TestResults/Test1_mask.nii.gz`, . . . – Predicted infarct masks for test images.

3. You have been provided with a dataset in the directory `Q3`. This dataset consists of train and validation data with binary labels (i.e. 0 and 1) and test data without labels in the `.npz` format. Your tasks are as follows (10 Marks):

   i. Apply Principal Component Analysis (PCA) to reduce the dimensionality of the data while retaining at least 95% of the variance. **You must implement PCA using only the NumPy library.** After applying PCA, use t-SNE (from scikit-learn or any other suitable library) to further reduce the data to 2 dimensions and plot the results to visualize the spread of your positive (labeled 1) and negative classes (labeled 0). Include this plot in your report and clearly mention the number of principal components selected to achieve the 95% variance threshold.

ii. Next, on the PCA reduced data, apply logistic regression from scratch, **using only NumPy**. You may use batch gradient descent, mini-batch gradient descent or stochastic gradient descent for optimization. You are also free to define a convergence criterion of your choice (e.g., maximum number of training iterations completed, or the training/validation error falling below a certain threshold). Plot and include the training and validation loss curves in your report, showing how the model converges during training. Also report the F1-score, precision, recall and accuracy on both train and validation samples.

iii. Once the model is trained, perform inference on the test data. Store your binarized predictions (values must be either 0 or 1) in an array of shape (624, 1). Save this array in the file: `Assignment1/Q3/LogisticRegressionPreds.npz` under the key `test_preds`. **Predictions not meeting this format will result in zero marks for this question.**

Note: We will be calculating the F1-score of your predictions on the test data. Grading for this question will be partly relative after considering the mean F1-score of all the students who submitted the assignment.

Submission format:
`Assignment1_{name}_{last_5_digits_SRNO}/Q3/PCA_Logistic_Regression.ipynb` – should contain code for both PCA and logistic regression with all cell outputs.
`Assignment1_{name}_{last_5_digits_SRNO}/Q3/LogisticRegressionPreds.npz` – An array of binary predictions under the key `labels`.
All the plots and evaluation metrics must be added in the report.

Your directory structure must be as follows:

```
-Assignment1_{name}_{last_5_digits_SRNO}
    |-- Q2
    |-- Q3
    |-- report.pdf
```

After this, create a `.zip` file of the main directory, save it as `Assignment1_{name}_{last_5_digits_SRNO}.zip` and upload it on teams.