Task 1

Coding strategy:

We Implemented one function to compute dice coefficient, precision and recall as the three have common or overlaps in the implementation (the computation of true positives, false positives and false negatives). The function returns a matrix for each metric (dice coefficient, precision and recall) per patient per class and inside Dice, precision and recall functions, I compute the mean and standard deviation over the classes.

The formula for each metric

$$Dice\ Coefficient\ = \frac{2\,TP}{2TP + FP + FN}$$

$$Precision\ = \frac{TP}{TP + FP}$$

$$Recall\ = \frac{TP}{TP + FN}$$

a. What is the relationship between the Dice coefficient, precision and recall? What are their pros and cons?

Relationship between Dice, Precision, and Recall:

- The Dice coefficient can be seen as a balance between precision and recall because it considers both false positives and false negatives.
- High precision corresponds to a low rate of false positives, while high recall corresponds to a low rate of false negatives.
- The Dice coefficient is high when both precision and recall are high.

Pros and Cons:

	Dice Coefficient	Precision	Recall
Pros	Suitable for imbalanced datasets because it considers both false positives and false negatives and balances precision and recall.	Useful when minimizing false positives is a priority.	Important when minimizing false negatives is a priority.

Cons	It may not be the best choice when precision and recall need to be individually optimized. In other words, It does not distinguish between precision and recall and may not be suitable for all applications.	It may not provide a complete picture of a model's performance, especially when false negatives are important.	It may lead to a high number of false positives and precision may suffer as a result.

d. Other metrics. What other evaluation metrics are there for image segmentation? How do they differ from the ones implemented in Task 1a?

Intersection over union (IOU): measures the overlap between the predicted and ground truth regions.

$$IOU = \frac{TP}{TP + FP + FN}$$

Difference: While the Dice coefficient is the harmonic mean of precision and recall, IoU is their arithmetic mean.

Accuracy: measures the overall correctness of the segmentation, considering both true positives and true negatives.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Difference: Accuracy considers true negatives, making it suitable for balanced datasets but may not be informative in the presence of class imbalance.

Task 2 a

Coding strategy

Image Loading and Reshaping: The code loads medical images using load_nii and reshapes them to a flat 1D array.
KMeans Clustering: Utilizes the KMeans clustering algorithm from scikit-learn to cluster the voxel intensities into four clusters.
Finding Right Cluster Indexes: Generates combinations of four values (indexes) and applies them to the clustered labels, selecting the combination that maximizes the Dice similarity coefficient with the ground truth segmentation.
Performance Evaluation: Computes the Dice score by comparing the predicted segmentation with the ground truth segmentation.

Task 2 b

Coding strategy

Image Loading and Reshaping: The code loads medical images using load_nii and reshapes them to a flat 1D array.

Gaussian Mixture Model: Four different Gaussian distributions were created using the Gaussian Mixture Model algorithm from the scikit-learn.

Model Training: Training data has been used for training modeling and saved.

Model Testing: The test data was applied to the trained models one by one. The segmentation results were recorded. The best model results are shown.

Performance Evaluation: Computes the Dice score by comparing the predicted segmentation with the ground truth segmentation.

Task 3

Coding strategy:

The chosen UNet for this task is taken from the MONAI open-source library. This was done because the models in this library are focused on medical imaging tasks and the data loading and preprocessing is quite intuitanal. The UNet is configured to handle 3D spatial data with a single input channel (grayscale images). The output consists of four channels, corresponding to the four distinct classes in the segmentation task. The number of channels in each layer of the UNet is set as (4, 8, 16, 32) with residual connections (num_res_units=2). A stride of (1, 1, 1) is used, ensuring that the spatial dimensions are preserved throughout the network and no resizing of the images is needed. The DiceLoss function was used as it is common for segmentation tasks and is good at dealing with unbalanced datasets. The Adam optimizer was chosen with a learning rate of 1e-3, providing adaptive learning rate adjustments during training. No public pre-trained model weights are loaded, however, if there are available model weights from previous training, from this task, the training method provides the possibility to load these weights. This allows for the continuation of training from a saved checkpoint. During validation, the mean Dice coefficient is calculated for each iteration and compared to the best observed metric. The model with the best Dice coefficient on the validation set is saved. This allows us to fall back on the best parameter configuration after training for x epochs. It is important to note that the UNet was only trained for 3 epochs and not until convergence. This was due to no GPU

access and only training on CPU and due to the necessity of other tasks requiring computing the available CPU-time was limited.

Task 4 Analysis

a) What is the most intuitive approach to segment the images based on the density plot of the input?

Watershed Algorithm: Using thresholding and morphological operation to identify markers. Then apply a predefined watershed algorithm e.g. by cv2 library, then use the markers to segment the original image.

Thresholding:

- Global Thresholding: Choose a global intensity threshold based on Hounsfield Units to separate different tissue types. Pixels with intensities above the threshold belong to one class, while those below belong to another. One possible example, how to implement this, would be the Multi-Otsu's method, with k set to 4 in order to segment the 4 different classes.
- Adaptive Thresholding: Adjust the threshold locally based on the image content. This is useful when there is significant variation in intensities across different regions of the image.

(b) How did method 2a perform? Comment based on the quantitative and qualitative results.

=> It's important to note that the effectiveness of this unsupervised method heavily relies on the assumption that the ground truth segmentation can be represented by a combination of four clusters and on knowing beforehand how the clusters have to be indexed. The code iteratively tests various combinations of indexes and selects the one that best matches the ground truth.

Quantitative Results:

- Best Dice score: 0.87, Worst Dice score: 0.78.
- A Dice score of 0.87 suggests a substantial overlap between the predicted and true segmentation masks.

Qualitative Results:

- Strengths: KMeans clustering is a simple and widely-used unsupervised clustering algorithm. It tends to work well when the underlying data can be separated into relatively distinct clusters. The method is computationally efficient and easy to implement.

- Weaknesses: KMeans assumes that clusters are spherical and equally sized, which might not be the case for all types of image data. A big problem is also that the correct indexing of the clusters is normally not known through which performance measures as the Dice score lead to bad results.
- Sensitivity to initialization: Different initializations of the centroids may lead to different results.
- Possible Improvements: Experiment with different clustering algorithms that might better capture the structure of medical images. Consider incorporating spatial information or using more advanced techniques that take into account local context.

In summary, while the method shows promise with high Dice scores, there will be a variability in performance across different images when the groundtruth and therefore the correct indexing is not given. This suggests that there may be room for improvement, potentially through algorithmic adjustments or exploration of alternative clustering methods.

c) How did method 2b perform? Comment based on the quantitative and qualitative results.

=> Gaussian mixture model offers a probabilistic method for modeling data. Furthermore, Gaussian Mixture Models assign probabilities to the data points corresponding to each cluster to enable uncertainty estimation and confidence measures for each assignment. Test sets were evaluated for each model, and the model that yielded the best results was considered.

Quantitative Results:

- Best Dice score: 0.80, Worst Dice score: 0.46.
- Despite conducting experiments across all models, there is a noticeable difference.

Qualitative Results:

- Strengths: Gaussian Mixture Model (GMM) clustering adeptly captures intricate data distributions through a blend of Gaussian components. It enables soft assignment, allowing each data point to potentially belong to multiple clusters with varying probabilities.
- Weaknesses: Testing all the train data in each model has taken a considerable amount of time as well. As a result of this method, there is a significant difference between the highest and lowest performances.
- Possible Improvements: Using image indexing, similar to what we do in the K-means algorithm, can lead to higher success rates.

As a result, when we examine the Dice scores, although there is a difference, the Gaussian Mixture Model proves to be a good algorithm for prediction without labeling and for obtaining fast results. Better results can be achieved by increasing the dataset or using different machine learning or deep learning models.

d) Which unsupervised method performed better? Why?

Both methods perform equally well. The first approach relies on knowing beforehand how the clusters have to be indexed. That's why there is not such a variance in the performance in contrast to the second approach. In this method, we tried to test how the Unsupervised Learning strategy would perform on completely unseen images. Therefore, we first trained the entire training set and later, we tested the test set without any indexing. As a result, we could not achieve as good results as the K-Means algorithm.

(e) How did method 3 perform? Comment based on the quantitative and qualitative results?

Quantitative Results:

- Best Dice score: 0.91, Worst Dice score: 0.81.
- For 2 epochs trained the mean Dice score on the validation set still increased, but the resources were exhausted limiting further training

Qualitative Results:

Strengths:

- Segmentation Accuracy: The UNet model demonstrated commendable performance in segmenting brain structures from 3D MRI data, as indicated by the Dice coefficients ranging from 0.86 to 0.79. The segmentation accuracy, especially with the best Dice coefficient of 0.86, reflects the model's capability to capture intricate anatomical details.
- Architectural Flexibility: The architectural parameters are easy to adapt and provide a wide possibility of hyperparameter tuning -> Fine-tuning of Hyperparameters like stride, number of residual units, and number of channels should be explored in order to improve the Dice score.
- Small Dice difference between the highest and lowest Dice on test set (0.07) displays the strong robustness and generalizability to unseen data.

Limitations:

- Strides configuration: The stride configuration of (1,1,1) limits fast training and therefore training for longer epochs and training until convergence
- The high amount of learnable parameters require GPU access for efficient training which was not available for out team

(f) Which approach (classical or DL) performed better? Why?

Quantitatively the supervised learning method performed better, based on the comparison of worst-dice scores and considering that UNet was not trained until convergence. The better quantitative performance can be attributed to UNet being a deep learning architecture designed for semantic segmentation tasks, making it well-suited for this exercise. Unsupervised learning methods like KMeans and GMM are known to struggle with capturing nuanced and hierarchical structures, which are common in MRI brain scans. Both unsupervised methods group pixels based on intensity. The lack of explicit knowledge of the classes could explain the lower performance as understanding complex anatomical structures is crucial and not feasible without labels. Furthermore, the unsupervised learning methods rely on distance metrics for clustering instead of loss functions like UNet does, the distance metrics might not be sensitive to capturing detailed boundaries in the segmentations.

In this comparative analysis of unsupervised and supervised segmentation algorithms on brain MRI data, both KMeans (0.87) and UNet (0.86) outperformed the GMM approach (0.8) in terms of the best Dice coefficient. KMean's worst prediction on a test slide showed a lower Dice score (0.78) compared to UNet (0.79), suggesting an overall better quantitative performance across instances by the UNet as the range between best and worst Dice score is smaller for UNet. This indicates a more stable performance across different scenarios, whereas GMM showed larger differences (0.34), implying greater variability in segmentation quality.

The UNet's segmentation was observed to be grosser and less fine-grained compared to unsupervised approaches. Both KMeans and GMM produced very fragmented predictions, making it challenging to analyze and interpret the segmentation results effectively.

The lack of hyperparameter tuning and only small number of epoch training (3 epochs) for the UNet could explain why the Dice scores are not outperforming the unsupervised method KMeans. Fine-tuning hyperparameters, such as learning rates or the number of layers, could potentially enhance the UNet's segmentation accuracy.

The qualitative comparison shows that UNet's segmentations are easier to interpret as they are not as fine-grained and better group main structures together. To accurately evaluate which approach qualitatively performs best, a professional opinion needs to be considered. Our assessment is only based on shallow information about brain anatomy.

(g) What additional information in the volumes is used by the DL models compared to the unsupervised approaches in Task 2? Why is it helpful?

Additional Information in DL Models:

DL models, especially convolutional neural networks (CNNs) designed for medical image segmentation, can leverage several advantages compared to unsupervised methods:

- Learned Hierarchical Features: DL models automatically learn hierarchical features from the data, capturing complex patterns and relationships that may not be evident in voxel intensities alone.
- Spatial Context Awareness: CNNs can exploit spatial relationships between voxels through convolutional operations, capturing contextual information that may be crucial for accurate segmentation.
- Adaptability to Variability: DL models can adapt to variations in image appearance, pathology, and anatomy, making them more robust across different datasets and scenarios.
- End-to-End Training: DL models are trained end-to-end, optimizing for the segmentation task directly, whereas unsupervised methods may require manual tuning of parameters and assumptions.
- Incorporation of Multimodal Data: DL models can easily incorporate information from multiple imaging modalities, providing a more comprehensive view for segmentation tasks.

Why DL Models Are Helpful:

DL models excel in tasks where complex patterns, spatial relationships, and context play crucial roles. They can adapt to diverse datasets and capture intricate features that might be challenging for unsupervised methods, especially in medical imaging where subtle variations are significant.

In summary, while unsupervised methods like the ones provided can be useful in certain scenarios, DL models offer a more powerful and adaptive approach for medical image segmentation by automatically learning and leveraging complex patterns in the data.