

Practical 2 - Brain tissue segmentation

Team number: 21

Team members: Emil Alizada, Sofiene Boutaj, Felix Hsieh, Daniil Morozov

The problem statement:

The goal of the coursework is to implement classical and deep learning approaches for segmentation of different tissue types in MRI scans of the brain, i.e., background, cerebrospinal fluid (CSF), white matter (WM), and gray matter (GM). We were provided with data from a total of 652 healthy subjects (552 train / 65 validation, and 65 test).

Task 1:

The goal: to implement the Dice score to evaluate the performance of the segmentation algorithms.

Solution: As we didn't know which class for each tissue we get after feeding the model, we have implemented the Dice score based on the following idea:

- We have computed the Dice score for each class (4 classes) in predicted segmentations regarding each class in ground truth segmentations. Finally we got a 4x4 matrix for DS score.
- We chose the maximum value for each class in DS matrix, as it meant that the majority of pixels in predicted segmentation had the same class as ground truth segmentations.

```
def Dice_score(predictions, gt):
    TP = np.sum(np.logical_and(predictions == 1.0, gt == 1.0), axis=(1,2,3))
    FP = np.sum(np.logical_and(predictions == 1.0, gt == 0.0), axis=(1,2,3))
    TN = np.sum(np.logical_and(predictions == 0.0, gt == 0.0), axis=(1,2,3))
    FN = np.sum(np.logical_and(predictions == 0.0, gt == 1.0), axis=(1,2,3))
    dice = 2 * TP / ((TP + FP) + (TP + FN))
    mean, std = np.mean(dice), np.std(dice)
    return mean, std
```

Fig.1 - Function which computes Dice score.

```
def evaluate_prediction(prediction, segmentation):

    dice_score = {'csf': [], 'white matter': [], 'gray matter': [], 'back': []}

    for c in np.unique(prediction):
        dice_score['csf'].append(Dice_score(prediction == c, segmentation == 1.0))
        dice_score['white matter'].append(Dice_score(prediction == c, segmentation == 3.0))
        dice_score['gray matter'].append(Dice_score(prediction == c, segmentation == 2.0))
        dice_score['back'].append(Dice_score(prediction == c, segmentation == 0.0))
    # take max Dice-values
    for k in dice_score.keys():
        dice_score[k] = max(dice_score[k], key=lambda x: x[0])
    return dice_score
```

Fig.2 - Function which evaluates performance of the segmentation algorithms.

Task 2:

The goal: to implement two different unsupervised methods to leverage the different intensity profiles of the tissues.

(a) Task 2Q1: What is the most intuitive segmentation approach? (based on the intensity density plot of the input)

Based on the intensity plot we can clearly see that there is a considerable intersection area for the intensity across different tissue types. Given that different tissue types tend to be closer to one another, an algorithm that has spatial awareness should be the most intuitive. We can also add that the distribution that best describes the intensity density plot is Gaussian (we can see three gaussians with different means and standard deviation for each tissues)

(b) Task 2Q2

For the first unsupervised method, we used a clustering approach using Kmeans++ algorithm and got the following results for the test dataset :

Table 1 - Results of segmentation using Kmeans.

| | CSF | White matter | Gray matter |
|--------------------|------|--------------|-------------|
| Mean | 0.65 | 0.51 | 0.56 |
| Standard deviation | 0.04 | 0.4 | 0.29 |

(c) Task 2Q3

For the first unsupervised method, we used the most intuitive approach with density estimation approach using Gaussian Mixture model and got the following results for the test dataset :

Table 2 - Results of segmentation using Gaussian Mixture model.

| | CSF | White matter | Gray matter |
|--------------------|------|--------------|-------------|
| Mean | 0.67 | 0.76 | 0.78 |
| Standard deviation | 0.08 | 0.32 | 0.23 |

We can clearly see that we have better the results than with Kmeans

(d) Task 2Q4

Since Kmeans handles anisotropic data poorly, Gaussian Mixture model clearly outperforms Kmeans. We argued that an algorithm that can handle spatial information should perform better. Due to our data being highly anisotropic, Gaussian Mixture Model is more able to incorporate spatial information, thus performing better.

Task 3:

The goal: to use deep supervised training to segment the different tissue types. Implement and train a deep network architecture to segment the 4 tissues (CSF, WM, GM, and background).

(a) Task 3Q5: Implement, train, and evaluate a deep learning architecture for segmentation. Evaluate the Dice scores (separately for every tissue type) for the whole test set. What results do you get?

Solution: We have decided to choose the model 3D U-Net, which has the same architecture as original U-Net, but with 3D convolutions instead of 2D ones (<https://arxiv.org/abs/1606.06650>). It gets 3D MRI scan as an input and outputs one-hot 3D matrix which defines the class for each pixel. While training the model train set (552 scans) was fed to the network one time with 1 scan per iteration. Every 50th iteration the performance was evaluated using validation set (65 scans) and Dice score from 1st exercise. The results for each tissue were evaluated on the test (65 scans) set and are presented in table 3.

Table 3 - Results of segmentation using 3D Unet.

| | CSF | White matter | Gray matter |
|--------------------|------|--------------|-------------|
| Mean | 0.71 | 0.90 | 0.74 |
| Standard deviation | 0.09 | 0.02 | 0.05 |

(b) Task 3Q6: Which of the three approaches above (classical and DL) obtains better results? Why?

Solution: The results for each method are presented in table 4. Plots of tissue densities are shown in figure 3.

Table 4 - Comparison between models.

| Method | Parameter | CSF | White matter | Gray matter |
|------------------|--------------------|-------------|--------------|-------------|
| KMeans | Mean | 0.65 | 0.51 | 0.56 |
| | Standard deviation | 0.04 | 0.4 | 0.29 |
| Gaussian Mixture | Mean | 0.67 | 0.76 | 0.78 |
| | Standard deviation | 0.08 | 0.32 | 0.23 |
| 3D Unet | Mean | 0.71 | 0.90 | 0.74 |
| | Standard deviation | 0.09 | 0.02 | 0.05 |

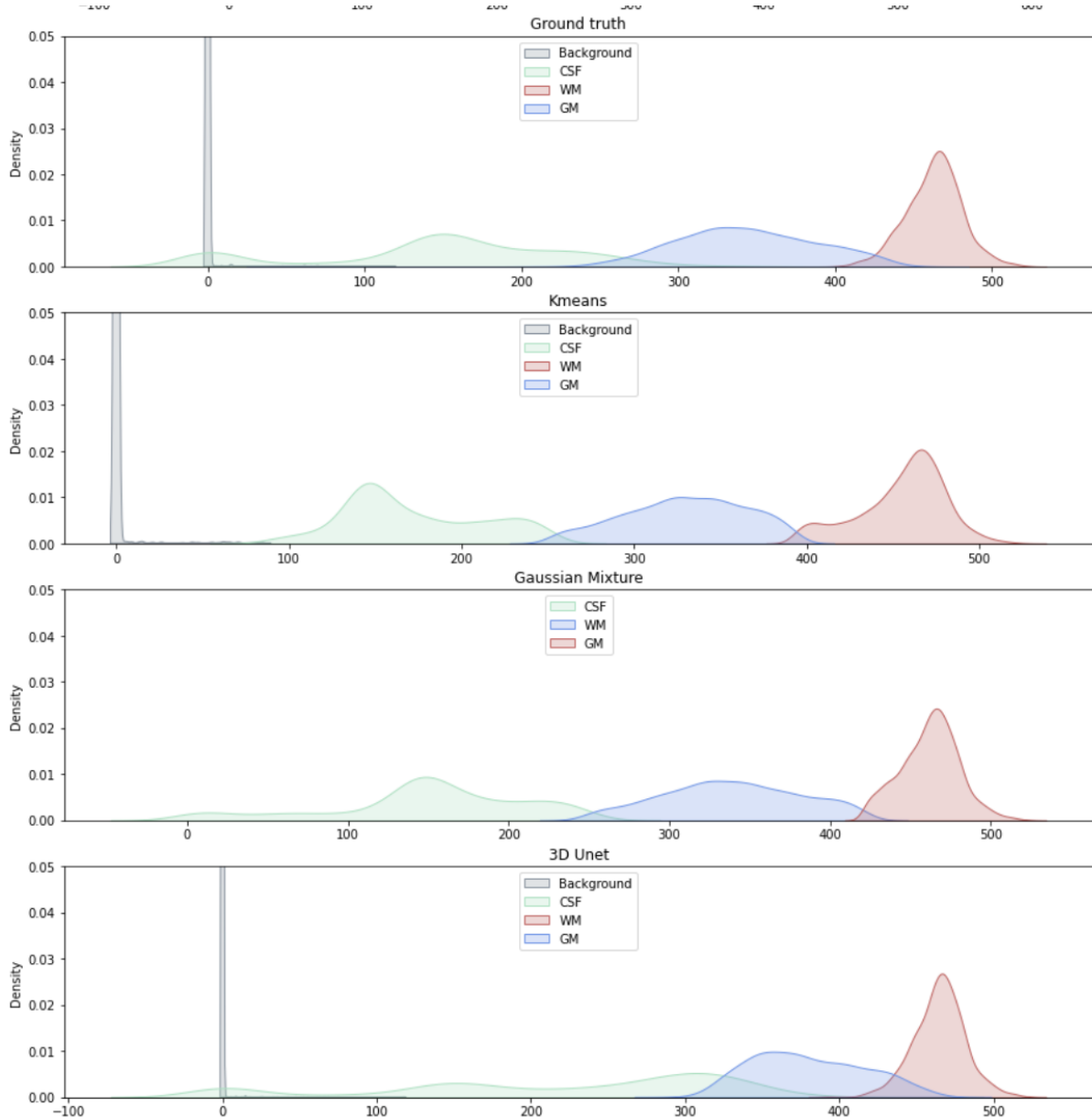


Fig.3 - Density estimations for each model.

From table 4 and fig.3 we can see that the performance of the deep learning model shows better results predicting CSF and WM in comparison to other approaches, though in our implementation works less better predicting GM. The reason why deep learning outperforms conventional machine learning models is because conventional models in our implementation use only one feature (pixel intensity). While ML models demand flattening the 3D volume to 1D vector for the input, DL approach allows to save all the spatial information due to use of convolutions.

(c) Task 3Q7: What extra-information in the volumes is used by the DL models compared to the unsupervised approaches in Task 2? Why is it helpful?

In that unsupervised approach we clearly saw improvement of the result when more spatial information is incorporated into the model. The extra information that CNNs

use is the spatial information that allows it to perform better than the latter.