Transfer Learning

Group 27
Ha Young Kim, Malek Al Abed, Marta Hasny,
Pascual Tejero Cervera, Ruxandra Petrescu

Task 1 : MONAI and Low Label Regime

1a : Full Data vs Low data

When there is not enough data, the training overfits quickly leading to faster convergence. This can be seen through the experiment shown in the table below that low data early stopped while the full data iterated through all the steps, which also gave a faster training time. However, since there is not so much data, the performance is worse than having the full dataset.

Additionally, when using low number of data, it can be seen that the performance is better when we have less weights (less number of layers). Therefore one can see that with less number of data, it is unnecessary to use many weights.

Performance was better with more number of layers with full data while with low data, the performance was better with a less number of layers.

Moreover, the residual unit improved the performance than without residual units.

	Full data	Low data
5 layer UNet num_res_units = 5 Patience = 5	Test loss: 0.2314 Test Mean Dice: **0.8647** Test Dice_CSF: **0.8456** Test Dice_WM: **0.8813** Test Dice_GM: **0.8673** 5000 steps (full)	Test loss: 0.5337 Test Mean Dice: 0.7304 Test Dice_CSF: 0.7093 Test Dice_WM: 0.7570 Test Dice_GM: 0.7249 550 steps (early stop)
5 layer UNet	Test loss: 0.2659 Test Mean Dice: 0.8438 Test Dice_CSF: 0.8204 Test Dice_WM: 0.8620 Test Dice_GM: 0.8489 5000 steps (full)	Test loss: 0.5336 Test Mean Dice: 0.7079 Test Dice_CSF: 0.7202 Test Dice_WM: 0.7335 Test Dice_GM: 0.6701 2100 steps (early stop)
4 layer UNet	Test loss: 0.2774 Test Mean Dice: 0.8392	Test loss: 0.4997 Test Mean Dice: 0.7178

	Test Dice_CSF: 0.8174 Test Dice_WM: 0.8564 Test Dice_GM: 0.8438 5000 steps (full)	Test Dice_CSF: 0.7196 Test Dice_WM: 0.7383 Test Dice_GM: 0.6955 1500 steps (early stop)
3 layer Unet	Test loss: 0.3018 Test Mean Dice: 0.8223 Test Dice_CSF: 0.7987 Test Dice_WM: 0.8410 Test Dice_GM: 0.8272 5000 steps (full)	Test loss: 0.4303 Test Mean Dice: 0.7701 Test Dice_CSF: **0.7514** Test Dice_WM: 0.7994 Test Dice_GM: 0.7596 2950 steps (early stop)
3 layer Unet num_res_units = 5 Patience 2	Test loss: 0.2419 Test Mean Dice: 0.8565 Test Dice_CSF: 0.8315 Test Dice_WM: 0.8748 Test Dice_GM: 0.8631 5000 steps (full)	Test loss: 0.4100 Test Mean Dice: **0.7800** Test Dice_CSF: 0.7510 Test Dice_WM: **0.8050** Test Dice_GM: **0.7840** 500 steps

1b. Early Stopping *Patience parameter testing*

Here, we changed the patience parameter to 2, 5, and 7. For training, low number dataset was used for faster training, and UNet with 3 layers was used.
With the patience parameter, when the patience parameter is low, it stops early sometimes even when it doesn't converge fully. Therefore with lower patience parameters, the test accuracy can get worse. As shown in the table below, the results are better when patient parameters are high. However it comes with a tradeoff of longer training time since it goes through more steps.
However, higher patience doesn’t always mean the accuracy is going to improve since the model will start overfitting at some point. One can see that the result of patience 2 is better than patient 5. This might be because of the validation loss growing or there are some fluctuations in the training. Therefore, it is important to tune this parameter according to the model to get a better result.

patience = 2 Early stopped, Finished training after 1800 steps	Test loss: 0.4051 Test Mean Dice: 0.7671 Test Dice_CSF: 0.7490 Test Dice_WM: 0.7833

	Test Dice_GM: **0.7690**
patience = 5 Early stopped, Finished training after 2350 steps	Test loss: 0.4372 Test Mean Dice: 0.7674 Test Dice_CSF: 0.7484 Test Dice_WM: 0.7965 Test Dice_GM: 0.7574
patience = 7 Early stopped, Finished training after 3050 steps	Test loss: 0.4111 Test Mean Dice: **0.7732** Test Dice_CSF: **0.7602** Test Dice_WM: **0.7960** Test Dice_GM: 0.7633

Task 2 : Autoencoder and Transfer Learning

2a Train an autoencoder

Used a Unet with 6 layers, 3000 steps. As shown in the picture below, the result looked reasonable. This model was saved to further be used for transfer learning.
Results of the autoencoder are visualized in table in *section 2c*.

2b Transfer learning

Used Unet with 3 layers, Low data, num_res_unit = 5, 500 steps, patience = 5
Used small number of layers with res_unit since this was the model that performed the best during the training of 1a. Additionally with low data, it was performing better with less number of layers, therefore we chose this model.

Frozen	Non frozen
Test loss: 0.4308 Test Dice: 0.7507 Test Dice_CSF: 0.7400 Test Dice_WM: 0.7786 Test Dice_GM: 0.7337	Test loss: 0.3938 Test Dice: **0.7817** Test Dice_CSF: **0.7591** Test Dice_WM: **0.7897** Test Dice_GM: **0.7963**

We can determine that the model with non-frozen encoder weights makes a better performance than the model with frozen encoder weights. This is because in the model of frozen weights, we only allow training on the decoder weights but probably this model would need further training also in the encoder weights.

This fact can be seen in the test results from the non-frozen model, where the values are in fact lower than the model without transfer learning (trained from scratch). It is a good idea to initialize the encoder weights with the ones from the previous trained model, but more training is necessary afterwards to make a better segmentation.

2c Experiments

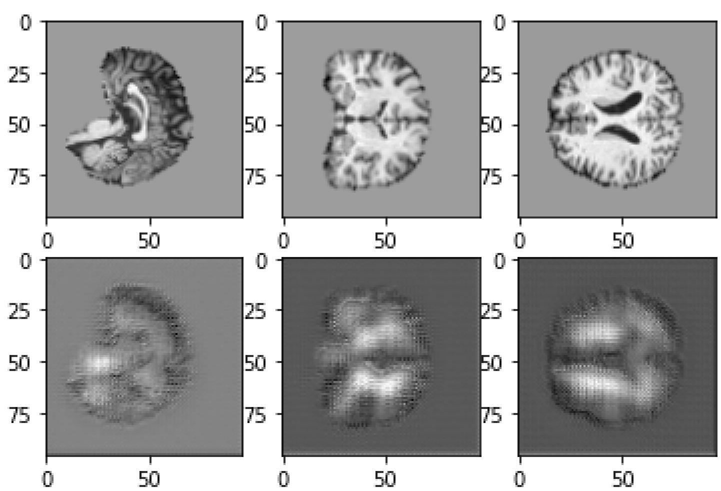
Autoencoder performance

The performance was tested on step 500, 1000, 1500, 3000. As seen in the result below, more steps ran it showed better performance in reconstruction.

However one should consider the tradeoff of having more steps with having longer training time. Therefore tuning the number of steps would be important.

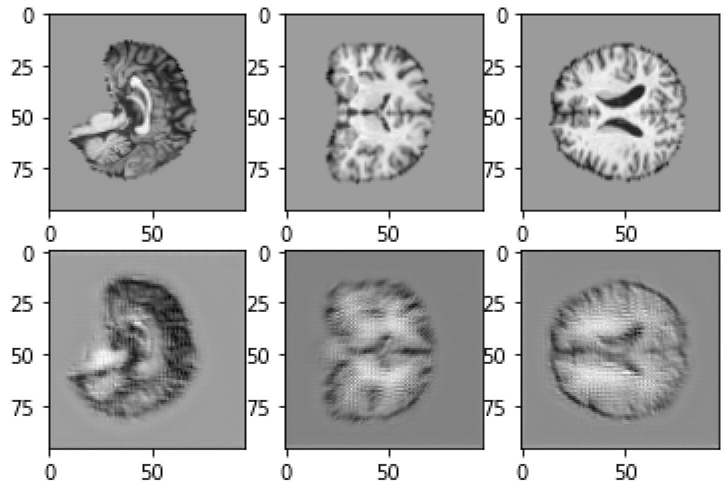
500 Steps

Train loss: 0.00840084102936089 - val loss: 0.0079 - val MAE: 0.0446

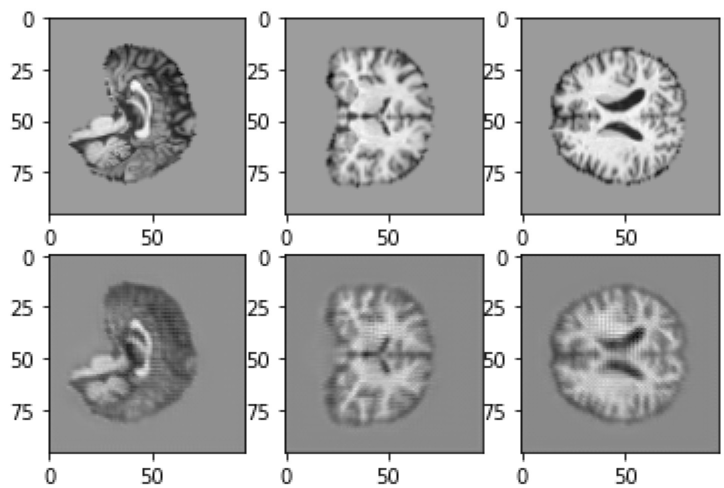


1000 steps

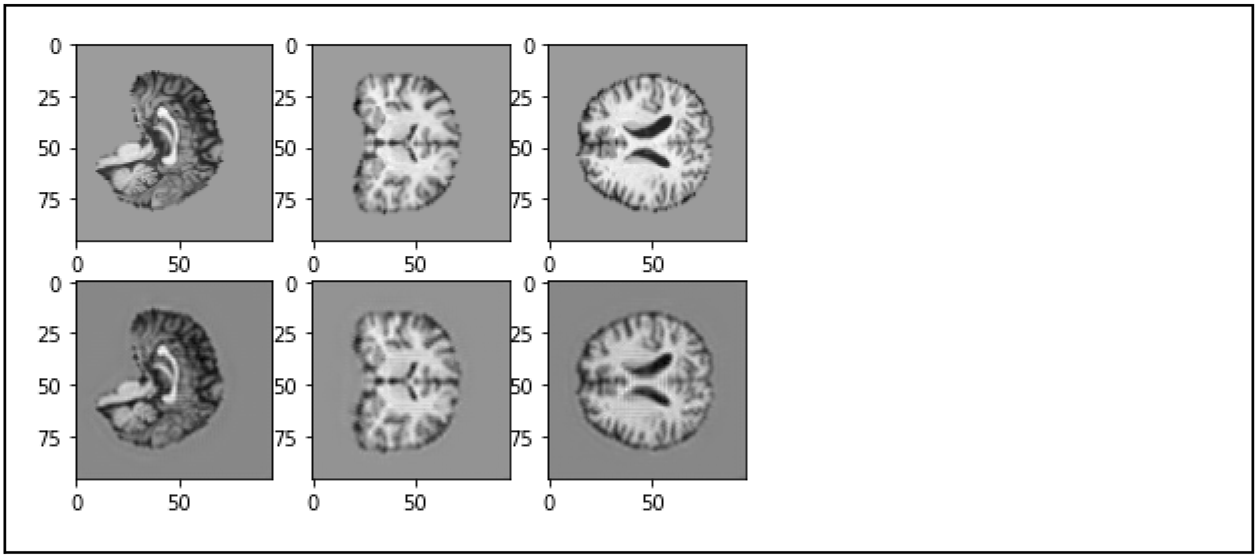
Train loss: 0.004552807547152042 - val loss: 0.0043 - val MAE: 0.0310



1500 steps
Train loss: 0.0021378234284929933 - val loss: 0.0021 - val MAE: 0.0203



3000 steps
Train loss: 0.0007841951330192388 - val loss: 0.0008 - val MAE: 0.0120



0
25
50
75
0
50