Machine Learning for Imaging – Coursework Report
Age Regression from Brain MRI

Group: **54**

Tanisha Mandre, Alexander Rader, Doruk Taneli
{tm520, apr20, dt420}@imperial.ac.uk

1 Part A

1.1 Task A-1: Brain tissue segmentation

We used a resolution of 96x96x96 for the MRI scans with a spacing of 2 and batch size of 2. Anything higher lead to GPU memory running out. We used the preprocessing pipeline given, which also normalises the image to zero mean and unit variance. A resulting sample image is shown in Figure 1.

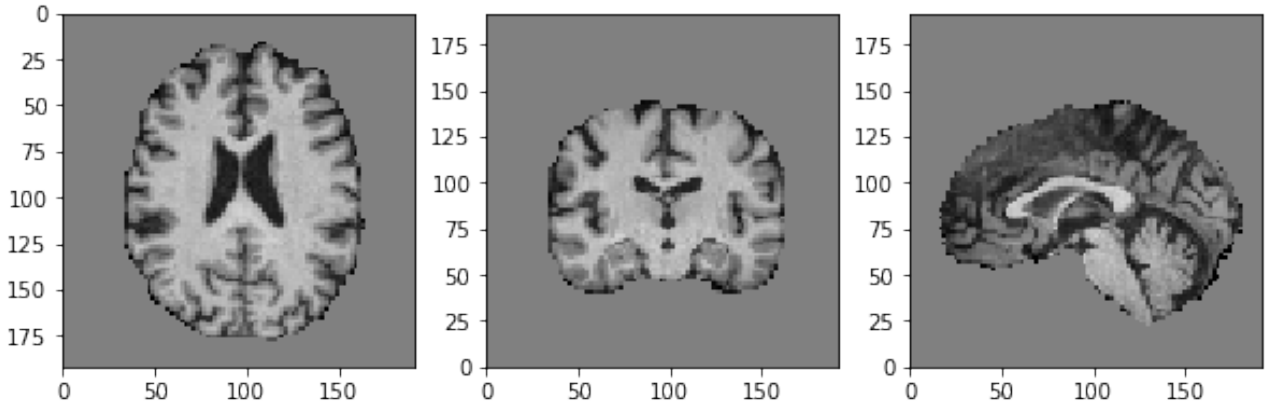

Figure 1: Example image after preprocessing

We trained for 100 epochs with a learning rate of 0.001. The network architecture is a U-net with 2 contraction and 2 expansion layers and a bottleneck in between. Each contraction and expansion block consists of a convolutional, batch norm and ReLU layer. The bottleneck includes 2 convolutional layers. Between blocks we used max pooling to downsample and transpose convolutions to up sample the image. We included two skip-connections between the contraction and expansion blocks. The skip connections make U-nets particularly suitable for segmentation, as higher resolution image data can be used in later layers. You can see the train and validation loss during training in Figure 2.

The loss consistently decreases and plateaus by the end of the 100 epochs. Moreover, the validation loss is very close to the train loss, so the network is generalising well. The total test cross entropy loss is 0.095532. We also calculated the dice scores for different tissues on the test set, which is shown in Figure 3. The average score was always above 0.8, showing high segmentation overlap. However, the box plots show that CSF was significantly harder to accurately segment than WM. An example segmentation is shown in 4.

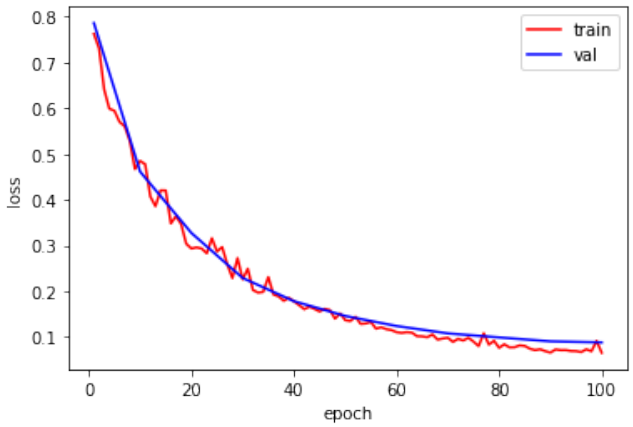

Figure 2: Segmentation loss for A1

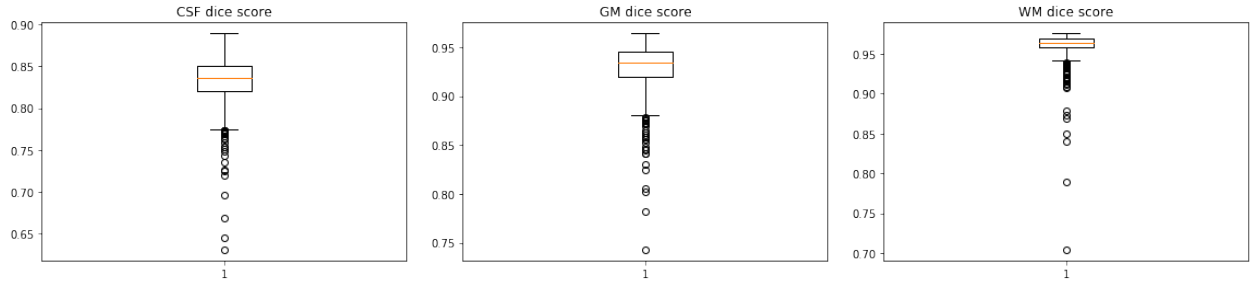

Figure 3: Dice scores from the test set

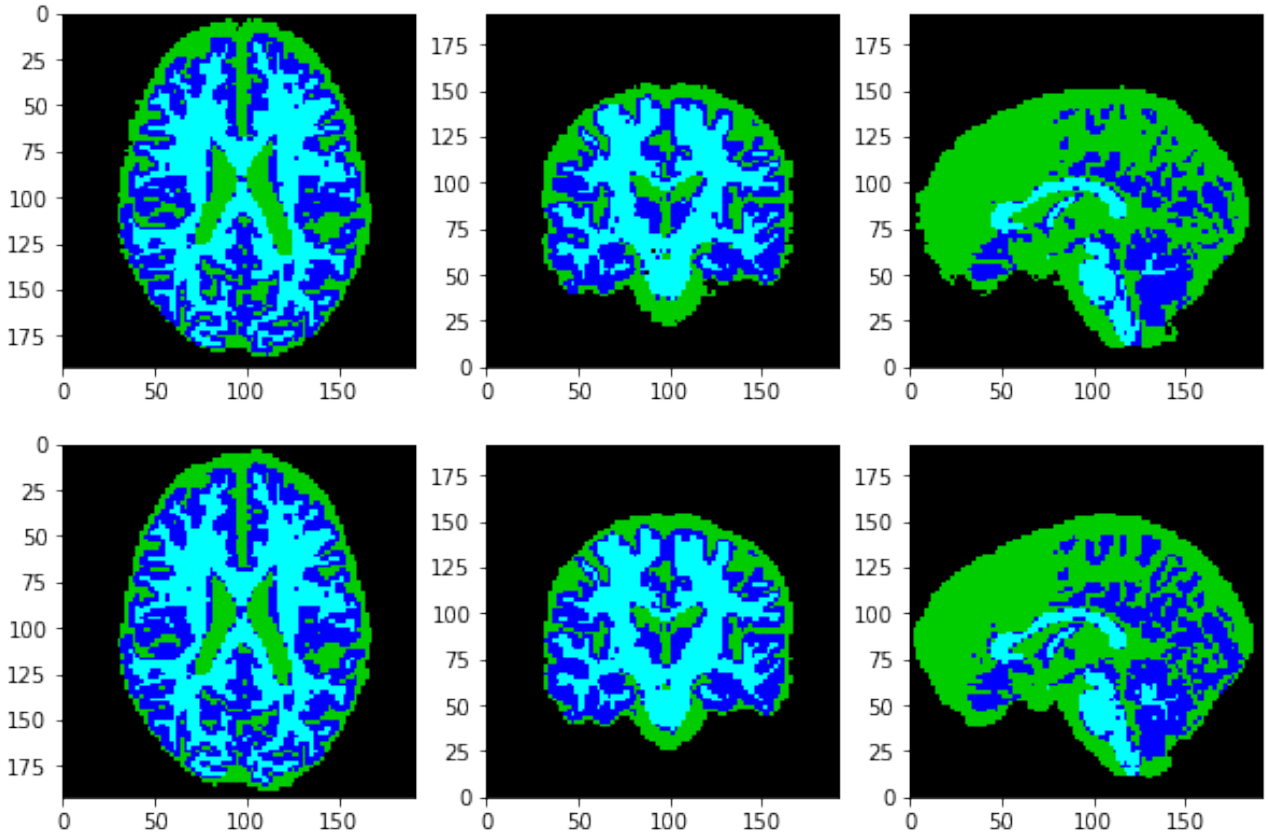

Figure 4: Reference(top) and Predicted(bottom) segmentations for a test sample

1.2 Task A-2: Feature calculation

Using the segmentations from A-1, we first calculate the total volume of CSF, GM, and WM, then plot the volumes vs real age. It is easier to see the trends when we normalize by calculating the volumes relative to the others, as seen in Figure 5.

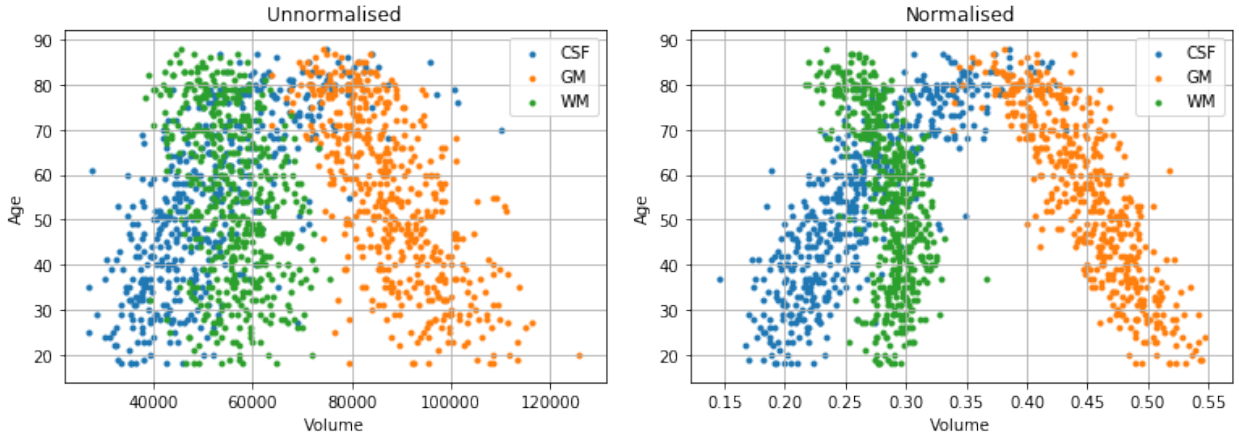

Figure 5: Feature calculation plots

There are clear trends for CSF and GM. WM volume doesn't give much information on its own, but it will help predict the age when used in combination with CSF and GM.

1.3 Task A-3: Age regression and cross-validation

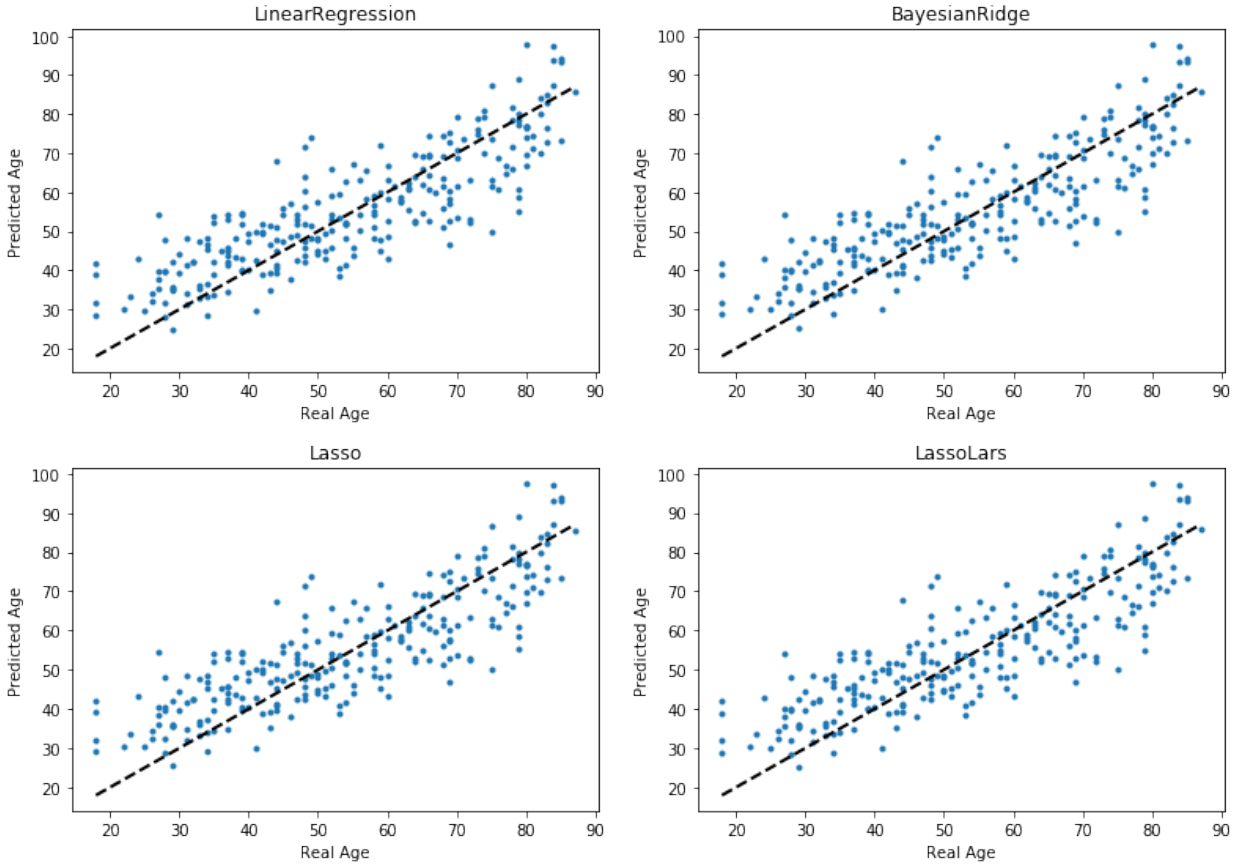

Figure 6: Predicted vs Real Age for training on Fold 1 and testing on Fold 2 for 4 regression techniques

Using the normalised X & y obtained from A-2, four different regressors were used to conduct

age regression including ***Linear Regression, Bayesian Ridge, Lasso(alpha=0.01) & Lasso-Lars(alpha=0.01)***.

Two fold cross validation was conducted by using the sklearn KFold library. Here the data was split into 2 folds. The data was randomly shuffled before the split to ensure that there were no biases in the training and test sets. The same split was used to test the performance of all 4 regressors. First Fold 1 was used as the training set while fold 2 was used as the test set to evaluate performance. This was then repeated using he opposite folds as training and test sets respectively.

We found that Linear Regression & Bayesian ridge had higher Mean Absolute errors and lower R2 Scores as compared the the other 2 regressors, which can be seen in figure 7. LassoLars(alpha=0.01) was chosen as the best regressor as it had errors that were more or less an average of the other regressors and didn't bias towards either of the 2 error metrics used.

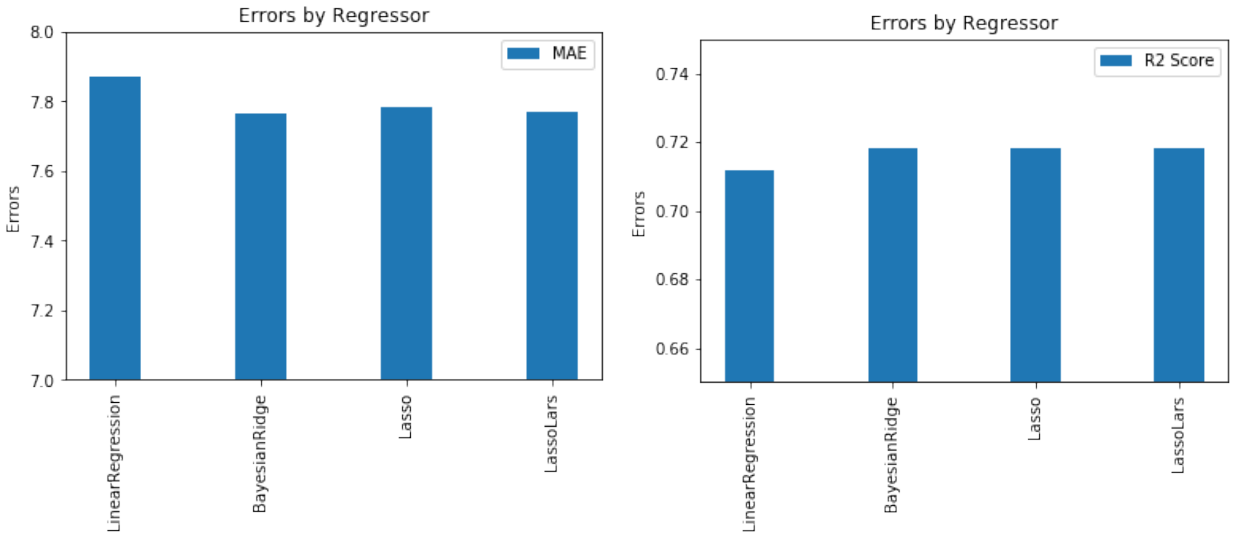

Figure 7: Comparing performance of 4 regressors using Mean Absolute Error (left) & R2 Score (right)

2 Part B

Model

We used the same preprocessing for images as in part A, so you can refer to Figure 1 for an example. To be able to access the reference age during training and testing, we created a class ImageRegressionDataset in the style of ImageSegmentationDataset, which also outputs the age in the getitem method. The model chosen was a simple CNN with 3 convolutional layers. Between each layer we used batch norm for regularisation, maxpooling for decreasing the image size and ReLU as the nonlinearity. We also increase the number of channels up to 256, before we use a linear layer to output the predicted age.

Training

To perform 2-fold cross validation we split the data set into 2 different dataloaders of 250 samples each and used a batch size of 2. The dataset was shuffled to ensure there were no biases in the training set.

The model was trained using a CNN for up to 30 epochs. For 2 fold validation, it was trained on 1 fold while testing on the other and vice versa. The results are as seen in Figure 8. It can be observed that the loss on the training data decreases at a steady rate however for the training data it plateaus much more quickly. At around epoch 25, it slightly increases, showing signs of overfitting. The 2 folds have a L1 loss (Mean absolute error) of 7.05 and 6.25 respectively giving the average 2-fold validation a loss of 6.65.

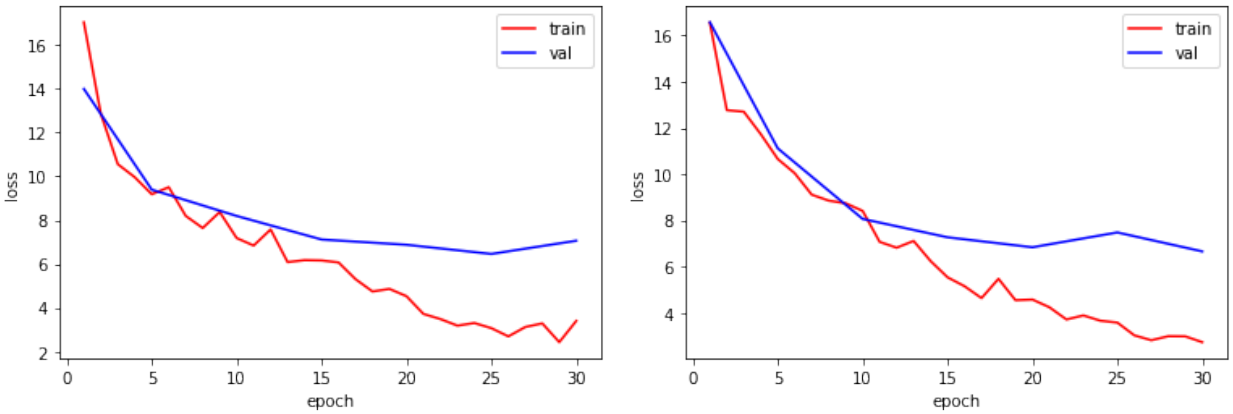

Figure 8: Loss on training Fold 1(left) & Fold 2(right) with epochs for 2 fold cross validation

On the other hand on training the CNN on the entire dataset of 500 samples, a much better loss of 4.69 was obtained on the final validation test set of 47 samples. This can be seen in Figure 9. Hence while 2-fold cross validation helps better estimate the performance, being able to use the entire training set improves the performance of the model further.

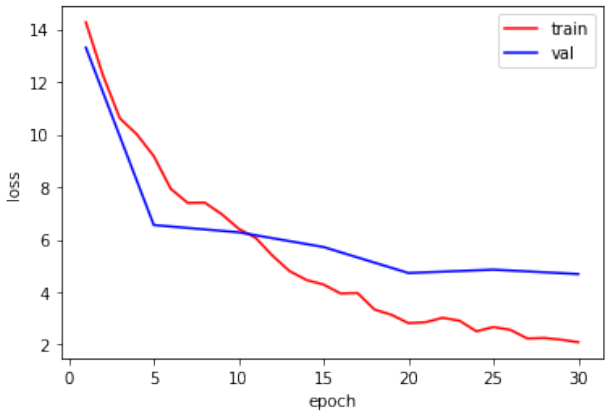

Figure 9: Loss after training on entire 500 training samples and testing on test samples

3 Age Regression Results

3.1 Part A: Final test on hold-out data

We ran the code for tasks A-1 to A-3 in succession using the hold-out dataset to get these final results. We segment the brain successfully as can be seen in Figure 10. We achieve dice scores comparable to the test set in A1. The full dice scores are shown below in Figure 11. Then we calculate the normalized brain volumes. When we compare these volumes versus age in Figure 12, we see the same trends we observed in Task A-2. We then use the best regressor we optimized in Task A-3 to predict the age from the normalized brain volumes. We achieve a mean absolute error of 7.996 and an R2 score of 0.762.

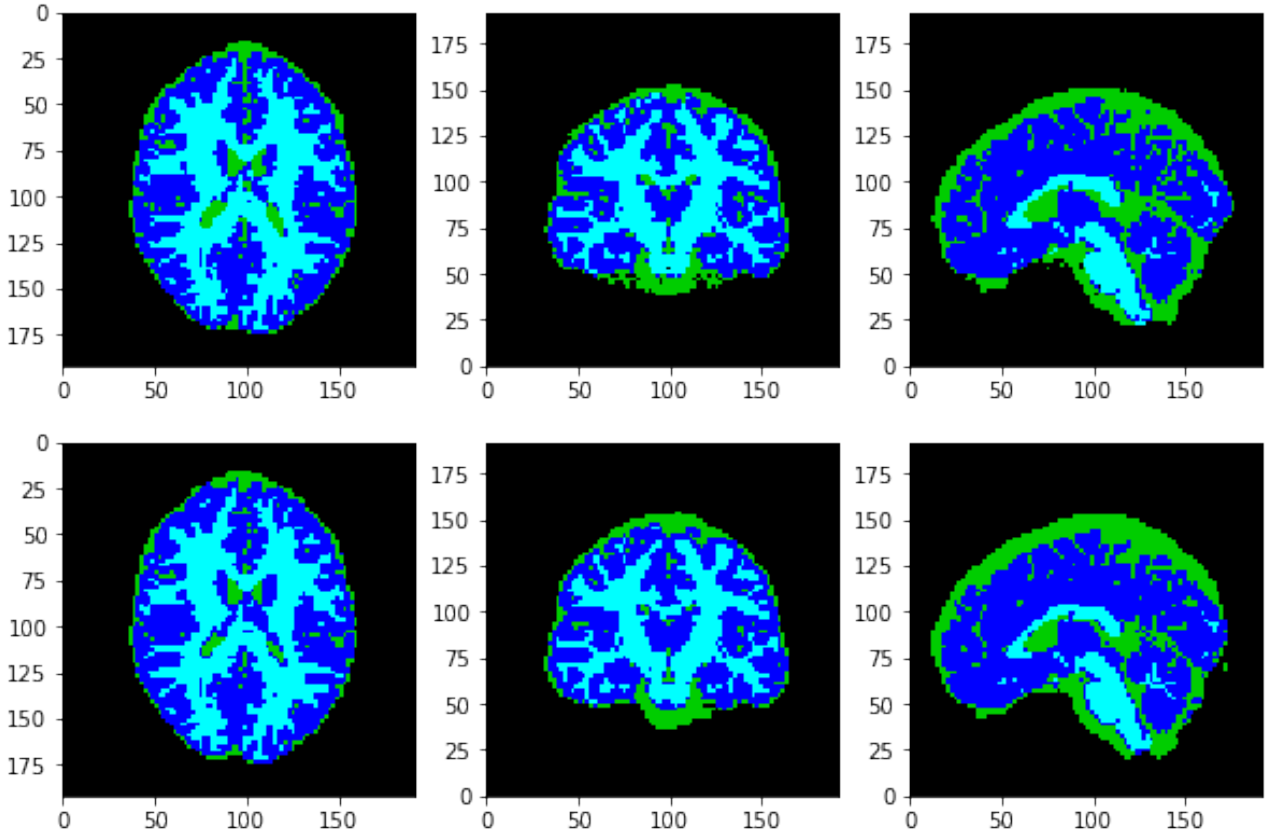

Figure 10: Reference(top) and Predicted(bottom) segmentations

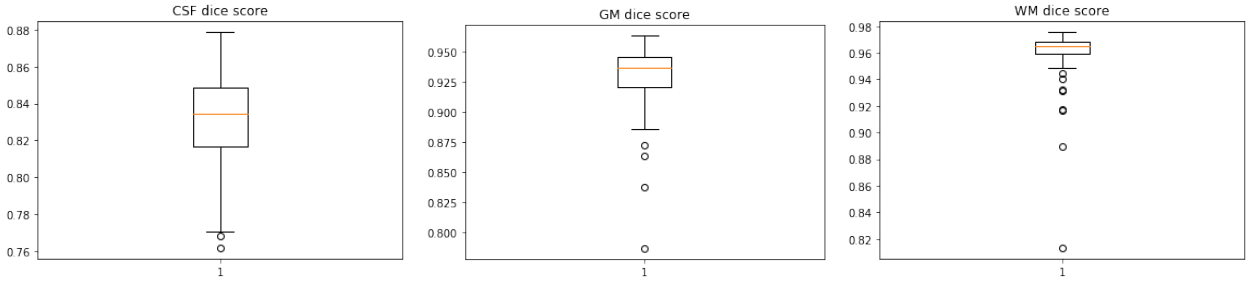

Figure 11: Dice scores from the hold-out set

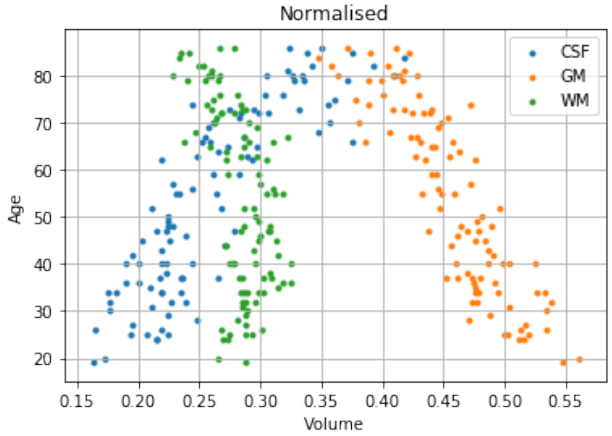

Figure 12: Normalized features vs Real Age

3.2 Part B Results

We achieve a mae of 5.746 on the test set. The scatter plot for these results can be observed in Figure 14.

3.3 Results Comparison and Discussion

After training the models of part A and B with the whole 500 instance dataset, we test it on the training dataset and get the plots in Figure 13, then test it on the held out data to get the plots in Figure 14.

From these figures, we can see that Part B is more effective. The correlation between real and predicted ages is visibly much stronger for the CNN approach. There are several reasons for this. First, the segmentation in Task A-1 is a bottleneck as it is only trained with 47 subjects. The CNN is trained on 500 samples, providing much more data. Second, In Part A, we manually select the features which are the normalized brain volumes that the regressor will be trained on. There might be better features for age regression other than volumes. But hand-crafting features is slow and difficult. Whereas in part B, the CNN is able to use the raw image data to directly predict ages. This showcases the advantages of end-to-end models: We do not have to hand craft any features, backpropagation will find good representations automatically and is able to decrease loss considerably compared to hand-crafted features.

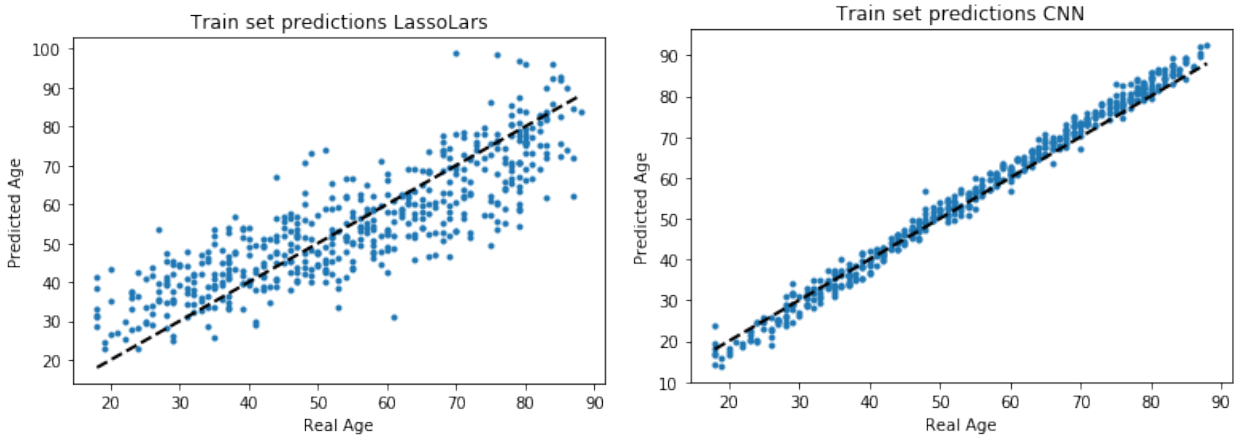

Figure 13: Predicted Versus real age for training set Part A(left), Part B(right)

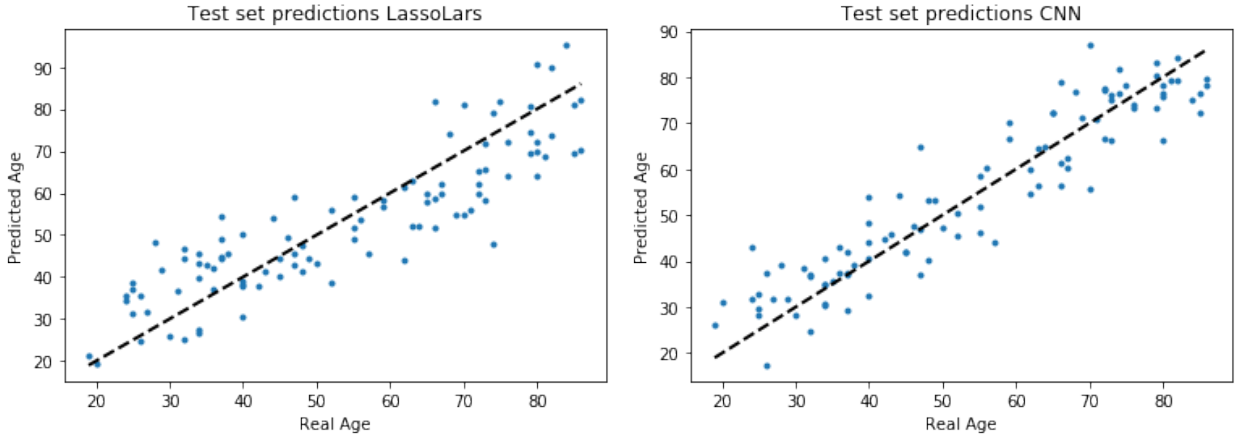

Figure 14: Predicted Versus real age for test set Part A(left), Part B(right)