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

Group Leader: **Wang**

Zhaoyu Wang, Yanda Tao
{zw1222, yt522}@imperial.ac.uk

1 Part A

1.1 Brain Tissue Segmentation

1.1.1 Model Selection

In the first step, in order to find the volume of the 4 types of brain tissues, the four-class brain segmentation is realised by a encoder-decoder based CNN model, the U-Net. Figure 1 shows the overview of the model architecture.

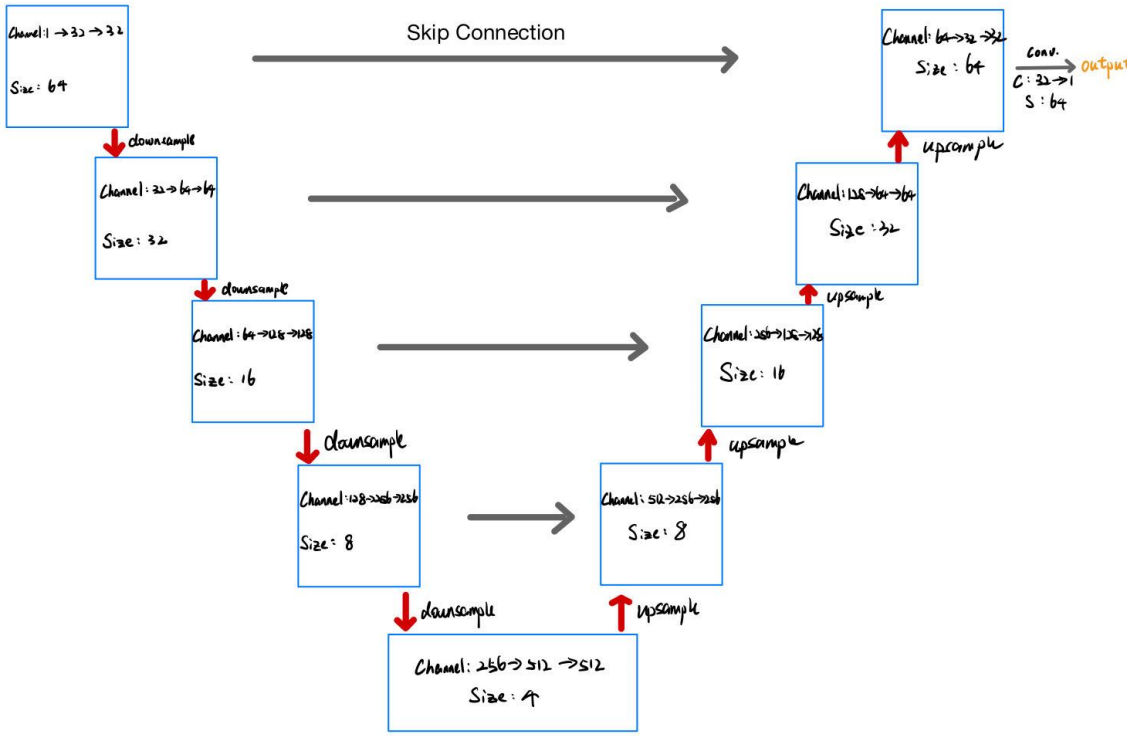
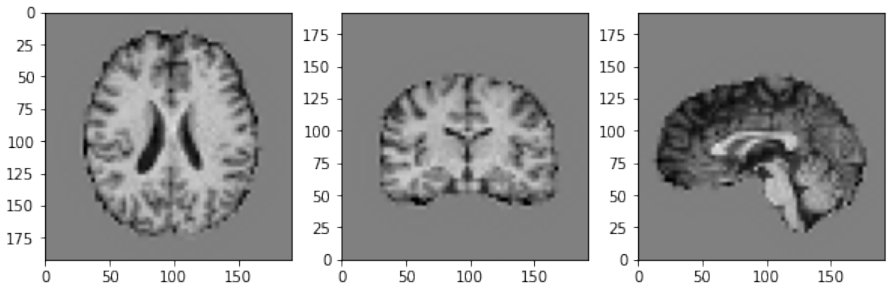


Figure 1: the overview of U-Net structure applied in this task

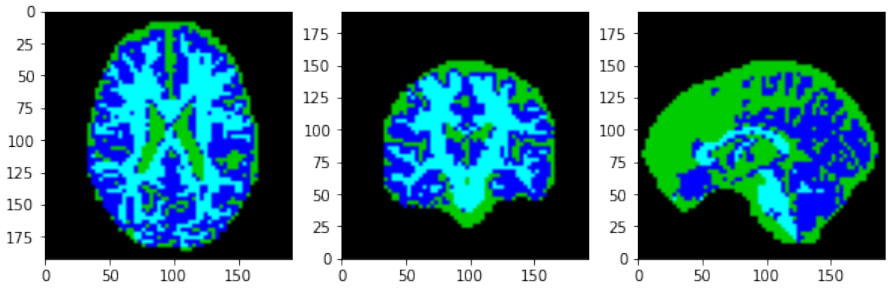
1.1.2 Hyperparameter Selection and Data Preprocessing

The model is trained with the provided dataset with the size of 64×64×64 and the spacing of 3×3×3. To prevent the model from over-fitting and under-fitting, the model is trained with 20 epochs, *batch*

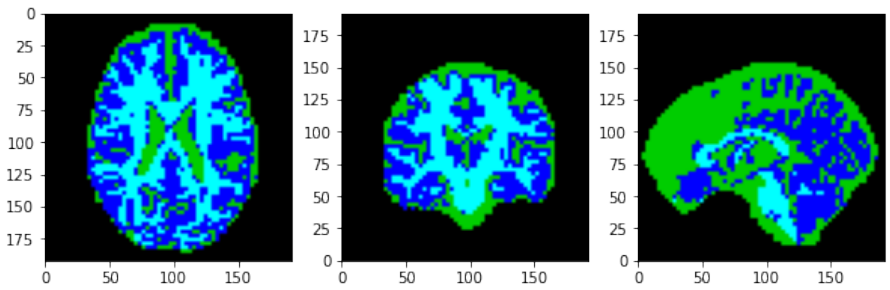
$size = 2$, using Adam optimiser with $learning\ rate = 0.001$. The data is loaded, normalised, resampled in the preprocessing step, and fed to the U-Net model. Figure 2 shows an example from the preprocessed dataset.



(a) Preprocessed Trained Image



(b) Segmentation Result



(c) Mask of the Visualised Image

Figure 2: Preprocessed Training Data Visualisation

1.1.3 Training, Validation and Testing

The learning curves of both training and validation are shown in figure 3. The same preprocessing method is applied to the 500 training set for the age regressor training which is used for testing the segmentation model in this stage, and the trained model is then tested with the preprocessed data. One of the segmentation result is shown in figure 4.

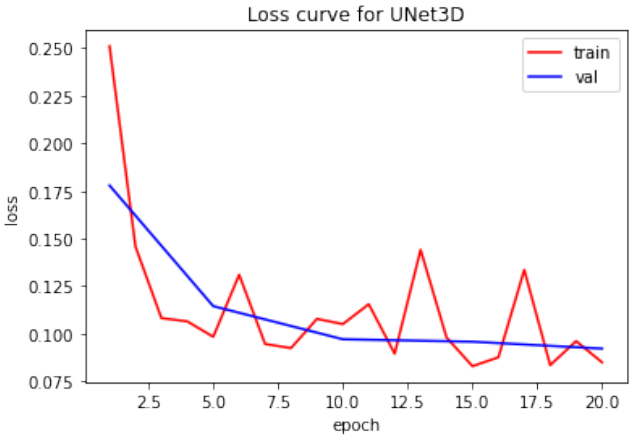
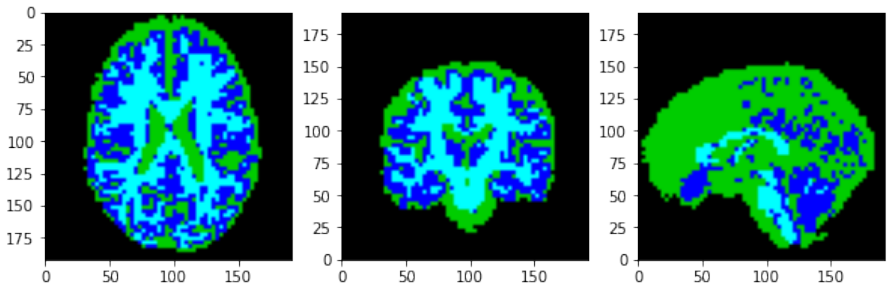
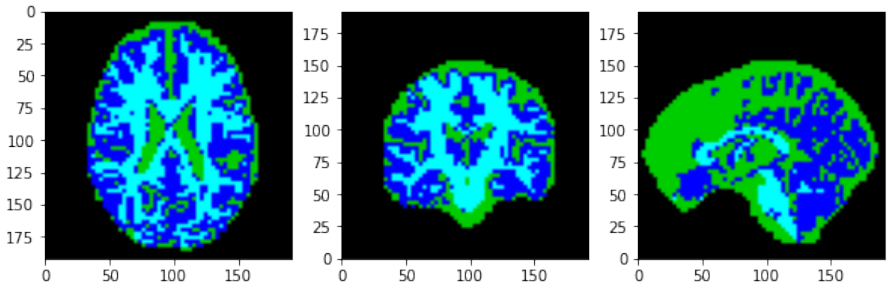


Figure 3: Learning Curve of the Selected Model



(a) Reference Tissue Segmentation



(b) Predicted Tissue Segmentation

Figure 4: Brain Tissue Segmentation Visualisation

Not only the cross-entropy loss (classification task, whether the pixel is classified into the correct segmentation) but also the Dice score are used to assess the model. We obtain the following error metrics on the training set for the age regressor:

- **Average Cross-Entropy Loss**: 0.109
- **Dice Score**: Illustrated with the box plot in figure 5.

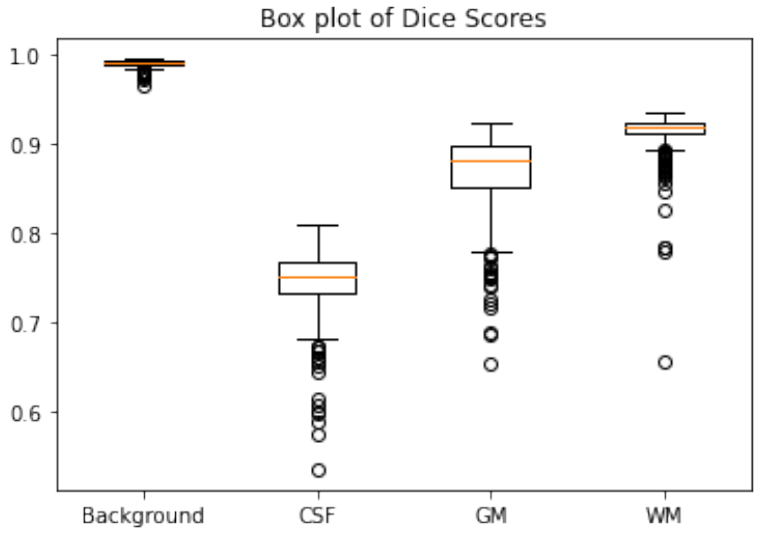


Figure 5: segmentation performance: the Dice Score representation

1.2 Feature Calculation

The volume of each piece of tissue can be treated as the total number of the occurrence of its class in the output segmentation image. Using the volumes of the three tissues calculated in the previous step, the relationship between patient's age and his/her volumes of brain tissues, CSF, GM, and WM are demonstrated by the scatter plot in figure 6.

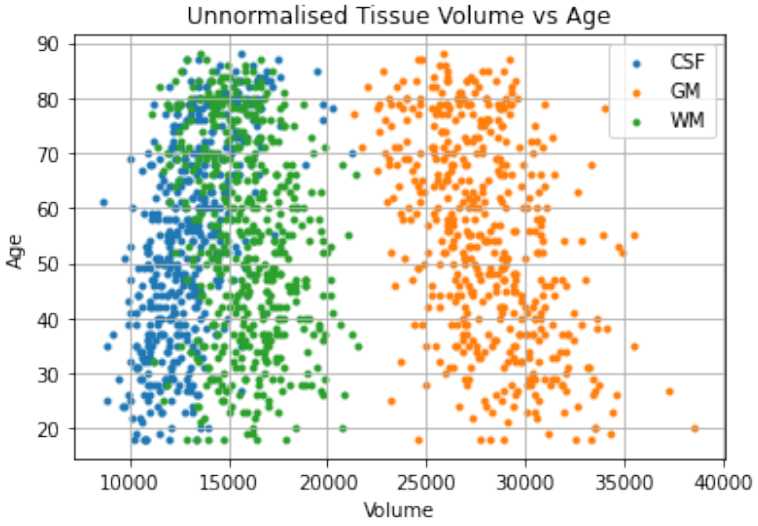


Figure 6: the relationship between tissue volumes and age

The relative brain tissue volume is a more reasonable feature and is then computed by calculating the ratio between each tissue volume and the overall brain volume. Its relationship with age is displayed by the scatter plot at figure 7. This relative volumes and the corresponding ages are used as the data for the following regression task.

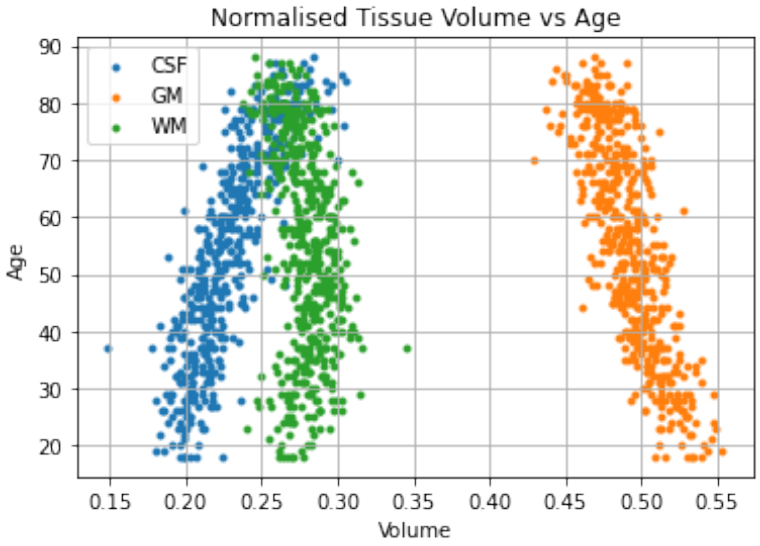


Figure 7: the relationship between relative tissue volumes and age

1.3 Age Regression and Cross-Validation

Different types of regressors are applied and their performance are indicated by cross-validation with mean-absolute-error and R2 score metrics. In this task, linear regression, polynomial regression with order of 3, support vector regressor (SVR), and the Bayseian Ridge regressor are trained and assessed in this regression task. We import these models, pre-process the data into polynomial features for polynomial model and perform 2-fold cross-validation . The best model is the polynomial regression with the order of 3 based on the mentioned performance metrics. The result of cross-validation is shown in figure 8.

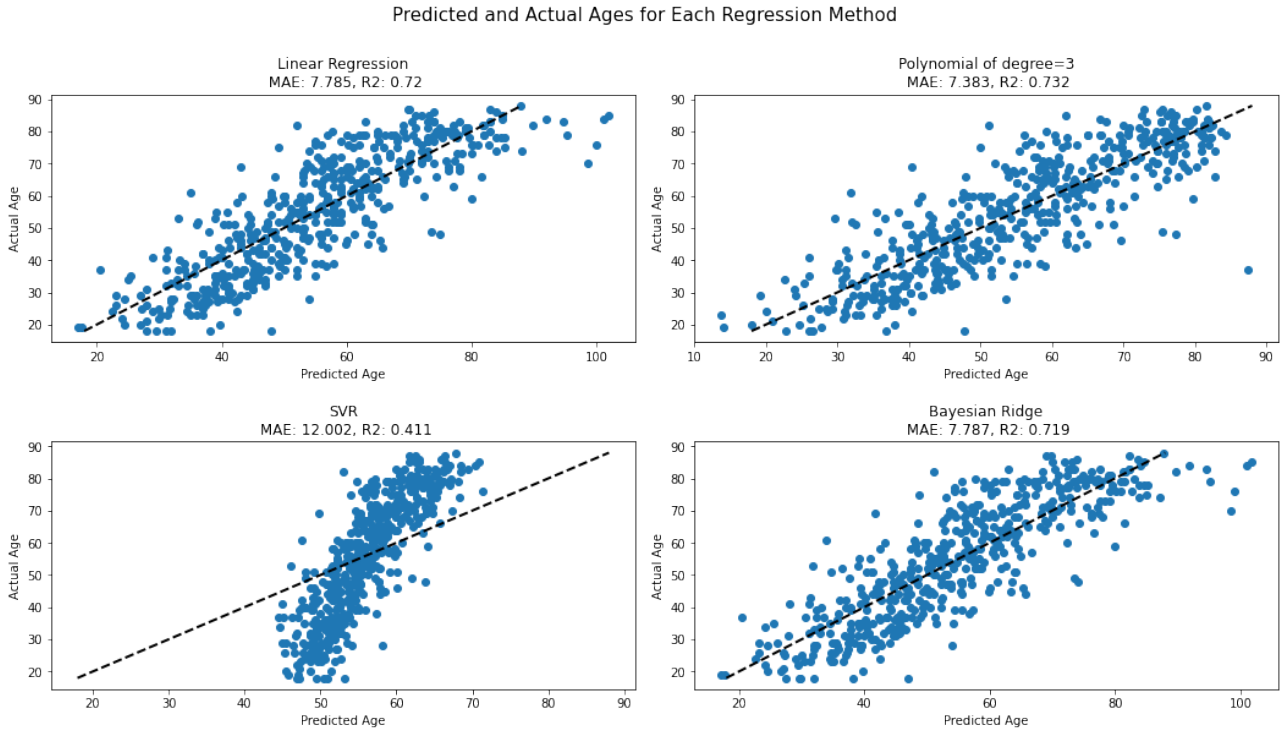
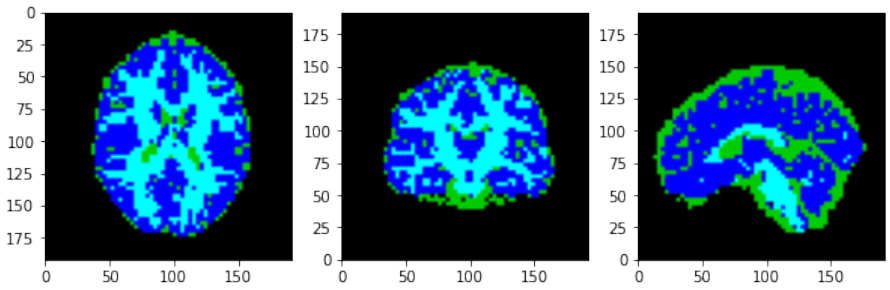


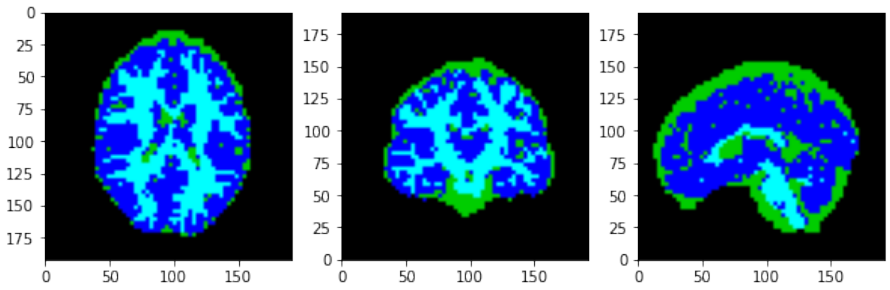
Figure 8: Predicted Ages vs Actual Ages

1.4 Final Test on Hold-out Data

Following the data loading and preprocessing steps, the segmentation model is tested with the final test hold-out data. One of the preprocessed test data and the corresponding predicted segmentation is shown at figure 9.



(a) Reference Segmentation from Final Test Data



(b) Corresponding Predicted Tissue Segmentation

Figure 9: Brain Tissue Segmentation Visualisation with Final Test Data

We obtain the cross-entropy loss and dice score from this testing for segmentation:

- **Average Cross-Entropy Loss**: 0.097
- **Dice Score**: The box plot is shown in figure 10.

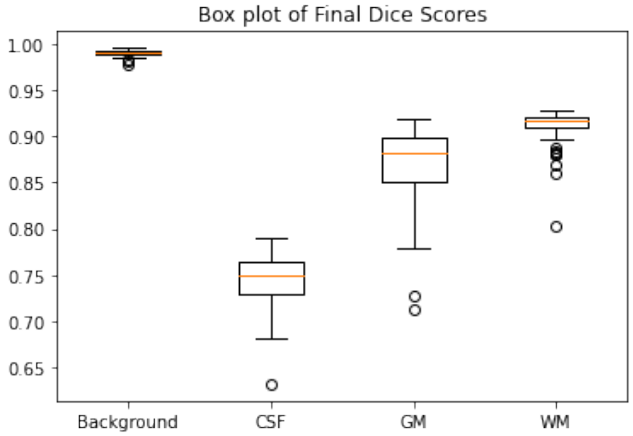


Figure 10: Box Plot of Dice Score with Testing on Hold-out Data

The relative volumes are then calculated by the selected regression model with the segmentation result above. The error metrics are shown below:

- **MAE Loss**: 6.924
- **R2 Score**: 0.803

The Regression result is shown in figure 11.

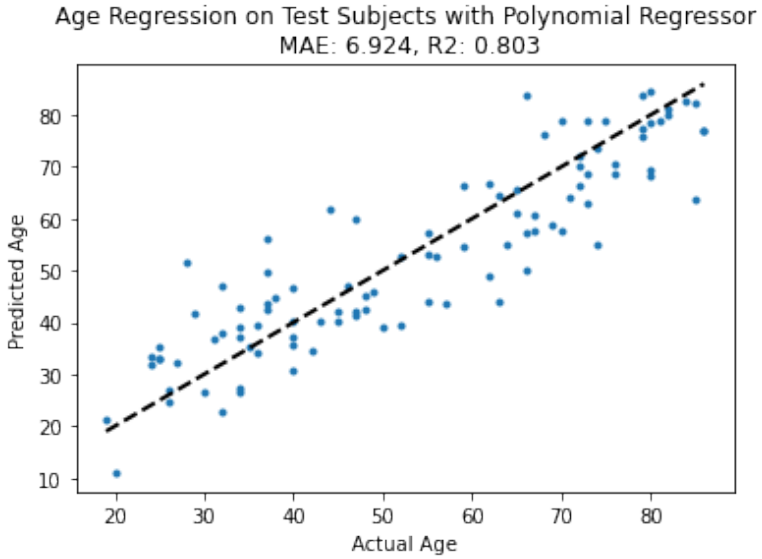


Figure 11: Age Regression on Final Test Data

2 Part B

2.1 Visualise training example after pre-processing

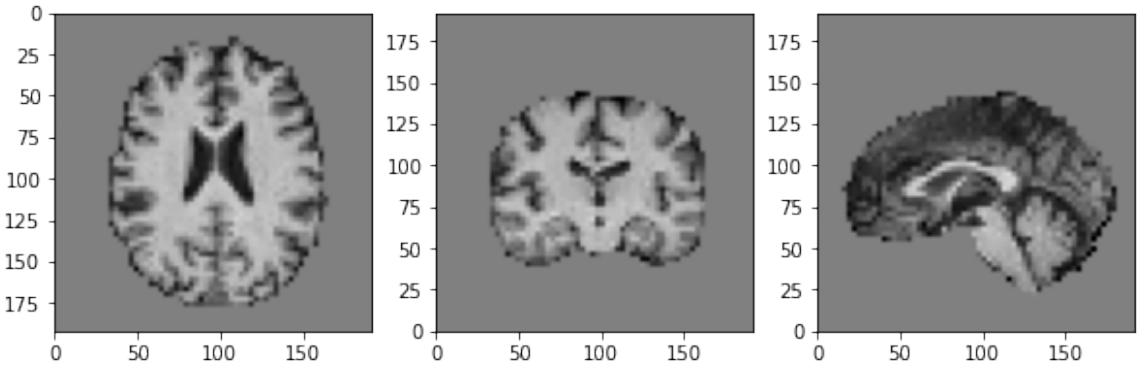


Figure 12: Pre-processed training image

Figure 6 shows a training image of our regression dataset after pre-processing. For uniformity, we have applied the same set of pre-processing operations that we have used during Task A on raw training images to obtain the above. This involves intensity normalisation and re-sampling of the input data.

2.2 Training and Validation

For training the regression CNN, we have decided to use a smaller batch size of 10 (early stoping) and a learning rate that is $1e^{-4}$. Since we are no longer separating the regression task in two sequential steps (with a regression task on straightforward 2D data), our network can be overly complex and encounter problems like diminishing return (more epochs/more complexity for small loss improvements) and overfitting.

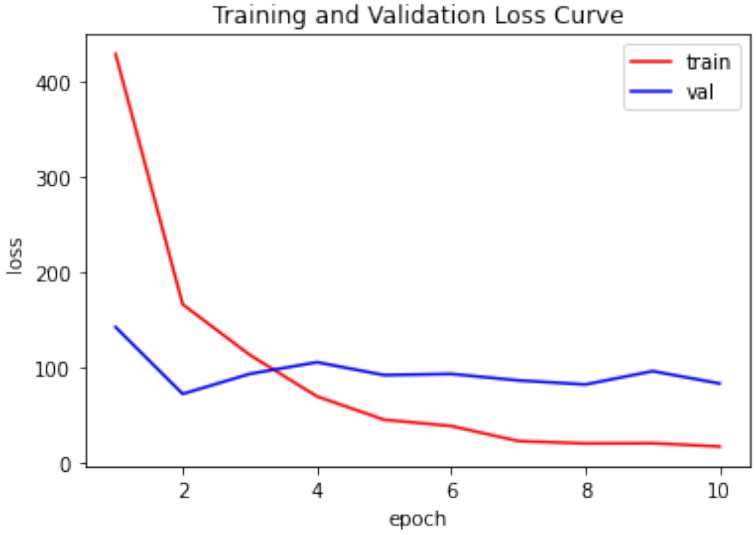


Figure 13: Training and validation loss curves of the regression CNN

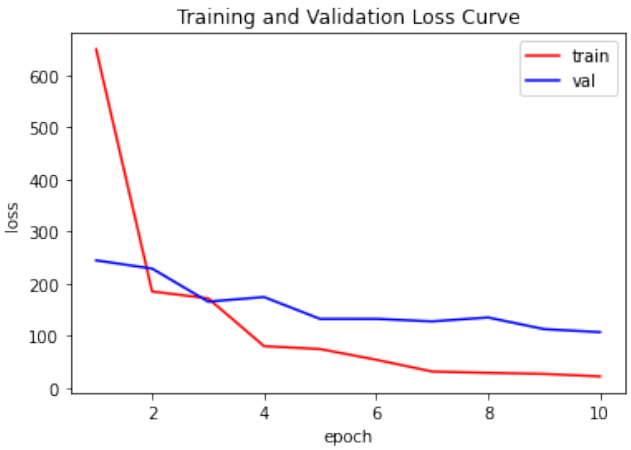
The above curves describe the loss behaviour of the model during training and validation. This figure illustrates and supports the decision we have made: our regression CNN model converges faster in training, but the validation loss seems to decrease slowly and fluctuate. This can be a sign of overfitting. However, we see in the following sections that we achieve good generalisability provided that we have enough training data.

We obtain the following error metrics on the validation dataset (47 subjects used for training the tissue segmentation model):

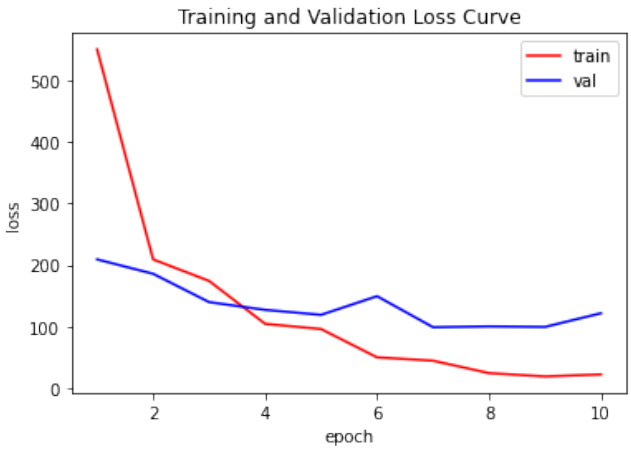
- **MAE**: 7.042
- **R2 score**: 0.771

2.3 2-fold Cross-Validation

We use 2-fold cross-validation to verify the generalisability of our model in a quick and simple manner. With this configuration, for each fold we have 2×250 subjects for training and for validation respectively.



(a) Loss curve of fold n°1



(b) Loss curve of fold n°2

According to the plots, we obtain the same loss behaviour with the 2-fold training dataset and the complete training dataset, though we also obtain error metrics that are slightly above the required bar.

Figure 14: 2-Fold Cross-Validation

- **Fold 1**: MAE=8.494, R2=0.673
- **Fold 2**: MAE=8.836, R2=0.647
- **Average**: MAE: 8.665, R2: 0.660

2.4 Final Test

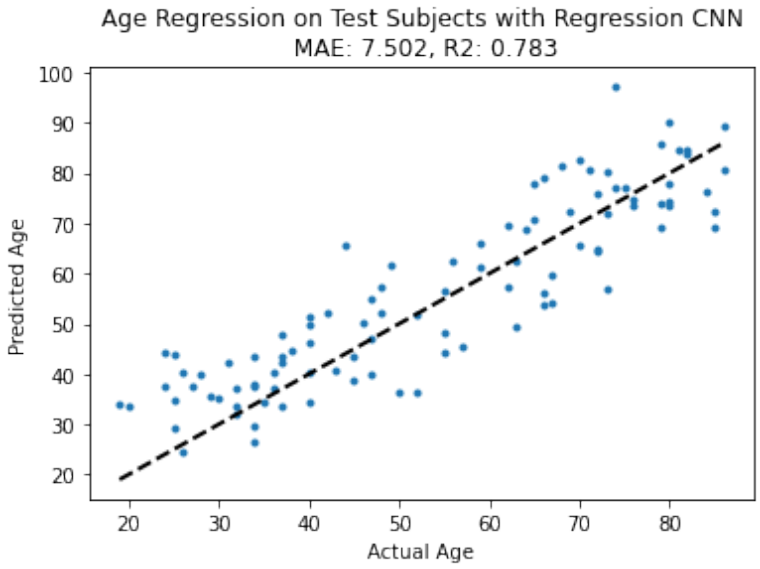


Figure 15: Regression results on Test dataset of 100 subjects

We collect regression results on the provided test dataset of 100 subjects in the figure above. Using the full training dataset of 500 subjects, the engineering decisions we have made (simplifying the model, early stopping, and smaller learning rate) seems to have avoided further overfitting and have helped us obtain satisfactory MAE value and shorter training time.

3 Age Regression Results

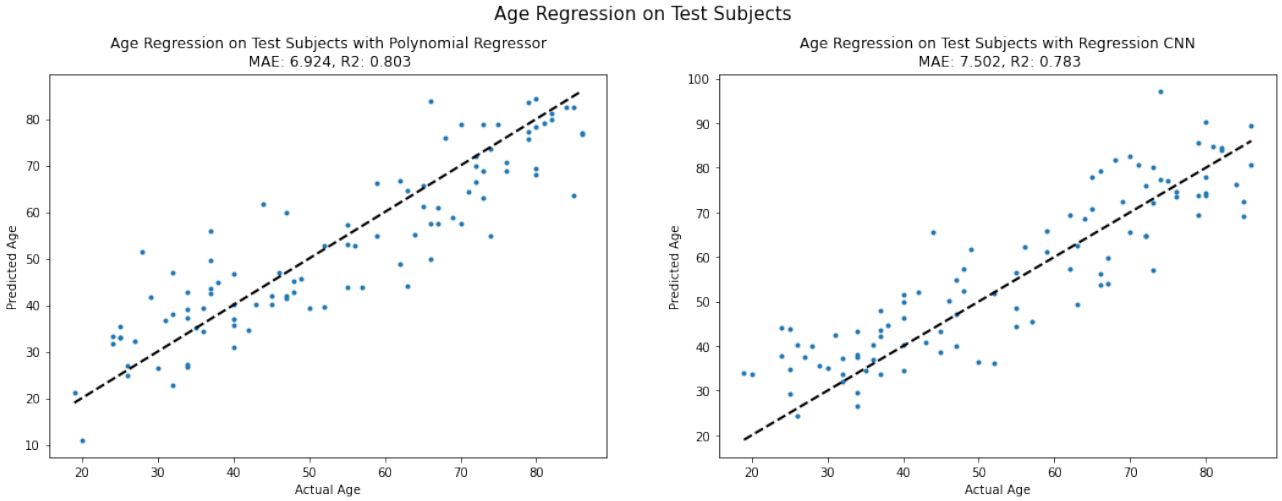


Figure 16: A side-by-side comparison of age regression results

In our case, the results obtained using Regression CNN is slightly worse than the two-step regression pipeline (Segmentation+Polynomial Regressor). There are several reasons behind this difference in performance.

- **Network complexity**: We have purposefully decided to use a simpler network in Part B. We believe that, because the Part B pipeline boils down to a single-pass regression, our model may be more prone to overfitting, which we have indeed observed signs of during model training. This

also means that with limited hyperparameter tuning our Regression CNN will achieve worse results than our attempt in Part A.

- **Choice of regression methods**: In Part A, we were able to compare multiple regression methods and choose the best-performing regressor to display the results, thanks to having brain image segmentation as a separate task. With our Regression CNN, we were unable to do the same. According to Part A, the best regressor is a Linear Regression with Polynomial Features, and not plain Linear Regression.