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

Group Leader: **Ye**

Joanna Ye, Chia-Hsin Lin
jdy15, cl222@imperial.ac.uk

1 Part A

The aim of part A is to perform age prediction from brain MRI scans by first extracting brain features from the scans using a CNN for segmentation, and then estimating ages based on the extracted features with a regression technique.

To do this, the images are first pre-processed by normalising them to zero mean and unit variance, and resampling them to an image spacing of 1x1x1 and size of 128x128x128. Figure 1 shows an example image after this pre-processing. It was chosen to reduce the spacing from 2x2x2 in the original scans to 1x1x1 to increase the resolution of the images and therefore improve the performance of the segmentation and regression. The output size of 128x128x128 was identified from hyperparameter tuning to give good results for both parts A and B, while also keeping the image size reasonable so that pre-processing, model training and inference are not too computationally expensive. The visualisation shows that although this image size does cut off a small part of the brain structure, it shows the ventricles well, which are one of the most important features for age prediction.

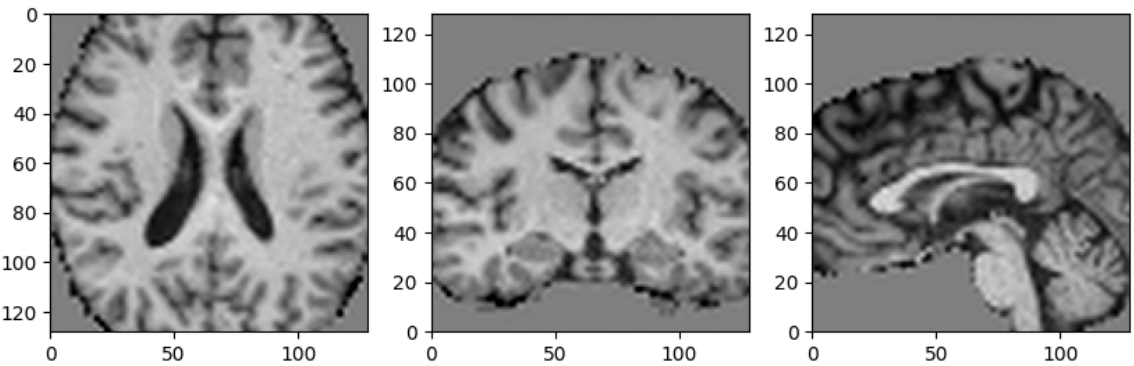


Figure 1: Example image after pre-processing.

The pre-processed scans are then passed through a CNN to segment out the volume of white matter, grey matter, and cerebrospinal fluid in the brain. The architecture of the CNN is shown in table 1; this is designed as follows: every convolution layer doubles the number of feature maps received from the previous layer, while keeping the size of the feature maps fixed. BatchNorm is inserted after all convolution layers except the last to regularise the training process. ReLU activation is applied after every Batchnorm layer and the final output layer. The output of the network is four feature maps corresponding to the logits for the three brain tissue types and image background to be classified at each input image pixel.

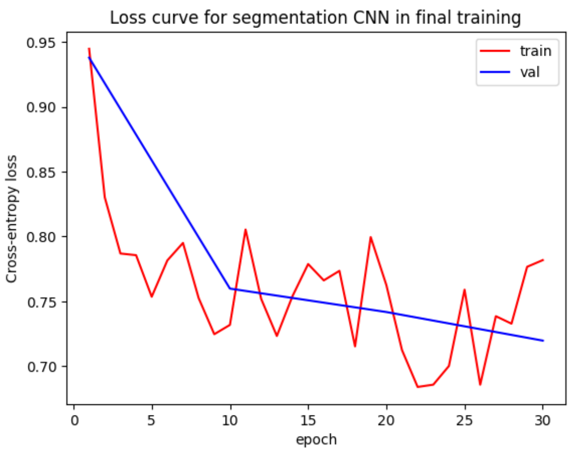
Layer	Layer type	Feature maps	Size	Kernel size	Padding	Activation
Input	Image	1	128x128x128	-	-	-
1	Convolution	8	128x128x128	3x3x3	1	-
2	BatchNorm	8	128x128x128	-	-	ReLU
3	Convolution	16	128x128x128	3x3x3	1	-
4	BatchNorm	16	128x128x128	-	-	ReLU
5	Convolution	32	128x128x128	3x3x3	1	-
6	BatchNorm	32	128x128x128	-	-	ReLU
7	Convolution	64	128x128x128	3x3x3	1	
8	BatchNorm	64	128x128x128	-	-	ReLU
9	Convolution	4	128x128x128	3x3x3	1	
Output	-	4	128x128x128	-	-	ReLU

Table 1: Segmentation CNN architecture.

Hyperparameter tuning was performed based on the loss curves and the resultant Dice score for the four classes for the 47 training and 5 validation images for segmentation. The hyperparameters yielding good results are:

- Optimizer: Adam
- Learning rate: 0.0002
- Batch size: 3
- Number of epochs: 30

In particular, the learning rate was lowered and the batch size was increased from the original script to allow the CNN to update more stably. To account for the slower learning schedule, the number of epochs was increased. The loss curve during training and the Dice score on the 500 regression images (used for testing of the segmentation) are shown in figure 2.



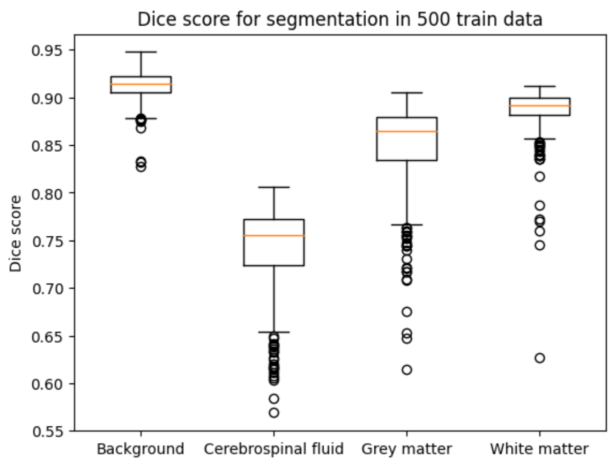


Figure 2: Loss curves for segmentation CNN training on 47 images (validation on 5 images) and the resultant Dice score on the 500 regression training images.

To estimate a patient's age from the segmented images, Ridge regression (linear regression with L2 regularisation) was used to fit the relative volumes of the three types of brain tissue. This method was identified via two-fold cross-validation on the 500 regression training images to give the best results;

table 2 shows the performance of the different techniques tried. Finally, the Ridge regression model was re-trained on the 500 regression training images and evaluated on the final 100 test images.

Regression technique	**MAE in years (cross-validation on 500 training subjects)**	**R^2 score (cross-validation on 500 training subjects)**
Ridge regression	7.55	0.73
Ridge regression with squared features	7.61	0.72
Lasso regression	7.59	0.73
Lasso regression with squared features	7.59	0.73
Support vector regression	8.18	0.69
Decision tree regression	9.94	0.49

Table 2: Performance of different regression techniques.

2 Part B

The aim of part B is to perform age prediction directly from the brain MRI scans using a CNN for regression. To do this, the images are first pre-processed similarly to part A by normalising them to zero mean and unit variance, and resampling them to an image spacing of 1x1x1 and size of 128x128x128 (see section 1 for why these values were chosen and an example image after pre-processing). The images are then passed through a regression CNN with the architecture summarised in table 3. This architecture is based on the architecture of LeNet with two main differences:

- There are two additional layers - one convolution and one max pooling - as it was found during hyperparameter tuning that this additional complexity enabled the model to achieve higher accuracy.
- The output layer is adapted for the regression task by having only one neuron with ReLU activation (since age cannot be negative).

Layer	**Layer type**	**Feature maps**	**Size**	**Kernel size**	**Padding**	**Activation**
Input	Image	1	128x128x128	-	-	-
1	Convolution	6	128x128x128	3x3x3	1	-
2	Max pooling	6	64x64x64	2x2x2	0	ReLU
3	Convolution	10	64x64x64	3x3x3	1	-
4	Max pooling	10	32x32x32	2x2x2	0	ReLU
5	Convolution	16	32x32x32	3x3x3	1	-
6	Max pooling	16	16x16x16	2x2x2	0	ReLU
7	Fully connected	-	120	-	-	ReLU
8	Fully connected	-	64	-	-	ReLU
Output	Fully connected	-	1	-	-	ReLU

Table 3: Regression CNN architecture.

Hyperparameter tuning was performed using two-fold cross-validation on the 500 training images. As well as the above image spacing, size and network architecture, this hyperparameter tuning identified that the following parameters produced good results:

- Optimizer: Adam
- Learning rate: 0.0005
- Batch size: 16
- Loss function: MSE

In particular, it was found that using MSE as the loss function during training gave better results than MAE; the average MAE on the validation set across the two folds was 6.95 years with MSE loss, and 7.07 years with MAE loss.

Finally, the model was re-trained on the full training set using the tuned hyperparameters, and evaluated on the test set of 100 images. The 47 segmentation training images from part A were used as a validation set during this final training to identify a suitable number of epochs; training was stopped after 40 epochs as the loss curves in figure 3 show that the performance on the validation set has stopped improving after this amount of training.

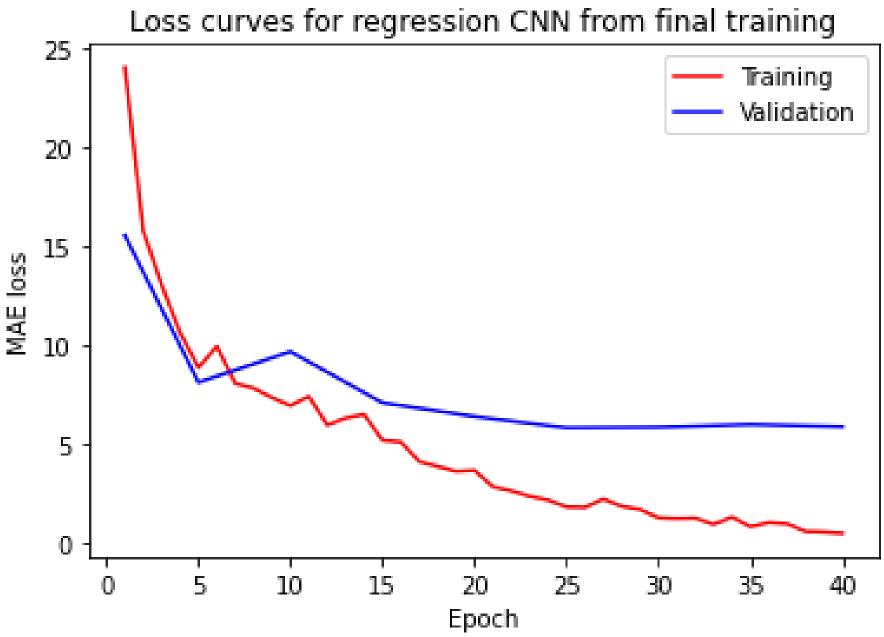


Figure 3: Loss curves for regression CNN from final training on 500 images (validation on 47 segmentation images from part A).

3 Age Regression Results

Table 4 summarises the performance of the age prediction for the models selected in parts A and B. Two evaluation metrics are considered: mean average error (MAE) for accuracy and R^2 score for the amount of variability in that data explained by the model.

In part A, the model obtains a better accuracy in cross-validation than in the final testing. This result is unexpected, as the model is trained with more data (500) in the test case than in cross-validation (250). However, the difference in MAE is not large, so we suspect that the lower accuracy may be attributed to noise in the data. On the other hand, the model achieves a better R^2 score in the test set as expected, as more training data allows the model to explain more variability in the unseen data.

In part B, the model shows a better accuracy and explains more variability in the test data than in cross-validation. This result aligns with our expectation for the same reasons as mentioned above.

Table 4 suggests that the part B model outperforms the part A model in both accuracy and the amount of variability captured in the data. This may be for two reasons:

- Single stage models (part B) accumulate less error than two-stage models (part A), as two-stage models have an additional error term from the extra stage.

- Single stage models learn the best features for regression directly, while two-stage models are forced to use features learnt from the previous stage.

In other words, the part A model has both the segmentation error and the regression error, while the part B model has regression error only. Therefore the error accumulation is less severe in part B. Moreover, the part B model can learn the best features for regression directly from the input images while the part A model is forced to used the segmented brain volumes to predict the age, resulting in less accurate results.

	Part A	**Part B**
MAE in years (final results on 100 test subjects)	7.73	6.23
MAE in years (cross-validation on 500 training subjects)	7.55	6.95
R^2 score (final results on 100 test subjects)	0.77	0.85
R^2 score (cross-validation on 500 training subjects)	0.73	0.77

Table 4: Results from parts A and B.

The scatter plots in figures 4 and 5 agree with table 4 and suggest that the model from part B outperforms the one in part A. The predicted ages from the regression CNN lie more closely to the true age (dashed line) and show lower variance than the model composed of the segmentation CNN and linear regression.

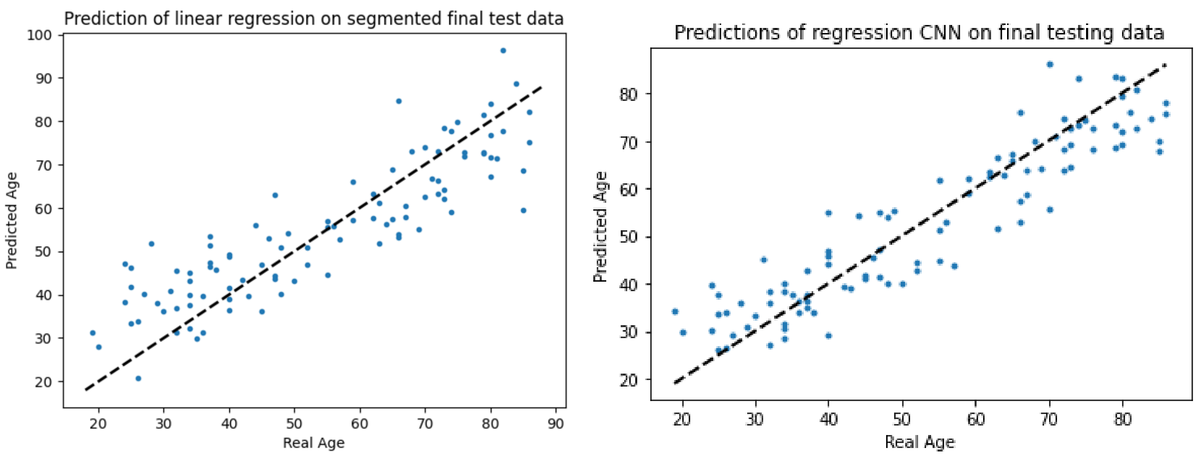


Figure 4: Final predictions on 100 test images (left - part A; right - part B).

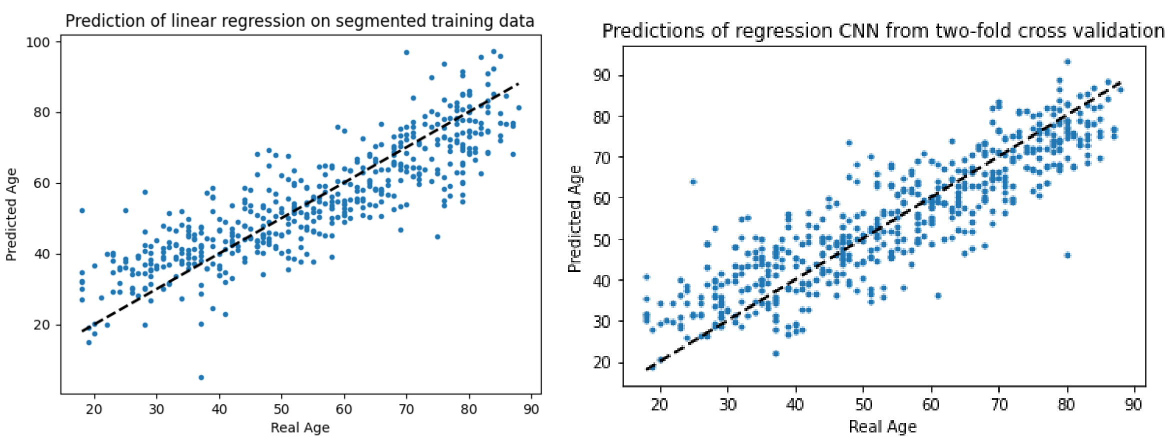


Figure 5: Predictions from cross-validation on 500 training images (left - part A; right - part B).