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

Group: **29**

Joël Tang, Eric Chen, Victor Tempé
{jt620, ecc20, vt520}@imperial.ac.uk

1 Part A

Describe your approach and the individual steps of part A (up to 300 words). Add figures or plots to illustrate your approach and show any results from pre-processing steps.

For the brain segmentation task, we decide to implement a modified version of 3D U-Net used in the BRATS challenge for tumor segmentation. We keep the default parameters with an input of size 64x64x64 and spacing 3x3x3. We train the network on 20 epochs with a learning rate of 0.001.

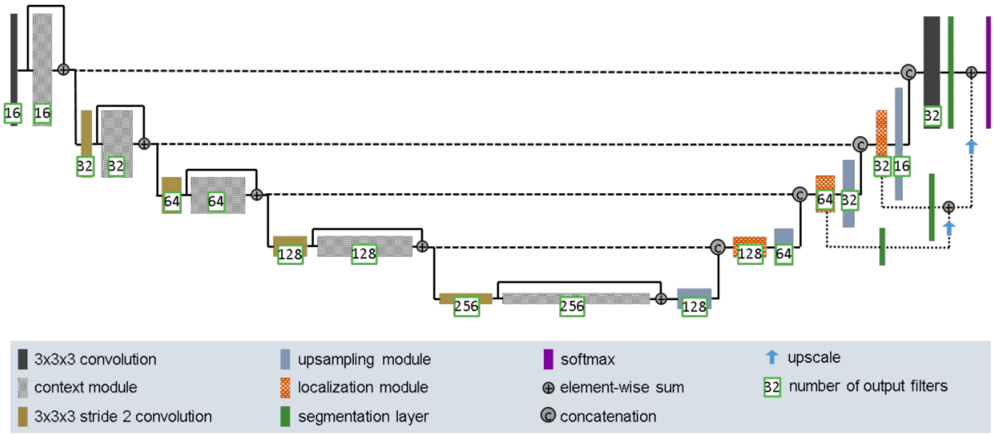
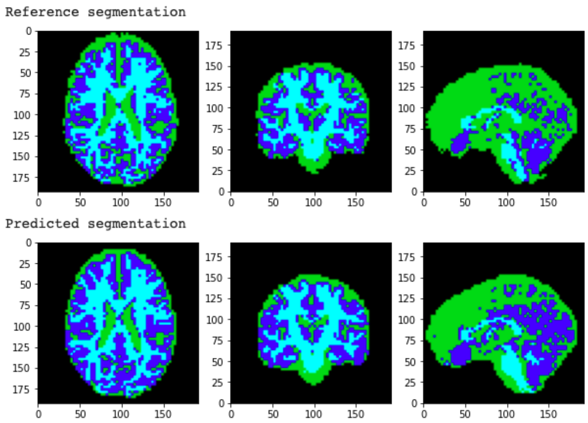


Figure 1: Modified 3D U-Net (context pathway : left ; localization pathway: right)

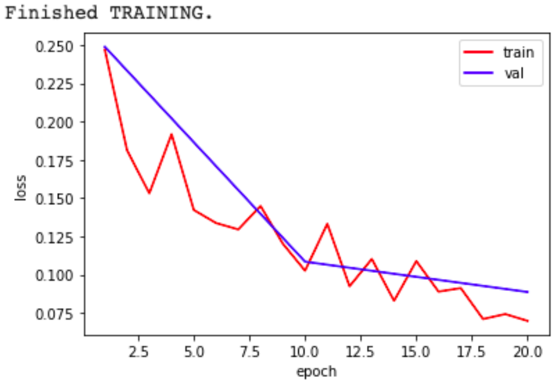
The network consists of a context pathway which aggregates high level information that is localized in the localization pathway. This is done by using a residual block of 3x3x3 convolutional layers and dropout layer. Each input before the block is added to the results and this element-wise sum will also be used in the localization pathway. Then, for the localization pathway, we localize the structures of interest by taking features from the context pathway that encode contextual information at low spatial resolution and transfer it to a higher spatial resolution. This is done by upscaling followed by a 3x3x3 convolution. After that, we concatenate the upsampled features with the ones from the corresponding level of the context pathway and use convolutions to recombine the features together. For the network, we use LeakyReLU along with instance normalization.

This model architecture gives us some satisfying results, with a segmentation close to the reference despite some differences. Regarding the Dice score, we obtain a mean value of around 0.85. When we look at the Dice score for each brain tissue, we can see that the segmentation is better for WM than CSF:

	Mean	CSF	GM	WM
Dice scores	0.85	0.75	0.87	0.92



(a) Reference and Predicted segmentation on testing sample



(b) Training and Validation curves for segmentation task

After having extracted the segmentation from the modified 3D U-Net architecture above, we can compute relevant numerical features. We computed the proportion of the segmented volumes of CSF, GM, WM in the total brain volume. We then compute second-order polynomials: $CSF * GM$, $CSF * WM$, $GM * WM$, CSF^2, GM^2, WM^2.

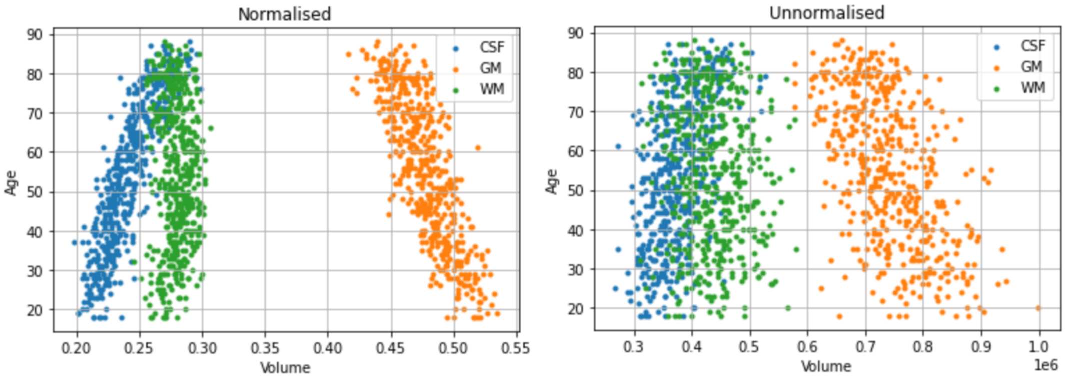


Figure 3: Normalised and unnormalised segmentation features.

The plot shows noticeable linear behaviour of the features with respect to the age.

Then, we use theses features to fit diverse regressors. Three models were tested here: linear regression, gradient boosting, and SVM for regression. Their performances were quite alike after some fine-tuning.

For linear regression, no particular hyperparameters were modified (closed-form solution, without regularization).

For the gradient boosting, we chose a low maximum of depth (2), given that the number of features is 9. 150 estimators were chosen. These hyperparameters are consistent with the idea that we want weak learners, and to prevent overfitting.

The SVR model was fine-tuned with a higher C (=5-6) to prevent under-fitting. The kernel is polynomial, even if we already produce lower-order features.

All the results are obtained from cross-validation scores, on two folds. These two folds are stratified, with respect to both age and gender variables. For this purpose, ages were considered into bins of 10 years (10-20 years old, 20-30 years old, etc.).

The MAE on the testing set were obtained by training the regression models on the whole dataset, which explains the slightly better results.

	Linear Regression	Gradient Boosting	SVR
MAE (2-folds)	7.452	7.397	7.363
MAE (test)	6.885	6.571	6.875

2 Part B

Describe your approach and the individual steps of part B (up to 300 words). Add figures or plots to illustrate your approach and show any results from pre-processing steps.

In this part the extraction of features as well as the regression are performed in a single CNN.

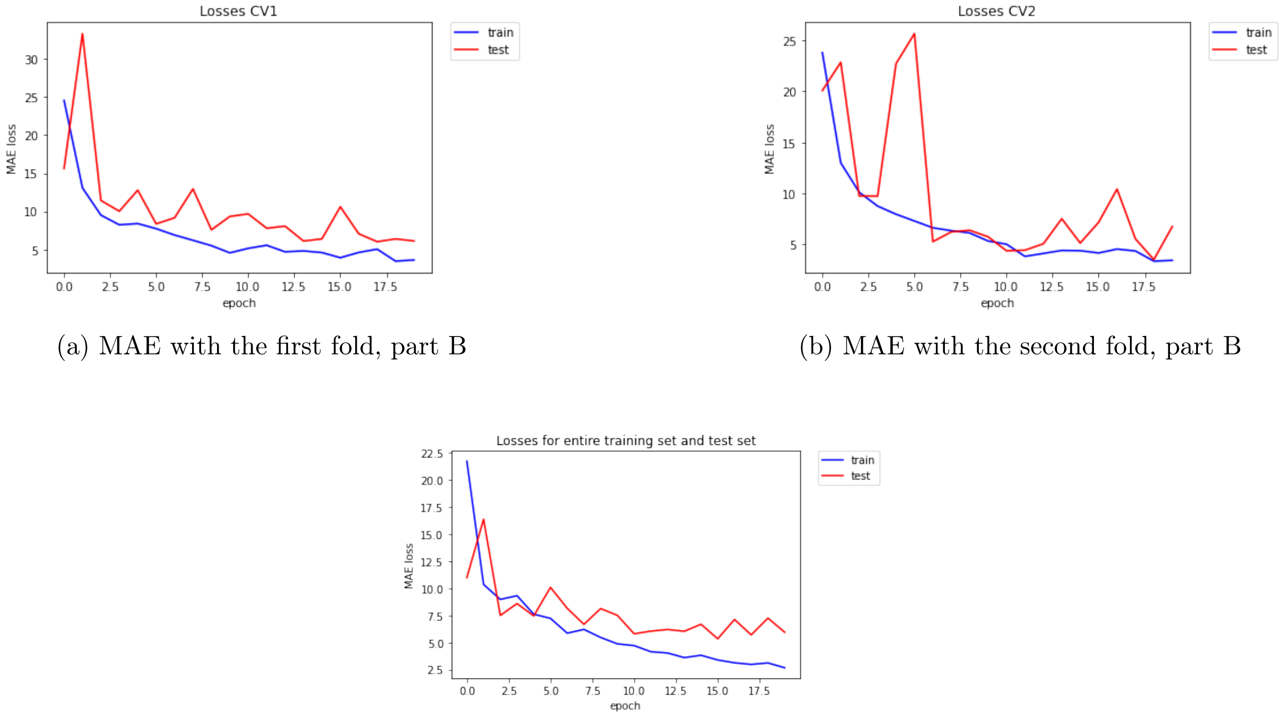
The CNN is composed of several blocks inspired by the ResNet blocks (two convolutions with kernel size 3 and two batchnorms) but without any skip connections in order to reduce the number of parameters. Between those blocks, which do not change the shape of the inputs except for the last one, down sampling is done using maxpooling. Convolutions do not include bias terms as they are followed by batchnorms. Finally, regression occurs in the last layer which is a 1x1x1 convolution with a single output channel.

All activations functions are ReLUs and the L1 loss or MAE loss was directly used during training to minimize this same metric in 20 epochs. Hyper parameter tuning was assessed with cross validation (the stratified 2-folds are obtained like in part A), we kept a batch size of 4 and chose quite a small learning rate for the Adam optimizer (0.0002 instead of the original 0.001) which is decayed by a factor of 0.92 from epoch 5 onwards with a scheduler. 0.92 because we wanted to divide the rate by 2 in 10 epochs and that $0.92^{10} \approx 0.43$. This configuration helped to reduce our initial overfitting issues. This issue was not a surprise as there are more than 3.5M parameters in the model.

About the inputs, the images are resized to (1, 64, 64, 64) shape with an equal spacing of 3x3x3. The model provides good results :

	2-folds	test
MAE	6.43	5.94

Although colab could not handle higher resolution images, it may have led to better results. Overfitting could also be addressed more with some other preprocessing steps, such as image rotations or scaling, that stabilize the training. Indeed, in the plot bellow, the validation curve oscillates. Yet it globally decreases over time.



(a) MAE with the first fold, part B

(b) MAE with the second fold, part B

Figure 5: MAE over time for Part B. Trained on 500 images, tested on 100

3 Age Regression Results

Add scatter plots in Figure 7 for the two approaches and discuss your results (about 300 words). Compare, for example, your results obtained via cross-validation on 500 subjects and final results on 100 test subjects. Feel free to add more figures.

Overall, the results were significantly better in the part B than in the part A. Three different regression methods were used in the part A: linear regression, gradient boosting, and support vector regression. With reasonable fine-tuning, all of the three approaches gave very close results in terms of validation MAE. Also, the data was augmented to include polynomials of order 2. The very close results suggest that we are limited by our input features.

On the other hand, the end-to-end approach of the part B is trainable from the features extraction to the regressor. We may assume that its better results are the result of the learning process of better representations useful for the task.

MAE are lower for the test set because the models were trained on 500 samples, instead of 250 for the 2-folds cross validation.

	Part A	Part B
MAE (2-folds)	7.363	6.432
MAE (test)	6.875	5.941

Also, it is possible to see that the points in the scatter plots of part B are more centered around a straight line than in part A, especially for the training set. Finally, in botch parts, we notice that the models generalize quite well to unseen data and achieve good age predictions.

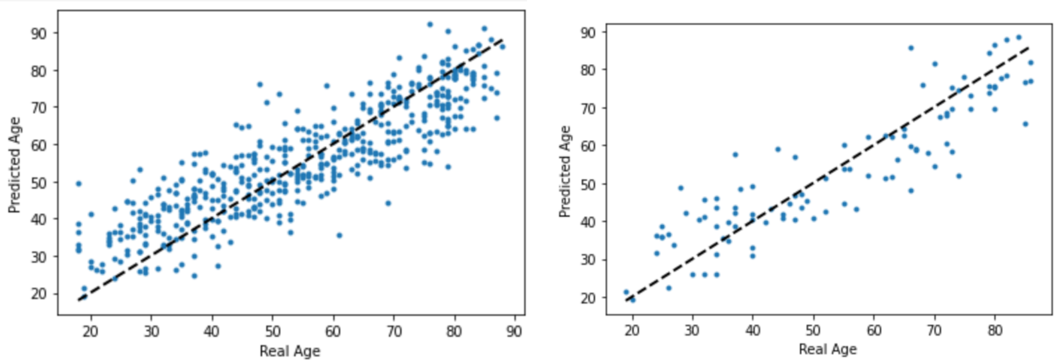


Figure 6: Part A: Left figure: Modified 3D U-Net + SVR regression on train (CV); right figure: On the test set (trained on the whole trainset)

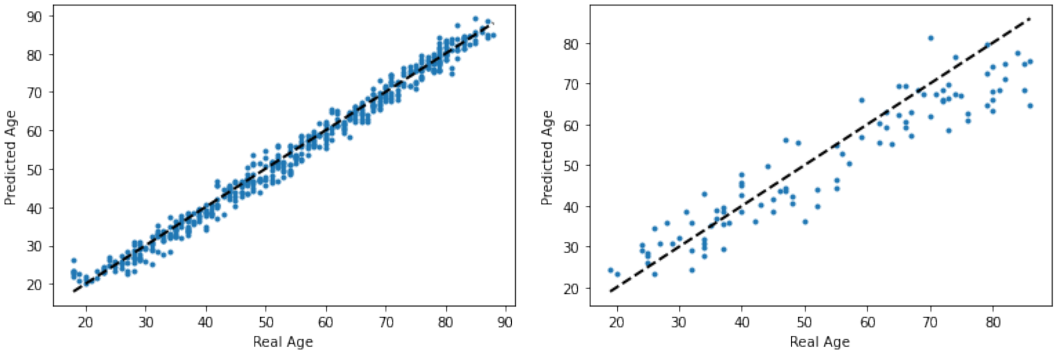


Figure 7: Part B: Left figure: Regression on the 500 train samples (CV); right figure: On the test set (trained on the whole trainset)