Brain Age Prediction Using Deep Convolutional Neural Networks

Zixian Ma
Stanford CS Undergraduate
zixianma@stanford.edu

Prabhjot Singh Rai
Stanford SCPD Student
prabhjot@stanford.edu

Harry (Xinxuan) Jiang
Stanford CS Undergraduate
xinxuan@stanford.edu

Abstract

In this project, we aimed at predicting the physiological age of a given subject's brain by training the model on Magnetic Resonance Imaging (MRI) data. By pre-processing the MRI datasets through FMRIB Software Library (FSL) prior to feeding them to models, we got the skull-stripped and normalized MRI data. We then fed the histograms of the data into a linear regression model as a baseline evaluation and the skull-stripped images into convolutional neural networks (CNN). Using transfer learning, we have implemented pre-trained models such as ResNet18 and ResNet50, fine-tuned the last fully connected layers and also created our own custom model architectures to train from scratch. Our approaches allowed us to achieve a validation error as low as 1 - 4 years and test error around 5 - 7 years. Also, we have tried identifying the most relevant features in brain age estimation by generating saliency maps of the MRI images.

1. Introduction

We investigated the application of computer vision in predicting brain age from MRI scans. Many neurodegenerative disorders are becoming very common in modern days with an increasing number of people suffering from them. Alzheimers disease (AD) is the most prevailing type, with a rate estimated to be around 5% for people older than 65 and a staggering 30% for those older than 85 in developed countries [7]. Using CNN, we have tried to predict the physiological age of brain through the MRI data so that prediction of such neurodegenerative disorders becomes possible.

2. Related Work

There have been initiatives before in addressing similar problem. We found a medium[6] post which precisely works on the same problem as this project, but uses different model configuration and doesn't use transfer learning. This resource was really helpful as it introduced many pre processing techniques that we weren't aware of (will discuss them explicitly below). There were some other papers as well, which we went through in order to have a better understanding of the problem statement. The Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker[2] paper talks about how to use the MRI scans to predict the chronological age and deviations from healthy brain aging. This approach additionally uses Gaussian processes regression (implemented using the Pattern Recognition for Neuroimaging Toolbox). As the multivariate Gaussians can reflect local patterns of covariance between individual points, the combination of multiple Gaussians in a GP can readily model non-linear relationships and is more flexible that conventional parametric models, which rely on fitting global models.

3. Dataset and Features

3.1. Data Collection

We collected the brain MRI data of both healthy subject and patients with Alzheimer Diseases from three sources. Specifically we obtained two publicly available and anonymized datasets, one of 619 healthy subjects from 3 London hospitals (brain-development.org, ixi-dataset) and the other of 1,206 subjects from 32 hospitals across the world (International Neuroimaging Data-Sharing Initiative, fcp-dataset). Additionally, on submitting a request to the Alzheimers Disease Neuroimaging Initiative (ADNI), we got access to 1724 subjects, including normal subjects and subjects suffering from 2 different types of diseases (Alzheimer's Disease and Mild cognitive impairment) (http://adni.loni.usc.edu, adni-dataset).

3.2. Data cleaning

In IXI-Dataset, 29 subjects with missing age, 25 subjects which were repeated and 2 subjects with missing data. For FCP-Dataset, 4 data sources didn't have demographic information, 5 sources and 1 file had MRI missing and data was curropted for Orangeburg(skull-stripped images under actual MRI images name). We discarded the cases where ages were ¿ 100 and less than 18.

After cleaning the datasets, we were left with 563 subjects (90.95% reduction) for IXI-Datasets and 1034 subjects

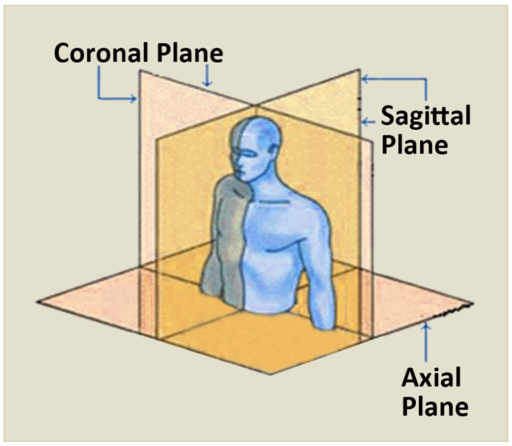


Figure 1. Different planes for viewing MRI images

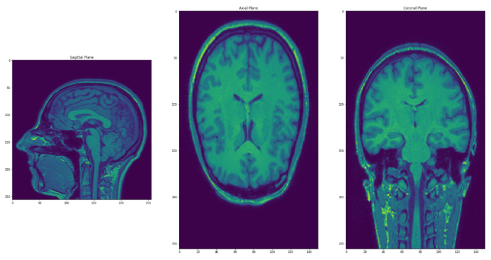


Figure 2. From Left to Right: MRI Scan in Sagittal Plane, Axial Plane and Coronal Plane for 35 years old female from Guys Hospital (IXI-Dataset subject IXI002)

(85.73% reduction) for FCP-Datasets and 1724 subjects for ADNI-Dataset (no reduction).

3.3. Data Exploration and Visualisation

The MRI images are 3D images. There are 3 different axis along which the MRI scans can be viewed. These planes are depicted in Figure 1. An example MRI scan visualisation is provided in Figure 2.

We created some basic stats around the data. Combining the two datasets, we have 26 different hospitals (22 from FCP-Dataset and 3 from IXI-Dataset and NA for ADNI dataset) out of which 47.18% are women. Maximum age is 92 and minimum is 18. Median age is 67.24.

We have also visualised the datasets together, and these are illustrated in Figure 3. We have plotted the patient distribution per hospital (where we can see that most of the data is from Guys hospital) on the left most side. In the middle, we have plotted the age distribution per hospital and it shows the interquartile range and the outliers in our dataset. We can see that ICBM, HH, Guys and IOP have data from almost all ages hence very less outliers. Lastly, we plotted the overall age distribution on the right most figure.

3.4. Data pre-processing

We used FSL [5] [10], which is a comprehensive library of analysis tools for FMRI, MRI and DTI brain imaging data created by the Analysis Group, FMRIB, Oxford, UK. There are a variety of techniques, such as brain-extraction, linear and non-linear registration, and tissue-type segmentation, we can use to process and normalize structural MRIs. After a careful and thorough investigation, we eventually chose to run the Anatomical Processing Script (fsl_anat), which is essentially a pipeline consisting of a range of tools, on our images with a few twists. Below are the **five main steps** in our data processing pipeline.

Step1: Automatic cropping This is achieved by the robustfov script, which is an automatic cropping step to remove the neck and lower head portions in the images.

Step2: Bias-field correction (RF/B1-inhomogeneity-correction) This process is done with FMRIB's Automated Segmentation Tool (FAST)[9], which besides its segmentation function, also corrects for spatial intensity variations (also known as bias field or radiofrequency field (RF) inhomogeneities).

Step3: Linear registration to standard space We applied FIRST, the linear registration tool, because the non-linear one is too time-consuming and doesnt have a significant effect. FIRST is a model-based segmentation/registration tool, which use model that are constructed from manually segmented images provided by the Center for Morphometric Analysis (CMA), MGH, Boston.

Step4: Brain-extraction This step is achieved by running the Brain Extraction Tool (BET) which simply deletes non-brain tissue from an image of the whole head.

Step5: Tissue-type segmentation This is again done by methods in FAST, which segment a 3D image of the brain into different tissue types (Grey Matter, White Matter, CSF, etc). It produces a probabilistic and/or partial volume tissue segmentation.[11]

3.5. Features

As discussed earlier, we are using MRI dataset 3d images as input to our models and leave the feature detection to the neural networks. We have also explored saliency maps, trying to detect which part of the images contribute significantly to the final scores (in the experiment section below). Since our inputs are just pixels or voxels of MRIs, we expect our features to be the grey and white matters of the brain.

3.6. Normalization and Data Augmentation

Before fitting the inputs into pre-trained models, we reshaped the MRI images, because they are black and white 3-dimensional images while the inputs of ResNets are 2-dimensional RGB images. Specifically, we sliced across the axial dimension to get a 2-dimensional slice. However,

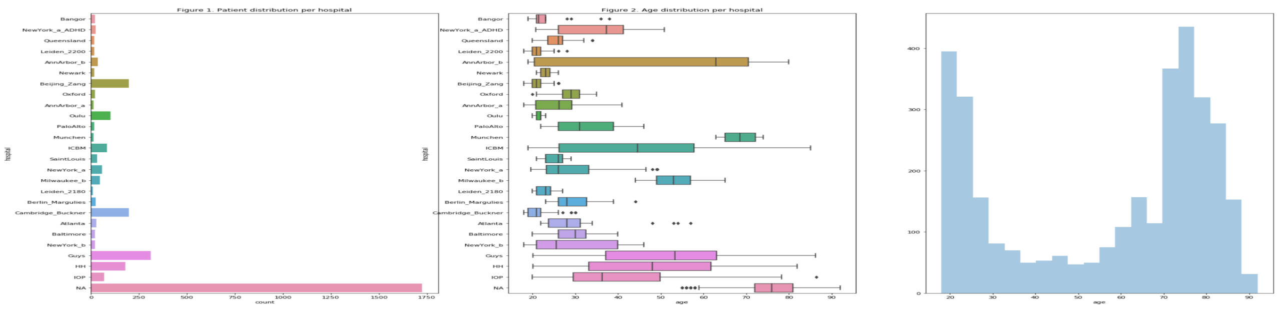


Figure 3. Left to Right: Patient distribution per hospital, Age distribution per hospital and Overall Age Distribution for combined datasets

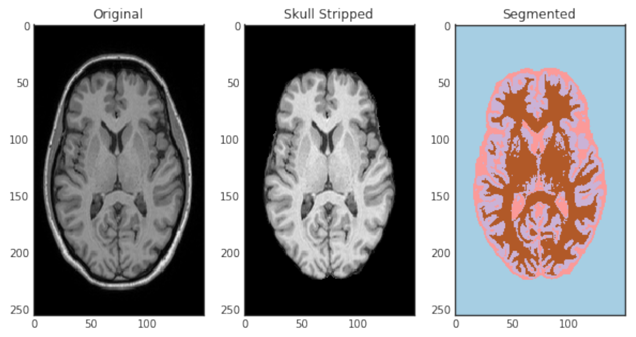


Figure 4. Image pre-processing - the left is the original MRI scan image, and center is the skull-stripped image used as input data, and on the right is the segmented image.

since not all the slices have the same sagittal and coronal dimensions, we still needed to reshape the slices into 224*224 images. If an existing dimension is greater than 224, we sliced the most central 224 pixels; if its smaller than 224, however, we evenly zero-padded that dimension to 224. Whats more, in order to generate the three channels, we followed two approaches. One was that we copied the slice three times and stacked them together, and in another approach, we used different dimensions of the 3d image to be fed as three different channels.

However, we are deeply aware of the limitations of this approach, since our brain MRIs look very different from the images in the ImageNet dataset, and brute forcing the black and white image into one that has three channels is also not the best idea to implement. Therefore, we also trained our own models on our limited data from scratch. We will elaborate on this approach in our experiment section.

3.7. Training, Test and Validation sets

For IXI dataset, we have a total of 536 examples; for FCP dataset, we have a total of 1034 examples. We first trained our models on IXI dataset and then used the FCP dataset for data augmentation. Specifically, in each training process, we divided the input data into 5 folds, 3 folds of which are training data, 1 fold validation and 1 fold test data. That means 322 (60%) training and 107 (20%) validation/test data points for IXI , 620 (60%) and 207 (20%) for FCP, and 341 (60%) and 114 (20%) for ADNI normal subjects. In the final data augmentation attempt, we had a total of 1283 examples for training and 428 examples for validating and testing.

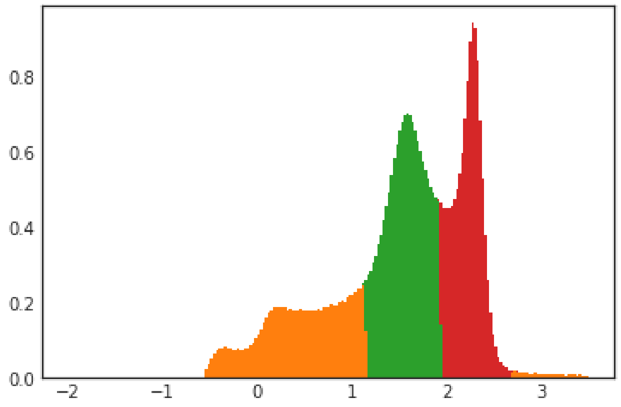


Figure 5. Histogram example - the x-axis represents the normalized voxel intensity of the skull-stripped image, y-axis the percentage of voxels. The green region represents grey matter and the red region is white matter.

4. Methods

We intend to explore both the performance of a baseline model using linear regression, as well as a more expressive model using a CNN.

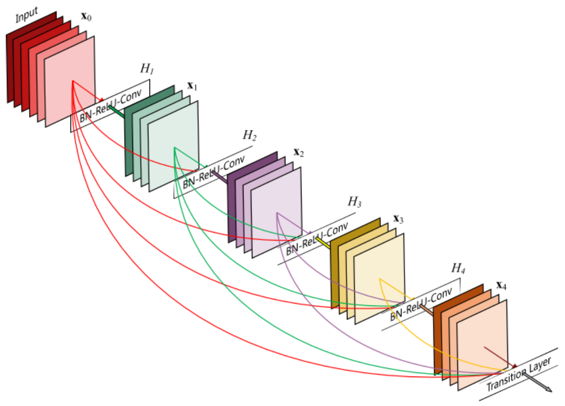
4.1. Baseline Model

The baseline model is a simple linear regression model. Since the amount of grey matter and white matter are reflective of a patient's brain age, we can plot a histogram of the patient's T1-weighted MRI images based on the relative brightness of the scanned image, as shown in Figure 5, and feed it into a linear regression model to predict the subjects' brain age.

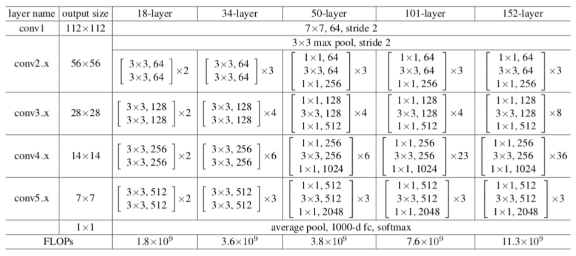
4.2. CNN Model

After we established the baseline model, we moved forward into training a much more expressive CNN model for brain age prediction. Our overall approach could be divided into two different categories: transfer learning and non-transfer learning. For the transfer learning approach, we experimented with the pre-trained models, Resnet-18, Resnet-50, as well as Resnet-101. Resnets have been shown to perform well on image classification tasks, since its residual nature allows the gradient flow to have ”shortcuts” during optimization. As a result, it has a much deeper representation of the input images, and performs much better on prediction tasks, compared to some earlier architectures such as VGG or AlexNet. [3]

Below is a conceptual sketch of ResNet’s residual convolutional layers:



The parameters for ResNet’s layers have been detailed below [3]:



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Since our pre-trained ResNet models were trained on the ImageNet datasets which consist of RGB images, the channel dimension is 3, which is inconsistent with our input data. Our input data is a three-dimensional array in which every value is the voxel intensity at that particular location. In order for our input data to work with the pre-trained models, we decided to take just one representative slice across the axial plane, and stack the 2 dimensional voxel intensities three times so that the ”color” channel is of dimension 3. In other words, the R, G, and B channel of our input to the pre-trained models are all identical. For the model output, we experimented with both a classification model and a regression model. For the classification model, we divided the entire age range into 13 bins, with 5 years per bin (age 18-22 belongs to one category, etc). We then train the model to classify which bin the input should fall under. For the regression model, we use a single-neuron fully connected layer to directly predict the age. we swapped out the last fully connected layers of those pre-trained ResNets with a) a Softmax layer for classification, and b) a one-neuron linear fully connected layer for regression. Two different models use different losses. The classification uses Softmax loss, whereas we use Mean Square Error for linear regression. The loss functions for Softmax and MSE are listed respectively below:

$$y_c = \varsigma(\mathbf{z})_c = \frac{e^{z_c}}{\sum_{d=1}^{C} e^{z_d}} \quad \text{for } c = 1 \cdots C \tag{1}$$

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}\Big(\hat{y}_i - y_i\Big)^2 \tag{2}$$

During training, we decided to perform hyper-parameter tuning and unfreeze all layers of the pre-trained models so that they could fit our data better.

Aside from transfer learning, we also constructed our own CNN models and trained them from ground up. We experimented with two models: a) a 10-layer 2D CNN, as well as b) 6-Layer 3D CNN. The model consists of some convolutional layers in the front, followed by a single fully connected layer as regression. Below are the architectures of our 2D CNN model and 3D CNN model, respectively:

Our 2D CNN Model:

```
model = nn.Sequential(
nn.Conv2d(64, 32, 5, 1)
nn.Conv2d(32, 32, 5, 1)
nn.BatchNorm2d(32),
nn.AvgPool2d(kernel_size = 2),
nn.Conv2d(32, 64, 5, 1),
nn.Conv2d(64, 64, 5, 1),
nn.BatchNorm2d(64),
nn.AvgPool2d(kernel_size = 2),
nn.Conv2d(64, 128, 3, 1),
nn.Conv2d(128, 128, 3, 1),
nn.BatchNorm2d(128),
nn.AvgPool2d,
nn.conv2d(128, 256, 3, 1),
nn.Conv2d(256, 256, 3, 1),
nn.BatchNorm2d(256),
nn.AvgPool2d(kernel_size = 2),
nn.Conv2d(256, 512, 3, 1),
nn.Conv2d(512, 512, 3, 1),
nn.BatchNorm2d(512)
```

```
    nn.AvgPool2d(kernel_size = 2),
    Flatten(),
    nn.Linear(512*1*2*2, 1),
)
```

Our 3D CNN Model:

```
    model = nn.Sequential(
    nn.Conv3d(1, 64, 5, 1),
    nn.Conv3d(64, 64, 5, 1),
    nn.BatchNorm3d(64),
    nn.ReLU(),
    nn.AvgPool3d(kernel_size = 2),

    nn.Conv3d(64, 128, 3, 1),
    nn.Conv3d(128, 128, 3, 1),
    nn.BatchNorm3d(128),
    nn.ReLU(),
    nn.AvgPool3d(kernel_size = 2),

    nn.Conv3d(128, 256, 3, 1),
    nn.Conv3d(256, 256, 3, 1),
    nn.BatchNorm3d(256)),
    nn.ReLU(),
    nn.AvgPool3d(kernel_size = 2),

    Flatten(),
    nn.Linear(256 * 8 * 8 * 8, 1),
)
```

5. Results

5.1. Linear Regression

We used the scikit-learn RidgeCV regression model to predict the input, and achieved a mean absolute error of about 7.99 years. We figure that the error is mainly due to the fact that we have not yet pre-processed a complete dataset, and that the linear model is quite limited in its capabilities to fit the data well. We also propose that perhaps increasing the number of bins when constructing the histogram would make a more powerful feature extractor, and would in turn make the linear model better.

5.2. Convolutional Neural Network

We formulated brain age estimation as both regression and classification problems and trained CNN models in both cases. For regression, the evaluation metric of our regression model is the mean absolute error between the predicted and true age of samples. For classification, the evaluation metric s accuracy, which is calculated by the number of correctly classified examples divided by the total number of examples. Since we have 14 classes, it would be a pain to plot all the possible AUC/ROC curves, so we mainly relied on accuracy to evaluate our model.

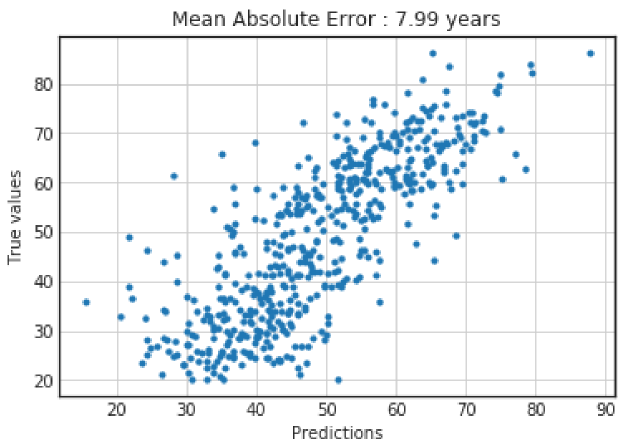


Figure 6. Linear Regression results - the X axis is the prediction, while the Y axis is the true values.

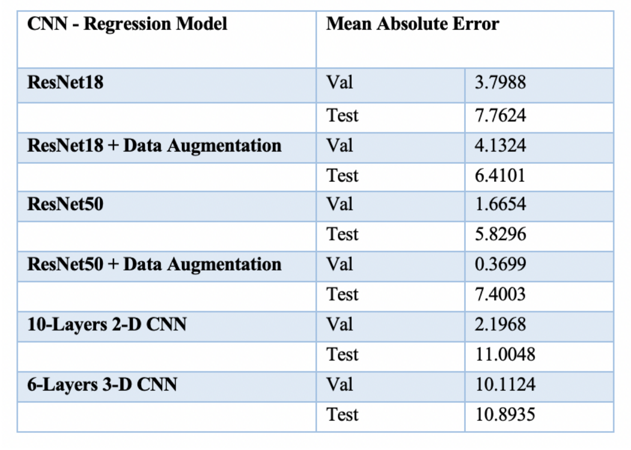
We applied transfer learning in training both of our regression and classification models. We mainly tested and compared the performance of pre-trained ResNet18 and ResNet50. We also tried training on our own dataset, but pre-trained weights produced more accurate results; so we are not going to discuss the former approach. Also, since the regression model was doing a better job, we also did a few experiments on it, which we are going to cover in the Experiment section.

As shown in the table in Figure 7., after extensive fine-tuning of the last fully connected layer in ResNet18 and ResNet50, we achieved results comparable to state-of-the-art performance (an age error between 4 and 7) [4] [6] using pre-trained ResNet18 and ResNet50 on our regression model, and ResNet50 had a slightly better performance than ResNet18. It is worth noting that, our validation errors were even a lot lower than state-of-the-art, which we think is a great success, although we still need to improve on our test errors. On our classification model, however, we failed to achieve very satisfying results (an accuracy of 60%-70% for the classification model) [6], and we are going to explain our failures in the discussion section.

6. Discussion

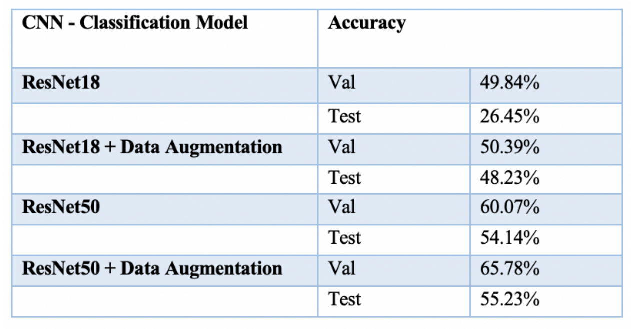
6.1. Approach Selection

Since we were not sure whether its better to treat brain age estimation as a regression or classification problem, we tried both approaches, and it turned out that the regression approach worked better. Most of the previous works were done using a regression approach, which means that the output of the model is a single age of a subject. However, we think it actually makes sense to use a classification approach, given that its super hard to estimate a single age correctly. Also, since our different datasets have different



CNN - Regression Model	Mean Absolute Error	
ResNet18	Val	3.7988
	Test	7.7624
ResNet18 + Data Augmentation	Val	4.1324
	Test	6.4101
ResNet50	Val	1.6654
	Test	5.8296
ResNet50 + Data Augmentation	Val	0.3699
	Test	7.4003
10-Layers 2-D CNN	Val	2.1968
	Test	11.0048
6-Layers 3-D CNN	Val	10.1124
	Test	10.8935

Figure 7. CNN results - Regression Model.



CNN - Classification Model	Accuracy	
ResNet18	Val	49.84%
	Test	26.45%
ResNet18 + Data Augmentation	Val	50.39%
	Test	48.23%
ResNet50	Val	60.07%
	Test	54.14%
ResNet50 + Data Augmentation	Val	65.78%
	Test	55.23%

Figure 8. CNN results - Classification Model.

age records the FCP one records all subjects' ages as integers while the IXI one uses decimal numbers classifying the ages into a fixed number of classes helps normalize the labels to some extent. The medium post we are referencing to divides the subjects ages into 3-, 5- and 7-years bins. To start, we divided the ages (starting from 18) into 14 classes, with each class spanning 5 years except for the last one, since there are fewer samples whose ages are above 83 (18 + 13 * 5) in the FCP and IXI dataset.

6.2. Hyperparameters

Since both ResNets were trained on the ImageNet dataset which classifies images into 10 classes, we need to swap in our own fully connected layer at the end and fine tune it. To fine tune our adapted model, we tried different combinations of hyperparameters and obtained the most optimal ones.

In terms of batch size, according to what we learned from CS231n's assignments, a larger batch size is more likely to result in overfitting. Since we dont have many data, we decided to try small batch sizes and then select the best size. Specifically, we tried 4 and 16, and the results clearly demonstrated that a batch size of 4 is better. Thus, we stuck to a batch size of 4 throughout the whole training process.

As for optimizers, we mainly tried two optimizers Nesterov gradient descent (GD) and Adam because these two have been proven to perform steadily well in a lot of research. After trying a range of learning rates between 0.0001 and 0.1, we decided to use 0.001 as our learning rate, and we decayed the rate by a gamma factor of 0.1 per 5 epochs. For Nesterov GD, we chose the standard momentum 0.9. We usually first trained each model for 10 epochs and had a look at the preliminary results, if it seemed to perform well, then we trained the model for 25 epochs. Sometimes, if our model was still improving after 25 epochs, we again increased the number to 50 epochs.

6.3. Address Overfitting

For each model, we first trained on the IXI dataset, given that its age distribution is more even. However, because the total number of examples is relatively small, our model overfitted the data and did not have good test results. Therefore, we attempted to use data augmentation. Nevertheless, when we tried to use data augmentation by incorporating the FCP dataset into our input to reduce overfitting, the model did not improve a lot due to the significant imbalance of age classes. However, when we finally integrated all three datasets (IXI, FCP and the normal subjects in ADNI) together, we were able to achieve a much lower validation error and a comparable test error.

6.4. Explanation of Failures

One example failure is shown in Figure 9. As shown, we did not generate very good results, which we think is mainly due to a great imbalance in the age classes of subjects in our datasets. As we can see in Figures 10 and 11, the histograms show the distribution of age and age classes of the IXI and FCP datasets. Although IXI has a relatively balanced age distribution, there are still a few missing classes. The FCP dataset obviously has a lot more subjects whose ages are in the first two classes (18 - 28 years old).

7. Additional Experiments

Because reducing a 3-dimensional image to a 2-dimensional one might lead to a significant loss of information, we decided to preserve the three dimensions of our input images and try training our own model from ground up. However, due to time limitation and the lack of an authoritative CNN architecture on 4-dimensional images, we were only able to design and train two mini-models, one of which resembles the 10-layers CNN implemented by [6], and the other uses 3-dimensional layers instead of the usual 2-dimensional ones. The architecture of the two models is very similar, except that one uses 2-dimensional Convolutional, BatchNorm and Average Pooling layers (where the

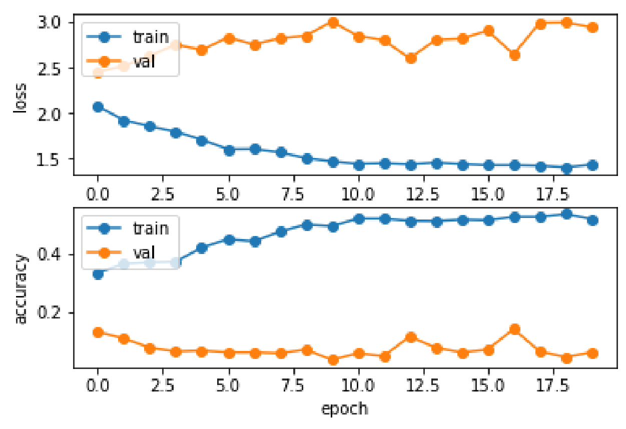


Figure 9. An example of failed learning curves.

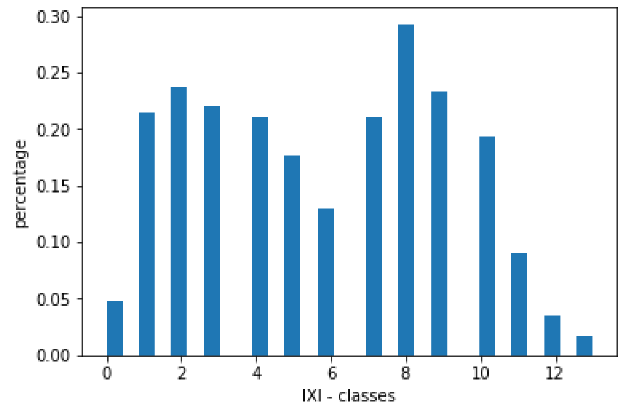


Figure 10. Histogram of the age classes distribution of the IXI dataset.

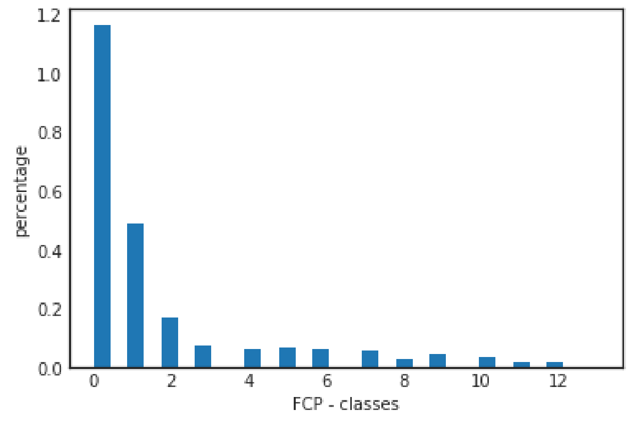
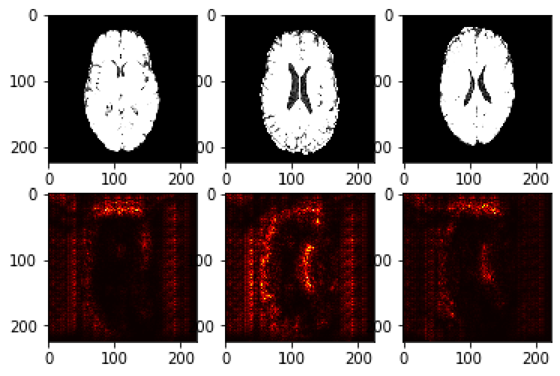


Figure 11. Histogram of the age classes distribution of the FCP dataset.

sagittal dimension of the image is treated as the equivalent of an RGB channel) and the other uses 3-dimensional layers. The architecture of the two models is illustrated in the methods section. The results are summarized in the tables in Figure 7 as well. Even though they didnt produce ideal results, we believe we will be able to improve them by adding more layers. We are still exploring better designs of these two models.

Additionally, we took the extra step to produce the saliency maps to understand what the network is actually learning. The saliency map is a visual representation of the locations of the input image that are considered ”most relevant” to the prediction of the model. To generate the saliency maps, we calculate the gradients of the input image with respect to the prediction. The intuitive understanding is that we will be able to see what portions of the input image have the steepest gradient, thus giving us an opportunity to understand which parts of the input are the most crucial. Here is a sample saliency map generated, with the first row being input images and second row being the corresponding saliency maps:



We see that the network is most interested in the gray matter, as well as some peripheral regions of the brain. Perhaps the most telling information about brain age is centered around the frontal cortex, as well as the gray matter area. In future research, we will also look into the connection between network activation and actual findings in neuroscience to corroborate the effectiveness of our deep convolutional networks.

8. Conclusion and Future Work

Throughout the project we explored different aspects of the brain age estimation problem and achieved results comparable to state-of-the-art performance. Specifically, in the data processing phase, we experimented with a few different methods and found the best pipeline - fsl_anat - to scientifically process the 3-dimensional brain MRIs. After massive data processing, we first tried our baseline model - linear regression - on our data (represented as histograms), and achieved quality results. We then took a step further by trying different CNN models using both regression and classification approaches. We implemented the pre-trained ResNet18 and ResNet50 models and fine-tuned the entire model. After comprehensive fine-tuning, using a regression approach, we were able to obtain an age error as low as 0 to 4 on validation dataset and 5 to 8 on test dataset. A pre-trained ResNet50 with data augmentation performed the best. Although our classification approach did not perform well, we figured out that it might due to the great imbalance in age classes in the dataset and will mitigate the errors later by collection and cleaning more data, and trying more complicated models to narrow the discrepancy between training and validation accuracy. Additionally, since most available models only take in 2D RGB images, and our data are 3D black and white MRIs, we believe it was worth trying our own models that preserve the input dimensions. Therefore, we also tried two mini CNN models, one using 10 2D Convolutional layers and the other using 6 3D Convolutional layers accompanied by Average Pooling and BatchNorm layers. Although they did not yield the best results, we believe they have the potential to perform better if we keep adding layers and fine tuning them.

Due to time limitations, we have not finished all of our goals. We have a list of things we plan to do in the future: a) improve our test errors; b) improve our classification accuracy by collecting more data and trying more complicated models; c) enhance the architectural design of our own mini-models (both 2D and 3D ones) ; d) test our trained model on patients with Alzheimer disease.

Most importantly, we learned a great deal from this project in terms of problem abstraction and formulation, data processing, model selection, CNN architecture design, model training, tuning and model evaluation. We are truly grateful for this valuable opportunity to learn and grow on our way to become experienced computer vision practitioners.

9. Contributions and Acknowledgements

We would like to thank Alzheimers Disease Neuroimaging Initiative to give us the access to their database of 1724 subjects. Also, we would like to thank brian-development.org website for providing us with the IXI dataset under the Creative Commons CC BY-SA 3.0 license. Finally, we would also like to thank FCP Classic Data Sharing Samples for providing us with datasets across different hospitals.

We would also like to thank the TAs who helped us at various steps during the project, specifically Nishith and Owen for helping us defining the problem statement and guiding us in taking the right direction with the project.

All three of us contributed equally towards the project and final modeling, with Zixian taking extra initiatives on understanding the data pre processing pipeline, Prabhjot making sure that the dev-ops for the project is planted right using AWS and Harry going a step further on analysing the established models and tweaking the architecture. We collaborated on github, the repository being https://github.com/raiprabh/brain-age.

References

[1] B. Avants, N. Tustison, and G. Song. Advanced normalization tools. *Electronic Distribution*, 2011.

[2] J. H. Cole, R. P. Poudel, D. Tsagkrasoulis, M. W. Caan, C. Steves, T. D. Spector, and G. Montana. Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker. *NeuroImage*, 163:115–124, 2017.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[4] J. Islam and Y. Zhang. Brain mri analysis for alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks. *Brain Informatics*, 6, 2018.

[5] M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich, and S. M. Smith. Fsl. *Neuroimage*, 62(2):782–790, 2012.

[6] S. Jgou. How to estimate the age of your brain with mri data. *Medium - The Launchpad*, 20192.

[7] T. Kaufmann, D. van der Meer, N. T. Doan, E. Schwarz, M. J. Lund, I. Agartz, D. Alnæs, D. M. Barch, R. Baur-Streubel, A. Bertolino, et al. Genetics of brain age suggest an overlap with common brain disorders. *bioRxiv*, page 303164, 2018.

[8] B. Patenaude, S. M. Smith, D. N. Kennedy, and M. Jenkinson. A bayesian model of shape and appearance for subcortical brain segmentation. *Neuroimage*, 56(3):907–922, 2011.

[9] S. M. Smith. Fast robust automated brain extraction. *Human brain mapping*, 17(3):143–155, 2002.

[10] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. Behrens, H. Johansen-Berg, P. R. Bannister, M. De Luca, I. Drobnjak, D. E. Flitney, et al. Advances in functional and structural mr image analysis and implementation as fsl. *Neuroimage*, 23:S208–S219, 2004.

[11] M. W. Woolrich, S. Jbabdi, B. Patenaude, M. Chappell, S. Makni, T. Behrens, C. Beckmann, M. Jenkinson, and S. M. Smith. Bayesian analysis of neuroimaging data in fsl. *Neuroimage*, 45(1):S173–S186, 2009.

[12] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57, 2001.