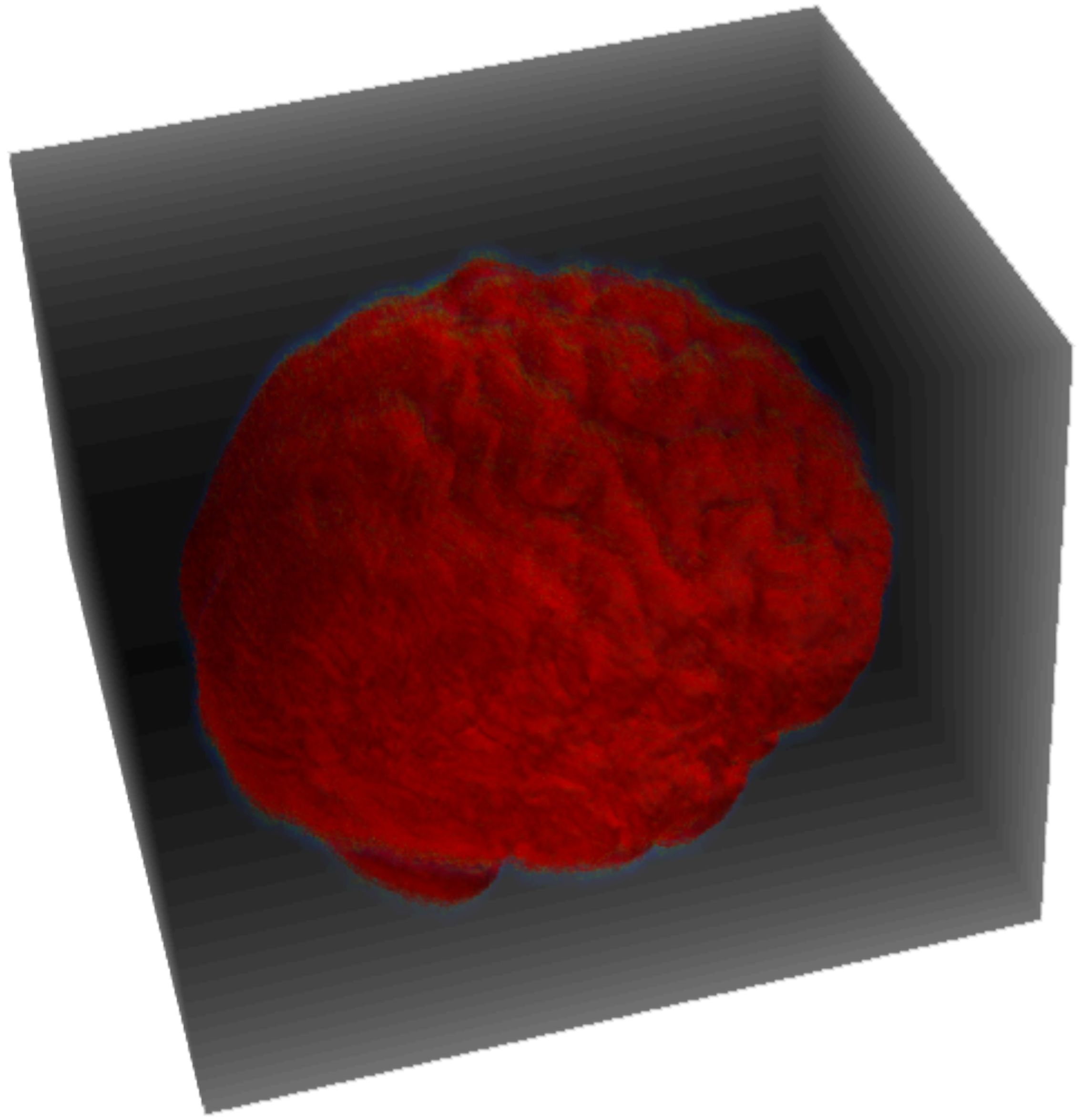
Predicting brain age from neuroimaging using Machine Learning and Deep Learning

Final Dissertation

Ziyang Wei

Computer Science

28th Aug 2019

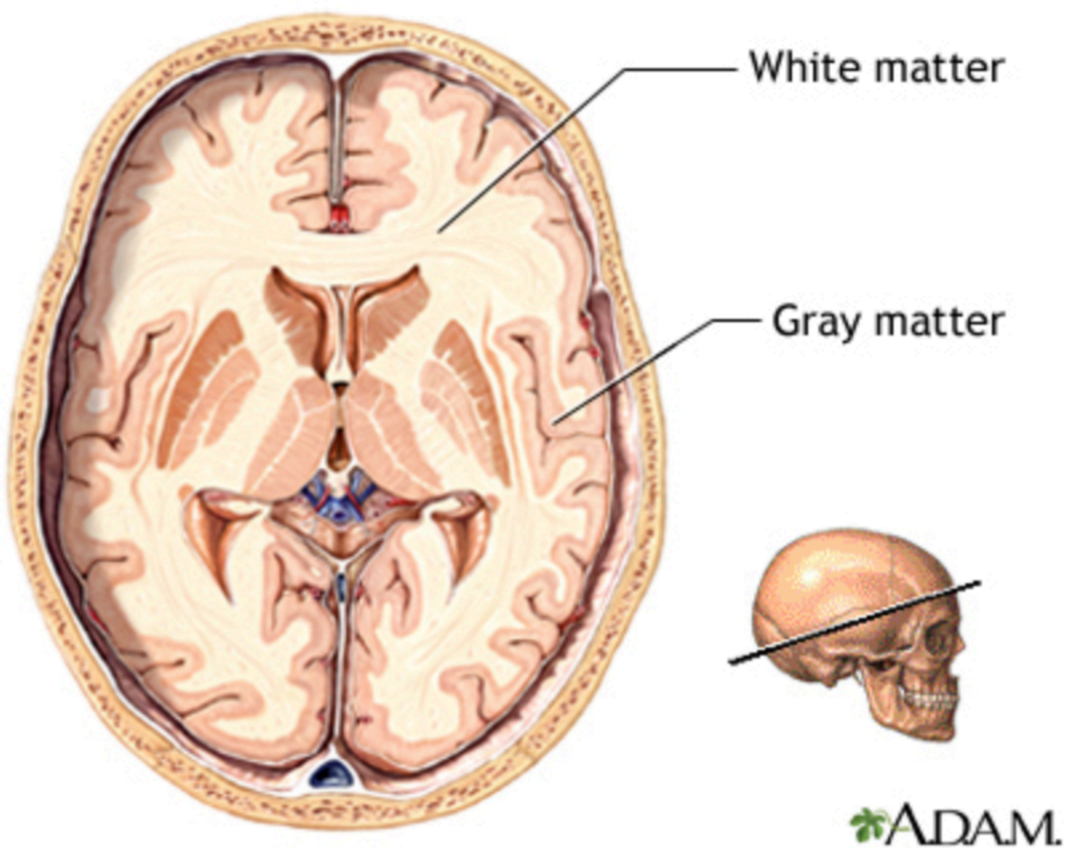


Outline

- **Background**
- **Dataset Overview**
- **Image Preprocessing**
- **Machine Learning Method**
- **Deep Learning Method**
- **Conclusion**

Background

What do we know about our brain?



Anatomy of human brain

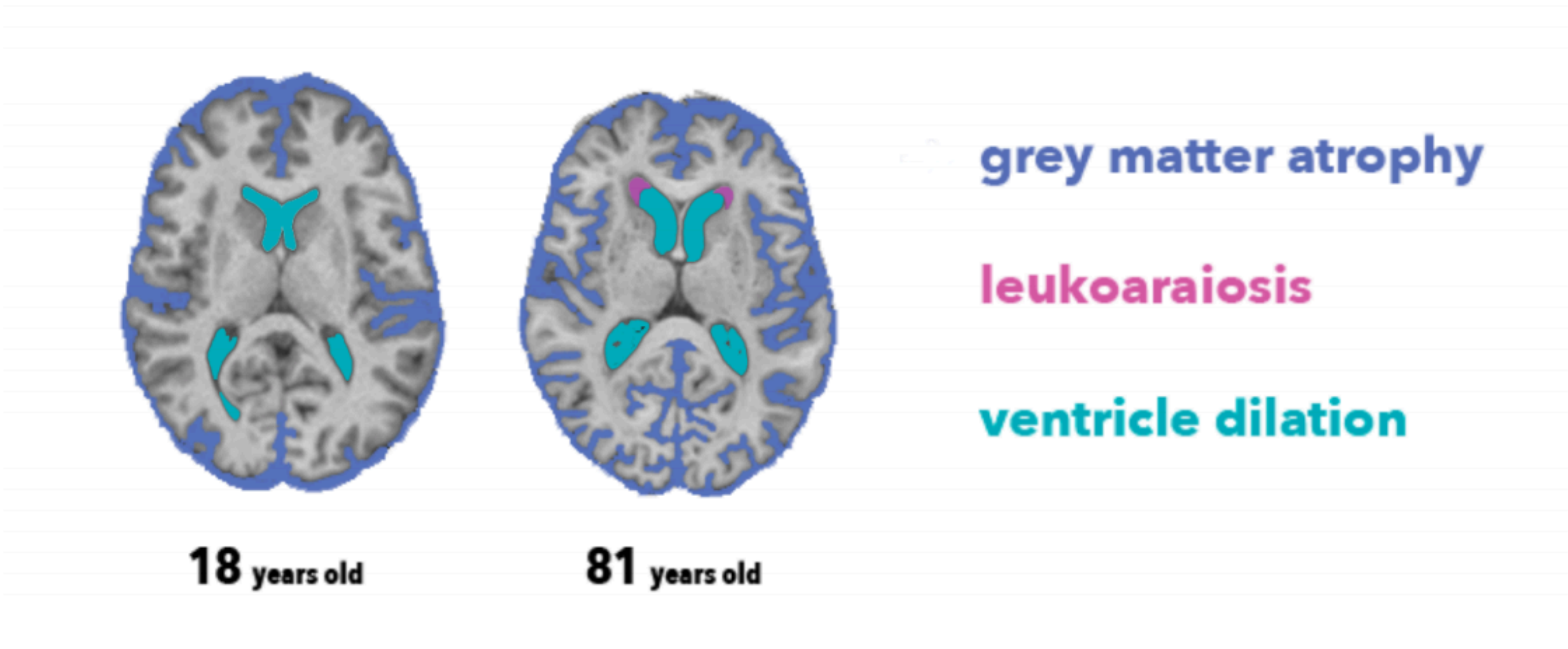


FIGURE 1.3: 18 year old brain versus 81 year old brain

Our task

Reveal the effect of aging on human's brain and predict the human brain age based on the captured feature.

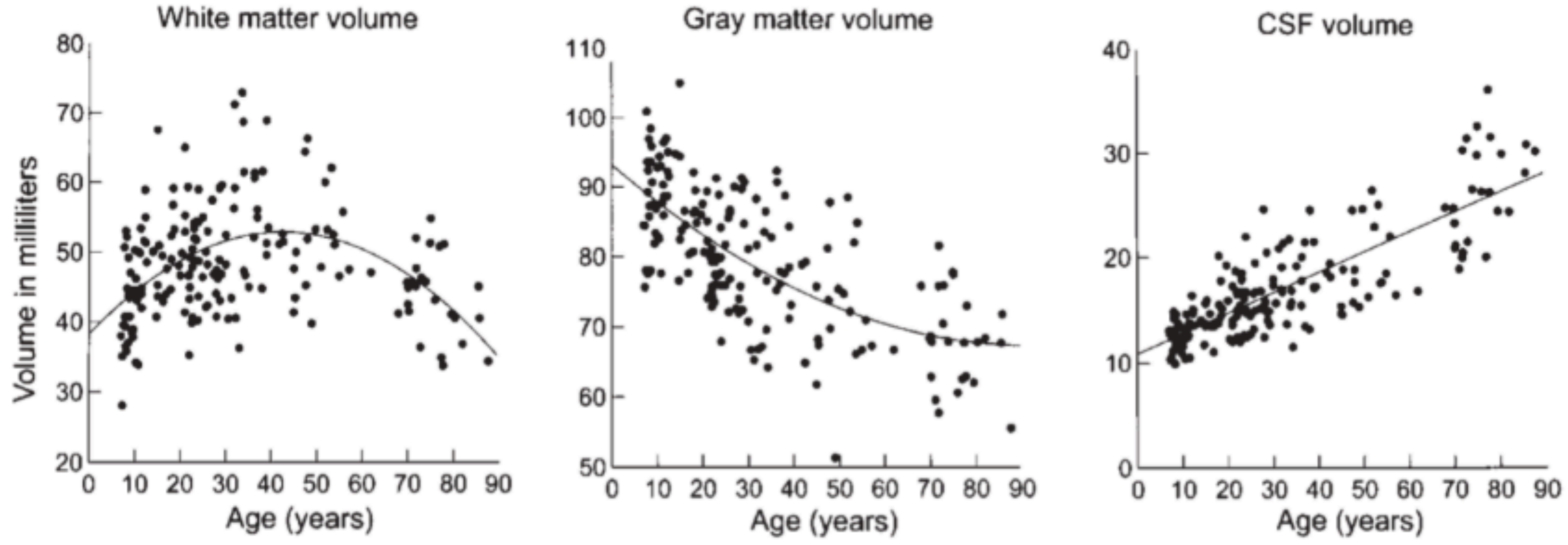
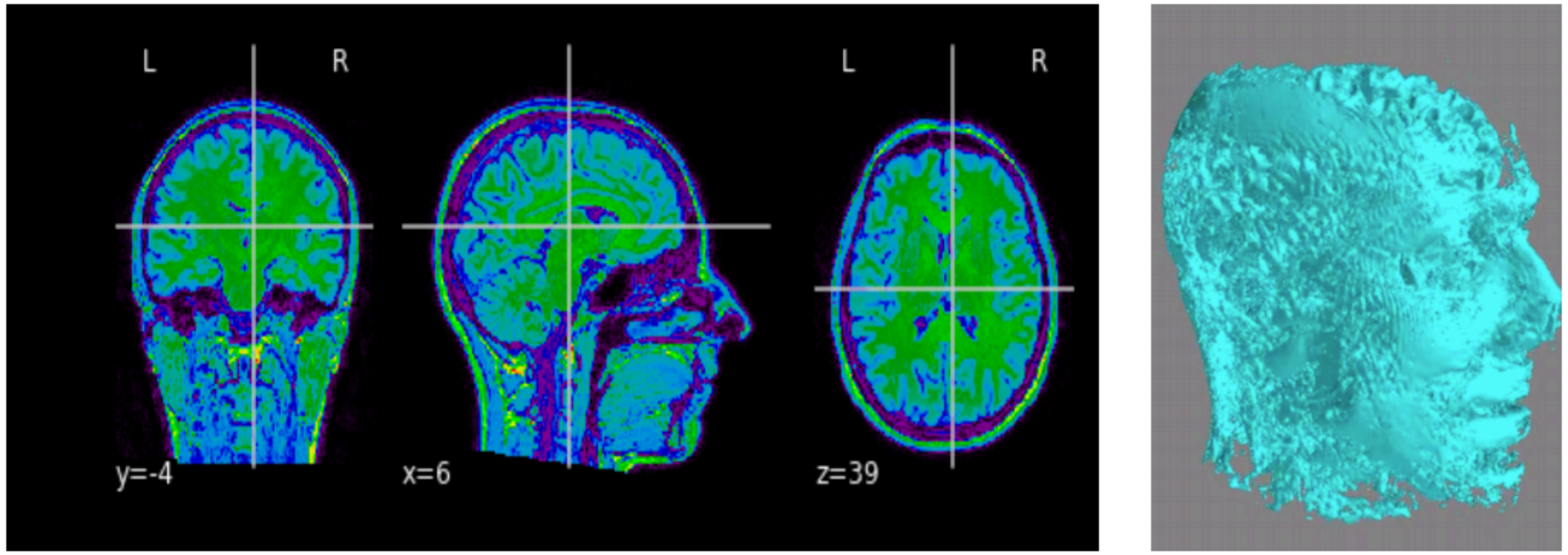


FIGURE 1.4: Scatterplots of the nonlinear effects of age on total brain white matter, total brain gray matter and total brain CSF volume.

Data Overview

.Nifti file format



(a) nilearn (b) mayavi

FIGURE 1.5: Visualize .nii file with thrid party library

$$\begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & a \\ m_{2,1} & m_{2,2} & m_{2,3} & b \\ m_{3,1} & m_{3,2} & m_{3,3} & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Affine Matrix contained in the Nifti file

Data Source

Our training-set are collected from 20 hospitals and universities all around the world.

Cohort	N	Age mean(SD)	Age range	Sex(male/female)	Repository details
Beijing	198	21.16(1.82)	18.0-26.0	76/122	INDI
Atlanta	28	30.89(9.72)	22.0-57.0	13/15	INDI
NewYork	104	25.41(10.17)	7.88-49.16	51/53	INDI
Queensland	19	25.95(3.78)	20.0-34.0	11/8	INDI
ICBM	86	44.19(17.81)	19.0-85.0	41/45	INDI
Oulu	103	21.52(0.57)	20.0-23.0	37/66	INDI
Cambridge	198	21.03(2.3)	18.0-30.0	75/123	INDI
Oxford	22	29.0(3.71)	20.0-35.0	12/10	INDI
PaloAlto	17	32.47(7.87)	22.0-46.0	2/15	INDI
Dallas	24	42.62(19.65)	20.0-71.0	12/12	INDI
NewHaven	35	29.11(8.72)	18.0-48.0	18/17	INDI
Baltimore	23	29.26(5.34)	20.0-40.0	8/15	INDI
AnnArbor	60	36.22(23.97)	13.41-80.0	38/22	INDI
Leipzig	37	26.22(4.94)	20.0-42.0	16/21	INDI
SaintLouis	31	25.1(2.28)	21.0-29.0	14/17	INDI
Bangor	20	23.4(5.18)	19.0-38.0	20/0	INDI
IXI	563	48.65(16.46)	19.98-86.32	250/313	IXI database
Pittsburgh	17	37.94(8.76)	25.0-54.0	10/7	INDI
Leiden	31	22.19(2.53)	18.0-28.0	23/8	INDI
Orangeburg	20	40.65(10.75)	20.0-55.0	15/5	INDI

TABLE 2.1: Data source table

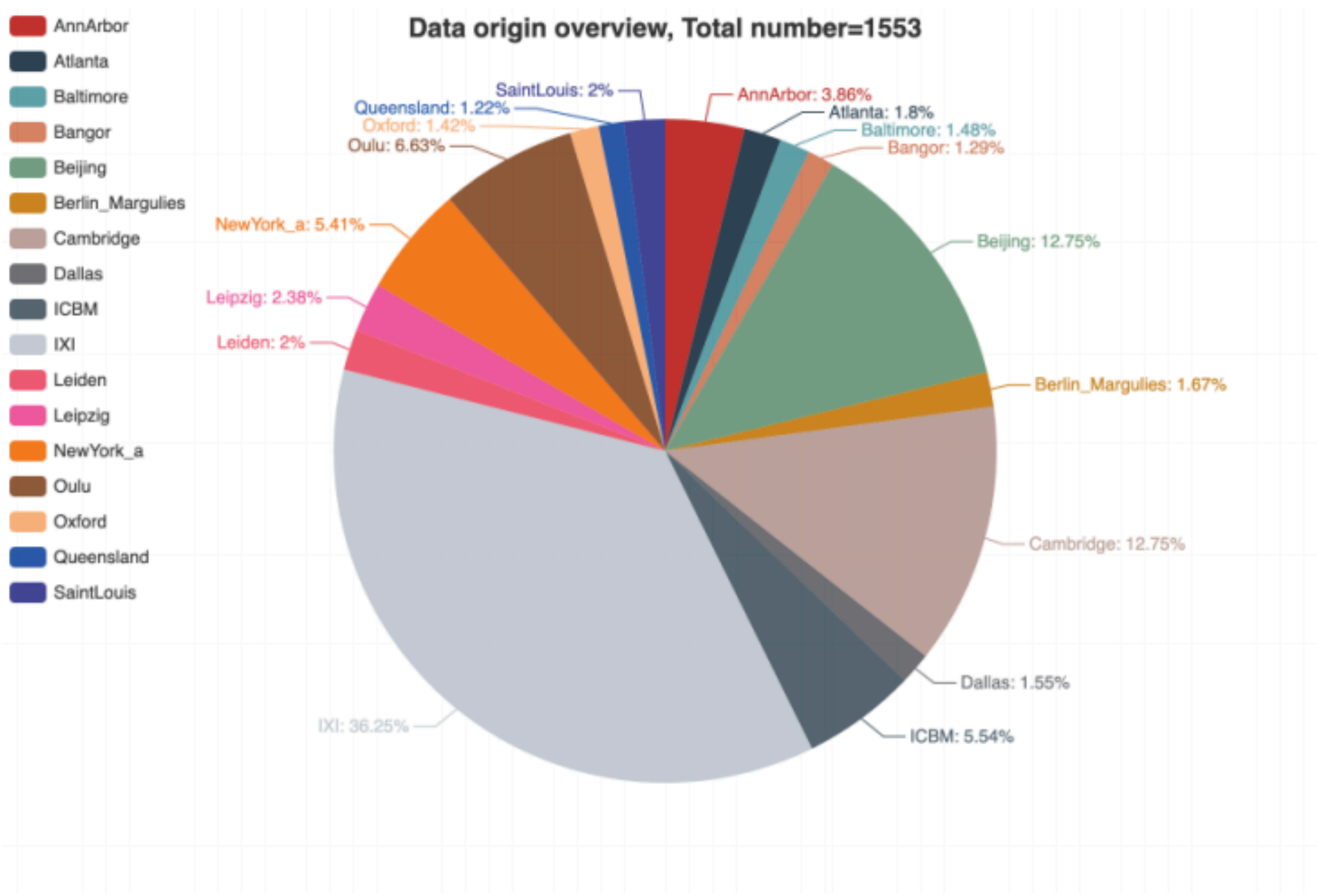


FIGURE 2.1: Data source

Data Distribution

Age Distribution

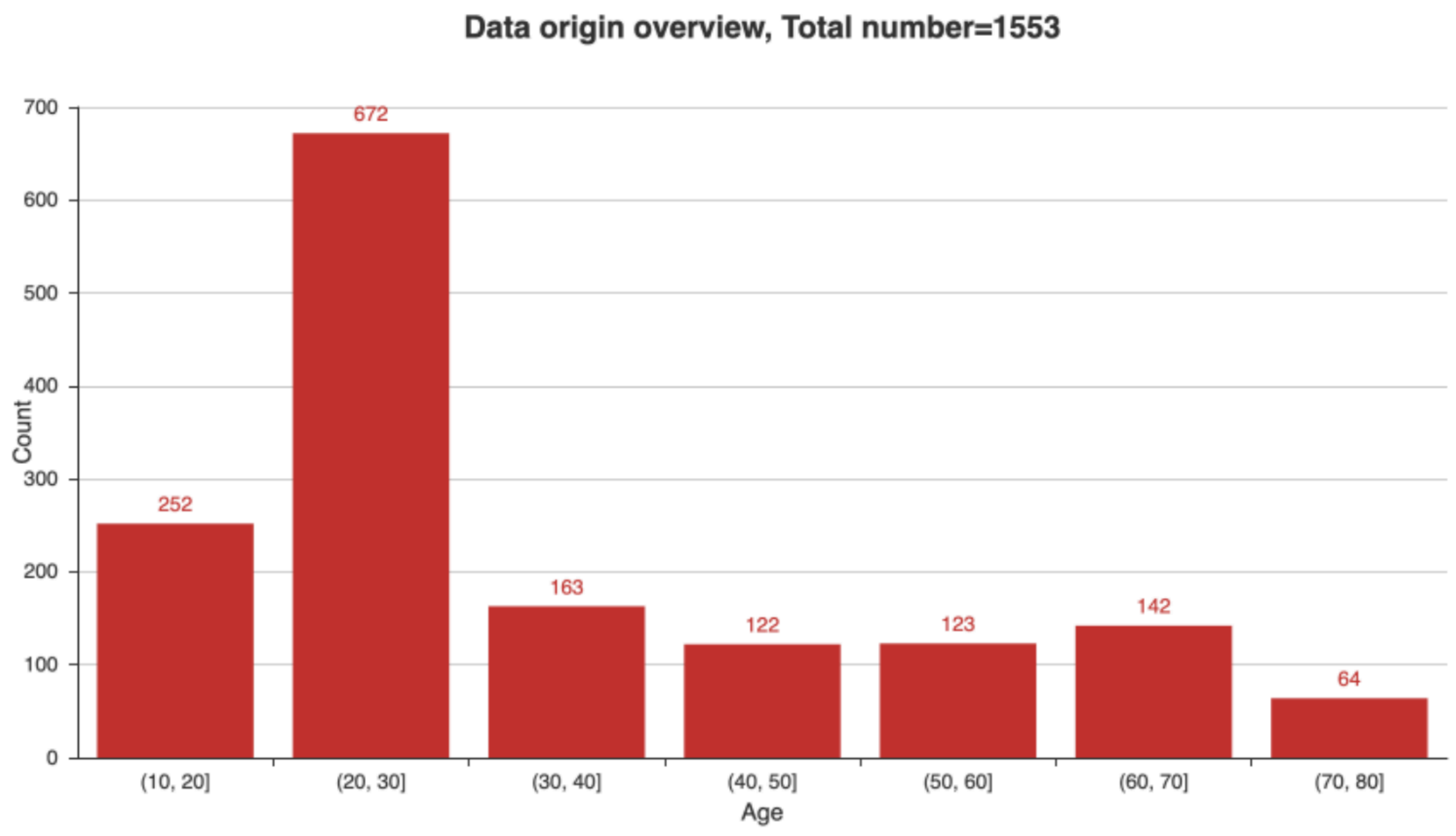


FIGURE 2.2: Age distribution

Gender Distribution

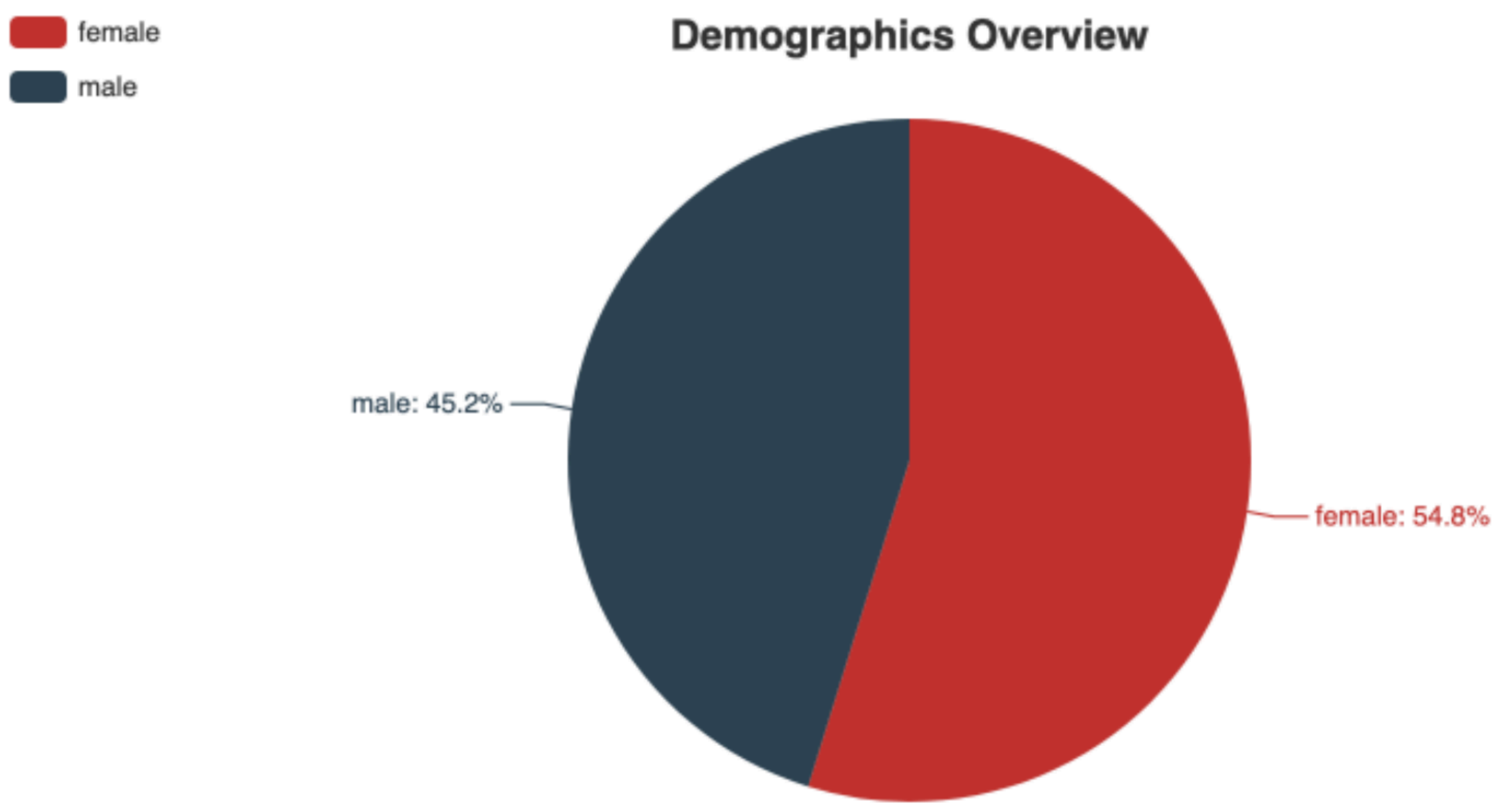


FIGURE 2.3: Sex distribution

Data Cleaning

The training data we collected have some problem raised by the transition of data through complicated data systems or unware manual modification:

- **Some of the subjects have the .nii file but missed in the .csv file**
- **Some of the subjects appear in the .csv file but miss the .nii file**
- **Duplicated problem is very common in the dataset.For example,two subjects may have the same ID but have different age or sex.**
- **Some subjects' age are obviously be marked wrongly.For example, I notice some of the subjects' old are less than 0, which is obviously impossible**

Image preprocessing

Image preprocessing is necessary

Images are collected from different devices, their:

- **resolutions**
- **contrasts**
- **voxel intensity ranges**
- **shapes**
- **field of views**
- **orientations**

are largely different with each other.

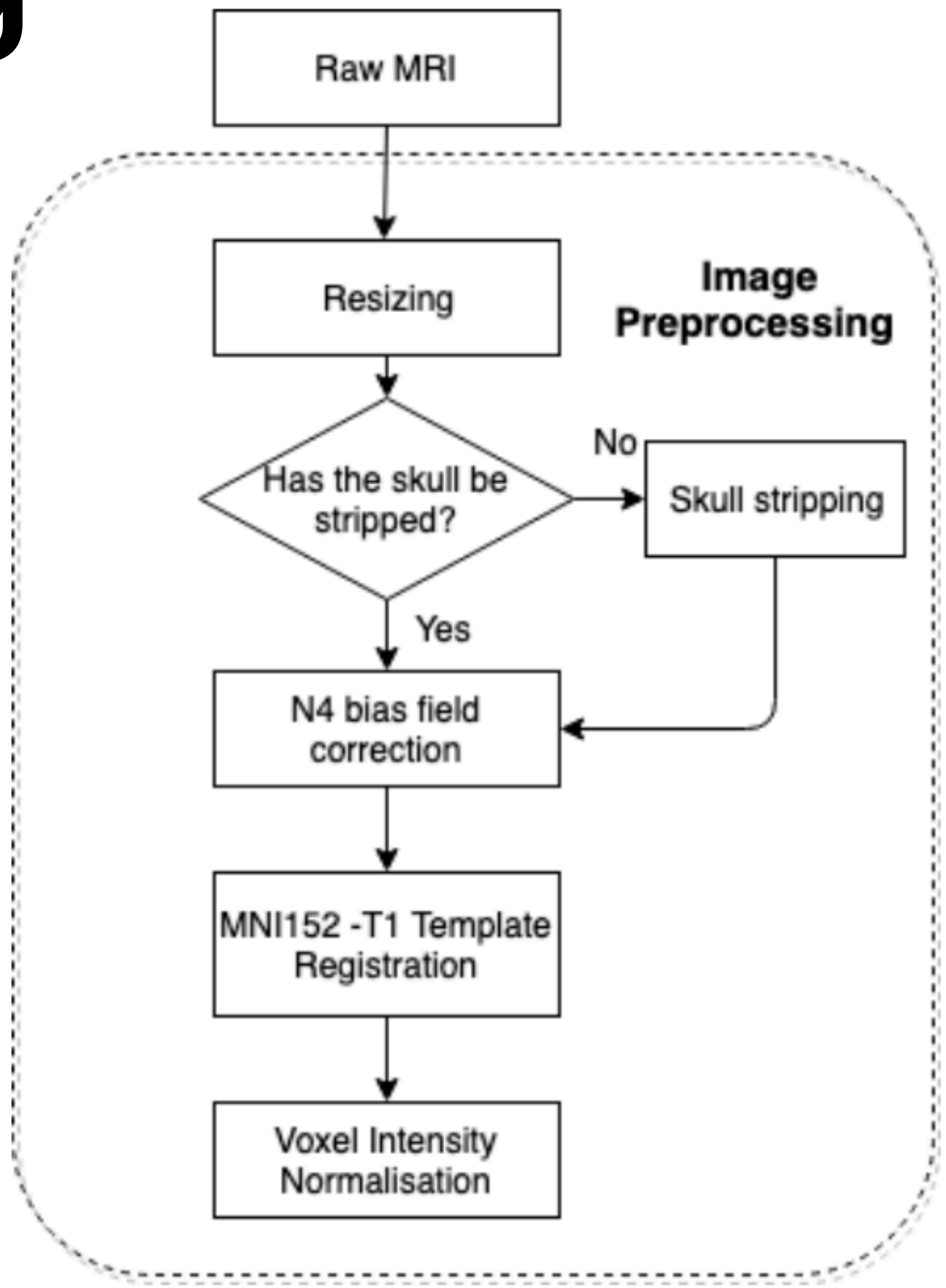


FIGURE 3.1: Overview of image preprocessing

Skull Stripping

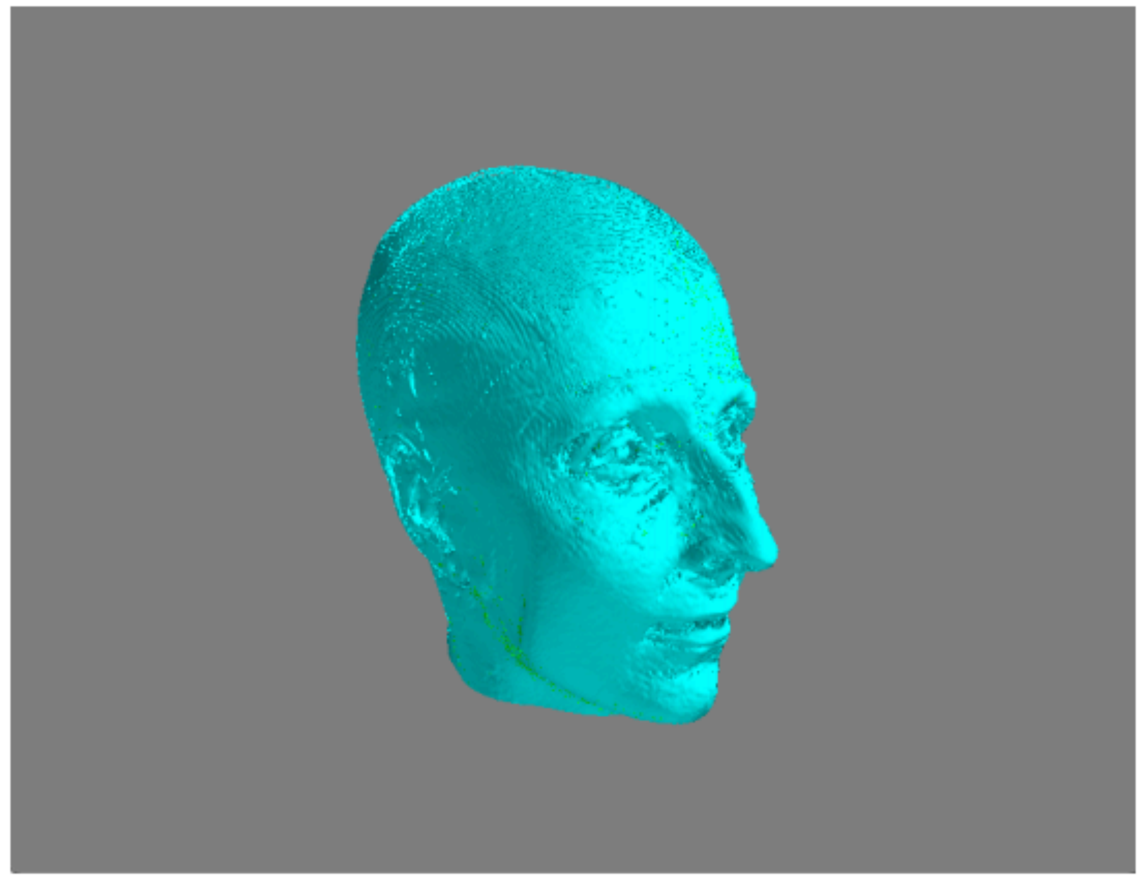


FIGURE 3.2: Image with no skull stripping

Pretrained CNN

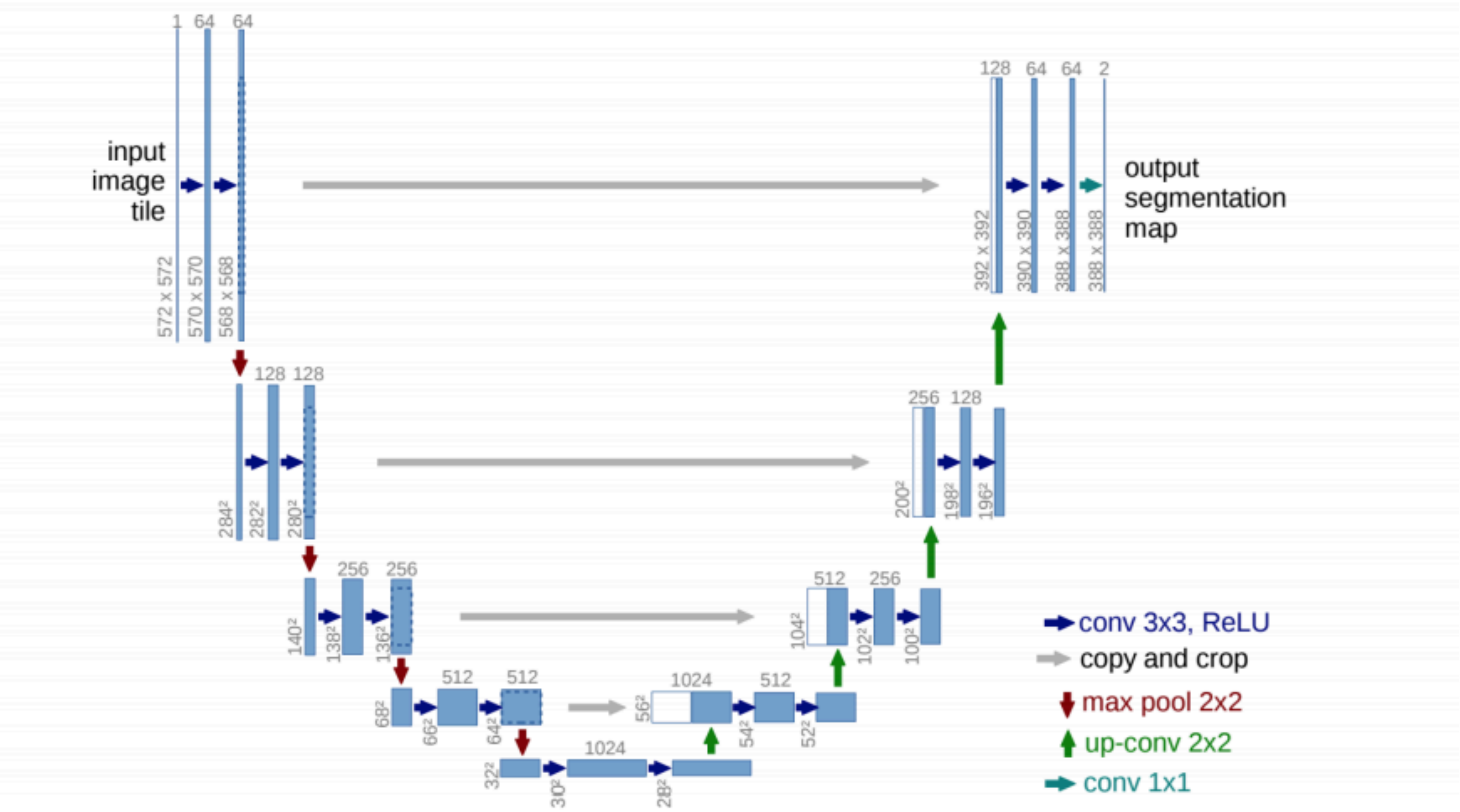


FIGURE 3.3: Unet architecture

Unet

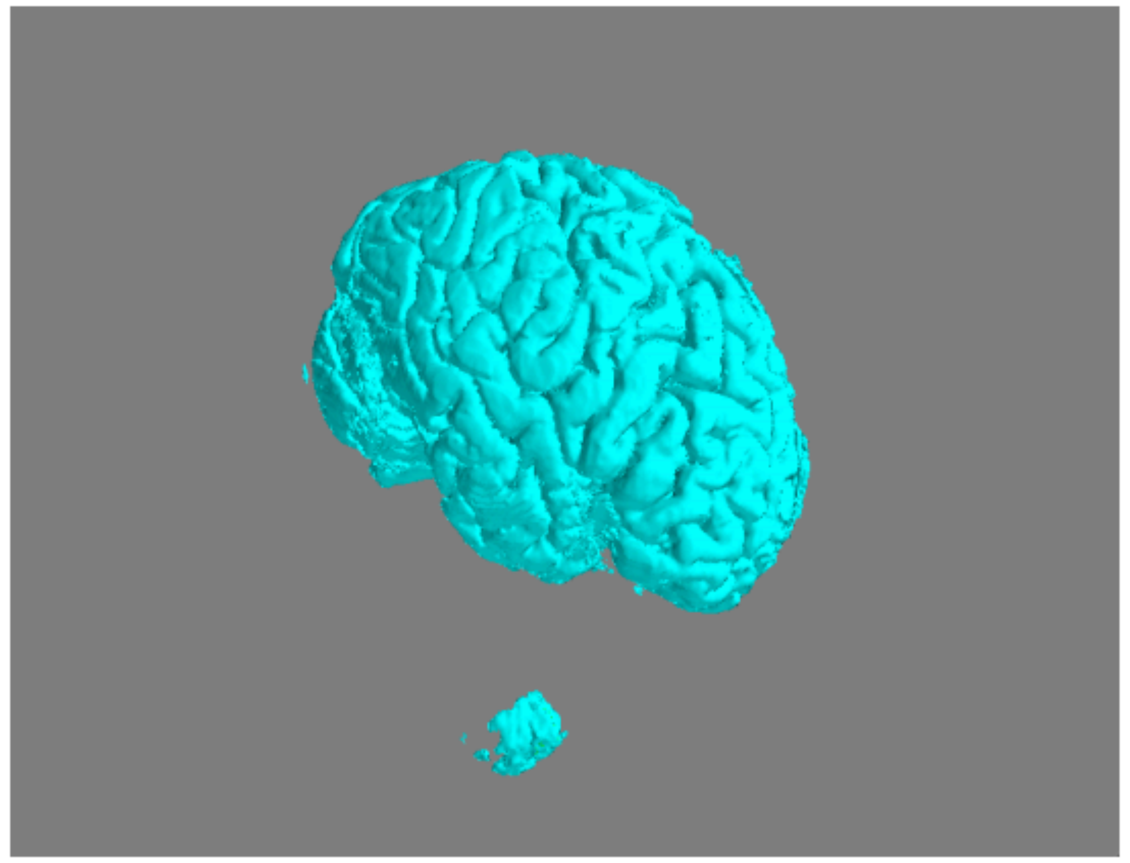


FIGURE 3.4: Output of DeepBrain

Remove small blobs with morphology method

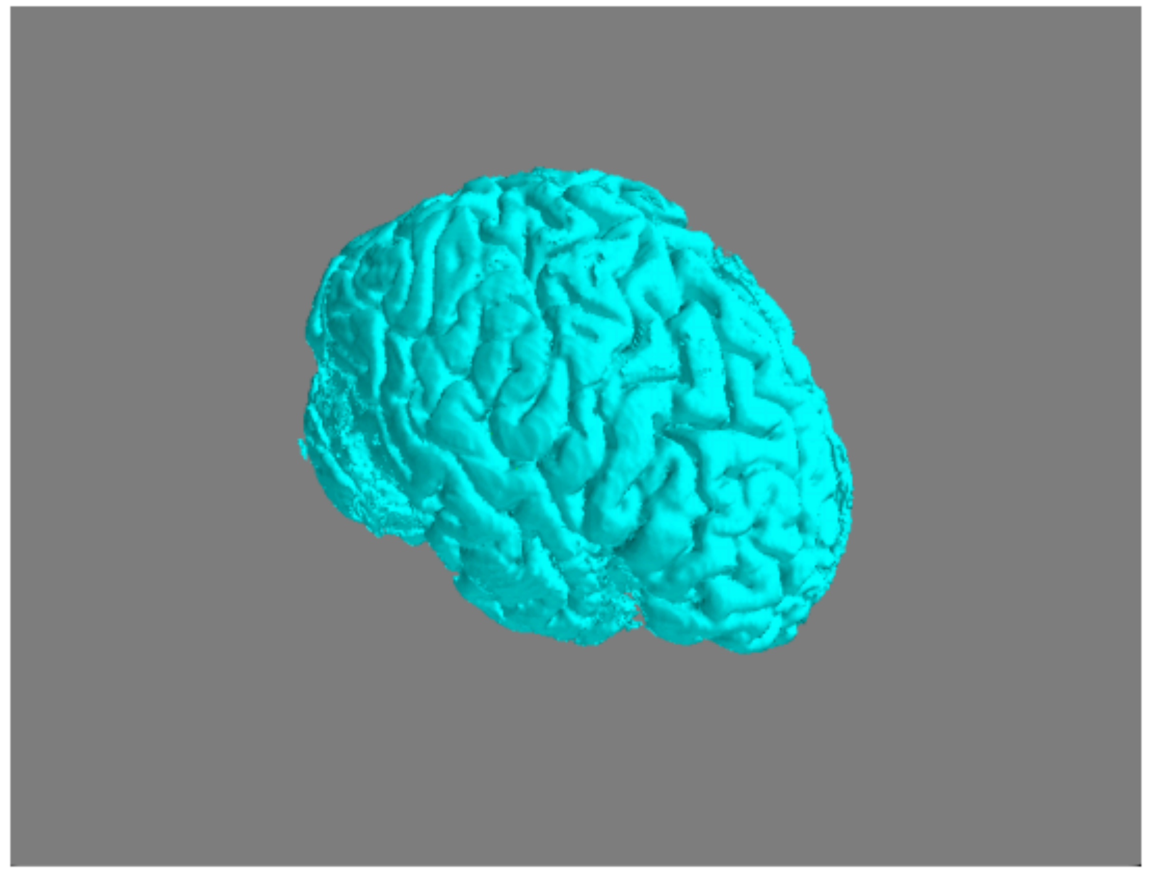


FIGURE 3.5: Final skull-stripped brain

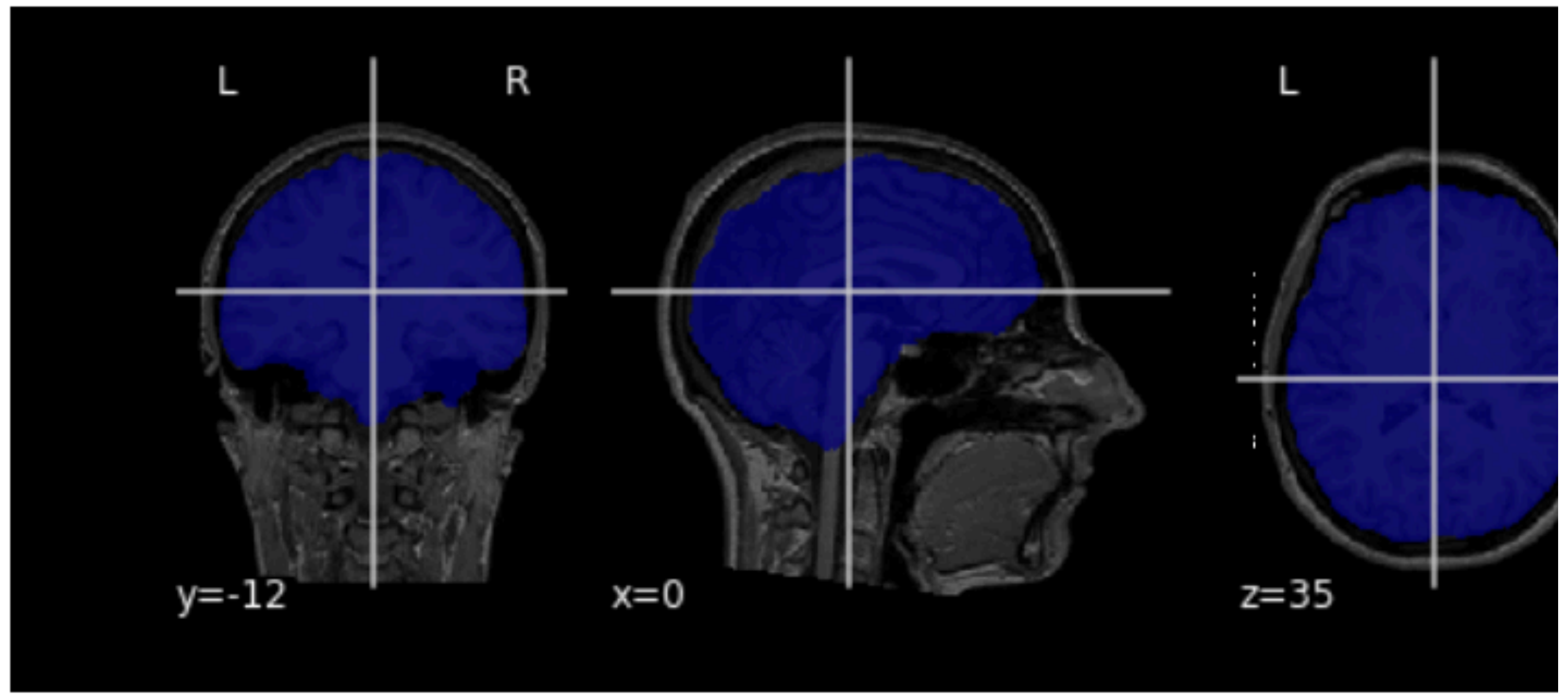
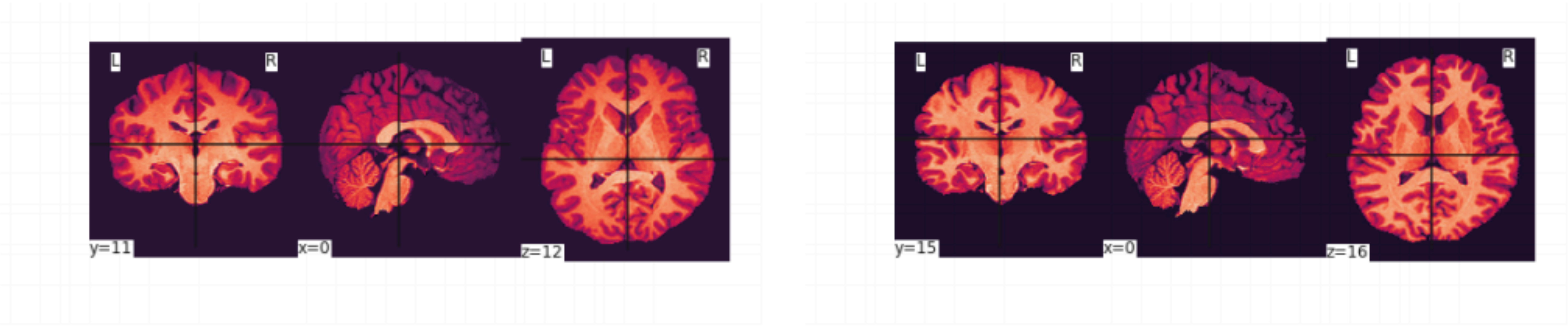


FIGURE 3.6: Brain mask of the whole head

N4 bias field correction

Why we need to apply bias field correction?

Images may be influenced by magnetic field like this:



(a) not corrected

(b) corrected

FIGURE 3.7: Difference between corrected image and not corrected image

I fix it with ‘Simple ITK’, which uses B-spline approximation to compute the magnetic field

It takes more than 5 days to process the training datasets even if I compute them with 5 servers through parallelizing operations!!

Template Registration

Our training dataset are various in shape, orientations.How to fix it???

We can set a standard template brain and align all the training data to have the same shape, orientation by computing the affine matrix .

We denote Y be the template, X be our training data, and T(x) is the affine transformation ,our aim is to search for the transformation (T_*) which gives the minimum cost

$$T^* = arg \min_{T \in S_T} C(Y, T(X))$$

The template for the registration that I used in the project is the MNI152 standard-space T1-weighted average structural template image.I use ‘FSL’ tool to do the registration directly.

Voxel Intensity Normalisation

The range of Images' intensity from our training dataset are different from each other!

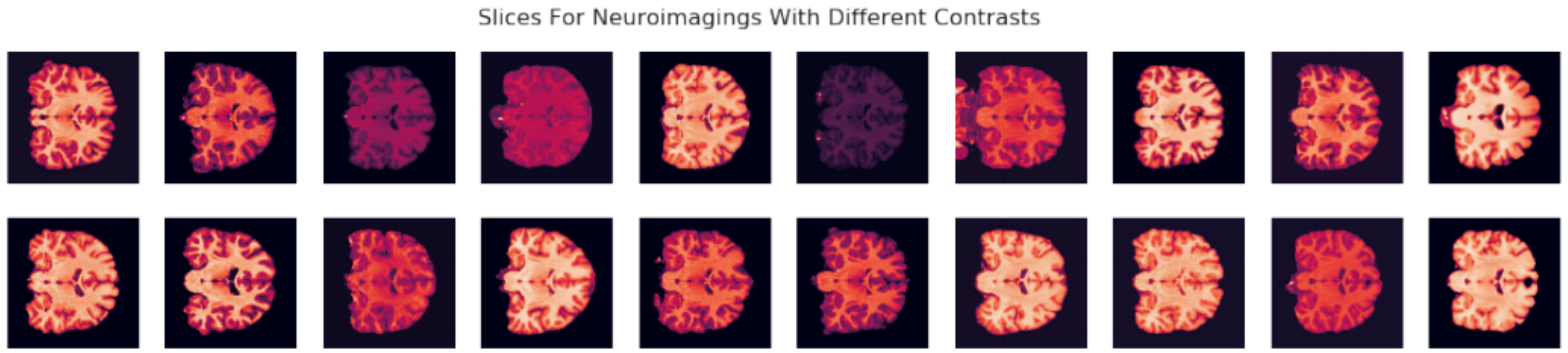


FIGURE 3.9: Slices for neuroimagings from different hospitals

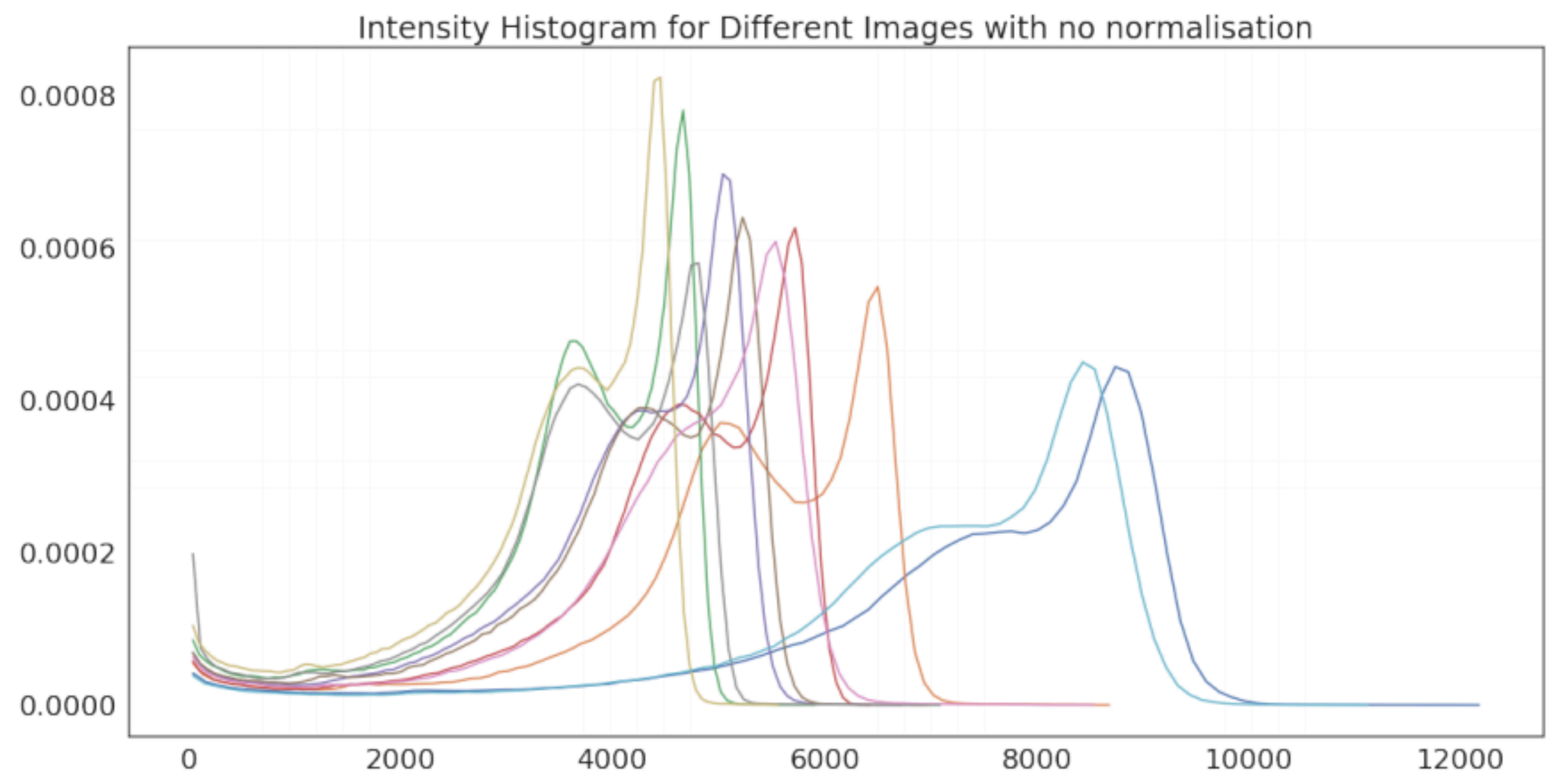


FIGURE 3.10: Intensity histogram for images with no normalisation

I implement three types normalisation method including Z-score normalisation, GMM based normalisation,white stripe normalisation, Simon Jgou's normalisation.

Voxel Intensity Normalisation

Z-Score normalisation:

$$I_{z-score}(x) = \frac{I(x) - \mu_{zs}}{\sigma_{zs}}$$

Gaussian Mixture normalisation:

$$I_{gmm}(x) = \frac{c_2 * I(x)}{\mu_{gmm}}$$

The value of μ_{gmm} can be computed by Gaussian component .

White-stripe normalisation:

$$I_{ws}(x) = \frac{I(x) - \mu_{ws}}{\sigma_{ws}}$$

Simon's normalisation:

$$c = \frac{\mu_{gm}}{\mu_{wm}}$$

$$a = \frac{c_4 - c^2}{c - c^2}$$

$$I_{sj}(x) = a * I(x) + (a-1) * I(x)^2$$

Voxel Intensity Normalisation

GMM normalisation:

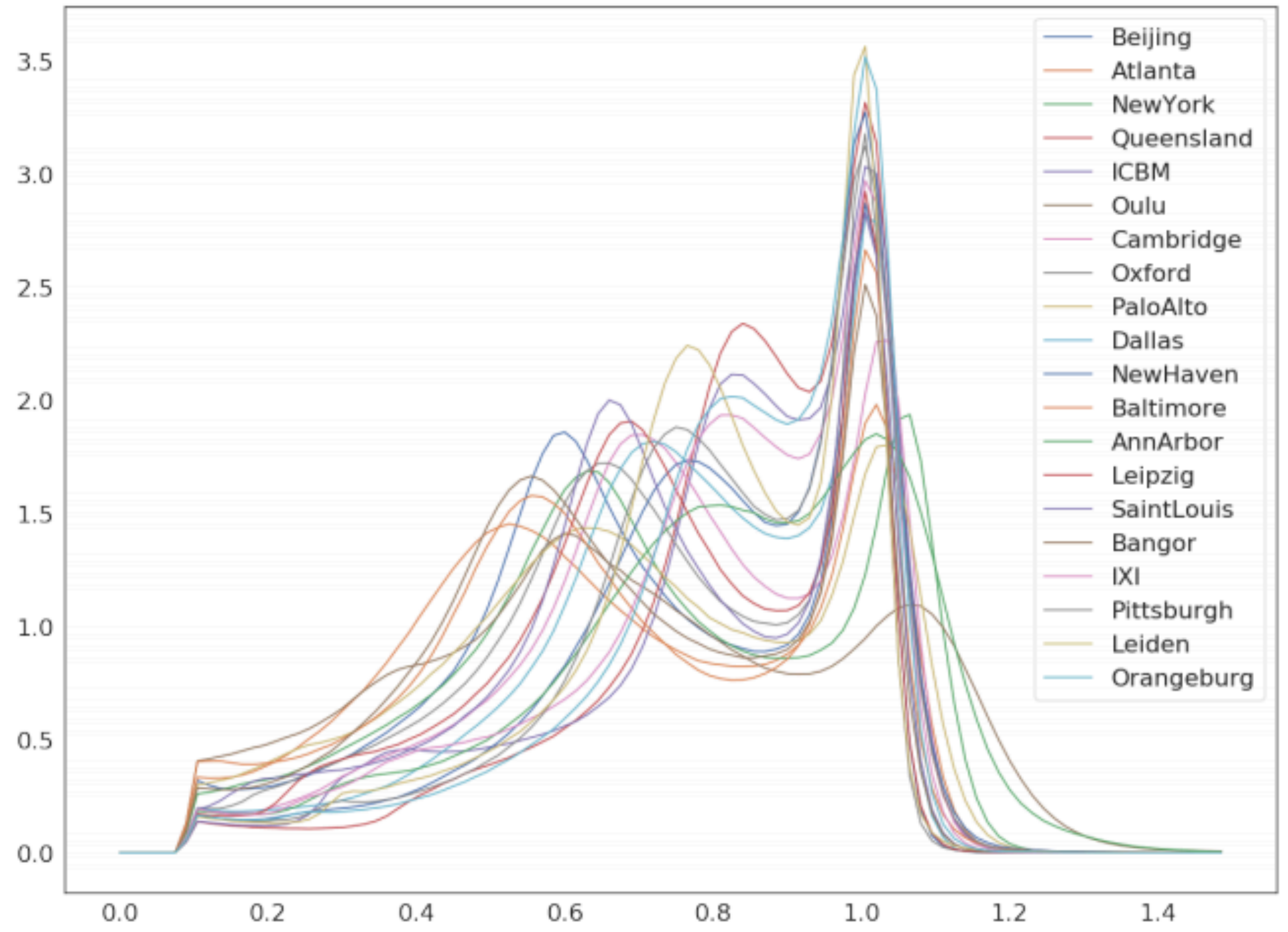


FIGURE 3.11: Average per-hospital intensity histogram with GMM norma

Homemade normalisation:

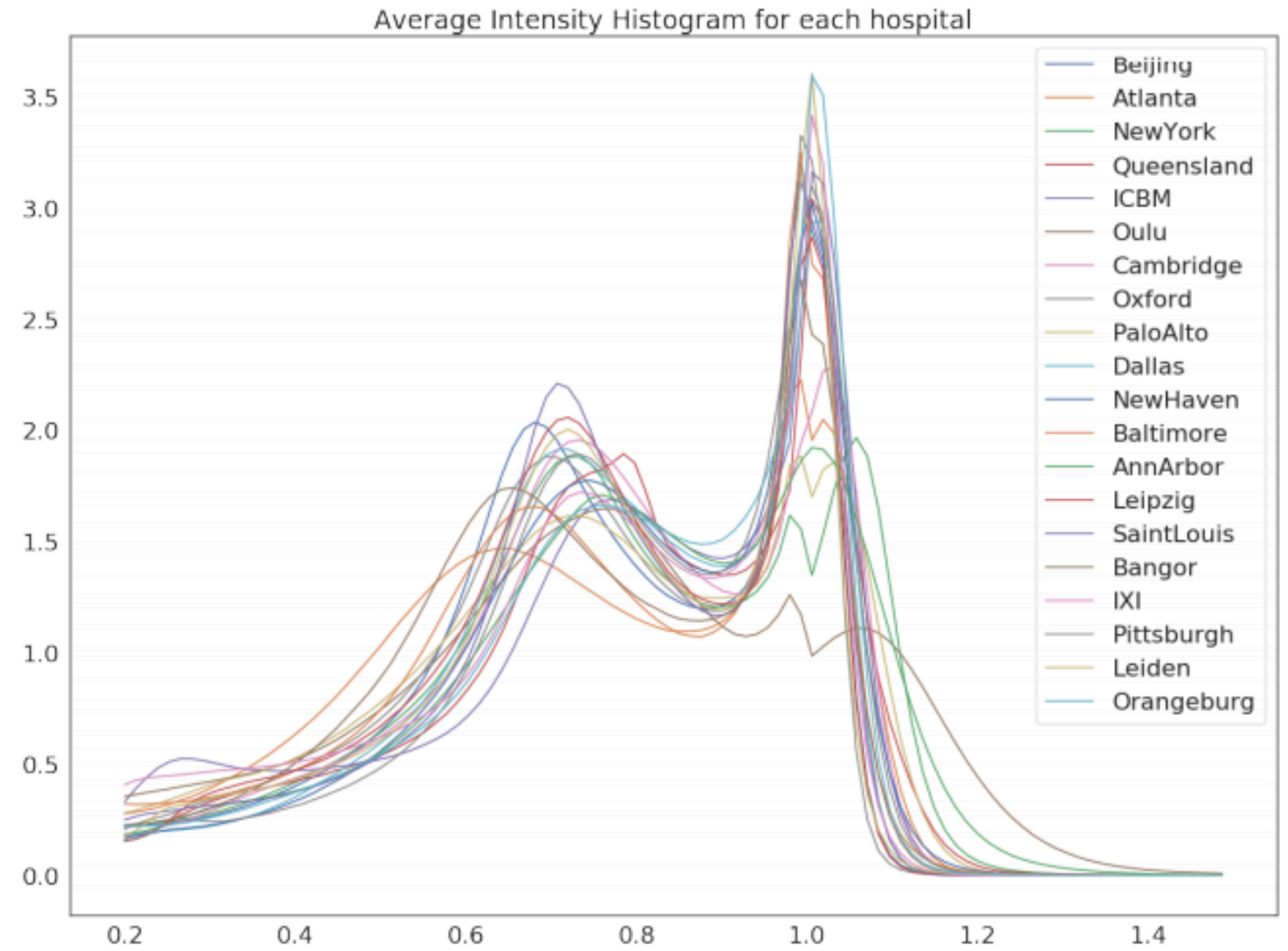


FIGURE 3.13: Average per-hospital intensity histogram with the homemade normali-

Normalisation Result:

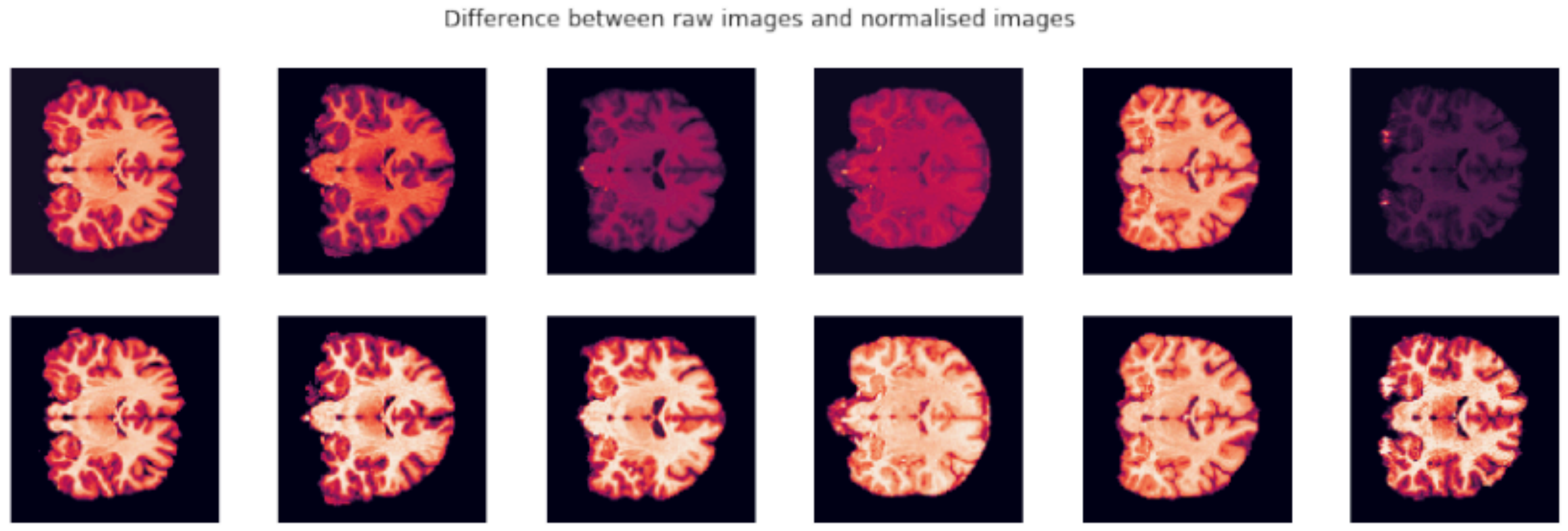


FIGURE 3.14: Difference between raw images and normalised images

Machine Learning Method

The intensity histogram can represent our brain!

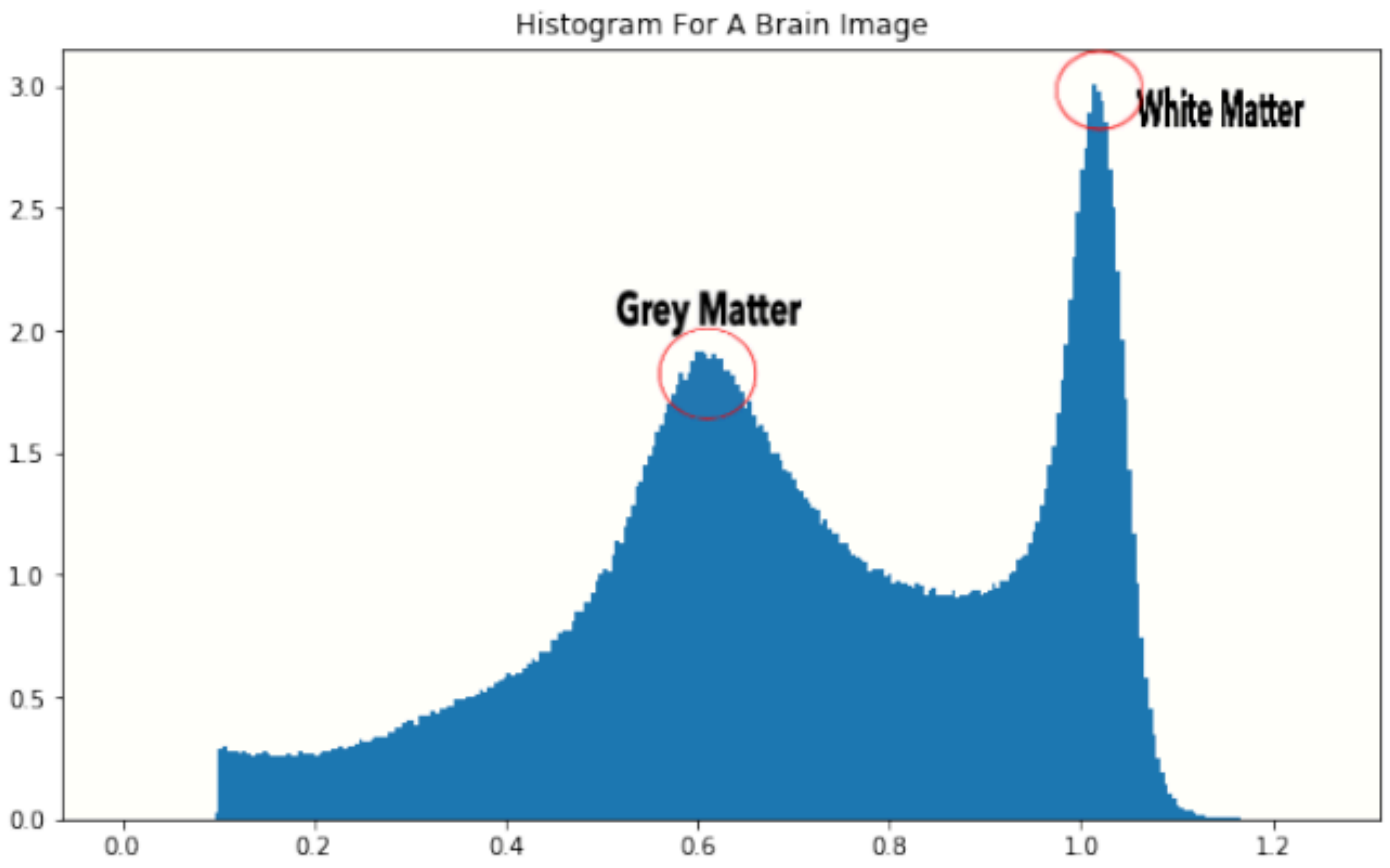


FIGURE 4.1: Histogram for a brain

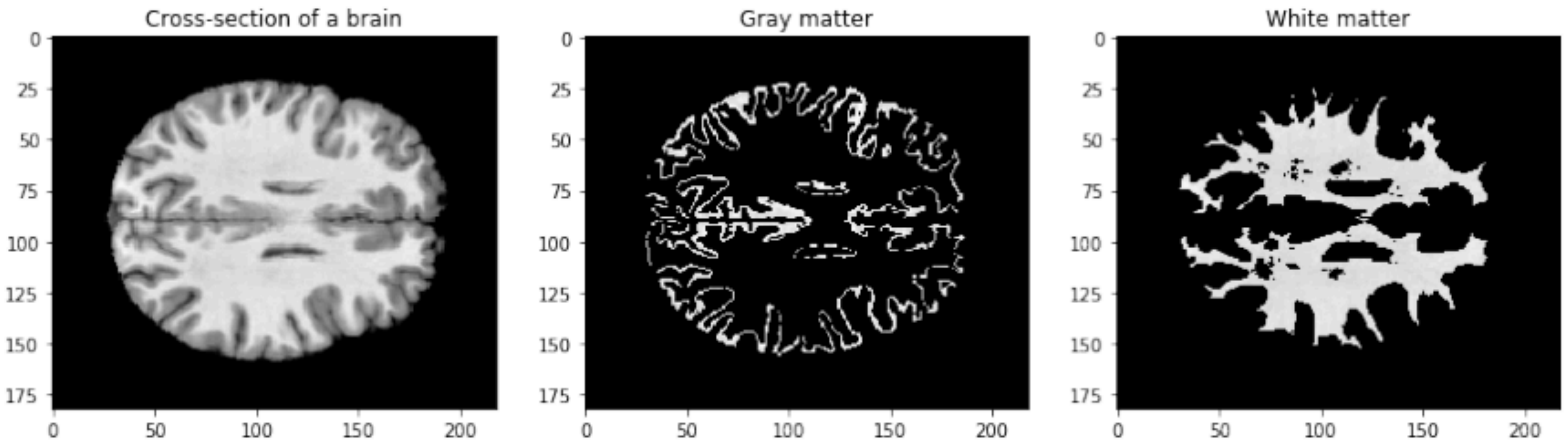


FIGURE 4.2: The cross-section image for grey matter and white matter

Different model yield different histogram!

Linear Regression

Linear model:

$$f(x_i) = \sum_{m=1}^{p} w_m x_{im} + w_0 = w^T x_i$$

Loss function:

$$J(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 = \frac{1}{n}\|y - Xw\|^2$$

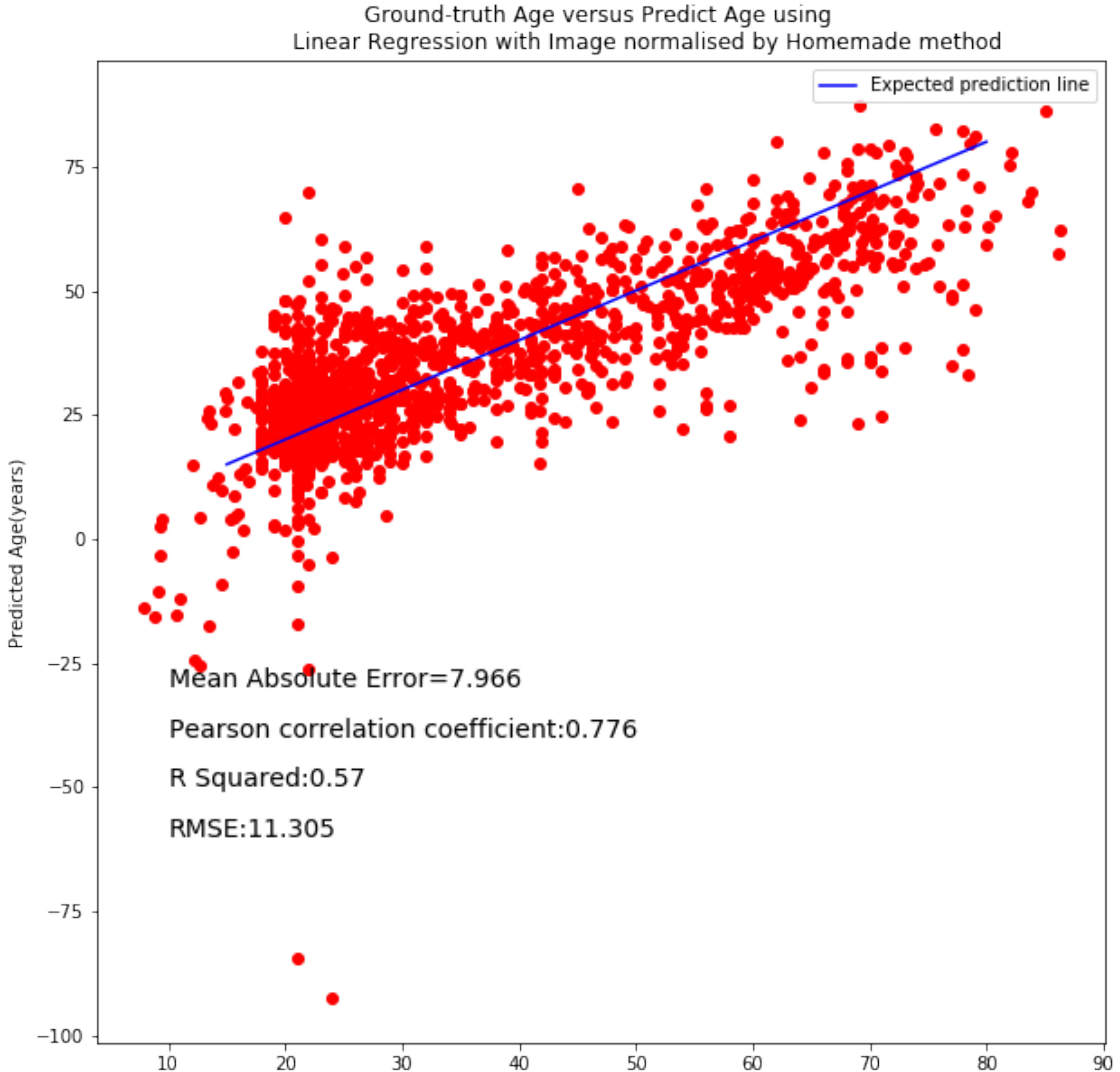
Gradient Descent:

$$w_i = w_i - \alpha\frac{1}{m}\sum_{j=0}^{m}(f(x_i) - y_j)x_i^{(j)}$$

Normal Equation:

$$\hat{w} = (X^T X)^{-1} X^T y$$

Sensitive to noise! And yield the worst result among all the method



Gaussian Regression

A Gaussian process generates data located throughout some domain such that any finite subset of the range follows a multivariate Gaussian distribution

Kernel function:

$$\kappa(x, x') = \exp\left(-\frac{(x - x')^2}{2}\right)$$

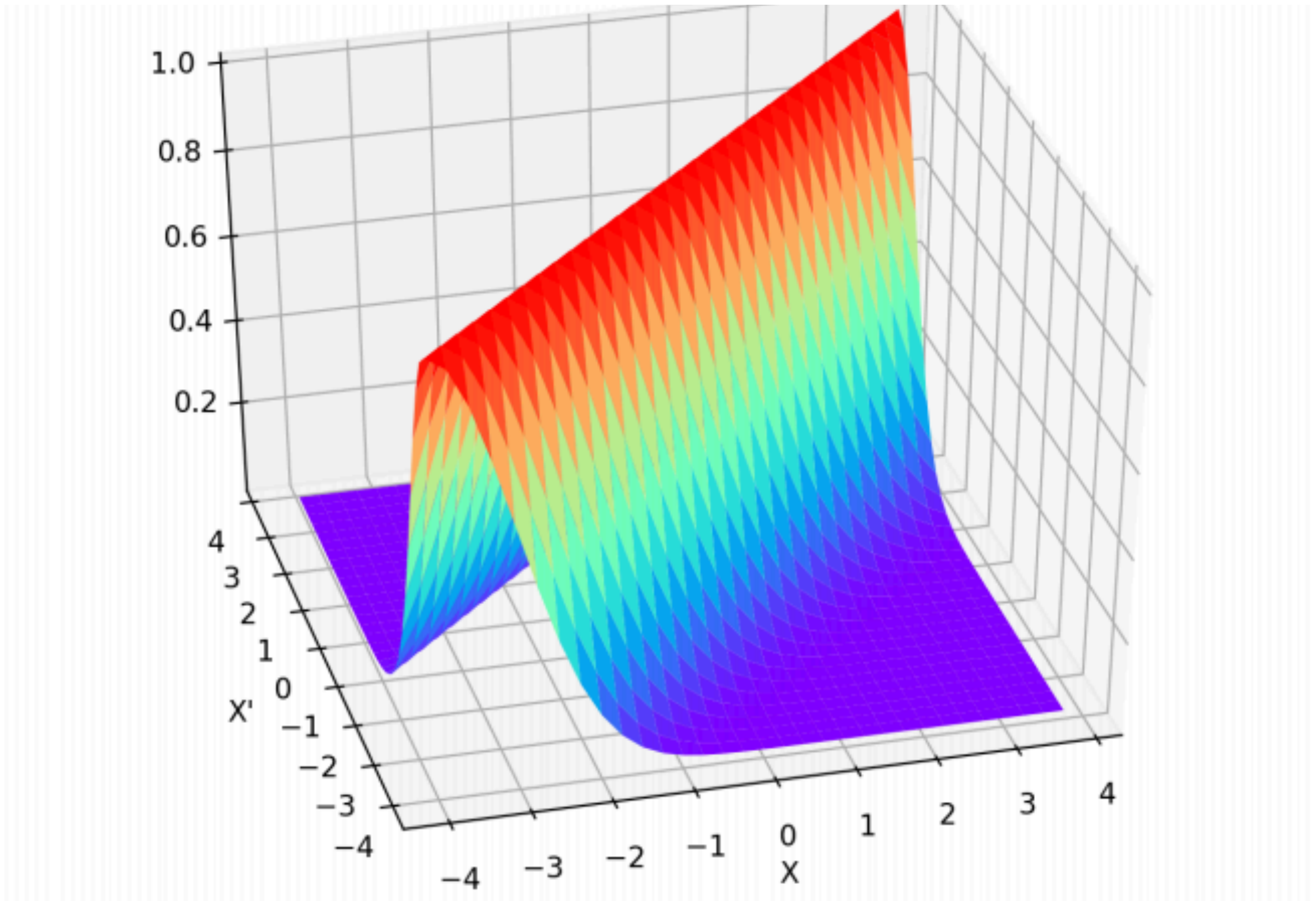
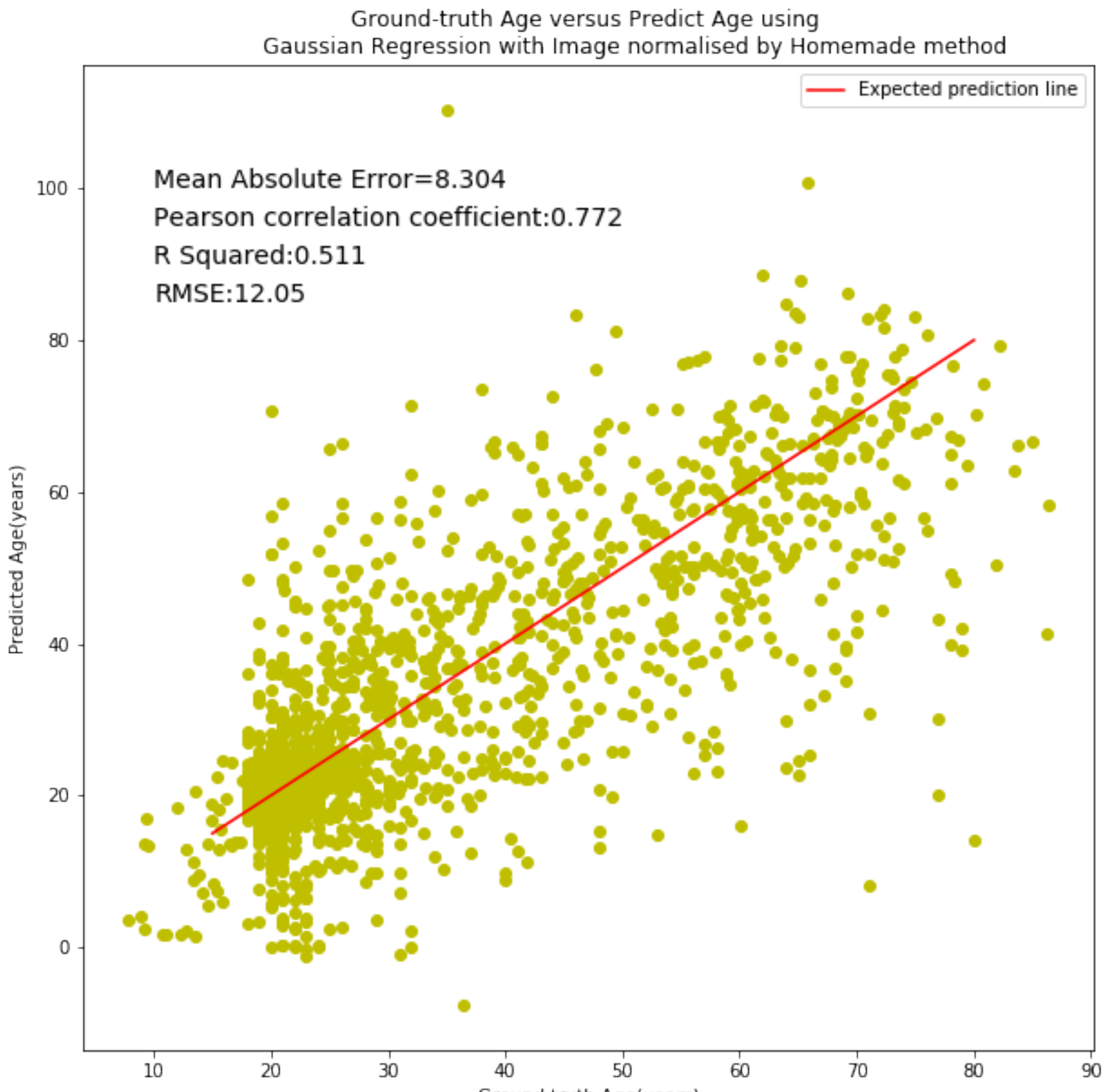


FIGURE 4.3: Image for Kernel function $\kappa(x, x')$

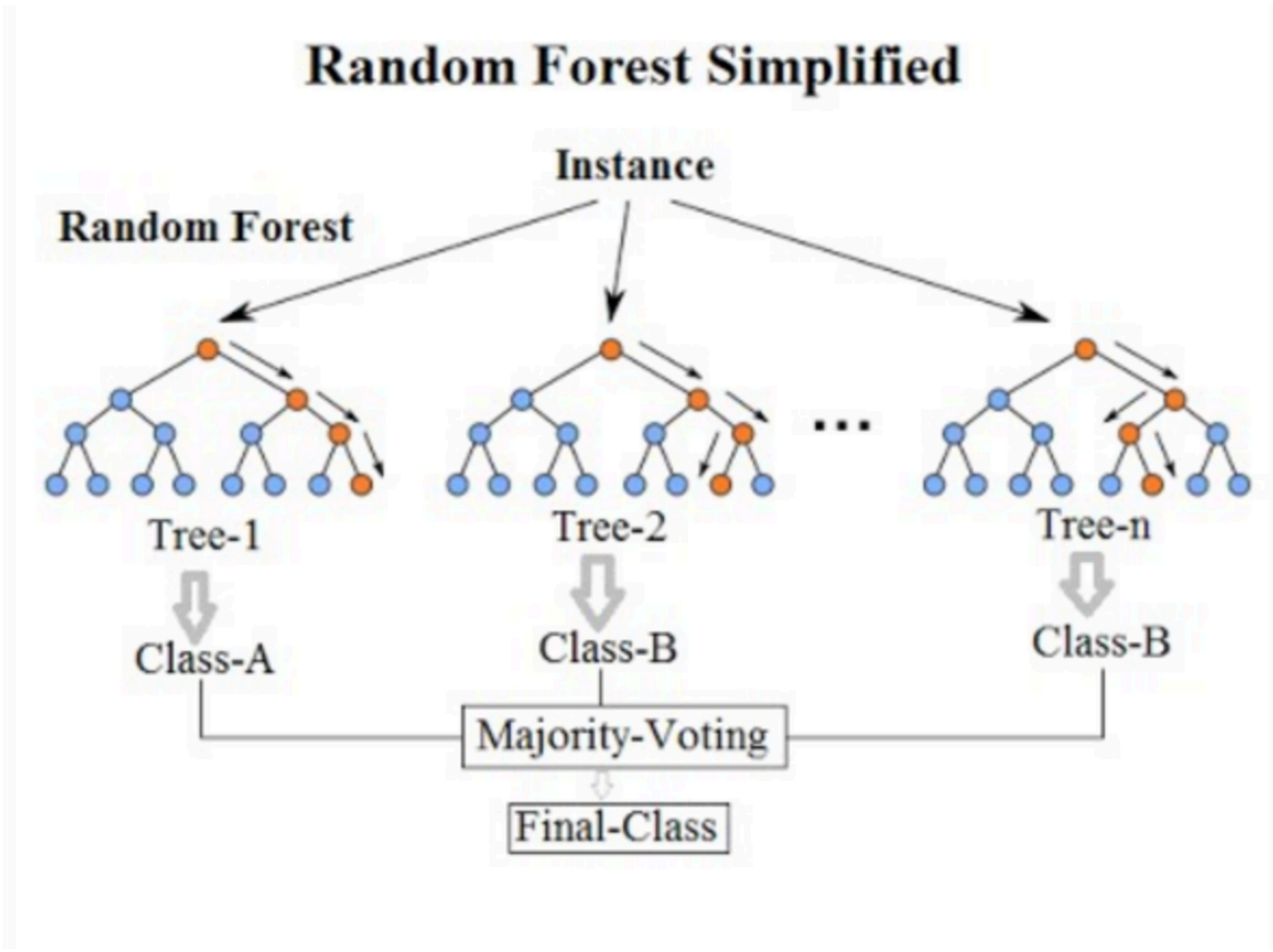
We expect that if our two varibale is close, their corresponding outputs should also have similar values either

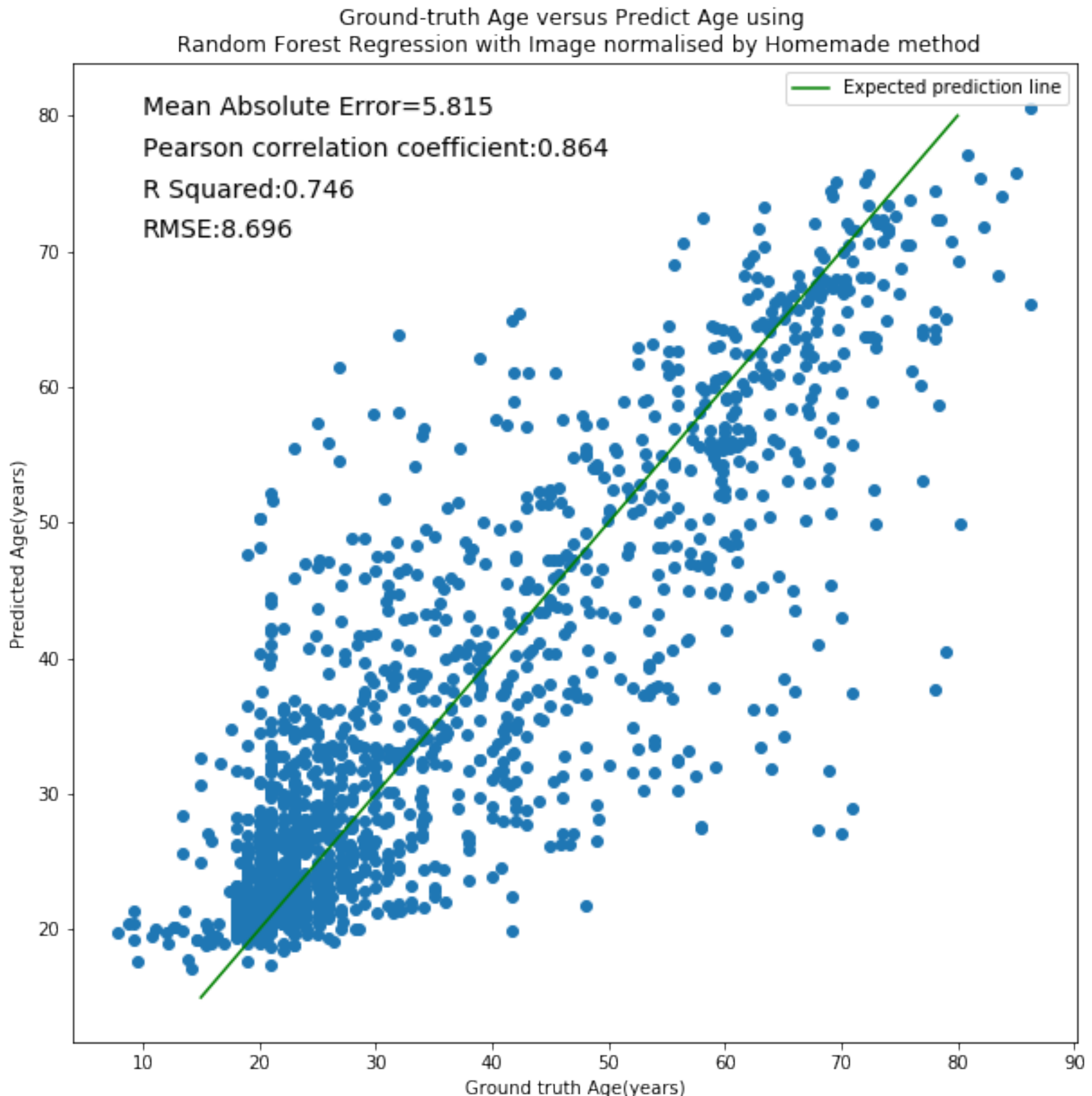
Result:



Random Forest Regression

Yield best performance with Machine learning method.





Deep Learning Method

With Machine learning method, we overlook the space information contained in the MRI!

CNN achieves many state of art performance in many tasks!

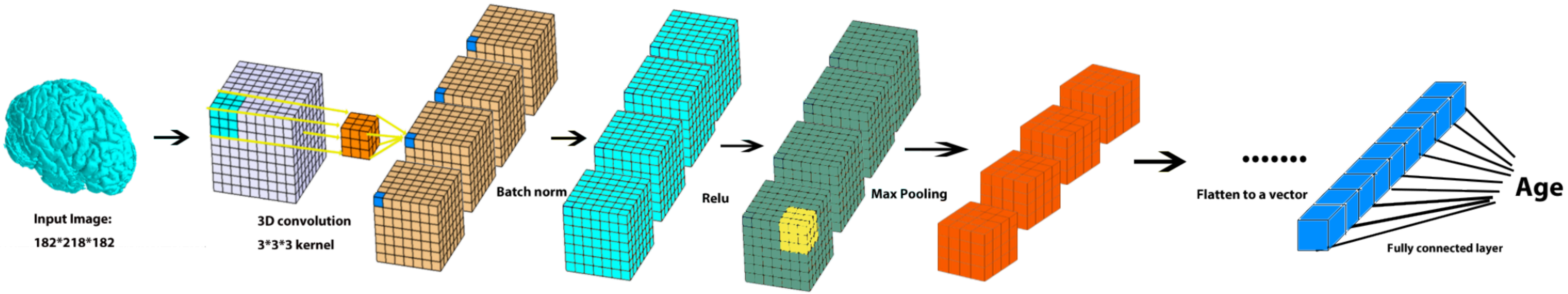


FIGURE 5.1: Overview of the 3D CNN architecture

Resnet

Our architecture will be:

Vanish gredient make it hard to train a deep neural network.

He Kaiming fix this by introducing residual mechanism:

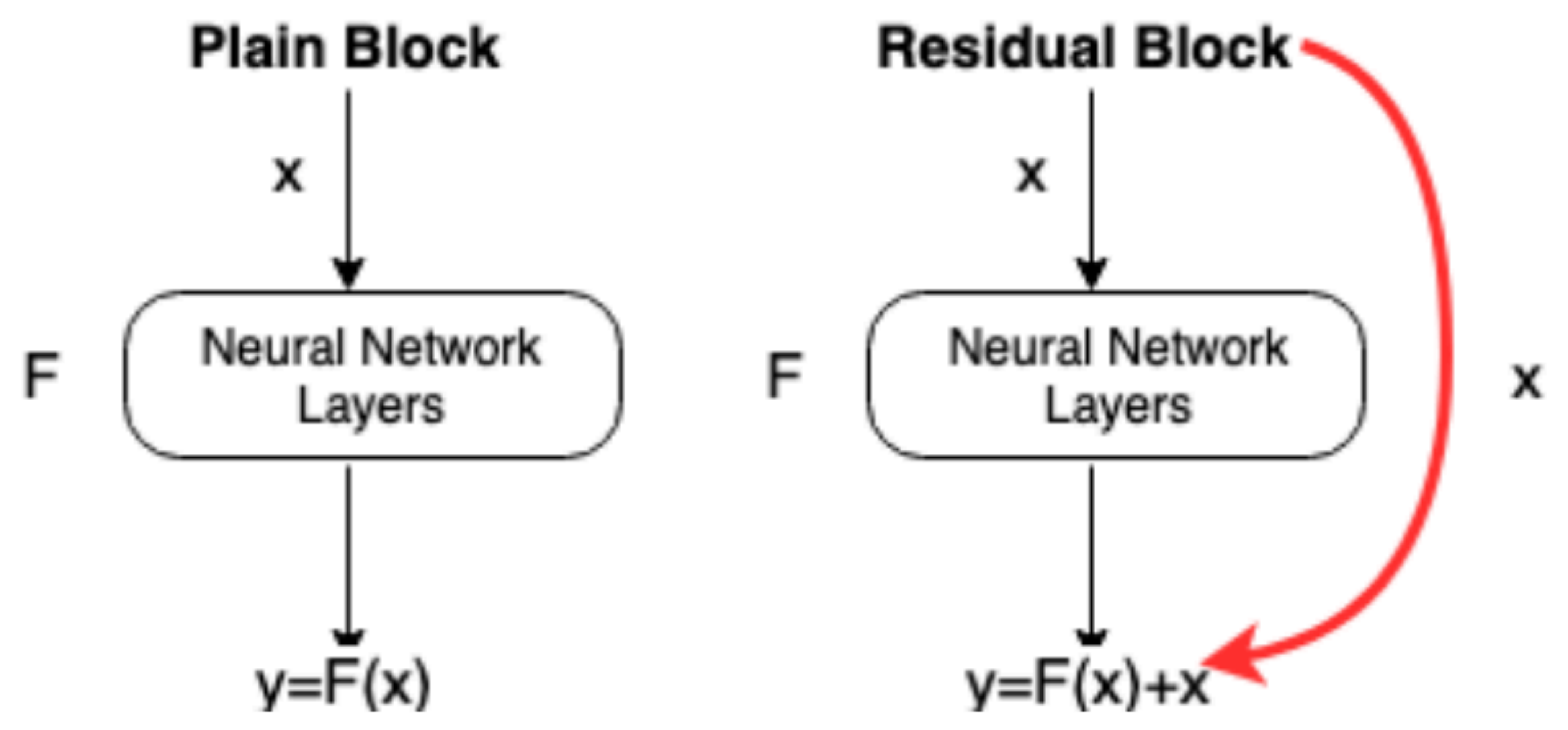


FIGURE 5.3: Residual blocks

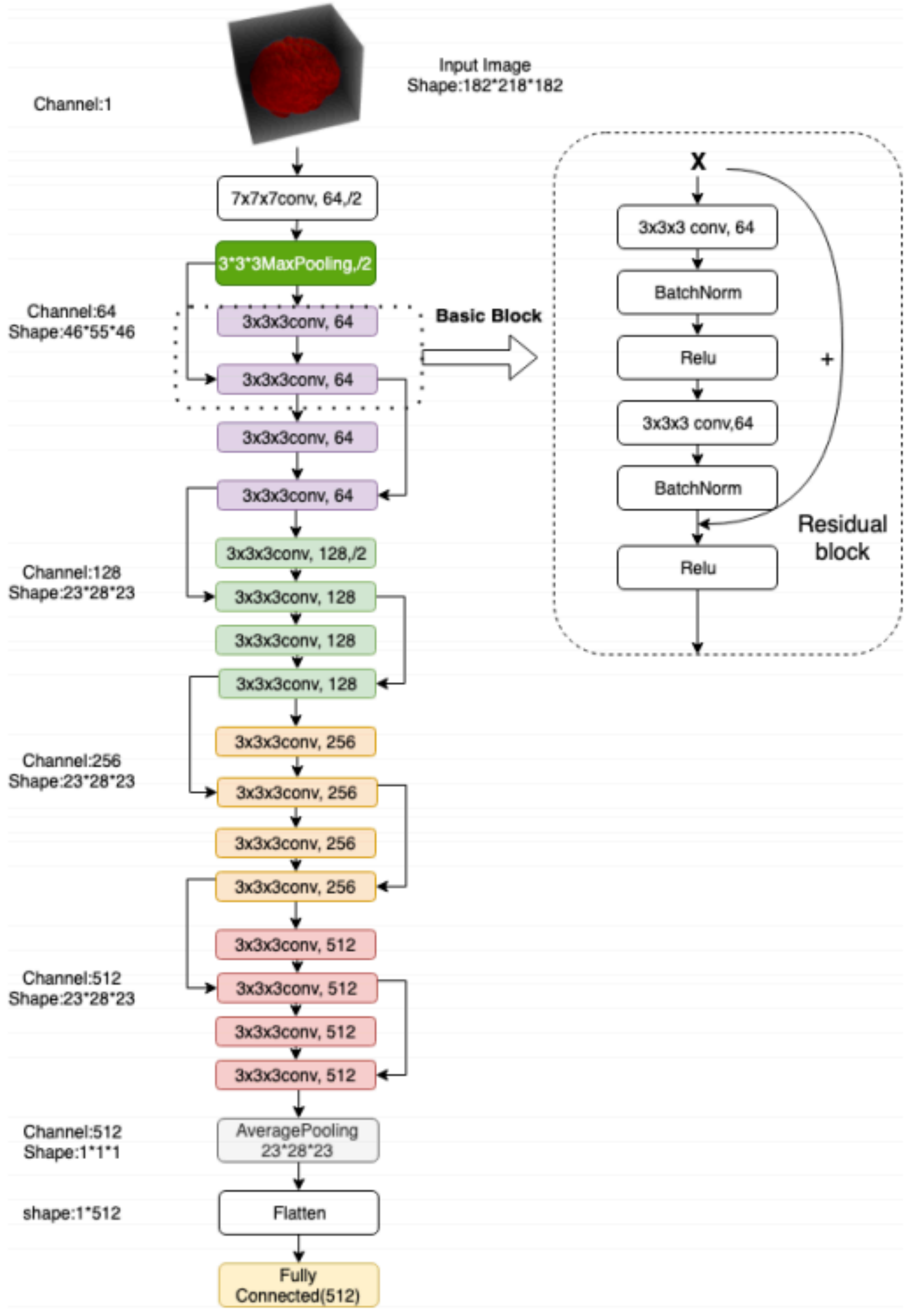
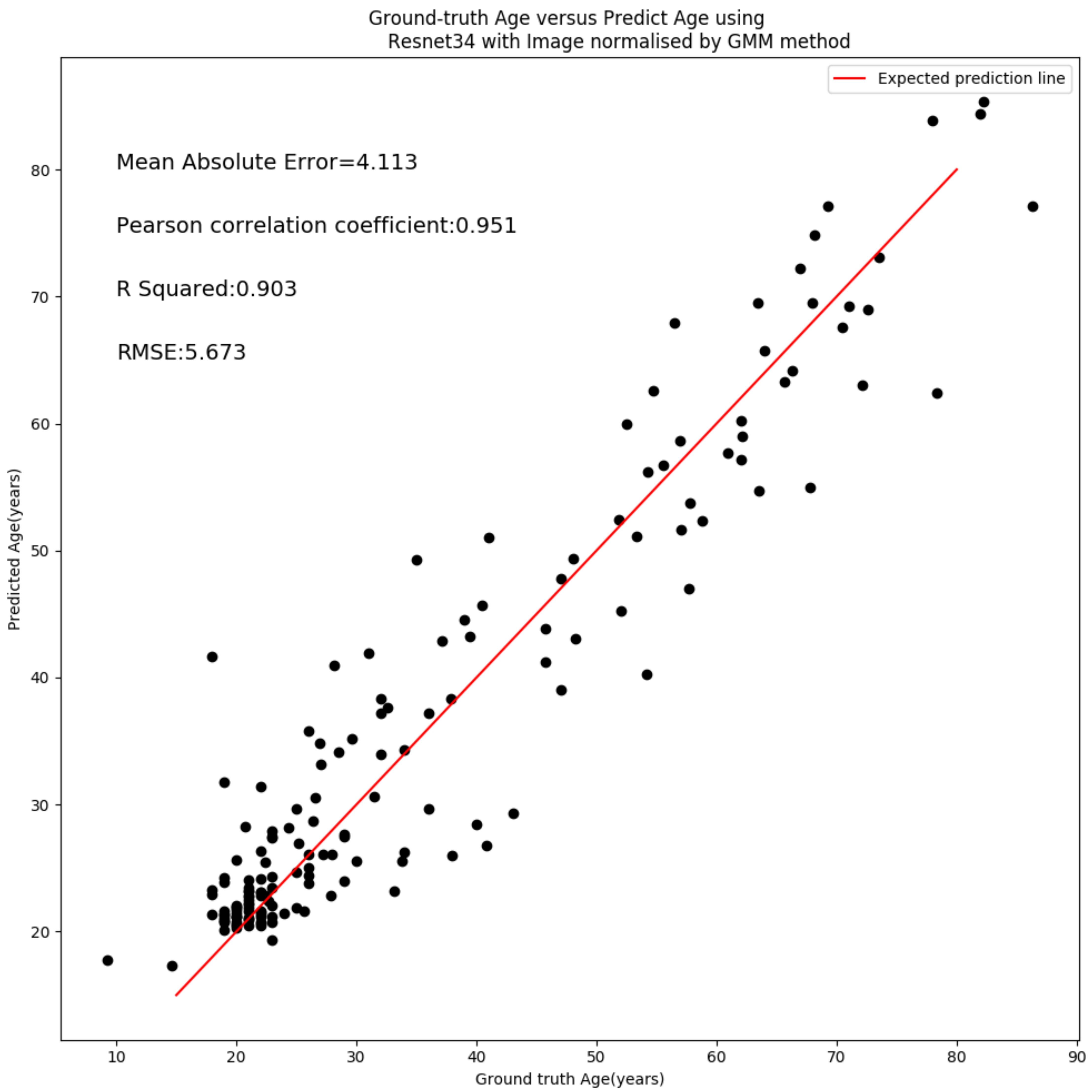
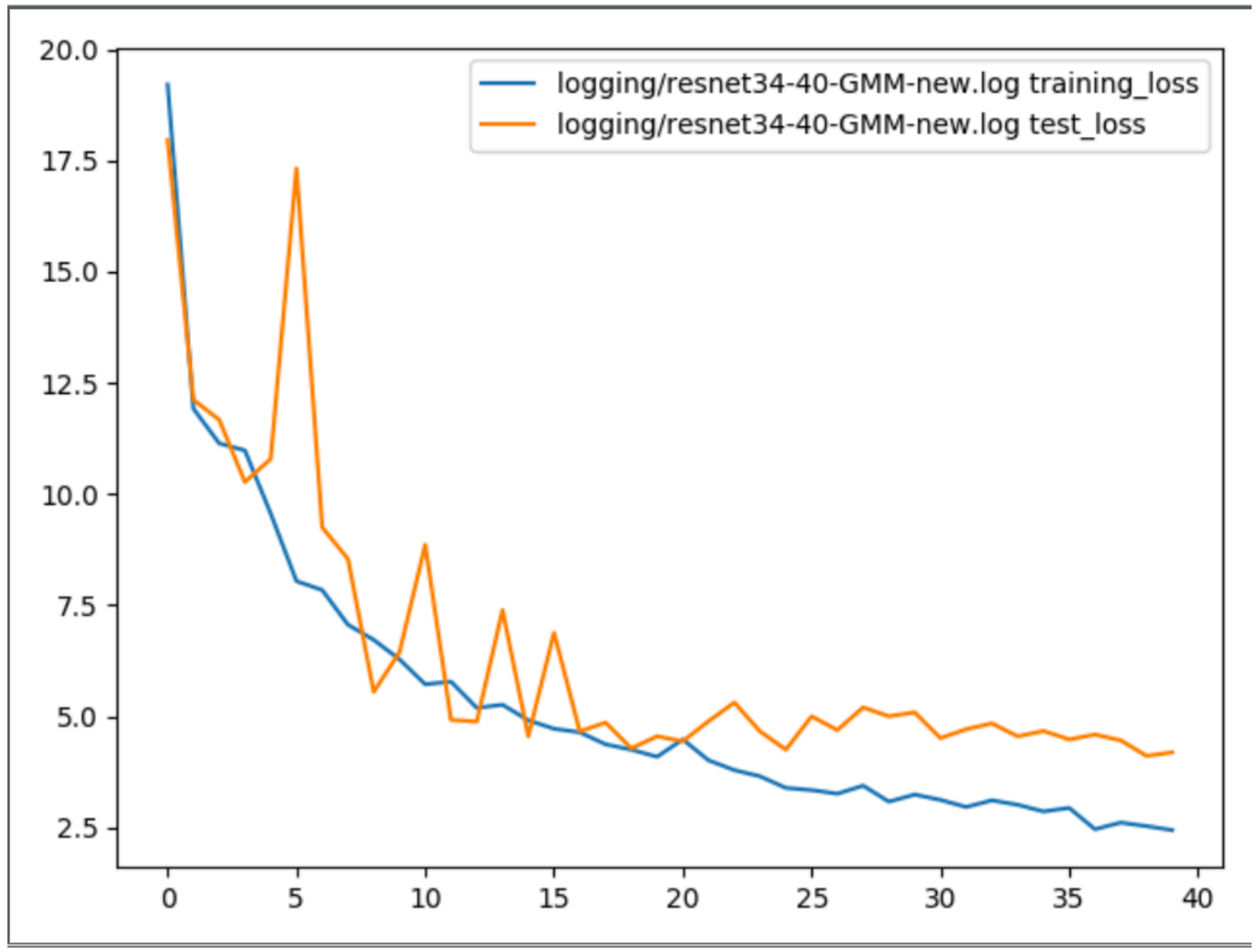


FIGURE 5.6: Architecture of 3D-Resnet18

The Best Result





Compare with the State of art performance

Table 1. Chronological age prediction accuracy

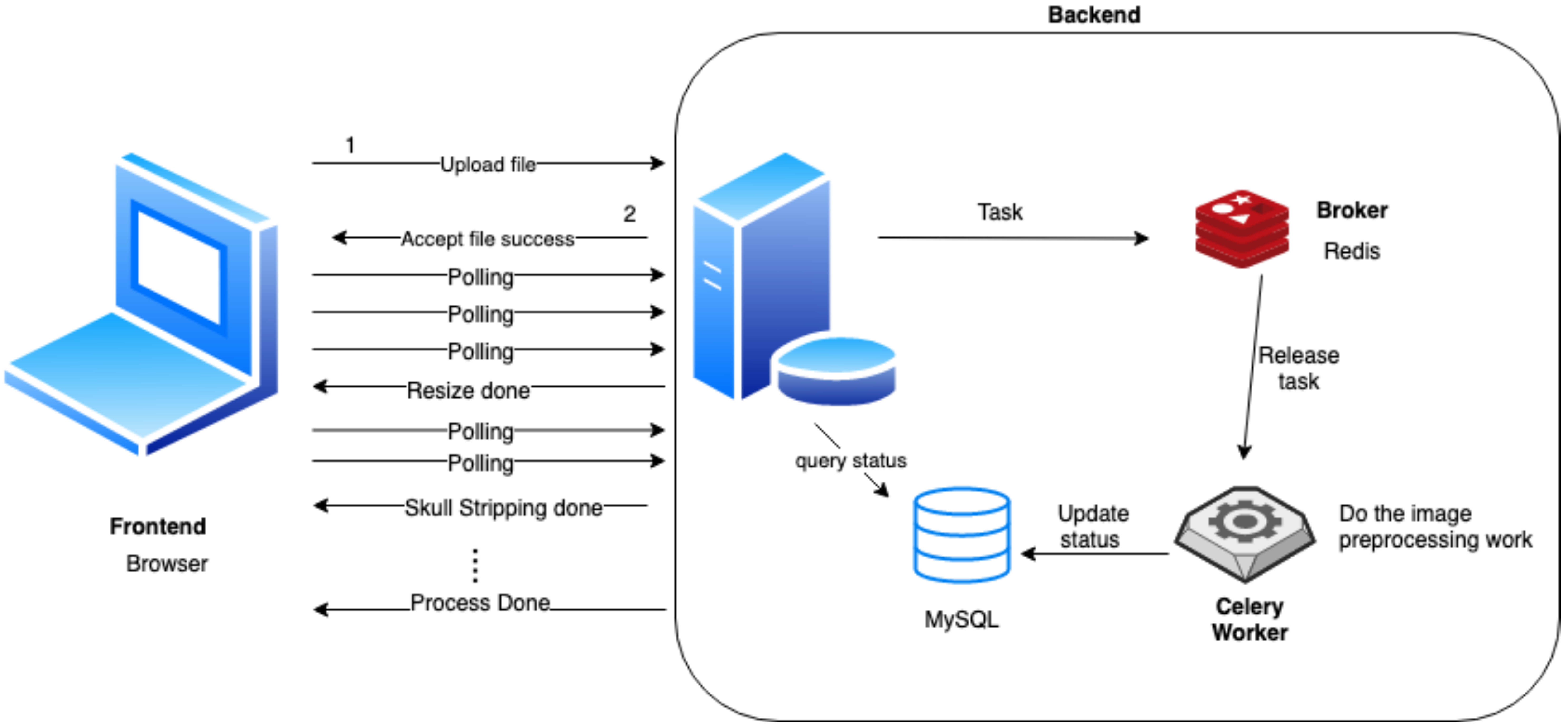
Method	Input data	MAE (years)	r	R^2	RMSE
CNN	GM	4.16	0.96	0.92	5.31
	WM	5.14	0.94	0.88	6.54
	GM+WM	4.34	0.96	0.91	5.67
	Raw	4.65	0.94	0.88	6.46
GPR	GM	4.66	0.95	0.89	6.01
	WM	5.88	0.92	0.84	7.25
	GM+WM	4.41	0.96	0.91	5.43
	Raw	11.81	0.57	0.32	15.10

Deploy the online service to web

Technology Stack

Frontend	Vue.js
Web server	Nginx
Backend Framework	Django Rest Framework
Cache	Redis
Distributed Task Queue	Celery

The deployment architecture of the service



video.mov

Resume
00:00 / 04:51

Favorites

Desktop
weiziyang
community
screenshot
mysite
life
study
Recents
Final_Project
Applications
Documents
Downloads
OneDrive - University of...

iCloud

iCloud Drive

Locations

Remote Disc
Network

Tags

红色
橙色
黄色
绿色
蓝色

IXI015-HH-1258-T1.nii.gz
IXI016-Guys...697-T1.nii.gz
IXI021-Guys...703-T1.nii.gz
IXI022-Guys...701-T1.nii.gz

1
Upload .n

7
Predict

100%

00:01
04:51

Options
Cancel
Open

Main Contribution

- Reproduce the image preprocessing pipieline
- Implement four kinds of normalisation methods and compare their performance.
- Compare the performance of different machine learning method for this task.
- Modify and Construct 3DResnet to make it adapt to our task.
- Train a model with its performance similar to the state of art
- Deploy our model on the website to make it accessible to public.

Drawback

- The training sample is still too few(1638).
- Did not try segment white matter, grey matter and CSF as first, which may make our model yield better performance.
- Fail in deploy the service since the shortage of the server’s memory