Faculty of Natural and Mathematical Sciences
Department of Informatics

Bush House, King's College London
Strand Campus,30 Aldwych
London WC2B 4BG
Telephone 20 7848 2145
Fax 020 7848 2851

7CCSMDPJ

Individual Project Submission 2018/19

Name:	Steven Vuong
Student Number:	1871066
Degree Programme:	MSc Data Science
Project Title:	Three Way Deep Learning Implementation for Parkinson's Diagnosis
Supervisor:	Dr. Hak-Keung Lam
Word Count:	10,224

RELEASE OF PROJECT

Following the submission of your project, the Department would like to make it publicly available via the library electronic resources. You will retain copyright of the project.

☑ I **agree** to the release of my project

☐ I **do not** agree to the release of my project

Signature: **Date:** April 28, 2020

Department of Informatics
King's College London
United Kingdom

7CCSMPRJ Individual Project

Three Way Deep Learning Implementation for Parkinson's Diagnosis

---

Name: **Steven Vuong**
Student Number: 1871066
Course: MSc Data Science

Supervisor: Dr. Hak-Keung Lam

This dissertation is submitted for the degree of MSc in MSc Data Science.

Acknowledgements

This project would not have been possible were it not for the support of my family and close friends; Thank you. Also, special thanks to the Informatics department, especially Guangyu Jia and my supervisor, Dr. Hak-Keung Lam for their expertise and guidance. Final show of appreciation go to the lovely folks at PPMI for helping me navigate their database and Google Colab for the usage of their GPU.

Additionally, here are some of my personal tips for those who want to try communicate their technical research/work to their non-sciencey friends/colleagues/family:

- Start with the high level digestible problem - "I'm trying build something to help to reduce the clinical misdiagnosis rate of Parkinson's Disease"
- When talking technical, try and relate the uses to other popular platforms they are familiar with and can easily acknowledge -"I'm working with computer vision, it's what Facebook uses to tag your face in pictures"
- Sandwich the technical bits in between things the other person can identify with -"Some of the best classifiers out there at the moment are above 97%! That's like an A*"

Whilst my mum still has no idea what I do with my time, I am optimistic in my efforts. Feel free to email me any of your tips at steven.vuong@kcl.ac.uk, I'd like to hear what you might suggest.

Abstract

In this study, we present 3 deep learning classification models in attempt to diagnose Parkinson's Disease (PD) using patient magnetic resonance imaging (MRI), age and gender data. Our first model is a 2D convolutional neural network (CNN) using MRI data which achieved a maximum accuracy on test set of 100% and 10-fold cross-validated score of 85.23%, so the former exceeds state of the art accuracy from a support vector machine (SVM) classifier with 97.5% accuracy from Adeli et al [1] by 2.5%. The second model is a 3D CNN in which we make an effort to validate the 100% accuracy reported by Soheil et al [2], who also used a 3D CNN. We achieve a cross-validated accuracy of 87.87% and a maximum accuracy of 93.10% on test set and therefore find it plausible. Finally, our tertiary model is an ensembled fully connected neural network (FCNN) using age and gender data, which has yet to be observed in literature and therefore is a novel contribution to achieve a cross-validated accuracy of 74.41% and a maximum accuracy of 81.67%. Thereby, it exceeds current clinical diagnostic accuracy of 73.8% by non-experts in PD [3].

Nomenclature

CCE	Categorical Cross Entropy
CNN	Convolutional Neural Network
DNN	Deep Neural Network
EEG	Electroencephalogram
FC	Fully Connected
FCNN	Fully Connected Neural Network
FN	False Negative
FP	False Positive
GA	Genetic Algorithm
HC	Healthy Control Patients (no Parkinson's Disease)
JFSS	Joint Feature Sample Selection
LDA	Linear Discriminant Analysis
LS	Least Squares
MLP	Multi-Layer Perceptron
MRI	Magnetic Resonance Imaging
NN	Neural Network
PD	Parkinson's Disease
PPD	Patients with Parkinson's Disease
PPMI	Parkinson's Progression Markers Initiative
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
ROI	Region of Interest
SN	Substantia Nigra
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
VBM	Voxel based Morphometry

Contents

List of Figures

List of Tables

1 Introduction

1.1 Parkinson's Disease

Parkinson's Disease (PD) is a progressive disorder of the central nervous system. Patients diagnosed with PD experience progressive neurodegeneration of the Substantia Nigra (SN) within the brain, resulting in increasing loss of motor control over time, including impaired balance and lack of coordination [9][10]. Approximately 6 million people worldwide are affected by PD and typically afflicts those over 60 years old [11]. Whilst there is currently no cure, early diagnosis can help patients make informed decisions to slow the progression of PD.

In a systematic review and meta-analysis, Rizzo [3] shows the current state of clinical diagnosis of PD by non-experts has a pooled accuracy of 73.8%, which improves to 79.6% when diagnosed by experts in movement disorders. This further improves to 83.9% with refined diagnosis and follow up.

The data used in this study was obtained from cohorts involved in the Parkinson's Progression Markers Initiative (PPMI) study [12], in which the baseline diagnosis was made by clinical evaluation according to the UK PD Brain Bank criteria [13]. PPMI itself is an extensive long-term study funded by the Michael J. Fox Foundation, spanning over 10 years in collaboration with multiple industry partners.

1.2 Computer Vision in MRI

Due to the low signal to noise ratio in MRI data, computer vision based diagnosis between patients with Parkinson's disease (PPD) and healthy control patients (HC) has been relatively difficult, especially with low resolution images, due to weaker magnetic field strengths. Now, however, advances in computer vision and the development of ultra high field strengths used in MRI procedures present exciting new opportunities for classification of MRI data [14]. Most notably, this includes the potential to correctly diagnose between HC and PD in prodromal stages, which is a period between initial symptoms occurring and full blown PD [15].

So far, effective and accurate diagnosis of PD by means of MRI biomarkers have gained increasing attention, with several biomarkers being important to the diagnosis of PD, such as abnormalities in the brainstem in medical contexts [16][17]. Studies examining the relevance of brain voxels in the context of classification analysis have identified critical regions involved in the pathophysiological mechanisms of PD. Thus, identifying the mid-brain, pons, corpus callosum and thalamus which are highly consistent in patients with progressive supranuclear palsy (PSP) [18].

This is important as PD has been only known to exist with certainty in PPD following post-mortem examinations upon discovery of lewy body accumulation within the brainstem, a proteic hallmark of PD, as well as consistent damage of the SN in a repeated pattern [19]. Consequently providing an important framework to investigate neurological disorders that affect a network of distributed regions thereby offering avenues for the application of computer vision-based diagnosis in clinical practice of PD; which may offer faster and more efficient diagnosis of PD in clinical practices. So as a result of better PD diagnostic accuracy, we may spare the financial and present expense that occurs because of misclassification, thus reducing the annual treatment expenditure of PPD, which in the EU only is estimated to be around €14b [20][21].

1.3 Aims & Objectives

In this paper we aim to present the effectiveness of using deep learning models trained using PPMI patient data and Magnetic Resonance Imaging (MRI) scans in correctly classifying patients into two groups: Healthy Control Patients (HC) and Patients with PD (PPD). Ergo, we aim to establish an end-to-end framework that improves upon benchmarks set by clinical diagnosis and traditional classifiers to compete with state of the art classifiers, thereby ultimately aiming to improve upon the clinical diagnostic accuracy in PPD.

The above goal is broken down into a number of steps, beginning with cleaning and preprocessing of raw data, followed by feature extraction and input into deep learning models constructed with Tensorflow 2.0 and Keras as a wrapper.

Lastly, we train our models and predict the classes of test sets, allowing us to evaluate the strength of our model graphically and numerically using objective metrics (displayed in equations (1.1) to (1.4)) of: accuracy; recall(sensitivity); precision(specificity) and F_1-Score. To calculate these, we determine the number of true positives, false negatives, false positives and true negatives as TP, FN, FP and TN respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{1.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{1.3}$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{1.4}$$

Using these metrics as evaluation, we will present three different deep learning models in this study. The results show a significant improvement compared to the results from the clinical diagnosis of PD, and are comparable to state of the art classification models.

This paper is structured as follows: chapter 2 discusses background of deep learning, leading up to chapter 3, which is a literature review into related works and current state of the art classifiers in diagnosing PD. Chapter 4 introduces the subsets of PPMI data used, the deep learning models applied and training processes. Chapter 5 presents and discusses the results. Finally, chapter 6 concludes our findings and suggests how might we improve our model further, leaving the key takeaways as the end note.

2 Background

A classification task typically involves splitting data into a training and test set. Each instance in the training set has a 'target value' or class label and one or multiple attributes/features. The idea is to produce a model using training data which can predict the target values of test data using only the test data's features [22].

2.1 Support Vector Machine

Given a training set of instance-label pairs $(x_i, y_i), i = 1, ., l$ where $x_i \in \Re$ and $y \in \{1, -1\}$, the SVM acquires the solution to the following optimisation problem [22]:

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \tag{2.1}$$

$$y_i\left(\mathbf{w}^T\phi\left(\mathbf{x}_i\right) + b\right) \geq 1 - \xi_i \tag{2.2}$$

$$\xi_i \geq 0 \tag{2.3}$$

where in (2.1) to (2.3), training vectors x_i are projected onto higher dimensional spaces by the function ϕ with b and C as constants. SVM then finds a linearly separating hyperplane with the maximal margin (greatest average/summed distance between points of each class) in this higher dimensional feature space. $\xi \geq 0$ is the penalty parameter for the error term.

Furthermore, $K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$ is called the kernel function. Though there are many kernels, the most common and basic are linear and radial basis function (RBF) kernels:

$$Linear : K(x_i, x_j) = x_i x_j \tag{2.4}$$

$$RBF : K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\gamma\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right), \gamma > 0 \tag{2.5}$$

there are also more advanced kernels, such as the Pearson VII Universal Kernel (PUK) [23].

2.2 Artifical Neural Networks

Artificial neural networks consist of simple neuruons that are interconnected in parallel, often with many layers to function as a collective system. One type of network 'learns' by adaptively updating their parameters during the training process [24].

An individual neuron, also know as a perceptron is the basic unit of a neural network. This effectively takes the weighted sum of its inputs into a threshold function which moderates the output. A simple linear threshold function is given by:

$$y = \begin{cases} 1 & \text{if } \mathbf{w}^{\mathrm{T}}\mathbf{x} = \sum_{i=1}^{n} w_i x_i \geq T \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

where, $x \in \{x_1, .., x_n\}^T$ and $w \in \{w_1, .., w_n\}^T$ [4]. A diagram of this with a simple threshold function is shown in Figure 1.

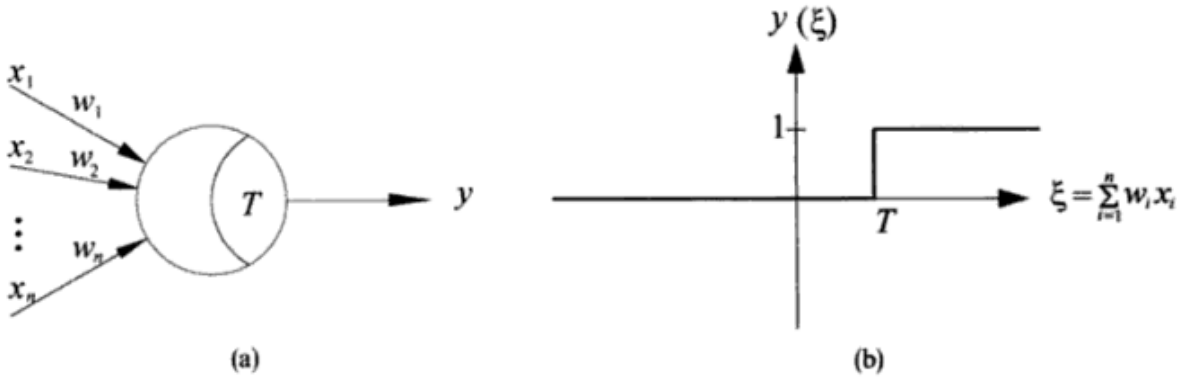

Figure 1: (a) Simple representation of a neuron and (b) it's transfer function [4]

By stacking perceptron units, such as in figure 2, we are able to form networks by passing the outputs of several perceptron units as weighted inputs to perceptrons unit in the subsequent layer. There are also various transfer functions which may be used, in this study we use the following functions:

$$Sigmoid : \phi(z) = \frac{1}{1 + e^{-z}} \tag{2.7}$$

$$ReLU : R(z) = \max(0, z) \tag{2.8}$$

$$Softmax : \sigma(\mathbf{z})_i = \frac{e^{-\beta z_i}}{\sum_{j=1}^{K} e^{-\beta z_j}} \tag{2.9}$$

The purpose of using (2.7) and (2.9) is that they are differentiable, meaning we are able to determine the slope of the curve at any two points. The softmax function is a more generalised logistic regression used for multi-class classification, and is also powerful for binary classification as it returns a class probability distribution.

Rectified Linear Unit (ReLU) as well as Leaky ReLU are very popular activation functions for deep learning and convolutional neural networks. $R(z)$ is 0 when z is less than 0 and is linear above 0. An issue is that a model fails to train a network as any input less than 0 will have an output of 0. Leaky ReLU resolves this by having a very slightly negative function for z below 0. In ReLU and Leaky ReLU both the function and their derivatives are monotonic [25].

Once we have the output from our model, we are able to calculate the error score using a loss function. In this study we use the categorical cross entropy (CCE) for binary classification as our loss function:

$$CCE = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \left(p_{ic} \log \left(y_{ic} \right) \right) \tag{2.10}$$

where p_{ic} is a binary indicator function that detects whether the i^{th} training pattern belongs to c^{th} category,

and output y_{ic} is the predicted probability distribution for i^{th} observation belonging to class c. This is used as it can improve the robustness of a model in comparison to categorical error, the current most popular loss function in classification tasks [26].

With our loss function, we are then able to perform backpropagation in order to recursively update our weights. The basic idea behind backpropagation is repeated application of the chain rule:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial net_i}\frac{\partial net_i}{\partial w_{ij}} \tag{2.11}$$

where w_{ij} is the weight from neuron j to neuron i, s_i is the output and net_i is the weighted sum of inputs of neuron i. Following this we can update our weight by performing simple gradient descent:

$$w_{ij}(t+1) = w_{ij}(t) - \epsilon\frac{\partial E}{\partial w_{ij}}(t) \tag{2.12}$$

with ϵ as our learning rate. If this is too small, our model may take a long time to converge, whereas if the learning rate is too large, we may have oscillation. One way to reduce convergence time is to introduce momentum, where parameter μ scales the influence of the previous step on the current [27].

$$\Delta w_{ij}(t) = -\epsilon\frac{\partial E}{\partial w_{ij}}(t) + \mu\Delta w_{ij}(t-1) \tag{2.13}$$

2.3 Convolutional Neural Networks

A CNN is a powerful model widely used in computer vision. Once trained, CNNs are a type of feed-forward network, meaning information moves forward only. There can be many different types of layers in a CNN, and the most important is the convolution layer, made of several filters which are updated during training. A given input is convolved with several filters which slide over it to produce an activation map. The activation map is then fed to the proceeding layer.

Typically, after the convolution layer comes the pooling layer, which down samples the activation map. The most common type of pooling layer is a max pooling layer, which selects the greatest value in a filter. The idea behind pooling is to reduce the image's spatial size and thus the number of parameters needed to be calculated, therefore, reducing the chance of overfitting [28].

Other ways of reducing overfitting, used by our model in this study, include introducing dropout layers [29]. The key idea behind dropouts is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much as the combination of units may be different with each pass as is shown in figure 2.

Regularisation can also reduce overfitting. In this study, we have proceeded with L_2 kernels, which Cortes et al. [30] have shown to have significant accuracy improvements with kernels and does not degrade in performance in large-scale cases, as L_1 regularisation might. L_2 / Ridge Regularisation essentially adds the sum of the weight's square to the loss function.

We also augment our dataset to increase the robustness of our model [31]. This increases our training set size by performing transformations on our original image set to produce a synthetic image set that is also possible in realistic scenarios. In this study, we mirror image the MRI scans. We also implement Gaussian masking to reduce the amount of noise in each MRI scan in our 3D dataset. Gaussian masking is a technique

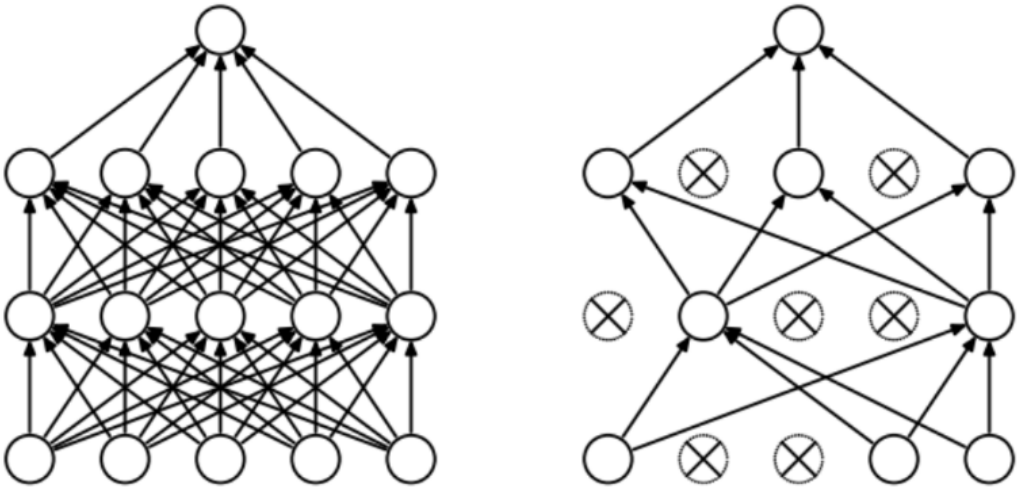

Figure 2: (a) Standard Neural Network (b) Neural Network with dropout layers [5]

which effectively blurs an image with the Gaussian function to reduce the amount of detail. This is useful for a CNN as it might remove noise that is being picked up initially [32].

Finally, for our 2D scans we normalise our images to increase the stability of our model, as well as reduce the amount of computational power required during the training process [33]. Normalisation was not done for the 3D scan, as the scale of normalisation would vary from layer to layer.

The above are all techniques implemented in our CNN to improve stability and reduce overfitting. Our CNN model is also connected to a FCNN to output a binary classification result for a given patient of having PD or not having PD.

3 Related Work

Attempts to diagnose PD using traditional classifiers has occurred for a number of years. For instance, a study in 2014 by Gil et al [34] used a multi-layer perceptron (MLP) classifier consisting of two fully connected layers, having one input; one hidden and one output layer in total to achieve an accuracy of 92.3%, sensitivity of 93.4% and specificity of 87.5% using a sigmoid activation function and a simple threshold function. This classifier was applied to speech data from the UCI Machine Learning Repository [35]. Additionally, another classifier mentioned within the same paper, a SVM with a PUK was able to achieve an accuracy of 93.3%, sensitivity of 94.5% and specificity of 89.6% . Although there were only 31 patients in the data sample, [34] demonstrates moderately good accuracy and concludes by proposing a hybrid model of MLP and SVM to produce better generalisation in test data. Whilst there has been no hybrid model encountered thus far in diagnosing PD with MRI data; there have been numerous cases of SVM classifiers applied to data derived from MRI. The accuracy of NN in other medical applications has also been compared to that of SVM classifiers such as predicting memory encoding using electroencephalogram (EEG) data, where Arora et al [36] found Recurrent Neural Networks to outperform SVM in 75% of subjects thus showing the applicability of neural networks (NN) in medical establishment.

Multiple classifiers have been used for diagnosis of PPMI MRI data. Table 1 shows that SVM classifiers have been used by Adeli et al, Huppertz et al, Pahuja and Singh et al in previous studies [1][37][38][8] to achieve test-set accuracy scores of 86.9%, 91.0%, 95.0% and 97.5% respectively. Furthermore, these studies vary in methods of feature extraction, validation methods and kernel function. For instance, [37] uses a RBF SVM classifier on age normalised, volumetric inputs of brain regions and feature extraction step via statistical parametric mapping and Atlas based Volumetry thus achieving an accuracy of 84.1%. The scoring function used in [37] was leave-one-out cross-validation, which has the advantages of a larger training set, simplicity and more realistically represents clinical diagnosis in practice (case-by-case basis). On the other hand, leave-one-out naturally has very high variance. This makes it an inferior cross-validation function when considering the possibility of occurrrencences of multiple MRI scans belonging to the same patient, taken at different time frames as is the case within the PPMI dataset. Therefore, models generally have high probability of overfitting, a situation whereby a model is too specific to a training set and is unable to generalise to unseen data [39].

Works in [1][8][40] instead validate with 10-fold cross-validation, which whilst more computationally expensive, has lower variance and provides a better model performance estimation for a larger sized dataset, such as the one used in this study by Wong and Kohavi et al [41][42].

Pahuja et al [38] performs voxel based morphometry for feature extraction and genetic algorithm for feature selection to achieve an accuracy of 91.0%, an improvement on [37] by almost 5%. This however, falls short of the study conducted by [8], which achieves an accuracy of 95.0% using a least-squares (LS) fitting SVM. The best result for a SVM classifier meanwhile, was achieved by Ehsan, Adeli et al [40] whose input parameters were brain densities of regions of interest (ROI), with a combination of linear and non-linear kernels.

Moving away from SVM classifiers, [40] uses Joint Feature Sample Selection (JFSS) for feature extraction and a robust linear discriminant analysis (LDA) classifier for MRI diagnosis and achieves an accuracy of 81.9% [43]; much weaker than all the SVM classifiers, hence we will not consider the LDA [40] studied as a baseline for comparison. Instead, we will use the 97.5% accuracy achieved by [1] as a benchmark for our classifiers.

Soheil et al effectively erases the need for a benchmark by achieving an accuracy of 100% on training and test data. Whilst this prompts questions of overfitting [39], the model is a 3D CNN which completely removes the need for the feature extraction process as the model learns the important features through multiple convolution and pooling layers [2]. In addition, Soheil et al pre-processes the MRI scans with steps including augmentation and skull-stripping, as well as strengthening the CNN model by implementing features such as regularisation and dropout layers.

Table 1 below provides a summary of the mentioned studies and their accuracy scores.

Study	Feature Extraction/Selection	Classifier	Accuracy on Test Set
[40], 2016	JFSS	Robust LDA	81.9%
[37], 2016	Atlas Based Volumetry	SVM with RBF	86.9%
[38], 2016	VBM and GA Selection	SVM	91.0%
[8], 2015	Kohonen Self-Organising Map	LS SVM	95.0% $\pm$ 0.6
[1], 2017	ROI Density From Kernel	Max-Margin SVM	97.5%
[2], 2018	None	3D CNN	100%

Table 1: Comparative overview of prior studies, classifiers & their respective accuracy applied to MRI data from the PPMI database. [8] is the only study which includes uncertainty in their accuracy measure.

Because it is not possible to achieve a score higher than 100% claimed by Soheil et al's [2] model, we will therefore endeavour to validate this accuracy. Especially as [2] did not report any cross-validation. However, CNN promises to be very powerful with high accuracy and removes the need for a feature extraction step. Additionally, [2] incorporated the age and sex of patients into his 3D model, whereas we will present another deep learning model using those parameters as explicit inputs in this study. Another model investigated in this study for the diagnosis of PD using clinical data is a 2D CNN model.

So in brief, there are three deep learning models investigated in this study in attempt to diagnose PD. The first and second are 3D and 2D CNN models respectively, both using MRI data. The final model is an ensembled FCNN using patient's age and gender. The 3D Model will attempt to validate the accuracy of [2] (claim of 100% accuracy) and perhaps streamline methodology. The 2D model will attempt to outperform state of the art standards and finally, the non-imaging based model is something not yet observed in current literature and will aim to beat the clinical diagnostic accuracy rate.

4 Methodology

4.1 Data Acquisition & Pre-Processing

We have constructed 3 different deep learning models, as explained in the previous sections. All the data derives from the PPMI database, however, each model has a slightly different subset of patient data, varying in type and is transformed after preprocessing. Therefore, for forward reference, we may refer to our ensembled FCNN as the Age/Sex model, the 2D CNN as the 2D model and the 3D CNN as the 3D model.

For the Age/Sex model, we had 596 patients, split randomly into 536 patients for training our model and 60 ($\tilde{1}0\%$) test patients. Of the 596 patients, 404 were PPD and 192 were HC, also 391 were males and 205 were females. A more broken down distribution of the training and test sets and their details regarding the number of PPD, HC, males and females can be seen in table 2.

Figures 5 and 6 show the patient's age distribution. From these figures, we can see that there are clearly more PPD in comparison to HC, and the age distribution is fairly similar between these two groups. Also, we observe a few outliers in the box plot for the control group. We attribute this to the fact that the age of patients with PD is naturally higher in reality and so is reflected in the study data. The HC group and PPD group also have similar upper and lower quartiles, both around 85 and 35 respectively. Although we could have selected an equal number of HC and PPD to improve our model, the trade off is having less data samples and so the benefit would be marginal, if any at all.

Before input into the Age/Sex model, the age and gender was one-hot encoded with 100 variables and 2 variables respectively (as the maximum age of patients did not surpass 100 and there are only 2 genders listed in PPMI - M and F). One hot encoding is a process where categorical variables are converted to a form interpretable by machine learning algorithms, typically binary strings e.g.'Male' becomes 10 and 'Female' becomes 01. We used this encoding method for our study because it is reliable. Although the encoding method is inefficient [44], we have adequate computing power to deal with this process so proceed with one hot encoding. This method was also used to encode all the class values for our patients (i.e. encoding HC or PPD for each patient).

Less patient samples were used for the 2D and 3D CNN models compared to the age/sex model because of the greater specification required for each MRI scan and thus less data available. The MRI scans were T1 weighted, 3T, 1mm thickness, flip angle of 9°, on the Sagittal imaging plane, of resolution (176, 256, 240) pixels and taken by a SIEMEN MRI Machine [12]. The images were also of description 'MPRAGE GRAPPA' [45].

The initial pre-processing for 2D and 3D MRI scans were the same. Firstly, the images were skull-stripped to remove non brain tissue as we want to focus on the Substantia Nigra (SN) and other key ROIs within the brain tissue. This was done with PyPi deep-brain [46], where the mask probability assesses the likelihood of the pixel belonging to brain tissue instead of skull tissue was set to $\geq$ 1e-3. Additionally, we transformed the scans to the axial view so it becomes (256, 240, 176) in pixel dimensions. The quantitative effect of this is unclear though it is suspected that changes in the SN would be easier to detect in the axial plane comparative to the sagittal plane from works in literature [47]. Finally, the scans were reduced in dimensions to (160, 160, 160) pixel dimensions as there was a lot of blank background around the brain image so was simply an act of cropping out empty background. The steps described above are shown with an example MRI brain scan in figure 3.

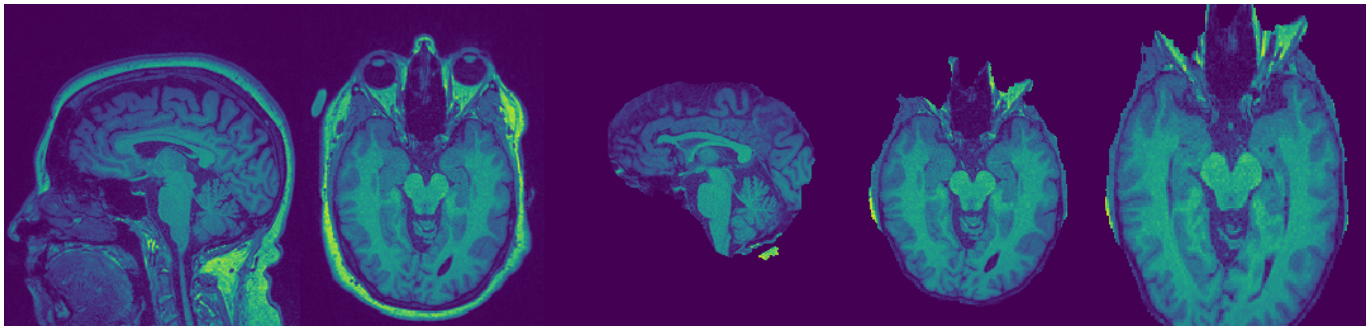

Figure 3: MRI Brain Imaging pre-processing. Left to right: pre-skull stripped brain in sagittal plane, pre-skull stripped brain in axial plane, post-skull stripped brain in sagittal plane, post-skull stripped brain in axial plane and post cropped brain in axial plane. We can see the large amount of blank space around the brain post-skull stripped and so justifies cropping the brain slices to increase the brain:background ratio.

Following this, the brain scans were augmented, effectively doubling the size of the imaging data. The augmentation process was simply a mirror-flipped image on the axial plane. By doing this, we have more (realistic) images to train our classifier on, thereby making it more robust and able to diagnose non-training data with better accuracy [29].

With the augmented data, we create a series of 2D images by extracting a single slice from the 3D scan. It was decided to select the 86th slice on the axial plane because it shows a clear portion of the SN in the centre of the MRI scan. An example of this slice is shown on the far right image in figure 3. This selection process was manual, though in future a segmentation model for each MRI scan that selects the axial slice with the clearest view of the SN or same relative brain-region with important ROI of each patient would be ideal.

220 imaging scans were used in training and testing the 3D CNN. 97 of which were HC and 123 PD, with 80 females and 140 males. This was split into 191 patients for the training set and 29 patients for test (13%), a larger proportion was selected for the test set compared to the age/sex data subset as the age/sex data subset has more patients so selecting a higher proportion of patients in the 3D CNN subset allows for better predictive estimation. A more granular inspection of the distribution of PPD, HC, males and females for the training and test set can be found in table 2.

The histogram of the ages of patients in the 3D CNN in figure 7 show there are almost as many HC as PD, though there is a spike in PD patients between the ages of 69 and 74. This distribution is reflected in figure 8 which also shows similar distributions of age for HC and PD in the 3D CNN data.

Of the 220 patients, only 191 were taken for the 2D CNN because some of the scans had non-clear images on the 86th slice. Although we could have left them in, non-clear images greatly increases the possibility that our model learns incorrect features, as a classification model can only go so far in terms of accuracy and classification ability with poor data [48][49]. In the 2D CNN dataset, 150 patients were used for training and 41 for testing (22%). A much larger portion was used for testing for the same reason as above and to validate more strongly the very high accuracy achieved in the Results section. 68/191 patients were females and 123/191 were males. 81/191 were HC and 118/191 were PPD. The distribution of which can be seen in figures 9 and 10. From these figures, it is observable that there are less HC than the 3D model, though the spread from figure 10 approximately reflects those observed in figures 8 and 6.

The final preprocessing step for the 2D model data involved normalising the images, this has the effect of

increasing training speed of our model without distorting the difference between pixels. For the 3D model, we did not normalise because the degree of normalisation vary on each slice in the axial plane as each slice is normalised relative to its maximum value. Instead, Gaussian masking/blurring was used to reduce the computational effort required and lessen the effect of noise in the image by reducing the image resolution. In addition to this, we can train our model with larger batch sizes thanks to the reduced amount of computation required to handle each image [32]. Figure 4 shows the effect of gaussian masking on brain MRI images. The final step involved reducing the pixel dimensions of each 3D MRI scan from (160, 160, 160) to (70, 160, 160) pixel dimensions after only considering imaging slices focusing on the mid to lower brain region where changes in the brain are suspected to occur for PPD [18][19].

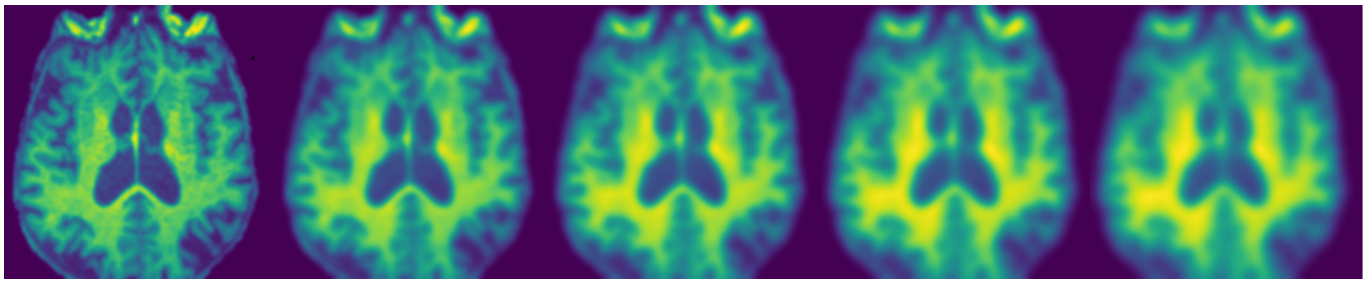

Figure 4: Gaussian Masking on MRI brain images. Left to right: 1σ, 2σ, 2.5σ, 3σ and 3.5σ.

	Parameter	Age/Sex FCNN	2D CNN	3D CNN
Training Set:	PPD	363	87	107
	Control	173	63	84
	Males	347	101	121
	Females	189	49	70
Test Set:	PPD	41	23	17
	Control	19	18	12
	Males	44	22	19
	Females	16	19	10
Total:	PPD	404	118	123
	Control	192	81	97
	Males	391	123	140
	Females	205	68	80
Training Set Size:		536	150	191
Test Set Size:		60	41	29
Total N.o Patients:		596	191	220

Table 2: Summary of Dataset Distributions of each Model, more pointedly, the distribution of PPD, HC, Males and Females in training and test sets

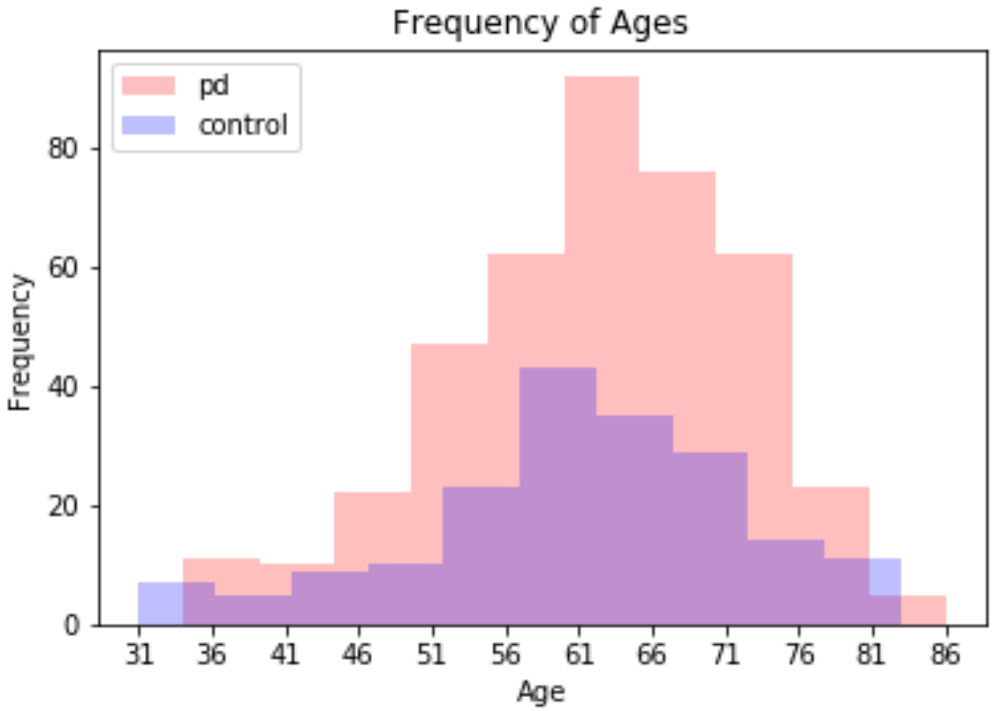

Figure 5: Histogram of the ages of patients in Age/Sex model, of PPD and HC. Here we see approximately equal distribution with many more PD than Control patients.

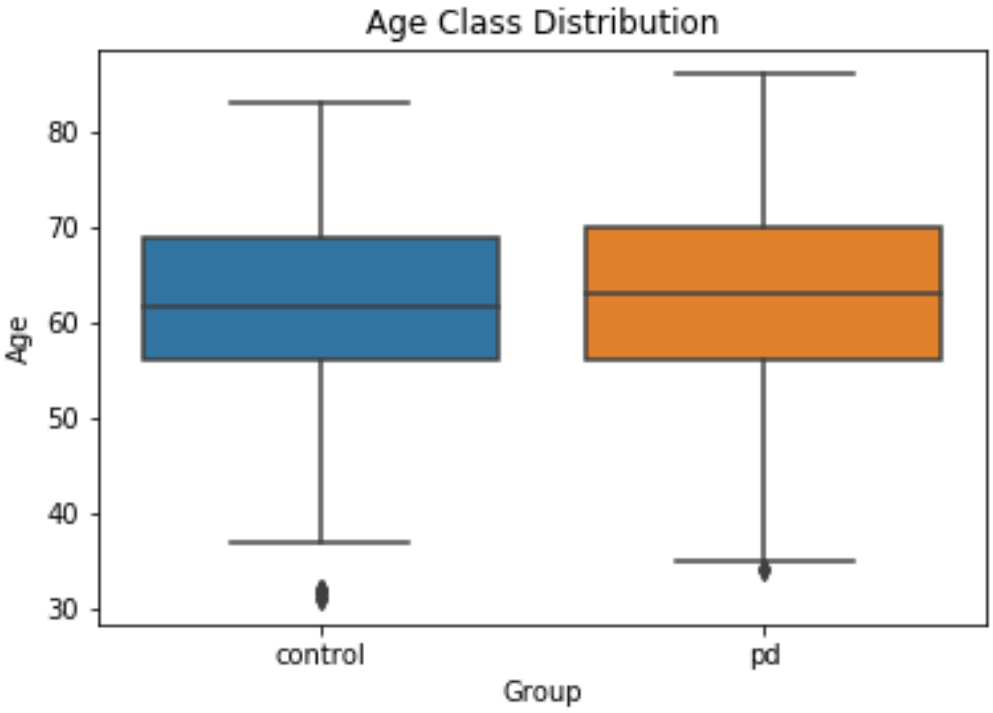

Figure 6: Boxplot of the Age Distributions in the Age/Sex model according to class of the patient having PD or not (HC/control). The box plots appear very similar in spread, with the control having a few outliers around the age of 30.

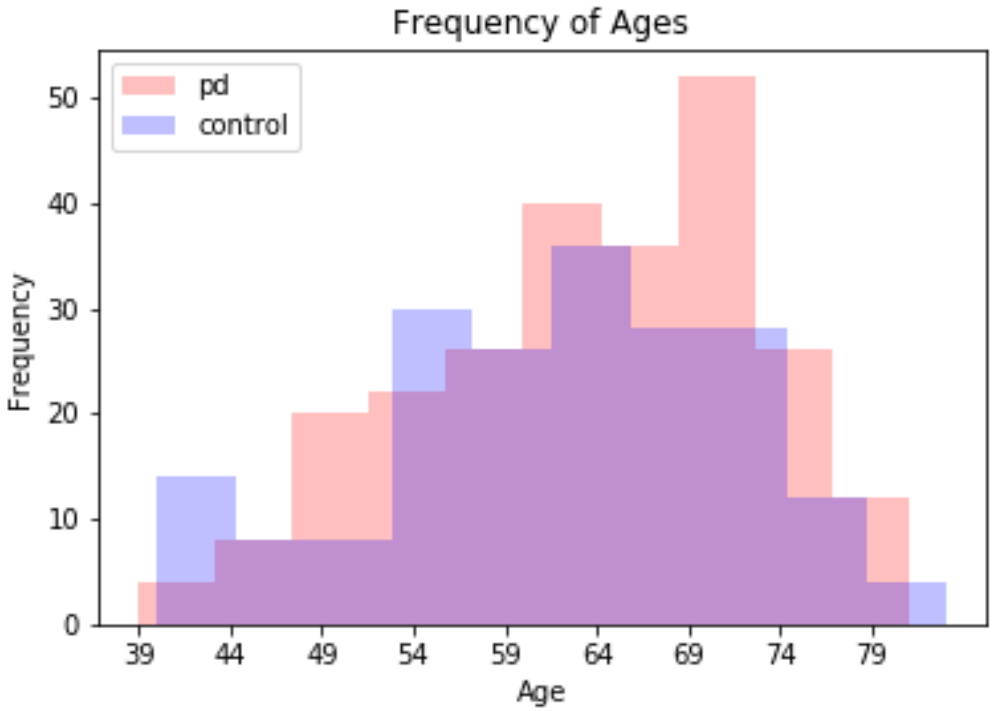

Figure 7: Histogram of the ages of patients in the 3D CNN model, of PPD and HC. Here there appear almost as many Control patients as PD, though PD has more patients in the ages of 59-74.

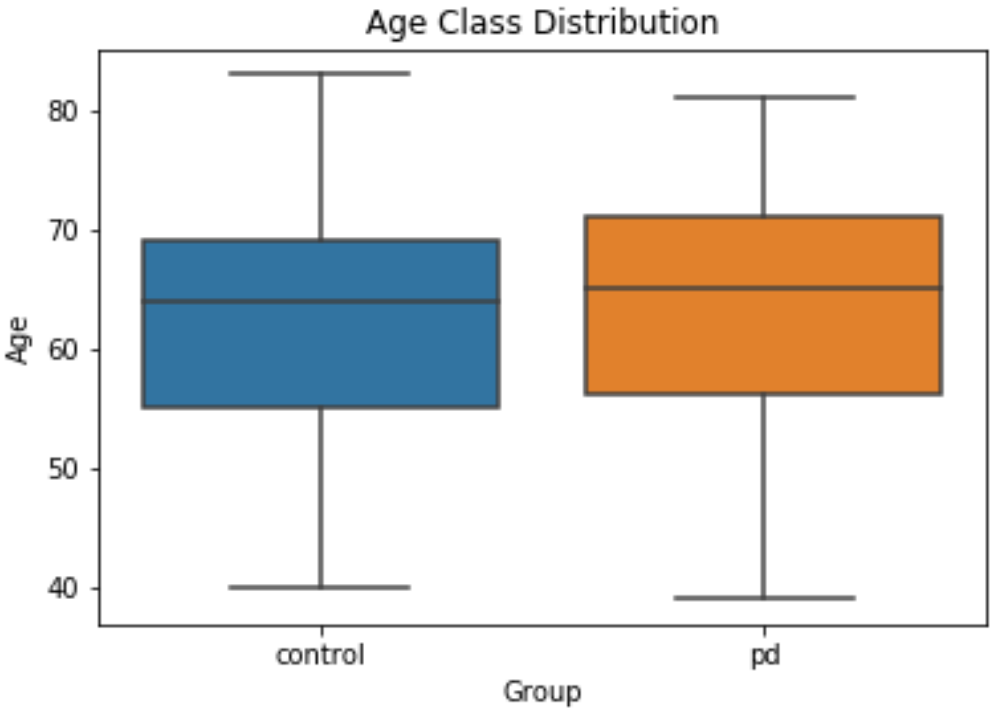

Figure 8: Boxplot of Age Distributions by class of the 3D CNN model data. The control and pd categories appear similar in distribution with the control having slightly higher upper ranges.

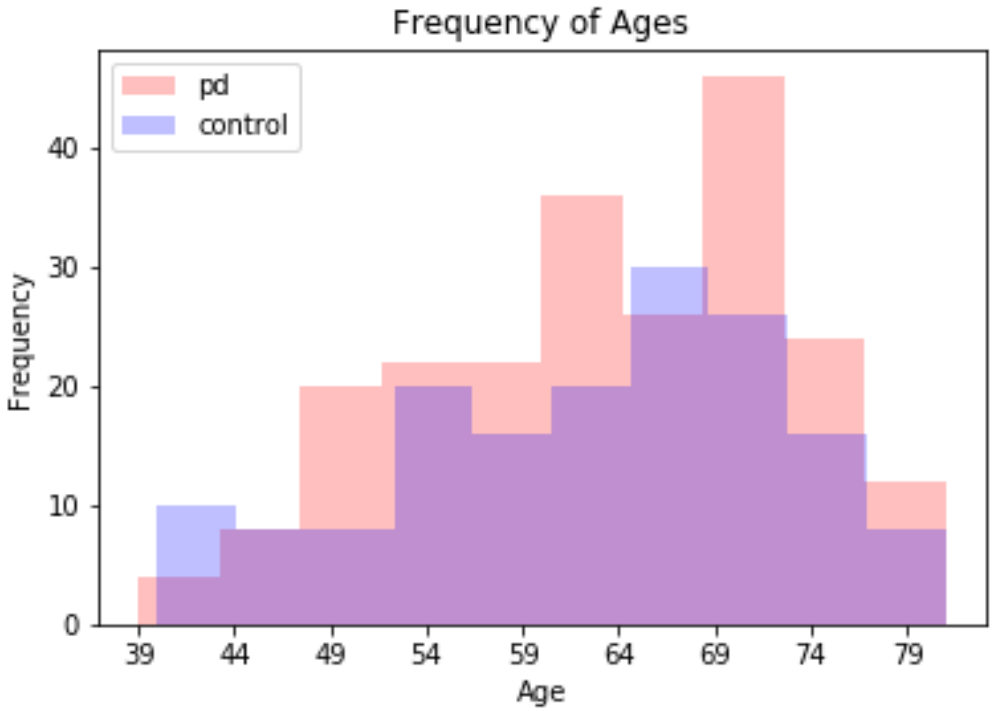

Figure 9: Histogram of the ages of patients in the 2D CNN model, of PPD and HC. There are more PPD than HC with particularly more in the age ranges of 59-64 and 69-74.

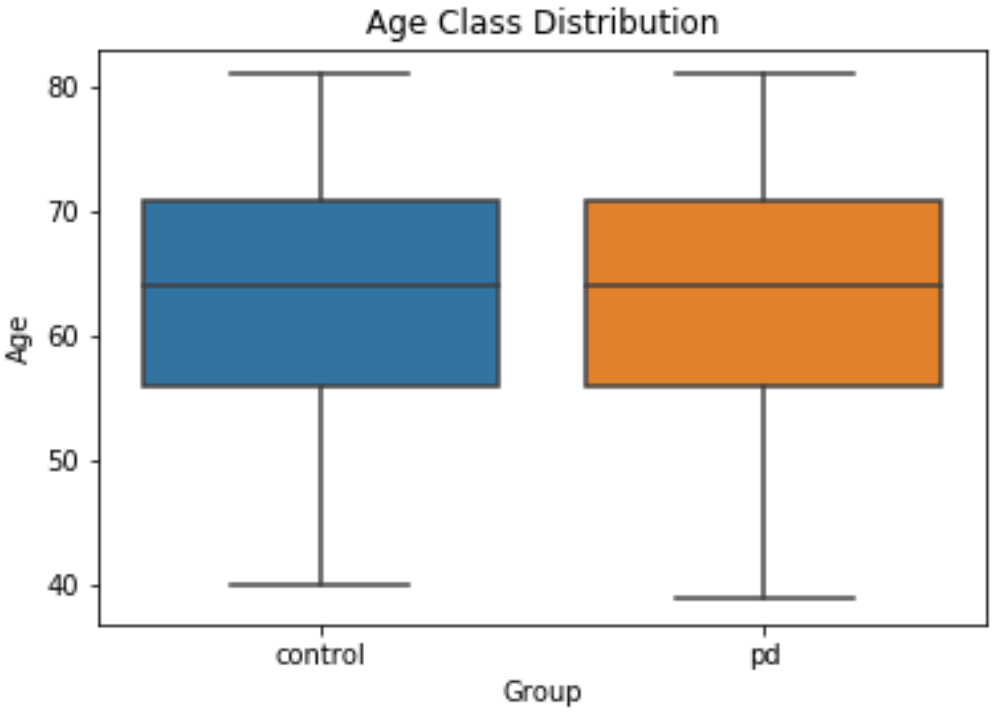

Figure 10: Boxplot of Age Distributions by class of the 2D CNN model data. These appear almost identical between each class.

4.2 Deep Learning Models

A deep learning model in this study refers to a neural network with several layers that can learn and automatically discover representations through multiple levels of abstraction for classification or detection [50]. Each of our models were trained using Google Colab GPU (1xTesla K80 , having 2496 CUDA cores, compute 3.7, 12GB(11.439GB Usable) GDDR5 VRAM with 12.6GB RAM) [51].

For the age/sex model, we input the one-hot encoded age and sex into individual (8, 2) FCNN with the class as the output. Then the two outputs are concatenated and put into another (8, 2) FCNN to form an ensembled deep NN, this can be seen in figure 11.

Each layer used ReLU activation function and the final layer used a sigmoid activation function as it introduces non-linearity and is differentiable [52][25]. Categorical cross-entropy was used as the loss function [26], with Adam optimizer (learning rate = 1e-3 and decay = 1e-3/200) [53], categorical accuracy was used as the accuracy metric. The Age/Sex model was trained for 500 epochs with batch sizes of 150 and a validation split of 0.1 to monitor the training progress. The training progress of this model can be seen in figure 14. Note that this model was the least computationally expensive, hence the ability to use very large batch sizes to increase the stability of our model in classifying the test set.

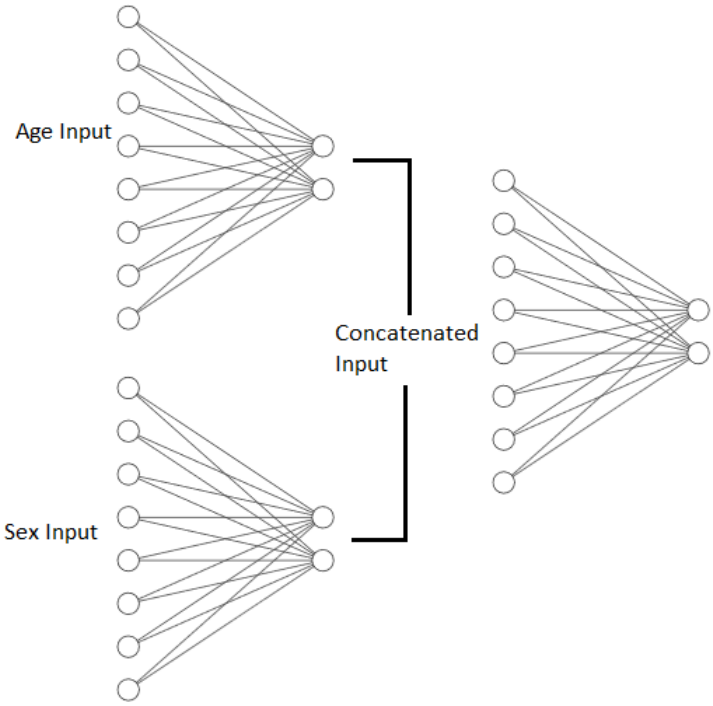

Figure 11: Triple (8, 2) FCNN model, where the one-hot encoded age and sex individually go through a (8, 2) FCNN before ensembling the output into another (8, 2) FCNN. Diagram was created with [6].

The 2D CNN model used in this study consists of 3 convolution layers consisting of 16, 32 and 64 nodes. Each convolution layer is followed by a max-pooling layer, each with pooling size 2 to finally be connected

to two FC layers consisting of 10 and 2 nodes respectively. This architecture can be seen in figure 12.

Each convolution layer has padding, kernel size = 3 and stride = 1. A L2 kernel initialiser and bias initialiser was used with parameter = 5e-3 to reduce overfitting and with Leaky ReLU as the activation function with $\alpha = 0.2$. We also use batch normalisation with momentum factor, $\mu = 0.5$ between max pooling and convolution layers. A dropout layer was also used between the two fully connected (FC) layers, with rate = 0.2 leading up to the final layer which used softmax as the activation function. The same optimiser, accuracy metric and loss metric was used as in the Age/Sex model. However, the batch size was 32 (smaller due to greater computational complexity demand with CNN models) and was only trained for 30 epochs as there was effectively no more learning occurring (evident from the flat line training accuracy graph in figure 12). Also, due to the limitation of disk space, only 100 images could be trained at once, so one epoch consisted of 3 batches (as our 150 training scans were augmented to 300 scans) where random shuffling occurred within each batch but not between batches. The test set was kept as a separate batch. This approach was not required with the 3D CNN training process as Gaussian masking reduced the required disk space enough to be able to not have to cyclically load batches for each epoch during model training. On the other hand, 3D CNNs are more computationally intense than 2D CNN models and so the batch size was set to 15 to optimise performance and speed.

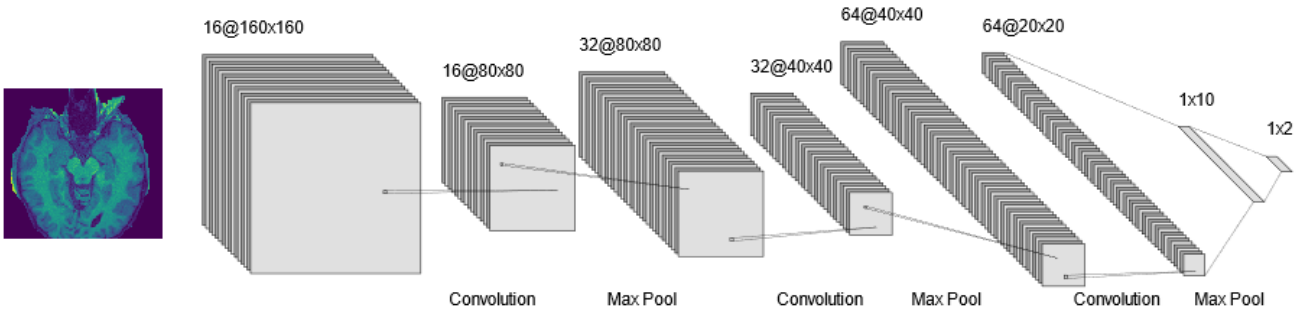

Figure 12: 2D CNN architecture of 3 convolution layers, 3 pooling layers and finally 2 fully connected layers. The initial input is a (160, 160) pixel image and the target output is a one-hot encoded class variable (PPD or HC). Diagram was created with [6].

Our third model is of 3D CNN architecture, an overview of which can be seen in figure 13. We used an architecture made of 5 3D convolution layers of 32, 64, 128, 256 and 512 nodes with 3D max pooling layers between them (L1-10 in figure 13). The purpose of doubling the number of nodes with each layer is to increase the number of abstractions with each pass before building and detecting features in our fully connected layers after, of which we have 3, consisting of 512, 64 and 2 nodes (L11-13 in figure 13) [50]. Each of our models have 2 node outputs, ultimately representing the possible one-hot encoded class outputs of whether the imaging scan is one of a PPD or HC to match predicted class outputs with actual class outputs.

L2 kernel and bias regularisers were also used with parameter=7e-3, as well as padding and stride=2. Each convolution layer had kernel size 3, except for L9 with kernel size 2. Leaky ReLU was used as the activation function after each convolution layer with $\alpha = 0.15$, this was also the activation function for the

FC layers (L11-13) and batch normalisation before each max-pooling layer with momentum factor μ=0.2. The max-pooling layers had pool size 2 and stride 2. Dropout layers were sandwiched between the FC layers with rates of 0.20 and 0.35 respectively. The final activation layer was softmax and choice of optimiser, loss and accuracy function were the same as the 2D CNN and the Age/Sex model. It should be noted that many of these parameters were chosen empirically after tuning and experimenting, the results from some of the prior experimentation can be found in the Results section and Appendix.

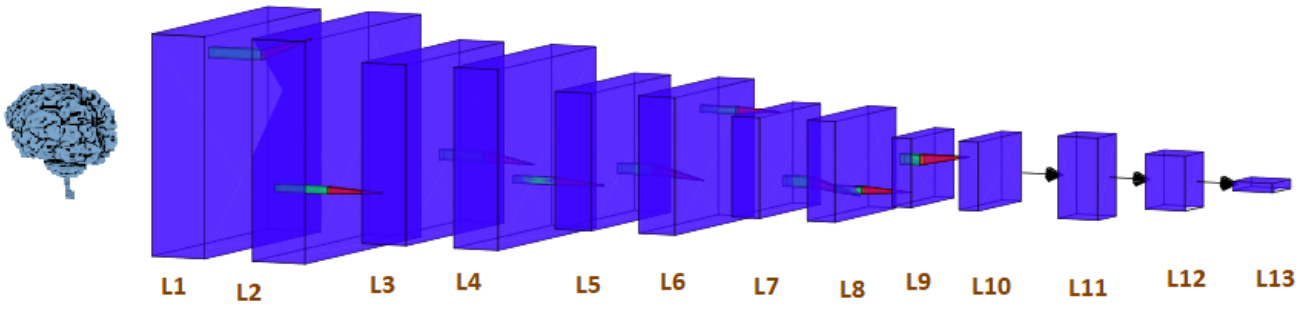

Figure 13: 3D CNN architecture, L1-10 are of alternating 3D convolution layers and 3D max pooling layers, with L11-13 being of FC layers. Thereby taking a (70, 160, 160) 3D scan as input and outputting one of two class values (PPD or HC). Diagram was created with [6], 3D Brain from Nevit Dilmen [7].

5 Results, Analysis & Discussion

5.1 Results

5.1.1 Model Training Plots

All the models were trained with categorical accuracy as the accuracy metric and categorical cross entropy as the loss metric. Both overfitting and underfitting are problematic for our models, as overfitting means that our model is unable to generalise to diagnose new data and can only classify the data it has been trained on, whereas underfitting is also a problem as it may lead to low accuracy and may not have learned all the important features within the MRI data set. Thereby highlighting the importance of factors such as number of epochs, batch size and model parameters. A key indicator we look for in a robust model here is a low validation loss, which indicates how well our model might perform against a test set [54]. Note the plots below have varying number of epochs, a concise assessment of each plot can be found in the captions.

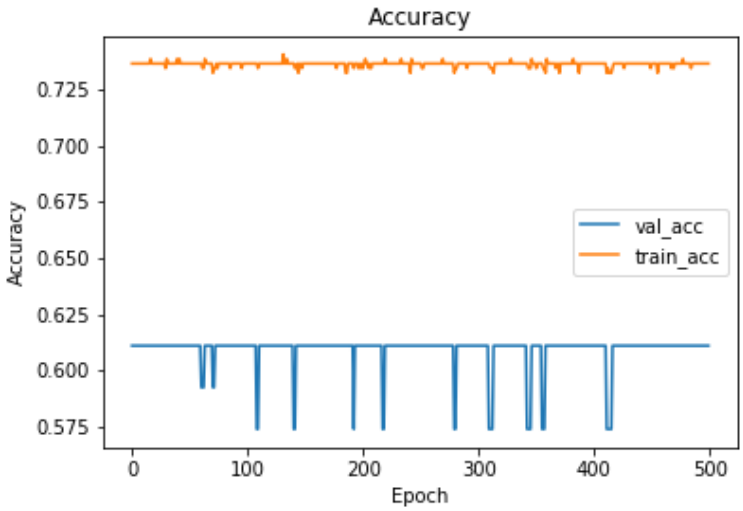

Figure 14: Age/Sex Model Accuracy Plot. The horizontal line for validation and training accuracy suggests that the model is no longer learning. It appears training for less number of epochs would have resulted in a similar outcome. The accuracy remained constant around 0.750 for the training set and around 0.610 for the validation accuracy.

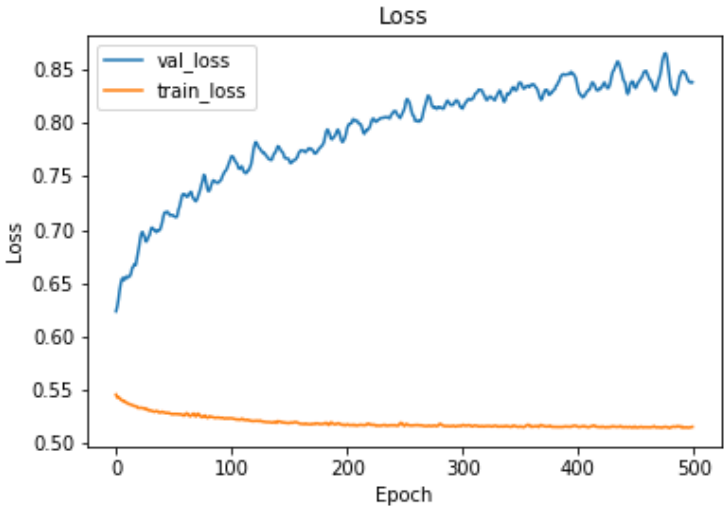

Figure 15: Age/Sex Model Loss Plot. The divergence of the training and validation curves suggests there is a high degree of overfitting occurring after 500 epochs (an epoch is a single pass through the entire training set). The loss here remains flat for the training around 0.52 and 0.85 for the validation set after 400 epochs, justifying the number of epochs after all. On the contrary, the validation loss should have a period of decline before increasing.

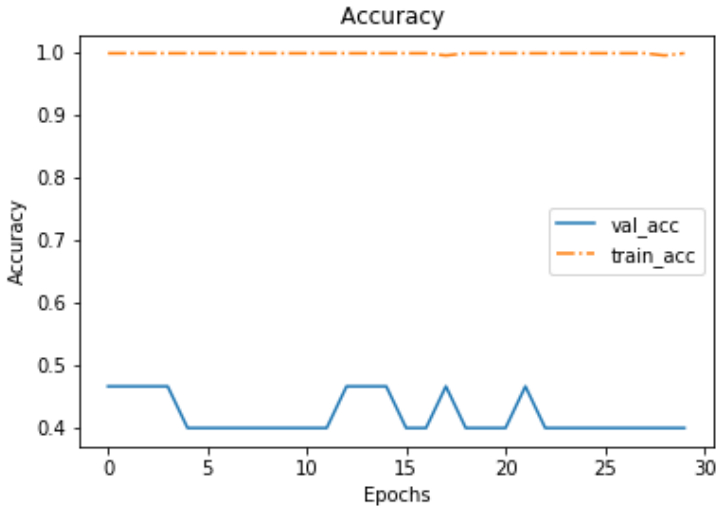

Figure 16: 2D CNN Model Accuracy Plot. This model was only trained for 30 epochs as the model has reached an accuracy close to 1.0 for the training set at the very beginning and did not learn any more in the sense that the accuracy remained constant. This implies that our model may have over fitted to our training set, especially as the validation accuracy remains between 0.4 and 0.5.

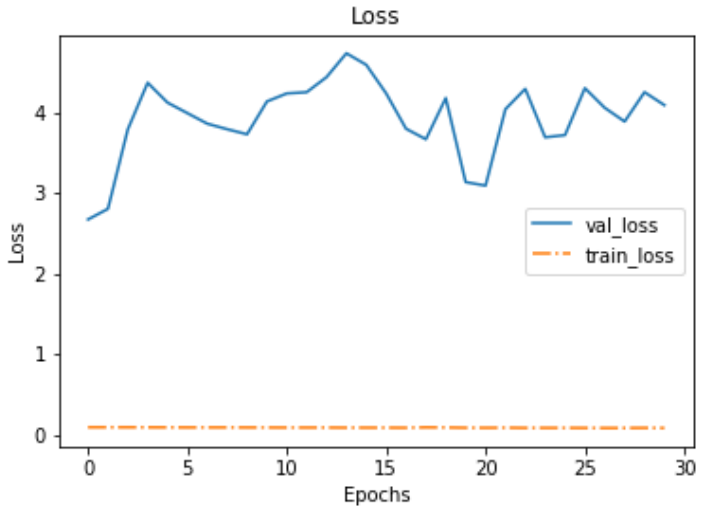

Figure 17: 2D CNN Model Loss Plot. Here the training loss appears constant around 0, showing a high degree of overfitting to our training set. The question-ability of the model robustness is furthered by the fluctuating validation loss between 2.5 and 4.5. On the other hand, perhaps it may be a simpler issue, such as not having randomly shuffled the dataset in proper manner.

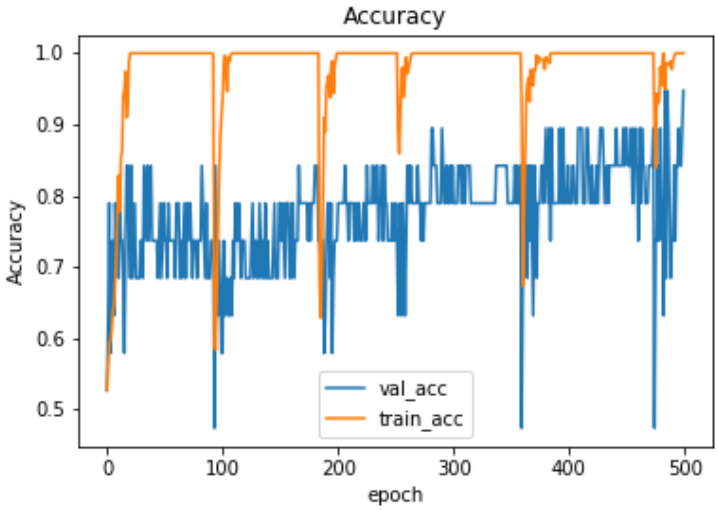

Figure 18: 3D CNN Model Accuracy Plot. Here we see periods of very high accuracy before a sudden drop, this indicates that we have over fitted our model because when it over fits to some parts of the dataset, it struggles to classify other parts of the dataset. This could also be indicative of a wild outlier in the dataset. Though in this case the accuracy of the training set is quite high (around 1) and the validation set reaches a maximum of around 0.9 showing some rigidity to the 3D CNN. It also appears the accuracy at the end of the training is a maximum for the training and validation set, which though may have been by coincidence, is timely to stop training the model for fear of the graph dipping again.

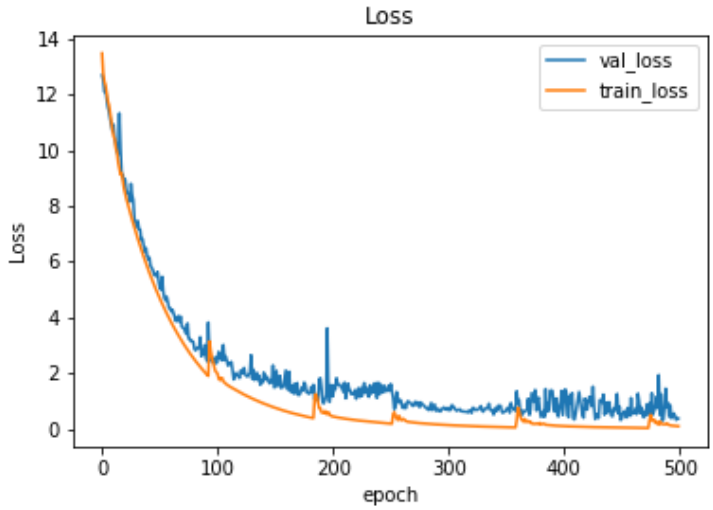

Figure 19: 3D CNN Model Loss Plot. Here, both the validation and training loss decrease together, with the training loss less than the validation loss showing that our model is not under fitting either. Training occurred for 500 epochs with the loss constantly decreasing, showing the improvement of our 3D CNN model and might not be overfitting after all so is actually a good fit, providing further evidence of an outlier in the 3D dataset and so calling for data quality checks. The loss function here appears to be within 0 to 1 for both our training and validation set; in cross-entropy, we look for loss values beneath 0.5.

Of the graphs above the 2D CNN model appears to have the lowest loss for training but the greatest loss for validation, indicative of overfitting. The loss of the age/sex model diverges for the validation and training sets, eventually plateauing after approximately 300 epochs, indicating that the model has reached convergence. This also appears to occur after around 300 epochs for our 3D CNN model by observing figure 19. Therefore, it appears the 3D CNN is the most stable of the three models by evaluating the loss graphs. This acknowledgement is further evidenced by the accuracy plots; of which the 3D CNN has the highest validation accuracy of above 0.9 at the end of training, in comparison to the rather low validation accuracy around 0.4 for the 2D CNN and above 0.6 for the age/sex model.

Whilst the graphs are all worthy indicators of how powerful and robust our models are, we fundamentally want to assess the performance against unseen (test) data. The results of which we will see in the next section.

5.1.2 Test Set Performance

Following training, we evaluate each of our models by measuring the accuracy of the results following 10-fold cross-validation, as well as the precision, recall and f_1 score of the strongest of each type of model. Therefore, we are able to get a comparative overview of each of our models to compare to state of the art in terms of accuracy and get a more complete picture of the performance of the strongest of each model and thus see their potential in diagnosing PD. Also note two things, firstly, the plots in the previous sub-section 'model training plots' are of the best performing models in terms of accuracy. And secondly, table 5 to table 8 (the CNN models) have double the number of patients due to augmentation of the data set where we doubled

the size of the imaging set by flipping each MRI axial slice.

Table 3 to table 8 below show the confusion matrix of the results of each of the highest performing model types on their respective training and test sets. In each confusion matrix, (0,0) are the true negatives, (1,0) are the false negatives, (0,1) are the false positives and (1,1) are the true positives for actual and predicted (pred) class values. The calculated summary metrics can be found in table 9.

		Actual	
		HC	PPD
Pred	HC	49	128
	PPD	20	339

Table 3: Age/Sex Model Confusion Matrix for Training Set. Here we can see a large number of false negatives (128/536), and the majority are the true positives (339/536).

		Actual	
		HC	PPD
Pred	HC	1	8
	PPD	3	48

Table 4: Age/Sex Model Confusion Matrix for Test Set. This is reflective of the training set, with the largest classification category being of true positives and second after (with a large difference) are false negatives. It is notable here that there is a heavy class imbalance between PPD compared to HC, this is not ideal and non-representative of the data set, so in future would be mitigated by obtaining the test set using stratified random split instead of just random split. The class imbalance problem is emphasised further as a classifier that only diagnoses as PD would be 93% accurate for this test set.

		Actual	
		HC	PPD
Pred	HC	121	5
	PPD	3	171

Table 5: 2D CNN Model Confusion Matrix for Training Set. Here the model appears quite promising given the very large proportion of true positive and true negatives to false positives and false negatives.

		Actual	
		HC	PPD
Pred	HC	36	0
	PPD	0	46

Table 6: 2D CNN Model Confusion Matrix for Test Set. Our model here has achieved a 100% accuracy on test set, appearing very powerful, even in spite of so much loss in validation from figure 17. We can test the reproducibility using cross-fold validation.

		Actual	
		HC	PPD
Pred	HC	176	1
	PPD	0	205

Table 7: 3D CNN Model Confusion Matrix for Training Set. This model has almost 100% accuracy on the training set, bar 1 false negative thus highlighting a very powerful model.

		Actual	
		HC	PPD
Pred	HC	16	1
	PPD	3	38

Table 8: 3D CNN Model Confusion Matrix for Test Set. This model is not 100% accurate, and has 4 misclassified results, which is less than the 2D CNN. However, appears to be a more robust model as typically the training set has a much higher accuracy than the given test set.

The high number of false negatives in tables 3 and 4 appear problematic, especially when diagnosing PD, as it is dangerous to diagnose a PPD as HC. The strength of this model is clearly not up to par with diagnostic methods and would not be recommended for use in clinical practice, though is representative of the potential usage of age and gender in clinical diagnosis and so could be integrated into hybrid models or otherwise as similarly suggested by [34].

Our 2D CNN appears very powerful with a 100% accuracy on test set and a lower accuracy on the training set from table 5 and table 6. Whilst the difference is not large, this prompts questions as to whether the training procedure is indeed correct. Further evidence is gathered by performing 10-fold cross-validation, the results of which can be seen in table 10.

Tables 7 and 8 show that our 3D CNN model trains to almost 100% and has 4 misclassifications out of 66 in the test set, with 3 of the 4 being false positives. The best case scenario would be to not have any type-I or type-II errors at all, such as the strongest classifier for our 2D CNN in diagnosing its respective test set, though requires cross-validation to test the reproducibility.

Table 9 shows the summary metrics for each of our models on the training and test sets. We can see that the highest of each metric is the precision in the training set, with values of 94.43%, 98.28% and 100% for

the Age/Sex model, 2D CNN and 3D CNN respectively. The Age/Sex model is the weakest by far, which is expected as one can only extrapolate so much information regarding whether or not a patient has PD from the gender and age, therefore was more of a venture out of curiosity. Despite that, the accuracy of the test set hit as high as 81.67%, with higher values of recall, precision and f_1 score by 4.04%, 12.45% and 8.05% respectively.

For the 2D CNN and 3D CNN models, it is worthwhile to note the considerable possibility of duplicate imaging scans as a result of there being multiple MRI scans belonging to any single patient which were taken at different points in time. We observe a 100% accuracy on the test set of the 2D CNN, though only a 97.33% accuracy on the training set. This is atypical and so raises questions around the robustness of the model, and to add is even more surprising considering the test set was relatively large compared to the training set (above 20% of the data).

Out of the three models, the 3D CNN appears the most robust as the test accuracy was less than the training accuracy at 93.10% with a high training accuracy of 99.74%. This is a more typical learning graph for a deep learning classification model, unlike what is seen by the age/sex model and the 2D CNN model. We can further assess the strength of each model by calculating the 10-fold cross-validation accuracy, which can be seen in table 10.

Model	Metric	Training (%)	Test (%)
Age/Sex FCNN	Accuracy	72.39	81.67
	Recall	72.59	85.71
	Precision	94.43	94.12
	f_1	82.08	89.72
2D CNN	Accuracy	97.33	100.00
	Recall	97.16	100.00
	Precision	98.28	100.00
	f_1	97.71	100.00
3D CNN	Accuracy	99.74	93.10
	Recall	99.51	97.44
	Precision	100.00	92.68
	f_1	99.76	95.00

Table 9: Overview of metrics of the best performing Age/Sex Model, 2D CNN and 3D CNN in terms of Accuracy, Recall, Precision and f_1 score for our training and test sets (2 d.p)

Below we see in table 10 that the 10-fold cross validation accuracy for our age/sex model, 2D CNN and 3D CNN are 74.41%, 85.23% and 87.87% respectively with accuracy ranges of [64%, 82%], [72%, 100%] and [82%, 93%] achieved in different training instances for the same model ordering.

Using the cross-validation accuracy as an objective measure of model performance over several runs, we can say that the 3D CNN is the most accurate of the three models and the most stable as it has the smallest range of accuracy results. Comparatively, the 2D CNN model also shows a high 10-fold cross-validation accuracy score though has a significantly larger range of accuracy values.

Therefore, there is potential in this space to have a very strong accuracy metric for both the 3D and 2D CNN models to push up the accuracy and simultaneously increase the reproducibility of each of the models.

The Age/Sex model performed as predicted, with a reasonable range of accuracy values and a 10-fold cross-validation accuracy of 74.41% as the raw data input itself does not offer much by way of foretelling the presence of PD. So this data combined with the maximum accuracy values for each model casts potential future work on increasing the stability of training of each model to achieve these accuracy values on our test set more consistently, as the accuracy values for certain runs of our 2D and 3D CNN models trump some state of the art models.

Model	10-fold Cross-Validation Accuracy (%)
Age/Sex Model	74.41
2D CNN	85.23
3D CNN	87.87

Table 10: 10-fold cross validation accuracy of each of our three models

5.1.3 Experiments

In this section we will highlight some experimental results (more of which can be found in the Appendix) from tuning variables and measuring the outcome as a result of changes to the model in attempt to optimise our created models for better performance as best as possible.

Effects of Gaussian Masking: Gaussian Masking significantly improved our 3D CNN. It is suspected this is because of the low signal to noise ratio initially which was alleviated by the Gaussian mask diminishing the effect of the noise. The highest accuracy before Gaussian Masking was 74%. The effects of varying σ, batch size and number of epochs can be seen in table 11.

σ	Batch Size	# epochs	Test Accuracy (%)
1.0	15	500	80.00
2.0	15	500	92.15
2.5	15	500	86.67
3.0	15	500	93.10
3.5	15	500	79.29
3.5	25	500	71.14
3.0	10	500	74.47
3.0	15	300	83.33
3.0	15	1000	87.88

Table 11: Effects of varying the degree of Gaussian Masking, then varying the batch sizes and number of epochs with different levels of Gaussian Masking. We can see the optimal value for σ is 3.0. Following this, we experimented with the batch size and epochs to find that batches of 15 and 500 epochs produced the optimum result with an accuracy of 93.10%.

Most of our experimentation occurred around the 3D CNN. The architecture changes and approaches learnt were then applied to the 2D CNN. The idea for batch normalisation and dropout layers improved the

accuracy of our model from 63% to 74% pre-Gaussian Masking. In addition, we also trained our 3D CNN with a non-augmented dataset to find a drop in accuracy by 9%, therefore providing empirical evidence justifying the usage of data augmentation.

Modification of architecture (in terms of number of nodes and layers) resulted in accuracy fluctuations around ±5% so was within the range of [0.53%, 0.63%] accuracy. More information can be found in the Appendix, though all of the architectures had between 4-7 convolution layers, each doubling the number of nodes from one layer to the next so are quite similar to that in figure 13. The size of our CNN models and batch sizes were limited by finite RAM. Other parameters also limited include the stride length and pooling sizes.

Below we list various trialled approaches and report their accuracy values. These were unsuccessful (as the useful approaches that improved accuracy made it into our actual models) but are worth mentioning.

- Ensembling the Age/Sex model with the 3D CNN. Accuracy: 62%
- Reducing the scan dimensions further to (128, 128, 128). Accuracy: 58%
- Training our 3D CNN model for 1000 epochs. Accuracy: 77%
- Two convolution layers before max pooling layer with 4 repetitions. Accuracy: 59% (with no Gaussian Masking)

5.2 Analysis & Discussion

Of the three models considered in this study, the 2D CNN has the greatest maximum accuracy value on a test set of 100%, followed by the 3D CNN with an accuracy of 93.10% on a test set and finally the Age/Sex model with a maximum accuracy of 81.67%. The original aim was to obtain a greater accuracy than state of the art, which is 97.75% set by [1] and has been met by the strongest 2D CNN. However, upon comparing the cross-validated accuracy, the greatest accuracy is of the 3D CNN at 87.87%, which is not stronger than state of the art models but rather trumps the accuracy of [37] without the need for a feature extraction/selection step. That is perhaps one of the key stand out points of a CNN, as it is able to learn the key changes between classes and although has not been done in this study, is something to be considered for future work.

Our Age/Sex model does not stand up to other accuracy values in literature, as it pales in ability to classify PD with a cross-validation accuracy of 74.41%. This is still better than random (50%) or if we had a classifier that only classified as PD (68% for our test set), thus suggesting implementation into future models which attempt PD diagnosis. Using the age and gender of patients in an ensembled deep learning model as is displayed in figure 11 is yet seen in literature and can therefore be considered as a novel contribution.

The original goal of diagnosing PD with 100% accuracy using a 3D CNN has been performed by [2]. However, they did not report any cross-validation method. We then took it upon this study as a goal to validate this result. This did not happen as we achieved a cross-validation accuracy of 87.87% with the same PPMI data source and our strongest 3D model achieving 93.10% accuracy on test set and so is less accurate by 6.90%. Because of the maximum result by our 2D CNN model of 100%, we firmly believe that a goal of 100% accuracy for PD diagnosis using CNN is achievable. Therefore, it seems that the focus of future work should be to make achieving this goal more reproducible in order to raise the cross-validation accuracy closer to 100%. In doing so, we may solve the initial problem of diagnosing PD at prodromal stages where dopamine producing cells have not yet deteriorated as well as cells around the SN.

By looking at figure 14 and figure 16, it is observed that there is a large gap between the training and validation accuracy. This implies a large degree of overfitting, meaning there is high variance in our model. This was offset slightly by introducing measures such as drop-out layers and regularisation, though without much success as the loss diverged then eventually stabilised in figure 15. This is indicative of an unstable model, which is evident from the wide range of accuracy values recorded of [64%, 82%] when cross-validating. Also, it appears from the accuracy curves of the 2D CNN and the age/sex model that the models did not learn over epochs as the graphs are horizontal, this appears indicative of high variance. Again, the instability of the 2D CNN is reflected in figure 17 with the fluctuating validation loss. Despite this, the 10-fold cross-validation score reached 85.25% which falls very short of its maximum classification accuracy of 100%. Though impressive that our 2D CNN may classify with 100%, it is unable to reproduce this result reliably, therefore, future work would be primarily in attempt to stabilise this model to produce strong test results repeatedly. That being said, the strongest performing 2D CNN can appear dubious as the training accuracy is less than the test accuracy, as seen in table 9.

Also, from table 9, we can see that our highest performing 3D CNN has accuracy, recall, precision and f_1 score of 100%, with only 1 reported false positive from table 7. This highlights a potentially very powerful model, though prompts questions of overfitting as the training accuracy is almost 100% and the test accuracy is slightly lower around 93.10%. We can also observe this from figure 18 where it becomes more clear that our 3D CNN has over-fitted because of the number of times the accuracy drops dramatically for both the validation and training accuracy, implying that it has fit so well to some MRI scans that it cannot correctly diagnose others. In brief, an overfitted model has learnt what to observe too precisely and cannot generalise. On the other hand, we may argue our model has not overfit from looking at figure 19 where we see the loss function constantly decrease for both the validation and training set, though has diminishing decreases after 200 epochs. This led to a lower number of epochs tested as is seen in table 11, (on the second to last row) obtaining a test-accuracy of only 83.33% with 300 epochs.

It was not expected for our Age/Sex model to exceed any accuracy values met in recent literature, especially one that exceeds clinical diagnosis of PD by non-experts of 73.8% accuracy [3]. Therefore, with our cross-validated accuracy of 74.41% accuracy, there presents the opportunity to incorporate these factors into other models to boost their classification ability or adapt other models to take into account patient metrics. It is also worth noting that the distribution of PPD to HC in real life clinical practice may not be reflected at all by the distribution of PPD to HC in the PPMI sub-samples chosen, or even in the PPMI dataset itself as figures 5, 9 and 7 show a much larger proportion of PD patients to HC which is extremely unlikely to be reflected in reality.

From table 1, we immediately notice the 100% accuracy achieved by [2]. Although he did not report any validation metric, he obtained an impressive result which this study has achieved with the best 2D CNN model also at 100% accuracy. All the accuracy values on table 1 used PPMI MRI data-set, though each other than the 3D CNN by [2] used methods of feature extraction followed by a classifier, of which SVMs were a popular (and powerful) choice. To restate, CNN models are advantageous as no feature extraction step is required, and so we report that the 2D CNN and 3D CNN in this study exceed the cross-validation accuracy of 81.9% with a LDA classifier by [40] as we recorded accuracy values of 85.25% and 87.85% respectively. One additional feature to consider for the future is a way to provide a quality of data assurance, or metric to weigh data points prior to model input as our 3D CNN may have suffered as a result of a fractional minority number of outliers of low quality imaging scans. Note, that our study used cross-validation over leave-one out validation, the latter of which may be more representative for usage in a clinical setting and would be

reasonable to also consider this for future works.

In terms of creating a model with greater accuracy, we have fine-tuned our parameters throughout this study via an experimental process of trial and error. One such example can be seen in table 11 where we adjust the degree of masking to find the best value for σ, then vary the batch size and epochs to further tune the accuracy. Unfortunately, due to limitations in computing resources, resolution may have been lost with larger stride lengths or larger filter sizes instead of smaller strides/filters. As a result, compromises were made in order to keep the number of tunable weights (nodes) below a certain threshold. So a future consideration includes having more granular filters and smaller stride lengths (more RAM). Despite this, workarounds were made such as reducing the batch size.

Our 3D CNN model in this study (which achieved our highest cross-validation accuracy) has better cross-validation results than [37] of 86.9% but falls short of 91.0%, 95.0% and 97.5% obtained by [38], [8] and [1] respectively. As we have obtained very high maximum classification accuracy of 100% and 93.10% on test-data for our 2D and 3D CNN models for which our CNN models exceed the accuracy values of [38] and [8] and [1] for our 2D Model in this study. The capacity for CNN models over SVM models in medical diagnosis is realised. In addition, by having no feature-extraction process, this form of unsupervised deep learning whereby our CNN models learn the important features in each MRI scan acknowledges the applicability of CNN models for computer-vision tasks in medical context. For good measure, we can then examine which features our CNN picks out in order to augment our understanding of such diseases in the medical field outside of just PD and MRI to further afield [34][36][2]. Though this has not been covered in this study, is scope for future work. Other obvious future work would be to improve the robustness of our CNN models, this includes hybrid ensembled deep learning models which have yet to be seen in literature [34]. In the Conclusion section we suggest some possible approaches for improving our model stability and other mentioned items.

6 Conclusion

6.1 Summary

The original aim of this study was to diagnose PD with MRI data from the PPMI database at an accuracy that exceeds above state of the art benchmarks set by prior SVM classifiers in the works of [1] who achieved a cross-validated accuracy of 97.5%. A prerequisite for the classifier models investigated in this study were that they implement a form of deep learning, this occurred by way of an ensembled deep FCNN, as well as 2D and 3D CNNs. In accordance, the second aim was to validate the 100% accuracy reported by [2] who also used a 3D CNN, though did not report cross-validation so raises questions about the reproducibility of their result and robustness of their model. Finally, as a curious venture, we input non-MRI patient data, namely the age and gender into an ensembled deep FCNN in attempt to classify PD. The benchmark for our tertiary classifier is set at 73.8%, which is the current state of clinical diagnosis of PD by non-experts [3].

SVMs are clearly a popular choice as we can see in table 1 and rightly so with high reported accuracy values above 90%. However, this requires an additional feature extraction/selection step that a CNN does not. Having a CNN learn which features are important is also valuable as it may detect features not considered in current clinical practice or literature. With our 2D CNN model, we report a maximum accuracy of 100% on a test set and a cross-validated accuracy of 85.23%. Therefore, the maximum accuracy does exceed state of the art classifiers, however, is unable to do so repeatedly and so has a cross-validated accuracy which trails by 12% in comparison to [1].

Nonetheless, we are able to see the potential of a CNN classifier in clinical practice, especially with ongoing improvements in the field of deep learning [50]. And so the next steps for our 2D CNN would be to implement a feature which is able to reliably select individual axial slices of the brain that offer a clear slice of the region where the SN is located. Or of the same relative brain region for each patient as the slice positioning may change relative to each patient scan due to the patient's initial positioning when taking the MRI scan as well as different head shapes and sizes. This could perhaps be incorporated by using another CNN classifier or SVM and select the maximum likelihood brain slices. This feature would be welcome because currently the selection method is to take the 86th slice, which was initially selected manually and not verified to be the same part of the brain; creating a problem and potentially giving outlier results. Therefore, highlighting the importance of an automated slice-selection method to offers a more end-to-end approach with dual CNN classifiers (technically 'deepbrain' package was also used to perform skull-stripping as is seen in figure 3, so would be a triple CNN classifier). The same suggested approach could also be used for 3D chunk selection, as currently we take a large section in the mid-brain where changes in the brain as a result of PD are expected to be [18][19]. Finally, another approach to test with the 2D MRI scans is Gaussian masking, as this resulted in a markedly large improvement in 3D classification accuracy as is seen in table 11 and is therefore worth pursuing.

Our 3D CNN did not meet the secondary goal of validating 100% reported by [2]. Instead we report a 99.97% and 99.10% maximum accuracy on training and test sets respectively (more metrics can be found in table 9), as well as a 10-fold cross-validation accuracy of 87.87%. This 3D CNN was found to be more robust than the 2D CNN and therefore has a higher cross-validation accuracy, thus exceeding the accuracy of [37] who achieved 86.9% using a SVM classifier but falls short of state of the art SVM classifier by 10%. Nonetheless, the upper margins of our 3D CNN run's accuracy value indicate that 100% is possible (as we got as high as 99.10% and firmly believe more can be done to improve this) and so infer that 100% accuracy

is achievable with a 3D CNN on MRI data for PD diagnosis.

To further improve our 3D CNN model, we would suggest future works include normalisation of the 3D MRI scan relative to the entire region taken. Another possibility is to map out CNN extracted ROI, which can supplement our understanding of PD in the brain. And though this feature was not implemented in this study, it would indicate how our model selects features and may even point out features yet considered in clinical practice. This could be hugely beneficial as so far, we only know whether or not for certain the existence of PD in patients following post-mortem examinations upon discovery of lewy body accumulation in the brain, as well as damage of the SN in a repeated pattern [19].

Lastly, our tertiary ensembled FCNN age/sex classifier scored a maximum accuracy of 81.67% on a test set and 74.41% upon cross-validation. Both of which indeed meet our third objective and so a future endeavour would be to assimilate this model into our other developed CNNs to boost their performance further. It should be noted that an ensembled deep NN using age and gender to classify PD has yet been observed in literature and is a novel approach, even more surprising as it exceeds the clinical accuracy benchmark of 73.8%.

These models serve and meet the goal of producing an end-to-end framework whereby we are able to classify PD for potential usage in a clinical setting. In the next sub-sections, we discuss future work and some pointers that would have been beneficial to know or at least acknowledge at the start of the project.

6.2 Future Work

There are non-specific ways to make a classification model better. Some obvious ones include having a larger data set, this could mean including other patient categories such as SWEDD (subject without evidence of dopamine deficiency) patients or those scanned with a different type of MRI machine. Another is to have equal amounts of PPD and HC, though the scale of this effect may not be significant and so should be tested.

Thirdly, a future endeavour would be to utilise a high performance computing cluster, of which we had access to Cambridge CSD3 and should have in hindsight have gone ahead with, instead of Google Compute Engine initially as we reached limits in terms of budget and therefore access to computing resources and subsequently switched to Google Colab. With greater computational power, we may have been able to increase the number of tunable weights (as we were restricted with stride, kernel size and number of scans we could process at any one time) to have a more powerful and accurate model.

In addition, it may be worthwhile comparing accuracy values across different software libraries. Tensorflow 2.0 was primarily used in this study, though there exist other powerful frameworks such as AlexNet and FastAi (built on top of PyTorch), which may offer better optimisation or more flexibility to experiment with different models and perform experiments. Eventually, the goal would be to have a classifier that can augment a medical professional's clinical diagnosis of PD when examining a MRI scan, especially as MRI machines become more powerful and so offer greater resolution and diagnostic potential with or without computer-vision [9]. For now, however, work should be focused on improving the stability and reproducibility of our models which achieve upper-margin accuracy values.

6.3 Lessons Learned

Below are some pointers that whilst are quite general, are subjectively very important and are things I wish were considered with greater appreciation during the start of the project and exponentially more so during

the analysis and write-up towards the end.

- Document (experimental) processes better, have a rigorous procedure and stick to it! This will save time in the future when it comes to sifting through historical results to present findings
- Also, the same can be said for documentation of source code and keeping a tidy repository in order to make the work to be as reproducible as possible, which is essential for research to be considered as repeatable and reputable.
- Make note of things or ideas that come to mind as they are often dispersed and can be at random times, like when you're at a museum or otherwise. It is better than sitting down later on and getting frustrated trying to remember what that 'bazinga' moment was about.

One last hard lesson learnt is about time optimisation. Too much time was spent chasing small improvements in accuracy for our model at early stages of the project, where gains were around 3% at a baseline accuracy of around 60% to 70%. This would have been more useful to do at the end where the accuracy began approaching 90%+. Such an activity for example is fine tuning hyper-parameters. Instead, more time should have been spent at the beginning to make leaps where possible and arguably should have been an ongoing pursuit even towards the end of the project. Examples of 'leaps' include adding dropout layers and batch normalisation, or applying Gaussian Masks to our data before input into our CNN. These represent horizontal changes where new approaches are trialled and implemented rather than vertical changes where fine-tuning occurs to what already exists. So perhaps the one takeaway I would like to part with our reader would be to make bold ventures and leave a traceable trail so others may follow suit, or so you can at least find your way back should you wander off-track.

References

[1] E. Adeli, G. Wu, B. Saghafi, L. An, F. Shi, and D. Shen, "Kernel-based joint feature selection and max-margin classification for early diagnosis of parkinson's disease," *Scientific Reports (Nature Publisher Group)*, vol. 7, p. 41069, 01 2017. Copyright - Copyright Nature Publishing Group Jan 2017; Last updated - 2017-08-18.

[2] S. Esmaeilzadeh, Y. Yang, and E. Adeli, "End-to-end parkinson disease diagnosis using brain mr-images by 3d-cnn," *ArXiv*, vol. abs/1806.05233, 06 2018.

[3] G. Rizzo, M. Copetti, S. Arcuti, D. Martino, A. Fontana, and G. Logroscino, "Accuracy of clinical diagnosis of parkinson disease," *Neurology*, vol. 86, no. 6, pp. 566–576, 2016.

[4] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*. Cambridge, MA, USA: MIT Press, 1st ed., 1995.

[5] A. Borad, "Regularization: Make your machine learning algorithms "learn", not "memorize"," Jan 2019.

[6] "Nn-svg diagrams." `http://alexlenail.me/NN-SVG/AlexNet.html`.

[7] "Nevit dilmen 3d brain." `https://commons.wikimedia.org/wiki/File:3DPX-003765_3DModel_of_Brain_Nevit_Dilmen.stl`.

[8] G. Singh and L. Samavedham, "Unsupervised learning based feature extraction for differential diagnosis of neurodegenerative diseases: A case study on early-stage diagnosis of parkinson disease," *Journal of neuroscience methods*, vol. 256, 08 2015.

[9] S. Lehéricy, E. Bardinet, C. Poupon, M. Vidailhet, and C. François, "7 tesla magnetic resonance imaging: A closer look at substantia nigra anatomy in parkinson's disease," *Movement Disorders*, vol. 29, no. 13, pp. 1574–1581, 2014.

[10] Y. T. S. E. Hirobumi Oikawa, Makoto Sasaki and K. Tohyama, "The substantia nigra in parkinson disease: Proton density-weighted spin-echo and fast short inversion time inversion-recovery mr findings," *American Journal of Neuroradiology*, vol. 23, no. 10, pp. 1747–1756, 2002.

[11] G. . Disease, I. Incidence, and P. Collaborators, "Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 1990–2015: a systematic analysis for the global burden of disease study 2015," *The Lancet*, vol. 388, no. 10053, pp. 1545–1602, 2016.

[12] K. Marek and D. J. et Al", "The parkinson progression marker initiative (ppmi)," *Progress in Neurobiology*, vol. 95, no. 4, pp. 629 – 635, 2011. Biological Markers for Neurodegenerative Diseases.

[13] W. R. Gibb and A. J. Lees, "The relevance of the lewy body to the pathogenesis of idiopathic parkinson's disease," *Journal of neurology, neurosurgery, and psychiatry*, vol. 51, no. 6, pp. 745–52, 1988.

[14] J. R. Polimeni and K. Uludağ, "Neuroimaging with ultra-high field mri: Present and future," *NeuroImage*, vol. 168, pp. 1 – 6, 2018. Neuroimaging with Ultra-high Field MRI: Present and Future.

[15] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. Behrens, H. Johansen-Berg, P. R. Bannister, M. D. Luca, I. Drobnjak, D. E. Flitney, R. K. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. D. Stefano, J. M. Brady, and P. M. Matthews, "Advances in functional and structural mr image analysis and implementation as fsl," *NeuroImage*, vol. 23, pp. S208 – S219, 2004. Mathematics in Brain Imaging.

[16] J. B. Schulz, M. Skalej, D. Wedekind, A. R. Luft, M. Abele, K. Voigt, J. Dichgans, and T. Klockgether, "Magnetic resonance imaging–based volumetry differentiates idiopathic parkinson's syndrome from multiple system atrophy and progressive supranuclear palsy," *Annals of Neurology*, vol. 45, no. 1, pp. 65–74, 1999.

[17] A. Quattrone, G. Nicoletti, D. Messina, F. Fera, F. Condino, P. Pugliese, P. Lanza, P. Barone, L. Morgante, M. Zappia, U. Aguglia, and O. Gallo, "Mr imaging index for differentiation of progressive supranuclear palsy from parkinson disease and the parkinson variant of multiple system atrophy," *Radiology*, vol. 246, no. 1, pp. 214–221, 2008.

[18] J. C. STEELE, J. C. RICHARDSON, and J. OLSZEWSKI, "Progressive Supranuclear Palsy: A Heterogeneous Degeneration Involving the Brain Stem, Basal Ganglia and Cerebellum With Vertical Gaze and Pseudobulbar Palsy, Nuchal Dystonia and Dementia," *JAMA Neurology*, vol. 10, pp. 333–359, 04 1964.

[19] K. Del Tredici, U. Rüb, R. A. I. De Vos, J. R. E. Bohl, and H. Braak, "Where does parkinson disease pathology begin in the brain?," *Journal of neuropathology and experimental neurology*, May 2002.

[20] L. C. Tan, N. Venketasubramanian, C. Y. Hong, S. Sahadevan, J. J. Chin, E. S. Krishnamoorthy, A. K. Tan, and S. M. Saw, "Prevalence of parkinson disease in singapore," *Neurology*, vol. 62, no. 11, pp. 1999–2004, 2004.

[21] M. Politis, "Neuroimaging in parkinson disease: from research setting to clinical practice," *Nature Reviews Neurology*, vol. 10, pp. 708–722, 2014.

[22] C.-w. Hsu, C.-c. Chang, and C.-J. Lin, "A practical guide to support vector classification," 11 2003.

[23] K. A. A. Abakar and C. Yu, "Performance of svm based on puk kernel in comparison to svm based on rbf kernel in prediction of yarn tenacity," *Indian Journal of Fibre and Textile Research*, vol. 39, pp. 55–59, 03 2014.

[24] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on Neural Networks*, vol. 3, pp. 683–697, Sep. 1992.

[25] S. Scardapane, S. V. Vaerenbergh, S. Totaro, and A. Uncini, "Kafnets: Kernel-based non-parametric activation functions for neural networks," *Neural Networks*, vol. 110, pp. 19 – 32, 2019.

[26] A. Rusiecki, "Trimmed categorical cross-entropy for deep learning with label noise," *Electronics Letters*, vol. 55, no. 6, pp. 319–320, 2019.

[27] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the rprop algorithm," in *IEEE International Conference on Neural Networks*, pp. 586–591 vol.1, March 1993.

[28] P. Tosteberg, "Semantic segmentation of point clouds using deep learning," 2017.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[30] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," *CoRR*, vol. abs/1205.2653, 2012.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, pp. 84–90, May 2017.

[32] A. Das, A. Roy, and K. Ghosh, "Proposing a cnn based architecture of mid-level vision for feeding the where and what pathways in the brain," in *Swarm, Evolutionary, and Memetic Computing* (B. K. Panigrahi, P. N. Suganthan, S. Das, and S. C. Satapathy, eds.), (Berlin, Heidelberg), pp. 559–568, Springer Berlin Heidelberg, 2011.

[33] Y. Wu and K. He, "Group normalization," in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), (Cham), pp. 3–19, Springer International Publishing, 2018.

[34] D. Gil and M. JOHNSON, "Diagnosing parkinson by using artificial neural networks and support vector machines," *Global Journal of Computer Science and Technology*, vol. 9, 01 2009.

[35] M. A Little, P. Mcsharry, S. Roberts, D. A E Costello, and I. M Moroz, "Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection," *Biomedical engineering online*, vol. 6, p. 23, 02 2007.

[36] A. Arora, J.-J. Lin, A. Gasperian, J. Stein, J. Maldjian, M. Kahana, and B. Lega, "Comparison of logistic regression, support vector machines, and deep learning classifiers for predicting memory encoding success using human intracranial eeg recordings," *Journal of Neural Engineering*, vol. 15, 09 2018.

[37] H.-J. Huppertz, L. Möller, M. Südmeyer, R. Hilker, E. Hattingen, K. Egger, F. Amtage, G. Respondek, M. Stamelou, A. Schnitzler, E. H. Pinkhardt, W. H. Oertel, S. Knake, J. Kassubek, and G. U. Höglinger, "Differentiation of neurodegenerative parkinsonian syndromes by volumetric magnetic resonance imaging analysis and support vector machine classification," *Movement Disorders*, vol. 31, no. 10, pp. 1506–1517, 2016.

[38] G. Pahuja and T. N. Nagabhushan, "A novel ga-elm approach for parkinson's disease detection using brain structural t1-weighted mri data," *2016 Second International Conference on Cognitive Computing and Information Processing (CCIP)*, pp. 1–6, Aug 2016.

[39] D. M. Hawkins, "The problem of overfitting," *Journal of Chemical Information and Computer Sciences*, vol. 44, no. 1, pp. 1–12, 2004. PMID: 14741005.

[40] E. Adeli, F. Shi, L. An, C.-Y. Wee, G. Wu, T. Wang, and D. Shen, "Joint feature-sample selection and robust diagnosis of parkinson's disease from mri data," *NeuroImage*, vol. 141, 06 2016.

[41] T.-T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Pattern Recognition*, vol. 48, no. 9, pp. 2839 – 2846, 2015.

[42] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, (San Francisco, CA, USA), pp. 1137–1143, Morgan Kaufmann Publishers Inc., 1995.

[43] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pp. 41–48, IEEE, IEEE, 1999.

[44] M. Cassel and F. L. Kastensmidt, "Evaluating one-hot encoding finite state machines for seu reliability in sram-based fpgas," in *Proceedings of the 12th IEEE International Symposium on On-Line Testing*, IOLTS '06, (Washington, DC, USA), pp. 139–144, IEEE Computer Society, 2006.

[45] F. Feliner, K. Holl, P. Held, C. Fellner, R. Schmitt, and H. Böhm-Jurkovic, "A t1-weighted rapid three-dimensional gradient-echo technique (mp-rage) in preoperative mri of intracranial tumours," *Neuroradiology*, vol. 38, pp. 199–206, Apr 1996.

[46] "deepbrain." `https://pypi.org/project/deepbrain/`.

[47] M. Hutchinson and U. Raff, "Parkinson's disease: a novel mri method for determining structural changes in the substantia nigra," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 67, no. 6, pp. 815–818, 1999.

[48] T. C. Redman, "The impact of poor data quality on the typical enterprise," *Communications of the ACM*, vol. 41, no. 2, p. 79–82, 1998.

[49] T. C. Redman, *Data Quality for the Information Age*. Norwood, MA, USA: Artech House, Inc., 1st ed., 1997.

[50] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436–444, 2015.

[51] "Google colaboratory." `https://colab.research.google.com/drive/151805XTDg--dgHb3-AXJCpnWaqRhop_2#scrollTo=vEWe-FHNDY3E`.

[52] R. HECHT-NIELSEN, "Iii.3 - theory of the backpropagation neural network**based on "nonindent" by robert hecht-nielsen, which appeared in proceedings of the international joint conference on neural networks 1, 593–611, june 1989. © 1989 ieee.," in *Neural Networks for Perception* (H. Wechsler, ed.), pp. 65 – 93, Academic Press, 1992.

[53] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[54] Y. Shibberu, "Introduction to deep learning: A first course in machine learning," *2017 ASEE Annual Conference Exposition Proceedings*, 2017.

A Appendix

A.1 Other Experimental Results

During the model tuning process, there was a lot of experimentation. I have recorded many of these in a Google spreadsheets folder, you can access the read only with the URLs below: Note: I have trialled over 100 models for the 3D CNN, and taken approaches that worked to the 2D CNN and Age/Sex FCNN.

- https://docs.google.com/spreadsheets/d/1nzSqv0be_ANGrEPMxZktjLKC5SwNmeGH7BIXbSF-wlo/edit?usp=sharing
- https://docs.google.com/spreadsheets/d/1rzmf_7p6sY1TnNdNPnWLwgYne-W6X8RwYMU4_Tnr9iQ/edit?usp=sharing

A.2 Source Code Listing/Readme

I have included the 'readme.txt' file below which explains the source code files used (mostly in jupyter notebook) run on Python.

```
The original code was executed in jupyter notebook on Google Colaboratory.
The .ipynb files can be found in the 'jupyter_notebook_files' folder. These
individual files were also exported as .py files and can be found in the
'python_files' folder.

Note that the patient data was downloaded into my local google drive
and can be found at https://www.ppmi-info.org/

There are a few jupyter notebook files, each of which have their own
function. These are listed below in the format 'name':'description' :

- 'Step1.ipynb' : Perform pre-processing, include loading the .DCM files,
  skull-stripping, cropping, augmenting and saving to array after checking
  the processed brain scan is of the right pixel dimensions (160, 160, 160).
  Also saving the corresponding patient information into array, including
  their age, gender, class (parkinson's disease or healthy) and finally
  saving these variables to a .pkl file to be used in a classification model

-  'misc2(age_sex_model).ipynb' : Loads the .csv file and extracts patient
   information. Here we have our age/sex model with data-preprocessing and
   training/testing the model

-  'misc3(2d).ipynb' : Here we run the 2D CNN model (ignore the description of
   using cv2 in the title). Steps include taking the 86th slice, normalising and then
   running it through a 2D CNN. With training and testing included.
```

```
- 'misc5(gaussian_mask).ipynb': We have here the 3D CNN model, along with methods to
  index the slices, taking the mid-brain region and concatenate the different
  3D MRI scans to train and test the 3D CNN on. This also includes the Gaussian Masking step

Things to Note:

- There is no code to perform cross-validation. This was conducted manually where
  the accuracy of each run was taken and averaged over 10 runs (with randomised training/test
  sets each time) to get the cross-validation score. Rather primitive but by definition is
  what k-fold cross-validation is.

Hope you enjoy, for any questions please feel free to contact me at 'steven.vuong@kcl.ac.uk'

Acknowledgement & thanks to:
- Google Colab for their great enabling tools
- Dr.Lam, Guangyu Jia for their wisdom
- Informatics department for the resources & teaching aid

Student ID: 1871066
Steven Vuong, 14/08/2019
```