Alzheimer MRI classification using Neural Networks

STAT3007/7007 Project Report
Group 16
Timothy Ryall (s46387334) · Cooper Janke (s4638729) · Nafis Riza (s4829902)
Torstein Korten (s4824968) · Paal Markus Bjoernstad (s4822011)

Abstract

Machine learning has been used to analyse medical images for many years, one such form of analysis that is gaining traction is the detection of Alzheimer's through brain MRIs (Moradi et al., 2015). We use a data set of 6,400 MRIs to develop Neural Networks that are capable of predicting the presence and severity of Alzheimer's, allowing for earlier and easier detection of the disease. We perform exploratory analysis of the brain MRI data, conduct several experiments using a range of machine learning models, with a focus on convolutional neural networks, and explore the practical and biological interpretation of our final model through the use of technologies such as Grad-CAM (Selvaraju et al., 2019a).

1 Introduction

1.1 Background

Alzheimer's disease is a brain disorder which results in the loss of various cognitive functions (Aging, 2023). It is the most common cause of dementia in the United States. It has a wide range of causes which range from genetic elements to lifestyle, but the causes are still not fully understood. There currently exists no cure, but diagnosis can help with managing the disease through various methods. Thus, it would be ideal to have a robust and non-invasive method for diagnosis as an earlier diagnosis will lead to better support through an earlier intervention. One promising biomarker which can be used for this purpose are brain images.

Magnetic Resonance Imaging (MRI) is a non-invasive imaging technique which can be used to produce 3-dimensional scans of a body (Biomedical Imaging and Bioengineering, 2022). It works by using a strong magnetic field to align protons combined with a radio frequency current to infer various properties about the surrounding tissue. MRI scans doesn't require radiation which makes it the desirable choice for regular screening and imaging. Thus, MRI is the method of choice for brain imaging where it can notably differentiate between white and grey matter in the brain.

1.2 Motivation

The dataset was originally shared with the intent "to design/develop an accurate framework or architecture for the classification of Alzheimer's Disease" (Kumar and Shastri, 2022). We seek to fulfill this intent through the investigation of various deep learning architectures. In addition, as dataset contains images from three different severities of Alzheimer's we seek to improve on previous models by creating a model able to distinguish between these groups. Finally, Alzheimer's disease has no currently known cure, but earlier diagnosis is beneficial. We believe that a deep learning model would be able to able to offer greater access to this diagnosis combined with a more consistent diagnosis, due to a reduced requirement for a highly skilled clinician which may not always be available.

1.3 Problem Statement

We seek to investigate various architectures with the purpose of producing an architecture which is able to accurately (with regard to both class-wise and overall accuracy) classify varying degrees of Alzheimer's disease. In addition, we seek to investigate the reasons behind the classifications the model makes.

2 Related Work

MRI imaging has previously been used as a method to diagnose Alzheimer's disease with reviews such as those by Reiman and Jagust (2012) and Pini et al. (2016) highlighting its successes so far. Other work such as that by Choi (2017) and Ahmed et al. (2019) has also highlighted the benefit of using deep learning based on MRI images for diagnosis of Alzheimer's.

Qiu et al. (2020) have had success with applying a 3D CNN coupled to a multilayer perceptron to perform diagnosis of Alzheimer's using 3-dimensional MRI images resulting in reasonable accuracy levels. However, their model has the limitation of only distinguishing between the presence or lack of the disease, while Alzheimer's disease presents as a spectrum. Feng et al. (2020) has shown the benefits of a 3D-CNN-SVM architecture which was able to distinguish between no disease and two severities of Alzheimer's with pairwise accuracies in excess 98%. Wang et al. (2018) have also investigated various deep learning architectures which can be applied to this problem, with their work notably highlighting the use of the leaky ReLU activation function.

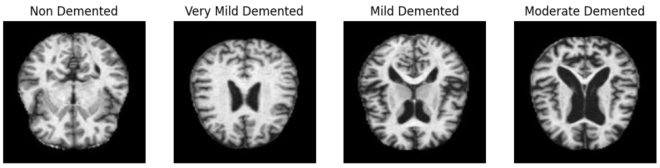
In addition, as a publicly available kaggle dataset others have already undertaken various classification attempts. Examples of these analyses include Tauhidi (2024) who have used an Efficientnet structure to achieve accuracies of more than 99%, while Fatima (2024) trained a range of models including LeNet, UNet and GoogLeNet to various degrees of accuracy.

3 Methods

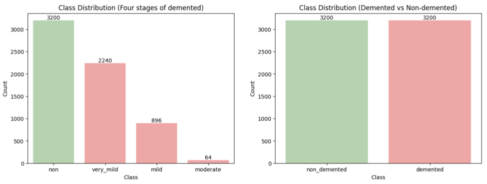
3.1 Dataset Extraction, Transformation and Formulation

The data set described above was sourced and downloaded from Kaggle (`https://www.kaggle.com/datasets/sachinkumar413/alzheimer-mri-dataset/data`) in the form of 6,400 png image files. These files were organised in a folder structure that sorted the data into the four respective classes (Non_Demented, Very_Mild_Demented, Mild_Demented, Moderate_Demented). There was no pre-segregation of testing vs training data within the provided data set, so this will have to be done when we conduct our experiments. In terms of data pre-processing we ensured all the images were 128 x 128 pixels and converted them all to 2D (data is grey scale, so adding another dimension is superfluous). Figure 1a shows a MRI image from each class.

It should be noted, that this report will focus on the Alzheimer's classification problem (classifying MRI's into 1 of the 4 classes). Equivalent formulations and experiments were produced for the Alzheimer's detection problem (classifying MRI's as Demented vs Non-demented), however the results and conclusions for these two problems were near identical. Thus for the sake of brevity we will focus on the classification problem as the practical interpretations of these models are more useful for the end users (i.e. doctors who wish to classify the stage of the disease). Note, there is a brief discussion on the detection based models in the model evaluation section below, and full detection experiments can be found in the Appendix.



(a) Example images for each class from the data set



(b) Class distribution for the 4 class data set and 2 class data set

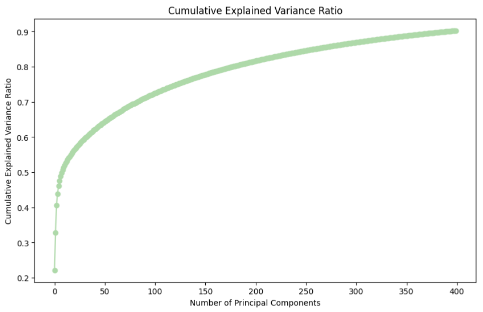
3.2 Exploratory Analysis

3.2.1 Class Distribution

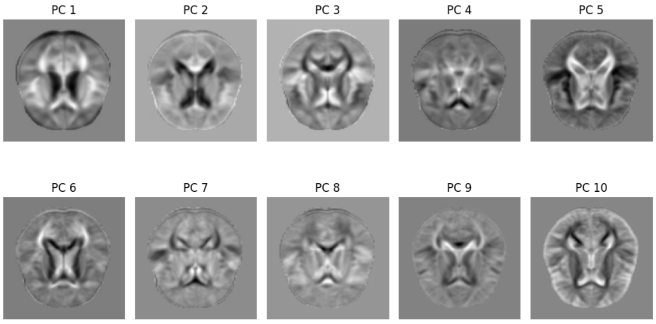
Looking at figure 1b we can see that there is a significant skew towards the less demented classes with very few data points belonging to the moderately demented class. Thus, to ensure correct predictions for the more demented classes, we had to ensure that the classes were correctly weighted in our model, and / or stratified sampling of train test data was preformed.

3.2.2 Principal Component Analysis (PCA)

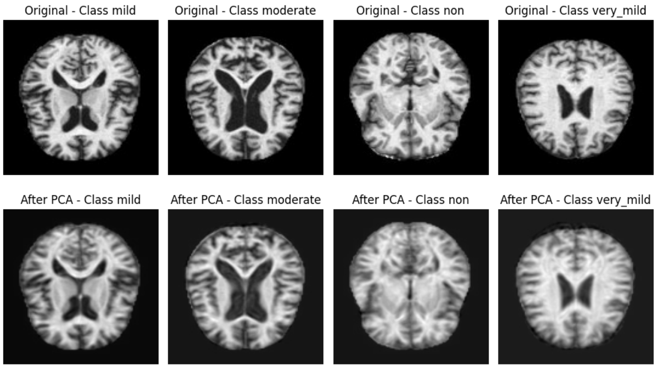
PCA is a dimensionality reduction technique used to gain insights and observe patterns from a high dimensional dataset by transforming it into a lower dimensional space while seeking to retain the variability in the original data. The original data consisted of 16,394 features, which is pretty high, and could slow the runtime of the ML algorithm that will be run. We used PCA to reduce the dimension to 400 principal components.



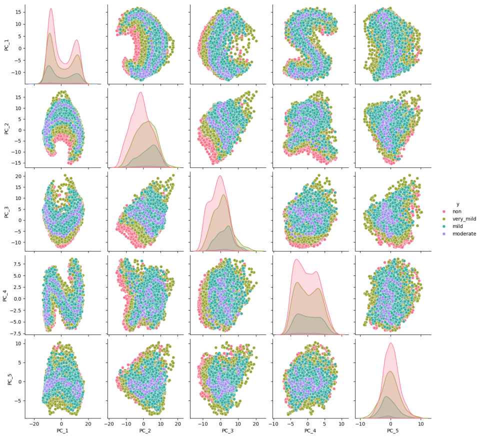
(a) Cumulative Explained Variance Ratio



(b) First 10 Principal Components


Original - Class mild
Original - Class moderate
Original - Class non
Original - Class very_mild
After PCA - Class mild
After PCA - Class moderate
After PCA - Class non
After PCA - Class very_mild

(a) Original VS PCA Images



(b) Pair-plot first 5 Principal Components

Figure 2a shows that 400 principal components cumulatively explain approximately 90.28% of the variation in the data which is a sizeable proportion of the variability. Additionally, to explain 85% of the variance, we need approximately 258 principal components.

We visualized the first 10 principal components (PC) to observe the dominant pattern in the data. These images show the most significant sources of the variability of the data, as shown in Figure 2b. The first 10 PC show various patterns, and displays the shape of MRI brain images, capturing different aspect of variation in the data. Figure 3a compared the original images to the reconstructed images using only 400 PC to assess how using fewer PCA components kept the important features of the images. This comparison images showed the trade-off between dimension size and image resolution. While the reconstructed images preserve the important features, there are some level of details that are missing from the original data.

Further analysis using PCA was conducted to show the pair-plot of the first 5 PC to see whether the data is separable in the lower dimension space or not. Figure 3b revealed that the data is not separable in lower dimensions, indicating that the data might not be efficiently captured by linear transformation providing guidance for selecting an appropriate models in the following tasks.

4 Experiments

In this section, we will discuss the various models tested for this problem. This includes some non-neural network based models, convolutional neural networks based on pre-trained models, as well as our own custom convolutional neural networks. Each section will present the details of the implementation and the results from the training. Discussion of the choices that were made, and the interpretation of the results, will also be provided.

4.1 Non-Neural Network based models

4.1.1 Implementation

Neural Networks models are generally considered best in class for many forms of image classification, especially in the medical field, however we would be amiss to immediately dismiss all other model architectures. We briefly explored a number of popular model types such as Support Vector Machines, Random Forests, and Gradient Boosting (List of models can be seen below). SVMs were investigated with several different kernels as it is well documented that certain SVMs are a close second to Neural Networks in image classification (Sun et al., 2015).

One benefit of these models is the configuration is greatly simplified when compared to Neural Networks, with generally only a handful of hyperparameters, compared to the highly customisable layers of neural nets. This allows for quicker experimentation at the cost of less flexibility so the models produced in this section may be generally good at classification but may not be the best in class for our specific problem. In these experiments, default hyper parameters are used unless specified.

4.1.2 Results

	Train accuracy	**Test accuracy**	**F1 Micro**	**F1 Macro**
XGBoost	95.21%	91.58%	0.916	0.878
Random Forest	94.11%	86.88%	0.869	0.756
Linear - SVM	94.57%	89.62%	0.896	0.909
Polynomial - SVM	93.98%	91.66%	0.916	0.911
RBF - SVM	93.94%	91.11%	0.911	0.907

Table 1: Model performance on the Alzheimer's classification data

The results show that for Alzheimer's classification, a SVM model with a polynomial kernel was the best out of the selection of models above. This matches the themes of (Sun et al., 2015) who states that SVMs are highly effective in many image classification applications. We can also observe considerable over fitting in all these models, with training accuracy being consistently larger than testing accuracy. This could be mitigated through the introduction and/or strengthening of each model's associated regularisation parameter. For example, in the case of the support vector machines, we could increase the C hyperparameter to increase regularisation and decrease over fitting. In summary, while not optimal for this problem, these non-neural network models can serve as a good baseline for the experiments to come.

4.2 Models based on pre-trained image classification models

We wanted to build a convolutional neural network to classify mri-scans to the correct degree of dementedness. To help us with this, and to save us valuable time and power consumption, we wanted to utilise a pre-trained classification network. Even though the pre-trained models are trained on other classes and images, we believe it will still be a good model to recognise basic shapes and relations.

Good candidates for such pre-trained models were the collection of EfficientNet models described by Tan and Le, 2020. These are 8 models with a differing number of parameters, and hence different capabilities. It classifies images to 1000 different output classes, and it's sizes are therefore very large even for the smaller models. We tested both `efficientnet_b0` and `efficientnet_b3`, which has `5.288.548` and `12.233.232` parameters, respectively.

To customize the model for our problem, we had to change the classifier to have the correct number of outputs. This meant that we had to do some additional training for the classifier.

We took two different approaches when implementing and training the pre-trained models. The first approach was to only train the classifier and fine-tune the weights of the pre-trained model. The second approach was to only train the classifier. The first approach is a prevalent transfer learning strategy, that gives both a general and task-specific network (Vrbančič and Podgorelec, 2020).

We are concerned about a limitation to the first approach. As the ImageNet dataset, which the EfficientNet models are trained on, differs significantly from the MRI scan dataset it might not be sufficient to train only the classification layer on top of the pre-trained model. The top-most layers of the pre-trained model will often output more domain-specific features (Vrbančič and Podgorelec, 2020). However, this information might not be sufficient for the classifier to correctly classify the MRI scans, because of the significant difference in domain.

4.2.1 Implementation of pre-trained model

We used the PyTorch framework to implement our models, and the pre-trained EfficientNet models are provided by the package `timm`. We implemented three different models shown in table 3, two based on the EfficientNetB0, and one on EfficientNetB3. One of the models based on EfficientNetB0 re-trained the already trained parameters in the whole network, instead of only the classifier as with the other models.

As for the hyperparameters, the Adam optimizer was chosen because of it's adaptive learning and position as a "jack of all trades" optimizer. The loss function used was cross-entropy loss which is suitable for multiclass classification problems. 15 epochs were used for training with a batch size of 64. For each epoch, the model was trained and the loss was backpropagated, with the optimizer used to update the model parameters. We had to provide a personalised classifier to the models, since the EfficientNet models classify to 1000 classes, and we only classified to 4 classes. The dropout layers were all set to randomly drop 30% of the outputs, to prevent overfitting. The architecture and hyperparameters are given in table 2. The slightly modified architecture for the detection problem is given in the appendix, table 14.

Architecture of classifier		
Layer	Output Shape	Num Param
Linear	(640)	819.840
ReLU	(640)	0
Dropout	(640)	0
Linear	(64)	41.024
ReLU	(64)	0
Dropout	(64)	0
Linear	(16)	1.040
ReLU	(16)	0
Dropout	(16)	0
Linear	(4)	68
Hyperparameters		
Learning Rate	0.001	
Loss function	Cross-entropy loss	
Optimizer	Adam	
Batch Size	64	
Epochs	15	

Table 2: Architecture of classifier and hyperparameters used on all models based on a pre-trained model.

Overview pre-trained models		
Model	Total param	Trainable param
EfficientNetB0	4.869.520	861.972
EfficientNetB0	4.869.520	4.869.520
EfficientNetB3	11.722.044	1.025.812

Table 3: Overview of models based on pre-trained EfficentNet models.

4.2.2 Results & Interpretation

The results for the three models based on pre-trained EfficientNet models are shown in table 4.

EfficientNet-b0 (weights frozen)	96.22%	73%	0.700	0.730
EfficientNet-b3 (weights frozen)	90.77%	66%	0.561	0.660
EfficientNet-b0 (not frozen)	97.41%	92%	0.887	0.916

Table 4: Model performance on Alzheimer classification data

We observed that the best model was the EfficientNetB0 with tuneable weights.

As talked about, we did not expect the frozen weight models to perform well, and this turned out to be true. Because of the high-level domain-specific output values from the pre-trained models, they did not translate well to our domain, without re-training the parameters in the EfficientNet itself.

Also, we see that the models with frozen weights have a high training accuracy. There can be many reasons for this overfitting, however it is likely due to the fact that the pre-trained model outputs features that that does not make sense in a MRI-domain. The model would then memorize distinct values from the model that maps to specific input values, and not actually learn the generalized patterns to recognize the classes. The confusion matrix for the best model are shown in table 5.

What we can see from the confusion matrix is that it very rarely misclassifies a demented input value as non-demented. These false negatives are very important to reduce when it comes to deceases, and our model manages this well.

		Reference		
Prediction	Non-Demented	Very Mild	Mild	Moderate
Non-Demented	655	12	0	0
Very Mild Demented	29	400	0	0
Mild Demented	40	25	105	1
Moderate Demented	0	1	0	12

Table 5: Confusion matrix for the ”EfficientNet-b0 with frozen weights” - on classification data

4.3 Customized CNN: AlzheimersNet

The next experiments that we conducted revolved around designing out own CNN architecture (which later will be referred as AlzheimersNet).

4.3.1 Architecture & Procedure

Table 6 shows the architecture of AlzheimersNet. We started with a simple architecture that consisted of three convolutional layers (each with 32 filters and kernel size of 3x3). ReLU activation functions were applied after each convolutional layer to introduce non-linearity which helps the model learn the complexity in the data. After the second and third convolutional layers, a 2x2 max pooling function is applied. We did not apply a pooling layer after the first convolution to avoid losing spatial information that contains important features. By keeping the full resolution after the first layer, the model can better capture and learn the basic features of the image, such as edges and fine details, which are essential for building a strong foundation for more complex feature extraction in subsequent layers. Finally, the output is flattened and passed through two fully connected layers. The first having 128 units, and the second 4 units, corresponding to the number of output classes for Alzheimer classificaiton problem.

Architecture of Multiclass Classifier		
Layer	Output Shape	Num Param
Conv2d	(64, 32, 126, 126)	320
ReLU	(64, 32, 126, 126)	0
Conv2d	(64, 32, 124, 124)	9,248
ReLU	(64, 32, 124, 124)	0
MaxPool2d	(64, 32, 62, 62)	0
Conv2d	(64, 32, 60, 60)	9,248
ReLU	(64, 32, 60, 60)	0
MaxPool2d	(64, 32, 30, 30)	0
Linear	(64, 128)	3,686,528
ReLU	(64, 128)	0
Linear	(64, 4)	516

Table 6: Multiclass Classifier CNN Architecture

The hyperparameters are the same as the ones described in 4.2.1. The training procedure follows the standard approach for the supervised learning with neural networks. The train loss and accuracy was computed and reported. The validation data was also used to compute the validation loss and accuracy on the validation data. These metrics were reported in each epoch to observe the model performance during training.

There were two experiments carried out with the AlzheimersNet. The first was Alzheimer's classification without weighting, the second was Alzheimer classification with class weighting to the target class in the cross-entropy criterion. The second experiment was done to mitigate the class imbalance of the dataset and better understand if models predictions were skewed.

4.3.2 Results & Interpretation

The model's performance on the multi-class Alzheimer classification was evaluated with and without assigning weights to the loss function to handle the class imbalance. The results are:

	Train accuracy	**Test accuracy**	**F1 Micro**	**F1 Macro**
Classification	100.00%	96.71%	0.970	0.967
Classification (weighted)	98.41%	97.70%	0.969	0.977

Table 7: AlzheimersNet Performance

		Reference		
Prediction	Non-Demented	Very Mild	Mild	Moderate
Non-Demented	649	18	0	0
Very Mild Demented	10	419	0	0
Mild Demented	0	9	162	0
Moderate Demented	0	1	0	12

Table 8: Confusion matrix of AlzheimersNet Classification Model

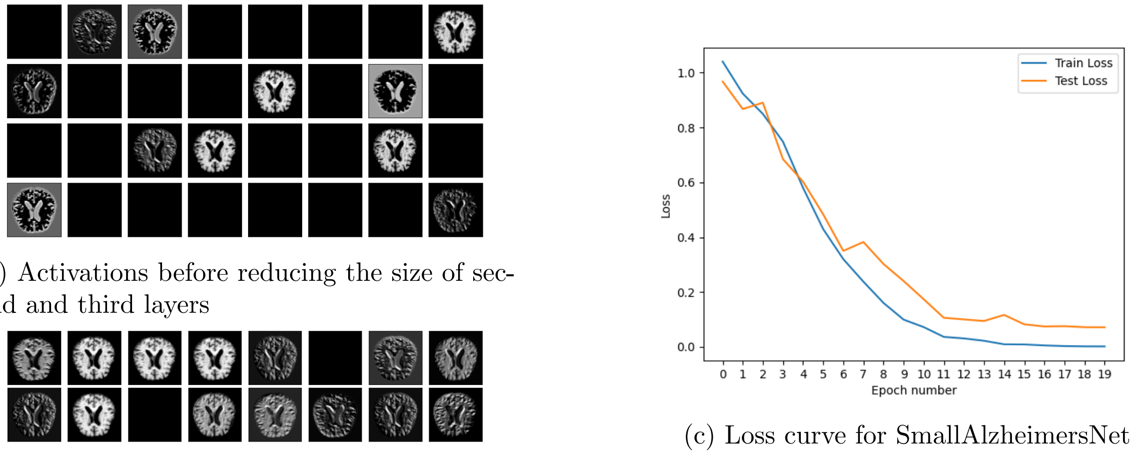
The results show that the AlzheimersNet performs well on both tasks. However, assigning weights to the cross entropy loss did not improve the F1 micro for the multiclass classification in this case. Although the overall test accuracy is slightly higher than unweighted CNN, this metric doesn't take into account the class imbalance of the data, therefore F1-micro is more appropriate metric to evaluate here as it takes into account the class imbalance.

One possible explanation for the lower performance could be suboptimal weights assigned to the cross entropy loss, so that it is unable to effectively handle the specific class imbalance in the dataset. While it may not directly improve the performance, there may be any other techniques to help deal with the class imbalance. For example, oversampling, undersampling, or ensemble methods.

The unweighted model have balance F1 macro and micro score, suggesting that the model is effectively distinguished between different stages in the Alzheimer and perform really well although the data is imbalanced.

The confusion matrix provided in table 8 shows that both the mild and moderate demented classes are all classified correctly, and only a few of the non demented and very mild demented images are misclassified. In the medical field, false negatives are commonly known as worse than having false positives thus the minimal false negatives is ideal for this problem.

Training and validation loss values were recorded for each epoch to understand the model's performance. The loss values tend to converge after around 10 epochs, this indicates that the model has effectively learned the underlying patterns within the data by this point. This convergence suggests that extending the training beyond 10 epochs does not significantly enhance the model's performance, thus making 15 epochs a suitable choice for training.



(a) Activations before reducing the size of second and third layers

(b) Activations after reducing the size of second and third layers

(c) Loss curve for SmallAlzheimersNet

Figure 4: Activations of the second convolutional layer state in SmallAlzheimersNet architecture and the loss curve from training the final model

5 Model Refinement

Focusing on our final AlzheimersNet model, we began hyperparameter tunning using 10-fold cross validation which was feasible due to the smaller size of the dataset. We first performed cross-validation on the classification accuracy beginning with the results shown below in Table 9. We also performed cross-validation for the detection model with results described in the appendix, but we focus on classification for reasons mentioned above. While further hyperparameters could have been tested the similarity in accuracy and f1-score between the final two tests suggests that it would have been ineffective, particularly as an accuracy of 99% was already achieved, and could potentially lead to overfitting. It was decided to use only 15 epochs as they achieved near identical accuracy and would reduce the risk of overfitting.

Learning rate	**Batch size**	**Epochs**	**Accuracy**	**F1 Micro**	**F1 Macro**
0.01	64	15	0.99	0.6884	0.6242
0.001	64	15	0.99	0.9891	0.9911
0.001	64	20	0.99	0.9892	0.9908

Table 9: Results of hyperparameter tuning using 10-fold cross-validation for own classification CNN

We then decided to view the activations of the hidden states for the classification model shown in Figure 4a and Figure 4b. Doing this revealed that a large amount of the convolution outputs in the second layer had close to zero outputs. This suggested that too many channels may have been used for the second and third layers. Indeed, a downsized model was tested with a reduced number of channels granting the results presented in Table 10. While we could have sought further improvements we decided against it for the reasons mentioned above. Looking at the hidden state activations for this model showed much fewer convolutions with values close to zero, while the model maintained its high accuracy. The loss curve shown in Figure 4c shows no to very little evidence of overfitting.

Learning rate	Batch size	Epochs	Accuracy	F1 Micro	F1 Macro
0.001	64	15	0.98	0.9847	0.9874
0.001	64	20	0.98	0.9844	0.9863
0.0005	64	20	0.99	0.9854	0.9873

Table 10: Results of hyperparameter tuning using 10-fold cross-validation for own classification CNN with reduced size of second and third convolutional layers

6 Model Evaluation and Comparison

6.1 Overall Results

	Train ACC	Test ACC	F1 Micro	F1 Macro
XGBoost	95.21%	91.58%	0.916	0.878
Random Forest	94.11%	86.88%	0.869	0.756
Linear - SVM	94.57%	89.62%	0.896	0.909
Polynomial - SVM	93.98%	91.66%	0.916	0.911
RBF - SVM	93.94%	91.11%	0.911	0.907
EfficientNet-b0 (weights frozen)	96.22%	73.00%	0.700	0.730
EfficientNet-b3 (weights frozen)	90.77%	66.00%	0.561	0.660
EfficientNet-b0 (not frozen)	97.41%	92.00%	0.887	0.916
AlzheimersNet	100.00%	97.00%	0.970	0.970
AlzheimersNet (weighted)	100.00%	94.00%	0.960	0.940
SmallAlzheimersNet	100.00%	99.45%	0.985	0.987

Table 11: Overall results of all experiments for the Alzheimer's classification (4 class) problem

6.2 Model Evaluation

When looking at the final results for the classification problem we can see that the non neural network models act as a good baseline as they are known to have good general performance over a range of problems but lack the fine grain customisation. The best of these models was a SVM with a polynomial kernel which achieved a testing accuracy of 91.66%.

This result was eclipsed by the 'pre-trained' CNN models which had much greater complexity allowing them to capture the intricacies of the data in much greater detail (Sharma, 2018). The best of these models was EfficientNet-b0 with weights unfrozen. The reason why the model with unfrozen weights preformed better is because it meant that the full set of parameters for the model could be trained to fit our data space, instead of just using the pre-trained weights that were trained on a different data set (Tan and Le, 2019).

Furthermore, when we customised our entire model architecture to the problem space by designing AlzheimersNet we saw another substantial increase in performance as this model architecture was a better fit for our dataset than the other prebuilt ones that were designed for just general images (not specifically brain MRIs). This model achieved a test accuracy of 97%.

After further optimisation the SmallAlzheimersNet achieved a cross-validated accuracy of 99.45%. This is comparable to the number one voted notebook on Kaggle for this data set (`https://www.kaggle.com/code/abdallahwagih/alzheimer-detection-efficientnetb3-acc-99`) which was only able to achieve a testing accuracy of 99.06%.

	Reference			
Prediction	Non-Demented	Very Mild	Mild	Moderate
Non-Demented	661	2	0	0
Very Mild Demented	6	427	0	0
Mild Demented	0	0	171	0
Moderate Demented	0	0	0	13

Table 12: Confusion matrix of SmallAlzheimersNet Classification Model

Table 12 is the confusion matrix for SmallAlzheimersNet and shows it had perfect classification for the mild and moderate classes (though this is likely due to the low number of samples in each) and only misclassified two very mild demented cases as non-demented. Six non-demented cases were classified as being very mild demented. Thus, this model presents a high overall accuracy combined with high class wise accuracies.

Particularly for medical tests it is important to consider class-wise accuracy. A large amount of samples incorrectly marked as not having Alzheimer's is concerning as it may lead to these individuals not seeking further aid. Meanwhile, a large amount in the other direction may prove expensive both financially and mentally due to the additional tests required or mental stress due to receiving a diagnosis. Our model seems able to avoid both of these pitfalls resulting in a model which can be used to effectively augment current medical diagnosis. Thus we can consider the aim of this report as a success as we were able develop an improved model to classify Alzheimer's disease.

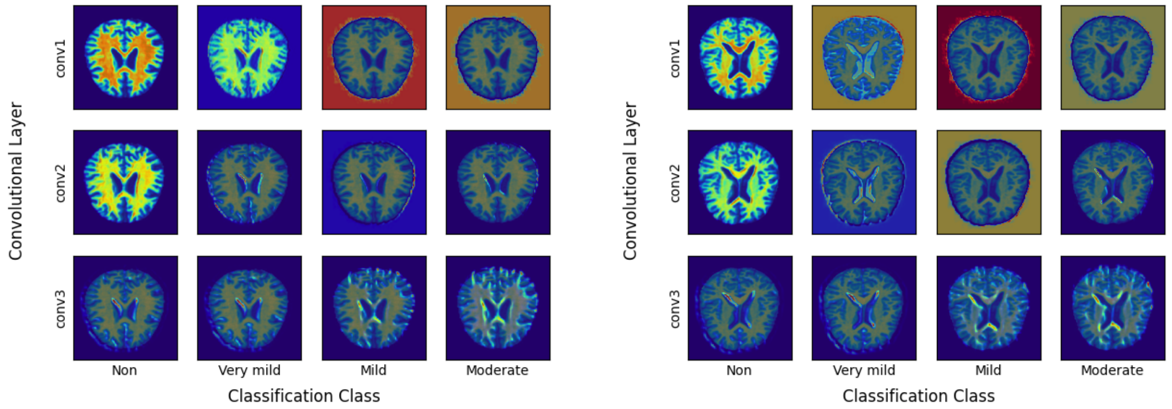
While not covered in depth throughout this report, the overall results for the detection based models can be found in the appendix, and will be discussed briefly now. A point of interest to note is that the accuracy's and metrics were considerably higher for the detection problem than the classification problem. This is expected, as the classification problem requires the learning of four separate classes and thus the complexity of the model is greatly increased, resulting in more complex decision boundaries (or similar). It should also be noted that the data distribution is quite uneven as seen in the EDA section of the report, thus, as some classes are under represented it may be harder for the model to get a good understanding of the smaller classes (e.g. moderately demented), impacting the overall accuracy.

7 Final Model Interpretation

We decided to investigate the architecture of SmallAlzheimersNet using grad-CAM. grad-CAM is a method which maps which regions of an image are "important" in determining what class an image is classified into (Selvaraju et al., 2019b).

Figure 5 shows this for two correctly classified examples one without and one with very mild Alzheimer's disease for all possible convolutional layer and classification combinations. Highlighted regions included the grey matter and the edges of the dura (outside of brain) and ventricle (the butterfly shaped region in the middle). An exception to this may be the layers where the background seems to have the largest effect, however an interpretation of this is it being used as a proxy for brain size. The dura and ventricle size are commonly used as a measure of atrophy and are used by human clinicians to diagnose Alzheimer's. Importantly. when constructing our architecture it was hypothesised that Maxpooling after the first CNN layer could be used to identify the major features which is confirmed by the grad-CAM results as for the first layer it highlights the larger structures present.

Interestingly, grad-CAM highlights several small regions while on other datasets it tends to highlight quite large regions. This suggests that there are specific regions which can be used



(a) Example of an MRI image without Alzheimer's Disease

(b) Example of an MRI image with very mild Alzheimer's Disease

Figure 5: gradCAM applied to very convolution layer and output class combination for Small-AlzheimersNet. Redder regions are deemed more important.

to generate an accurate classification. Alternatively, it could mark some form of overfitting as these small regions may be specific to our dataset, but this seems unlikely due to the high accuracies and f1-scores achieved during cross-validation.

8 Conclusion

We have tested a range of architectures with the goal of trying to accurately classify a the severity of Alzheimer's disease based upon MRI imagery. These models have had varying degrees of success, but overall we found the most effective to be our own customised architecture which achieved accuracies as high as 99%. We have also investigated what regions are deemed important by our network through the use of grad-CAM. Thus, we can consider this part of the project a success.

There are of course limitations and further pathway which can be explored. Namely, we have used 2D convolution while MRI images are truly 3D. We could further extend our model by using the full 3D data which we hope would help to further improve the our model. We also aren't surely about the demographics of our dataset and suspect that the majority may be older people. This makes us question the generalisability of our model and in future we would be interested in training it on a more general dataset. This leads into one of the most true problems for machine learning which is that we always desire more data and while our dataset was reasonably large we would like to have had access to a larger dataset. Additionally, there was some inconsistency in the training and testing datasets used across different models and architectures. Since, the work was conducted by multiple individuals, the data splits for training and testing were not uniform. Thus, while our model achieved very high accuracy we believe that there is still plenty room for improvement.

Overall, we believe we have been successful in developing an architecture to classify severity of Alzheimer's disease from MRI images. We imagine that something like this could be applied to as an augmentation to current medical diagnosis helping to improve both access to and consistency of care through the rollout of a relatively lightweight CNN. Particularly combined with tools like grad-CAM which is able to identify so called "important" regions we believe that this could be a valuable tool in the future of medical diagnosis.

Appendix

Appendix A: Alzheimer's detection problem

	Train accuracy	**Test accuracy**	**F1 Micro**	**F1 Macro**
XGBoost	95.98%	93.11%	0.931	0.931
Random Forest	96.11%	88.34%	0.883	0.883
Linear - SVM	96.72%	90.30%	0.902	0.902
Polynomial - SVM	95.51%	94.67%	0.946	0.946
RBF - SVM	95.90%	93.81%	0.938	0.938

Table 13: Non nural network model performance on the Alzheimer's detection data

Architecture of classifier		
Layer	Output Shape	Number of parameters
Linear	(640)	819.840
ReLU	(640)	0
Dropout	(640)	0
Linear	(64)	41.024
ReLU	(64)	0
Dropout	(64)	0
Linear	(16)	1.040
ReLU	(16)	0
Dropout	(16)	0
Linear	(2)	34
Hyperparameters		
Learning Rate	0.001	
Loss function	Binary Cross-entropy loss	
Optimizer	Adam	
Batch Size	64	
Epochs	15	

Table 14: Architecture for the output layers for the detector model and hyperparameters used on all models based on a pre-trained model.

Architecture of AlzheimerNet Detection		
Layer	Output Shape	Num Param
Conv2d	(64, 32, 126, 126)	320
ReLU	(64, 32, 126, 126)	0
Conv2d	(64, 32, 124, 124)	9,248
ReLU	(64, 32, 124, 124)	0
MaxPool2d	(64, 32, 62, 62)	0
Conv2d	(64, 32, 60, 60)	9,248
ReLU	(64, 32, 60, 60)	0
MaxPool2d	(64, 32, 30, 30)	0
Linear	(64, 128)	3,686,528
ReLU	(64, 128)	0
Linear	(64, 1)	129
Sigmoid	(64, 1)	0
Hyperparameters		
Learning Rate	0.001	
Loss function	Binary Cross-entropy loss	
Optimizer	Adam	
Batch Size	64	
Epochs	15	

Table 15: Architecture of AlzheimerNet Detection

	Train accuracy	**Test accuracy**	**F1 Micro**	**F1 Macro**
XGBoost	95.98%	93.11%	0.931	0.931
Random Forest	96.11%	88.34%	0.883	0.883
Linear - SVM	96.72%	90.30%	0.902	0.902
Polynomial - SVM	95.51%	94.67%	0.946	0.946
RBF - SVM	95.90%	93.81%	0.938	0.938
EfficientNet-b0 (weights frozen)	97.78%	81.00%	0.806	0.806
EfficientNet-b3 (weights frozen)	97.29%	76.00%	0.761	0.761
EfficientNet-b0 (not frozen)	99.58%	98.00%	0.984	0.984
Classification CNN	%	%		
Classification CNN with weight	%	%		
Smaller Classification CNN	%	%		

Table 16: Overall results of all experiments for the Alzheimer's detection (2 class) problem

Learning rate	**Batch size**	**Epochs**	**Accuracy**	**F1 Micro**	**F1 Macro**
0.001	64	15	0.98	0.9848	0.9848
0.0008	64	20	0.99	0.9901	0.9901
0.0005	64	20	0.99	0.9864	0.9864

Table 17: Results of hyperparameter tuning using 10-fold cross-validation for own detection CNN

	Train accuracy	**Test accuracy**	**F1 Micro**	**F1 Macro**
Classification	100.00%	96.71%	0.970	0.967
Classification (weighted)	98.41%	97.70%	0.969	0.977
Detection	99.34%	94.92%	0.949	0.949

Table 18: AlzheimersNet Performance

References

Aging, N. I. of (Apr. 2023). *Alzheimer's Disease Fact Sheet.* English. U.S. Department of Health and Human Services. URL: https://www.nia.nih.gov/health/alzheimers-and-dementia/alzheimers-disease-fact-sheet.

Ahmed, M. R. et al. (2019). "Neuroimaging and Machine Learning for Dementia Diagnosis: Recent Advancements and Future Prospects". In: *IEEE Reviews in Biomedical Engineering* 12, pp. 19–33. ISSN: 1941-1189. DOI: 10.1109/RBME.2018.2886237.

Biomedical Imaging, N. I. of and Bioengineering (Apr. 2022). *Magnetic Resonance Imaging (MRI).* English. U.S. Department of Health and Human Services. URL: https://www.nibib.nih.gov/sites/default/files/2022-05/Fact-Sheet-Magnetic-Resonance-Imaging-MRI.pdf.

Choi, H. (Nov. 2017). "Deep Learning in Nuclear Medicine and Molecular Imaging: Current Perspectives and Future Directions". In: *Nuclear Medicine and Molecular Imaging* 52.2, pp. 109–118. ISSN: 1869-3482. DOI: 10.1007/s13139-017-0504-7.

Fatima, I. (Jan. 2024). *AD-Early-Detection-using-Deep-Learning-Models.* Kaggle. URL: https://www.kaggle.com/code/iffat12/ad-early-detection-using-deep-learning-models.

Feng, W. et al. (May 2020). "Automated MRI-Based Deep Learning Model for Detection of Alzheimer's Disease Process". In: *International Journal of Neural Systems* 30.06, p. 2050032. ISSN: 1793-6462. DOI: 10.1142/S012906572050032X.

Kumar, S. and S. Shastri (2022). *Alzheimer MRI Preprocessed Dataset.* DOI: 10.34740/KAGGLE/DSV/3364939. URL: https://www.kaggle.com/dsv/3364939.

Moradi, E. et al. (2015). "Machine learning framework for early MRI-based Alzheimer's conversion prediction in MCI subjects". In: *NeuroImage* 104, pp. 398–412. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2014.10.002. URL: https://www.sciencedirect.com/science/article/pii/S1053811914008131.

Pini, L. et al. (Sept. 2016). "Brain atrophy in Alzheimer's Disease and aging". In: *Ageing Research Reviews* 30, pp. 25–48. ISSN: 1568-1637. DOI: 10.1016/j.arr.2016.01.002.

Qiu, S. et al. (May 2020). "Development and validation of an interpretable deep learning framework for Alzheimer's disease classification". In: *Brain* 143.6, pp. 1920–1933. ISSN: 1460-2156. DOI: 10.1093/brain/awaa137.

Reiman, E. M. and W. J. Jagust (June 2012). "Brain imaging in the study of Alzheimer's disease". In: *NeuroImage* 61.2, pp. 505–516. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2011.11.075.

Selvaraju, R. R. et al. (Oct. 2019a). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128.2, 336–359. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

— (Oct. 2019b). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128.2, pp. 336–359. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7.

Sharma, N. (2018). "An Analysis Of Convolutional Neural Networks For Image Classification". In: *Procedia Computer Science* 132. International Conference on Computational Intelligence and Data Science, pp. 377–384. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.05.198. URL: https://www.sciencedirect.com/science/article/pii/S1877050918309335.

Sun, X. et al. (2015). "Image classification via support vector machine". In: *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*. Vol. 01, pp. 485–489. DOI: 10.1109/ICCSNT.2015.7490795.

Tan, M. and Q. Le (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR, pp. 6105–6114.

Tan, M. and Q. V. Le (2020). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. arXiv: 1905.11946 [cs.LG].

Tauhidi, S. I. (Feb. 2024). *Efficient-AD::*$val_acc = 0.99$. Version 4. Kaggle. URL: https://www.kaggle.com/code/sunxyz/efficient-ad-val-acc-0-99/notebook.

Vrbančič, G. and V. Podgorelec (2020). "Transfer Learning With Adaptive Fine-Tuning". In: *IEEE Access* 8, pp. 196197–196211. DOI: 10.1109/ACCESS.2020.3034343.

Wang, S.-H. et al. (Mar. 2018). "Classification of Alzheimer's Disease Based on Eight-Layer Convolutional Neural Network with Leaky Rectified Linear Unit and Max Pooling". In: *Journal of Medical Systems* 42.5. ISSN: 1573-689X. DOI: 10.1007/s10916-018-0932-7.

Contribution

- Timothy Ryall: 20%
- Cooper Janke: 20%
- Nafis Riza: 20%
- Torstein Korten: 20%
- Paal Markus Bjoernstad: 20%

Acknowledgement

We give consent for this to be used as a teaching resource.