**CS 412 INTRODUCTION TO MACHINE LEARNING**

**Detection of Alzheimer’s using Deep Learning Models**

**Project Report**

**TEAM:**

**Kalyan Kumar Paladugula**

**(**[**kpalad4@uic.edu**](mailto:kpalad4@uic.edu)**)**

**Sudha Anusha Sagi**

**(**[**ssagi2@uic.edu**](mailto:ssagi2@uic.edu)**)**

**Vignesh Narayanaswamy**

**(**[**vnaray21@uic.edu**](mailto:vnaray21@uic.edu)**)**

**Contents:**

1. Abstract
2. Introduction

2.1 About Alzheimer’s

1. Dataset

3.1 3D MRI images data

3.2 Demographics data

1. Literature Review

4.1 Related Work

1. Data Analysis and Data Processing

5.1 Data Analysis – Demographics data

5.2 Data Analysis – MRI image data

1. TFRecords Creation
2. Deep Learning Models

7.1 Details of Architecture

7.2 3D Convolutional Neural Networks

7.3 3D Deep Convolutional Neural Network (Baseline)

7.4 Parallel Model

7.5 Mixed input model

1. Conclusions
2. Bibliography
3. **ABSTRACT**

Alzheimer’s disease, the most common form of dementia, is an irreversible, progressive brain disorder associated with permanent loss of memory and cognitive functioning. Mild Cognitive Impairment (MCI), another form of dementia, can be an early sign of Alzheimer’s. Diagnosing MCI from neuroimages is especially hard due to its similarity to AD. In this project, we explore different Deep Learning models that can clearly distinguish between patients who may be normal or have AD or MCI.

**2. INTRODUCTION**

**2.1 About Alzheimer’s:**

Alzheimer’s is the sixth-leading cause of death in the United States1. It is the only top-10 cause of death in the United States with no known cure. It is estimated the Alzheimer’s and other forms of dementia will cost $305 billion in 2020and the expenses are projected to reach $1.1 trillion (in 2020 dollars) by 2050. Currently, more than 5 million people in the US have Alzheimer’s and this number would reach 14 million by 20501. Someone in the US develops Alzheimer’s every 65 seconds2. By 2050 this is projected to go down to 33 seconds2. Alzheimer’s is not just a disease of old age: 200,000 people under age 65 have early-onset Alzheimer’s disease. It has a very high mortality rate: 1 in 3 seniors dies with Alzheimer’s or another dementia1. Although the mortality rates of the other major diseases decreased significantly, the deaths from Alzheimer's disease have increased significantly. Between 2000 and 2018, the number of recorded deaths from Alzheimer's increased by 246%, while the number of deaths from the number one cause of death (heart disease) decreased by 7.8%1.

AD is characterized by memory impairment, language dysfunction, and impairment of recognition, leading to AD patients having difficulty managing themselves with neurofibrillary tangles4. Though some treatments may temporarily suppress the symptoms of AD, there is still no powerful evidence that can tell the reason or can stop its progression5. As their condition gets worse, patients become more and more dependent on the help of others. This also puts a heavy burden on the caregiver, including social, psychological, physical, and economic factors.

Since MCI could be an early sign of AD, extensive research is being carried out to develop techniques for early detection at pre-symptomatic stages to slow or prevent the progression to AD. Advanced neuroimaging techniques, such as magnetic resonance imaging (MRI), positron emission tomography (PET), Diffusion Tensor Imaging (DTI), Cerebrospinal Fluid Flow (CSF) test, have been developed to identify structural and molecular biomarkers6. But their high dimensionality and multi-modality make these techniques difficult to analyze for humans6. Hence, machine learning approaches have attracted significant attention in the field of high dimensional image analysis6. Pattern analysis methods, such as linear discriminant analysis (LDA), Principal Component Analysis (PCA), logistic regression (LR), support vector machine (SVM), and support vector machine recursive feature elimination (SVM-RFE), have been used for early detection of AD and the prediction of AD progression6.

However, Machine Learning algorithms such as ones mentioned above require data preprocessing and feature extraction prior to application. Since those processes can be highly time-consuming, another family of machine learning called Deep Learning has become prominent due to their automated nature. These deep learning techniques extract the biomarkers (most important features) on their own for classification.

**3. DATASET**

The dataset we used contains two main parts of the data points.

1. 3D MRI images in the form of npy files (NumPy Arrays)

2. Demographics in the form of a csv file

The whole dataset has a total of 3013 rows of information where some subjects were scanned multiple times. These 3013 rows are again divided into training, validation, and test datasets. The training dataset has 2109 rows, the validation dataset has 435 rows while the test dataset has 469 rows of data.

|  |  |
| --- | --- |
| Dataset | Number of data points |
| Training | 2109 |
| Validation | 435 |
| Test | 469 |

Table: Dataset division for data modeling and testing

**3.1 3D MRI images:**

The MRI images are 3D images and each MRI image is represented in the form of 62 2D slices.

**3.2 Demographics:**

There are two main demographics values that are important in our dataset.

1. Age at Scan

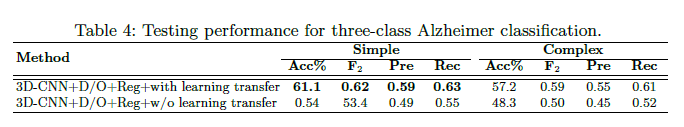
2. Sex

**4. LITERATURE REVIEW**

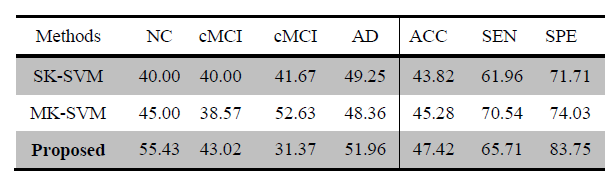
**4.1 Related Work:**

We divided the methods used by the researchers into two types: Pure Deep Learning, and Hybrid Deep Learning. In Pure Deep Learning, deep learning was used for both feature extraction and classification whereas in the Hybrid one, traditional machine learning models, like SVM, Logistic Regression, and Multi-Kernel SVM, were used for classification. The former one gave better results when used on large datasets and the latter one in case of small datasets.

Some researchers converted the multi-classification problem to binary classification problem and used transfer learning later7. They first built the models on only two classes: AD and NC. Then they added a new class MCI and fine-tuned the weights to accommodate this. This is possible because they used cross-entropy loss which could be extended to multi-class cases7. And they have shown that in case of limited training datasets, the models with the transfer learning strategy yielded better results than the ones that classify the input into three classes from scratch. As MCI is the intermediate stage between the cognitive decline of normal aging and the more pronounced decline of AD, first learning to separate the AD and the NC then adding the third class and fine-tuning the network transfers the learned knowledge to classify the middle condition, not jeopardizing the performance of AD Diagnosis7.

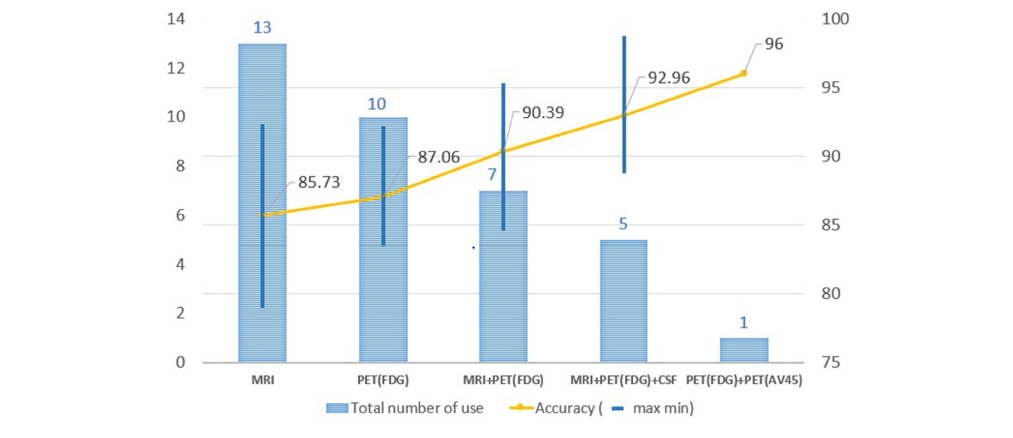
7

And some further split the MCI class into two: MCI non-converters (ncMCI) or MCI converters (cMCI) based on the risk of progression to AD. In [8], they converted the three-class problem to four-class problem (NC, ncMCI, cMCI, and AD) and applied Pure deep learning with stacked sparse auto-encoders and a softmax regression layer. But they got low accuracy.

8

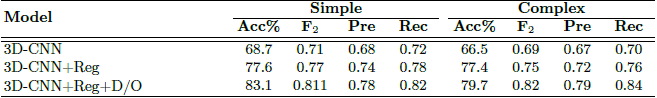
Few researchers combined the neuroimaging techniques (MRI+PET, MRI+CSF, MRI+PET+CSF) to improve the accuracy and F1 scores. In [8], they extracted the grey matter volumes from MRI and CMRGlc patterns from PET. And in [10], they used the fusion of MRI, PET, and CSF.

Two or more multimodal neuroimaging data types (MRI+PET, MRI+PET+CSF) resulted in better accuracy6.



Some researchers used the drop-out technique to avoid the “weight co-adaptation” problem which is a typical cause of overfitting in Deep Learning9. It is possible that only a few weights are involved in giving you the desired output. But in this scenario, you will also be looking at only part of the features in the input. This is called "overfitting" and is a result of co-adaptation. Your network has adapted to the dataset. Neural Networks show a gradual decay in performance rather than a complete failure when some of the neurons are disabled (or killed in the case of biological neurons). Basically, a subset of the connections is enough to give the desired output by the network.

To prevent this from happening, a solution is to introduce dropout: during training, disable (multiply by some value [0,1]) the output of some of the neurons. This makes sure the rest of the neurons are "forced" to do something or learn to classify the dataset. At every epoch, you dropout a random set of neurons with probability p (the only parameter of a dropout layer). This makes the remainder of the 1−p fraction of the neurons to also learn the feature set. Since you are no longer depending on only a few features or weights, you can escape the overfitting problem. The dropout technique improved the accuracy of the models7.

7

Some researchers used F1-Score instead of accuracy to improve the recall of AD and MCI classes. As it is difficult to differentiate AD and MCI classes, even if we get high accuracy, we might not get good accuracy on AD and MCI classes (Ex: 84% accuracy - 99% on NC, 70% on MCI, and 80% on AD). If the F1-score is used to pick the best model, we would get better recall scores on all the three classes.

**5. DATA ANALYSIS AND DATA PROCESSING**

**5.1 Data Analysis – Demographics data:**

The demographics dataset taken has five columns: train\_valid\_test, Age at Scan, data at scan, Sex, and diagnosis. The diagnosis column has three classes: CN (Cognitively Normal), MCI (Mild-Cognitive Impairment), and AD (Alzheimer’s disease).

The total dataset has a total of 3013 rows of information where some subjects were scanned multiple times. These rows are divided into training, validation, and test datasets using the train\_valid\_test column. We have performed data analysis on the training and validation datasets. The training dataset has 2109 rows while the validation dataset and test dataset have 435 and 469 observations respectively.

There are two main demographics values that are important in our datasets.

1. Age at Scan

2. Sex

We plotted the box plots of the distribution of age with respect to the Labels:

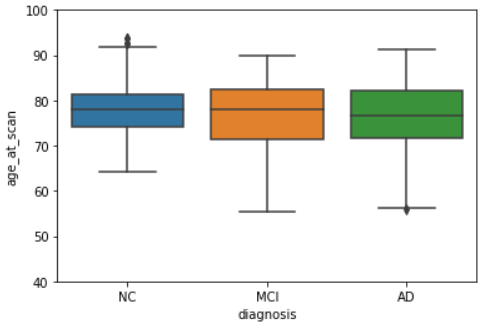


Figure 5.1: Box Plot of distribution of age with respect to the Labels in the Training dataset

In case of the Training dataset, the distributions of age for different labels are almost similar. So, age is not an important feature for the classification.

We then conducted an analysis to see the distribution of the subjects according to the labels i.e., CN, MCI, and AD. The results were as follows shown in figure 5.2.

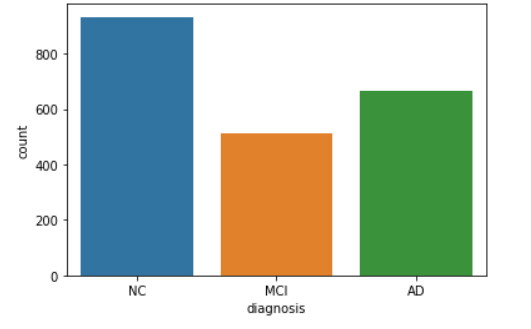


Figure 5.2 Distribution of the 3 class labels among the Training and Validation sets.

The classes are not balanced in the training data; there are more observations with NC class. So, we will choose the best model that has a higher F1 Score.

Further considering the gender into consideration, we found the following observations:

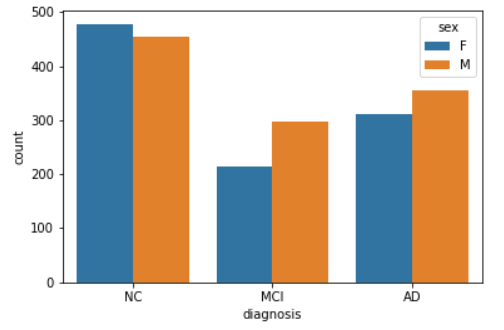


Figure 5.3: Distribution of 3 class labels among the Training dataset with respect to gender.

In case of observations with NC class, there is not much difference in the male and female ratio but for MCI and AD classes there is a noticeable difference.

Our data analysis on demographics has shown that gender feature is important for classification and age is not that important.

**5.2 Data Analysis – MRI images:**

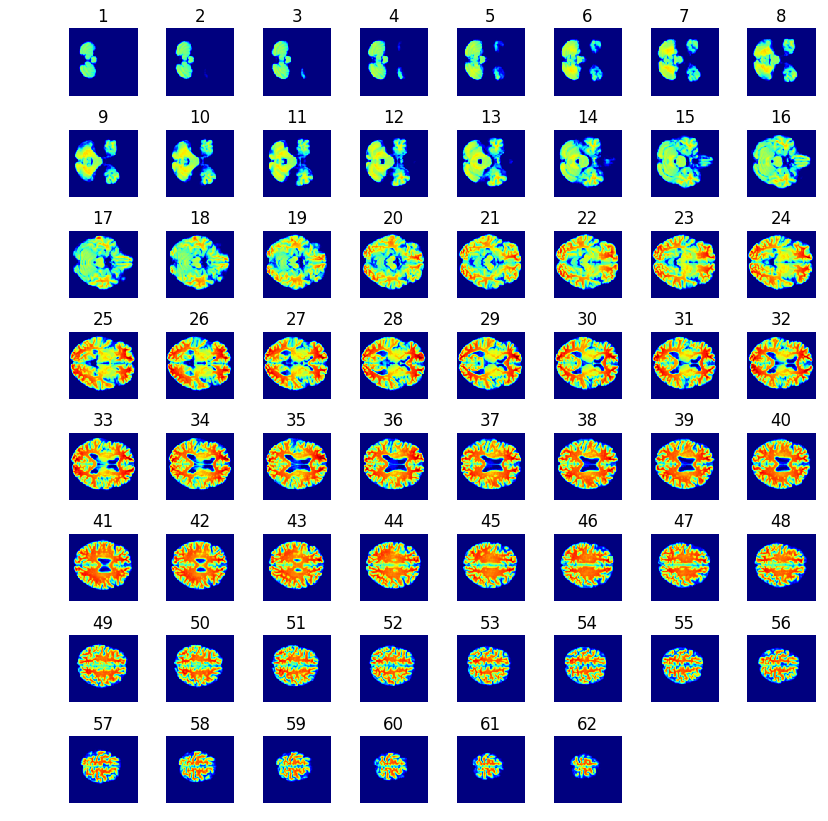
The dataset has a total of 3013 data points which includes all the subjects: CN(Clinically Normal), MCI(Mild Cognitive impairment), and AD(Alzheimer’s disease). The MRI images are 3D images and these MRI images are represented in the form of 2D slices. So, each subject has 62 2D brain slices corresponding to one MRI image. Consecutive 62 images are considered as one 3D image.

Figure 5.6: 2D slices of a 3D MRI image of the subject [0]. Each slice represents a section of the brain image cut horizontally11.

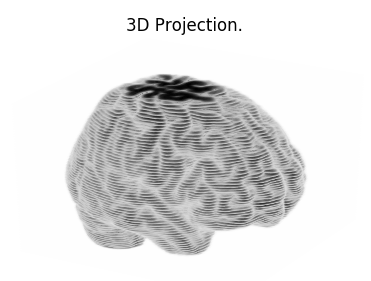
So, similarly, every set of 62 elements of npy files are mapped to the respective subject in the same order. From figure 5.6 we can identify that each 2D slice is a part of 3D MRI image which is sectioned horizontally. This way it provides a clear view of every part in the image which makes it easier to feed the data to a neural network. The 3D projection of the same MRI 2D slices for the same subject[0] can be viewed as the

Figure 5.7: 3D projection image of the subject[0]

**6. TFRecords Creation**

The MRI data folder has multiple training, valid, and test npy files. We combined all the training npy files into a training dataset, valid npy files into a valid dataset, and similarly for the test dataset. Then, we added the labels column to these datasets by mapping these datasets to the diagnosis column of the corresponding training, valid, and test demographics datasets.

Then, to reduce the running time, we converted the new datasets into TFRecords.

**7. DEEP LEARNING MODELS**

There are several pre-trained models including InceptionNets, ResNets, and VGG models that perform very well for specific datasets. However, two major limitations of such models did not allow us to utilize them for our own project.

1. They are usually pre-trained on some common image datasets like ImageNet and not on medical neuroimages.
2. They are constrained to 2D images rather than 3D

We also tested several recent 3D model architectures such as VoxNets and PointCNNs. These are state-of-the-art models for object recognition from 3D point clouds. But they were less efficient in performing our specific classification task which is more complicated and nuanced than simple object recognition.

This pushed us to design our own novel architectures that performed surprisingly well despite their simplicity relative to the models discussed above. In the forthcoming subsections, we discuss our models in greater detail.

**7.1 Details of the Architecture:**

Before we discuss the models, we would like to go over the “what” and “why” for each layer used in our model architectures.

*Input block:*

Accepts input of dimension 62x96x96x1 and feeds it to convolutional blocks.

*3D Convolutional blocks:*

Contain ‘n’ filters of size 3x3 that perform three-dimensional convolutions with a stride of 2 on the input. The value for ‘n’ varies with each model. Having convolutions with a stride of 2 allowed us to not include pooling blocks like MaxPool and AveragePool that are not capable of preserving details in the data.

*BatchNormalization:*

Used in conjunction with convolutional blocks to regularize outputs and reduce the need for explicit Dropouts.

*Activations:*

For all our convolutional blocks as well as Dense layers, LeakyReLU activations are used. For deep networks, it is paramount that gradients flow to each and every layer so the model trains properly. LeakyReLU activations have this property and can significantly reduce the risk of vanishing and exploding gradients. For our output, we have used Softmax activation as it more closely resembles a probability distribution.

*Flatten Blocks:*

Converts the output of convolutional blocks to a vector so it can be fed to a fully connected layer

*Dense blocks:*

Capture details from the output of convolutional layers and introduce further non-linearity.

*Dropout layers:*

Used with Dense layers to “dropout” a fraction of the outputs to reduce the chances of overfitting.

*Output block:*

Outputs a 1x3 vector specifying the probability that given input belongs to each of the three classes.

**7.2 3D Convolutional Neural Network:**

A 3D CNN[10]  remains regardless of what we say a CNN that is very much similar to 2D CNN. In a 3D CNN, the kernels move through three dimensions of data (height, length, and depth) and produce 3D activation maps.

The 3D convolutions are the generalization of the 2D convolution. Here in 3D convolution, the filter depth is smaller than the input layer depth (kernel size < channel size). As a result, the 3D filter can move in all 3-direction (height, width, breadth of the image). At each position, the element-wise multiplication and addition provide one number. Since the filter slides through a 3D space, the output numbers are arranged in a 3D space as well. The output is then 3D data.

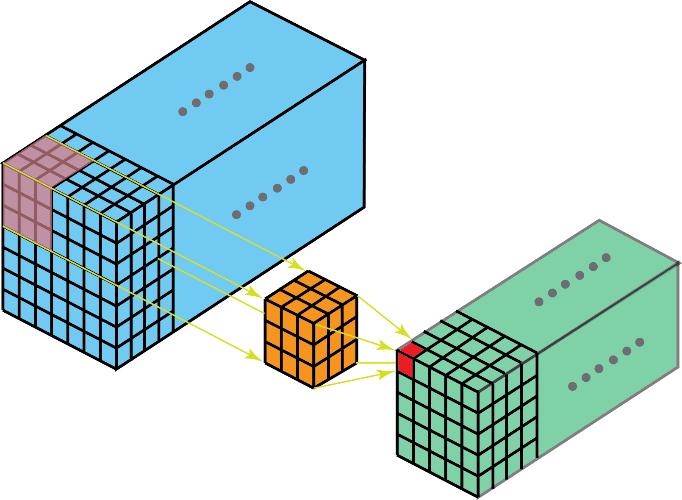


Figure 7.1: In 3D convolution, a 3D filter can move in all 3-direction (height, width, channel of the image). At each position, the element-wise multiplication and addition provide one number. Since the filter slides through a 3D space, the output numbers are arranged i*n a 3D space as well. The output is then a 3D data.*

Similar to 2D convolutions that encode spatial relationships of objects in a 2D domain, 3D convolutions can describe the spatial relationships of objects in the 3D space. Such a 3D relationship is important for some applications, such as in 3D segmentations/reconstructions of biomedical imagining, e.g. CT and MRI where objects meander around in the 3D space.

3D CNN varies from 2D CNN as follows:

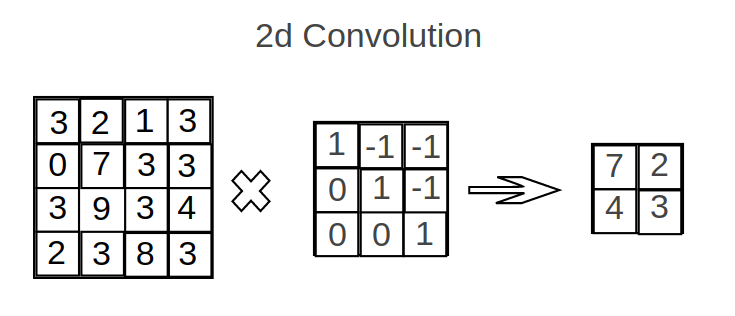
1. 3D Convolutional Layers:

Figure 7.2 2D Convolution

Originally a 2d Convolution Layer[10] is an inner product (entry per entry multiplication) of the input and the filters, where filters and inputs are 2d matrices. In a 3d Convolution Layer, the same operations are used. We do these operations on multiple pairs of 2d matrices. Padding options and slides step options work the same way.

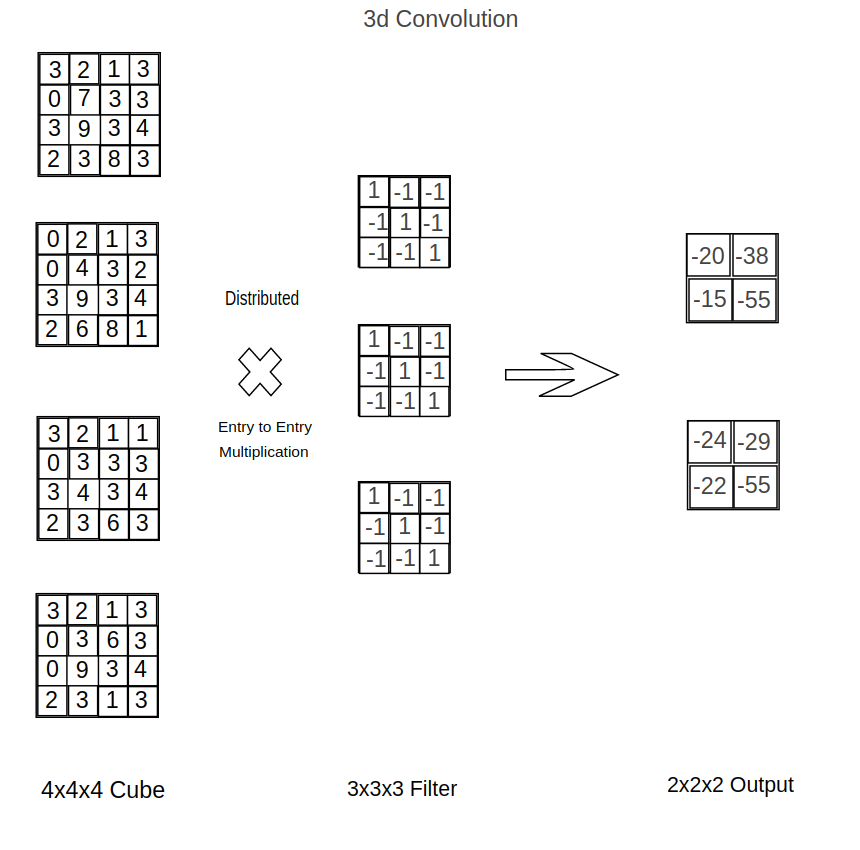
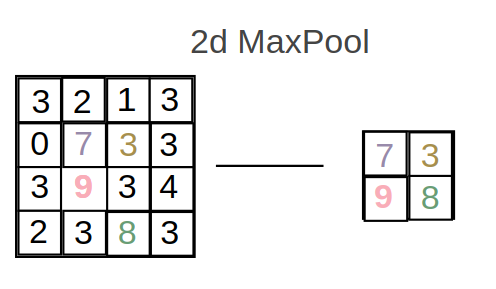


Figure 7.3 3D Convolution

2d Maxpool Layers[10] (2x2 filter) is about taking the maximum element of a small 2x2 square that we delimitate from the input. Now in a 3d Maxpool (2x2x2), we look for the maximum element in a width 2 cube. This cube represents the space delimited by the 2x2x2 zone from the input.

****

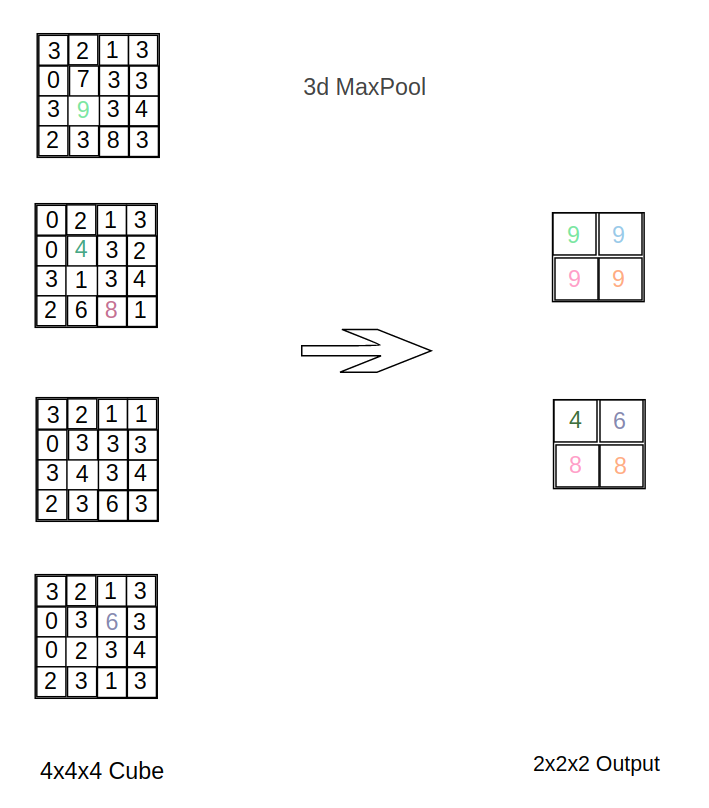


Figure 7.4 3D Max pool

**7.3** **3D Deep Convolutional Neural Network (Baseline):**

As a baseline to aid performance comparisons we created a straight-forward deep convolutional network capable of operating on three-dimensional images with multiple image channels. Visualization of the model architecture is given in figure 7.5.

From figure 7.5 we can observe that the architecture starts with 62 x 96 x 96 x 1 input layer nodes and reduces to 30 x 30 x 47 x 8 nodes after its first 3D convolutional layer and its normalization. Similarly, it reduces to 6 x 11 x 11 x 8 after 3rd batch normalization. Finally, after some more intermediate layers, the final output we get is in the form of 3 nodes corresponding to CN, MCI, and AD.

For more details about the architecture, please go through Figure 6.5 which shows a detailed design of the layers and nodes.

The scores for the Baseline 3D deep convolutional neural network are as follows,

We can see that the model was able to detect NC with almost close to 87% accuracy. The model finds difficulty in separating MCI and AD patients and this is because there are a lot of similarities in the critical regions of MRI images of patients with MCI and AD.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Normal Control | Mild Cognitive Impairment | Alzheimer’s | Average |
| Accuracy | 87.4% | 60% | 77.77% | 74.66% |
| F1 Scores | 83.23% | 64.56% | 74.6% | 74.1% |
| Recall | 87.42% | 58.59% | 77.77% | 74.6% |

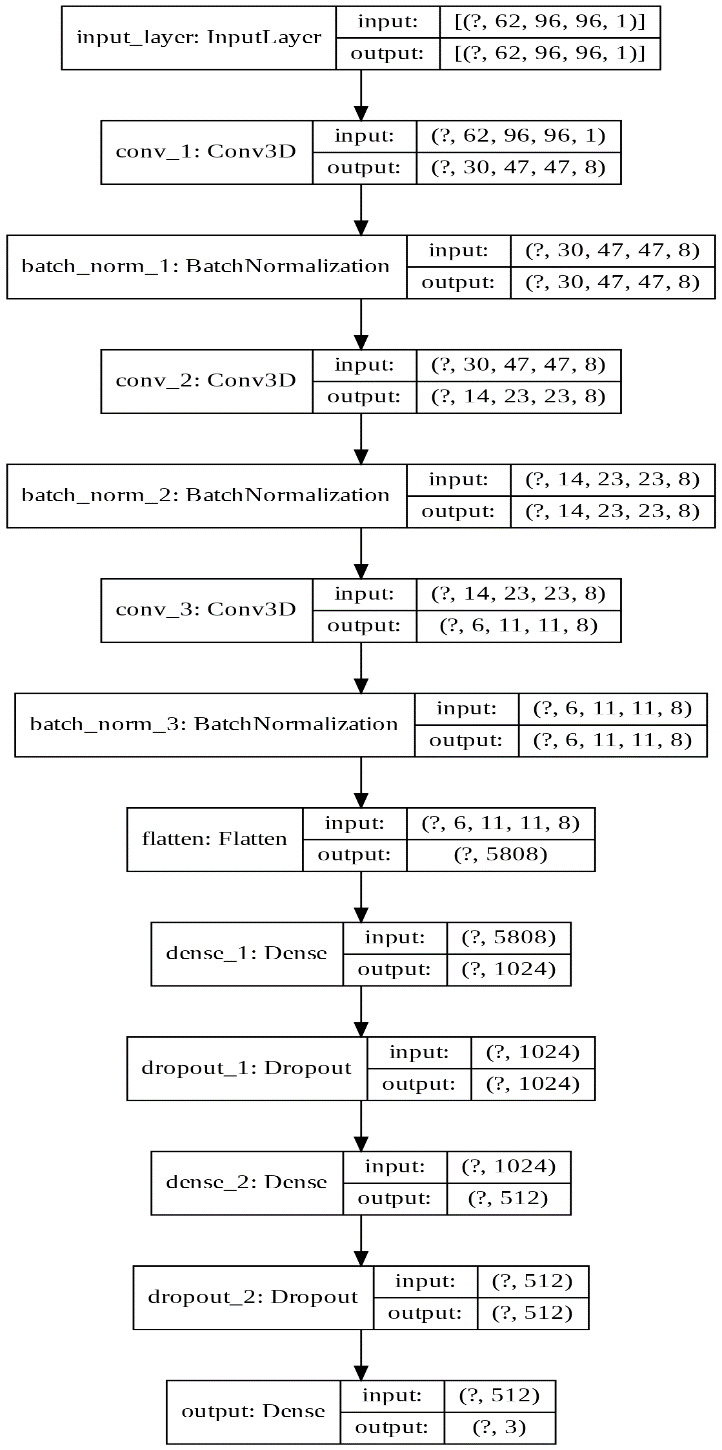


Figure 7.5 3D Deep Convolutional Neural Network – Baseline model

**7.4 Parallel model:**

This architecture uses parallel Convolutional neural networks. From figure 7.6 we can observe that the architecture separates out into two branches that are working parallelly. The architecture starts with 62 x 96 x 96 x 1 input layer nodes and reduces to 30 x 30 x 47 x 8 nodes after its first 3D convolutional layer and its normalization in both the parallel states. Similarly, it reduces to 14 x 23 x 23 x 16 after 2nd batch normalization in both the parallel states. Then we concatenate the two branches and merge the layers together. Finally, after some more intermediate layers, the final output we get is in the form of 3 nodes corresponding to CN, MCI, and AD. For more details about the architecture, please go through Figure 6.6 which shows a detailed design of the layers and nodes.

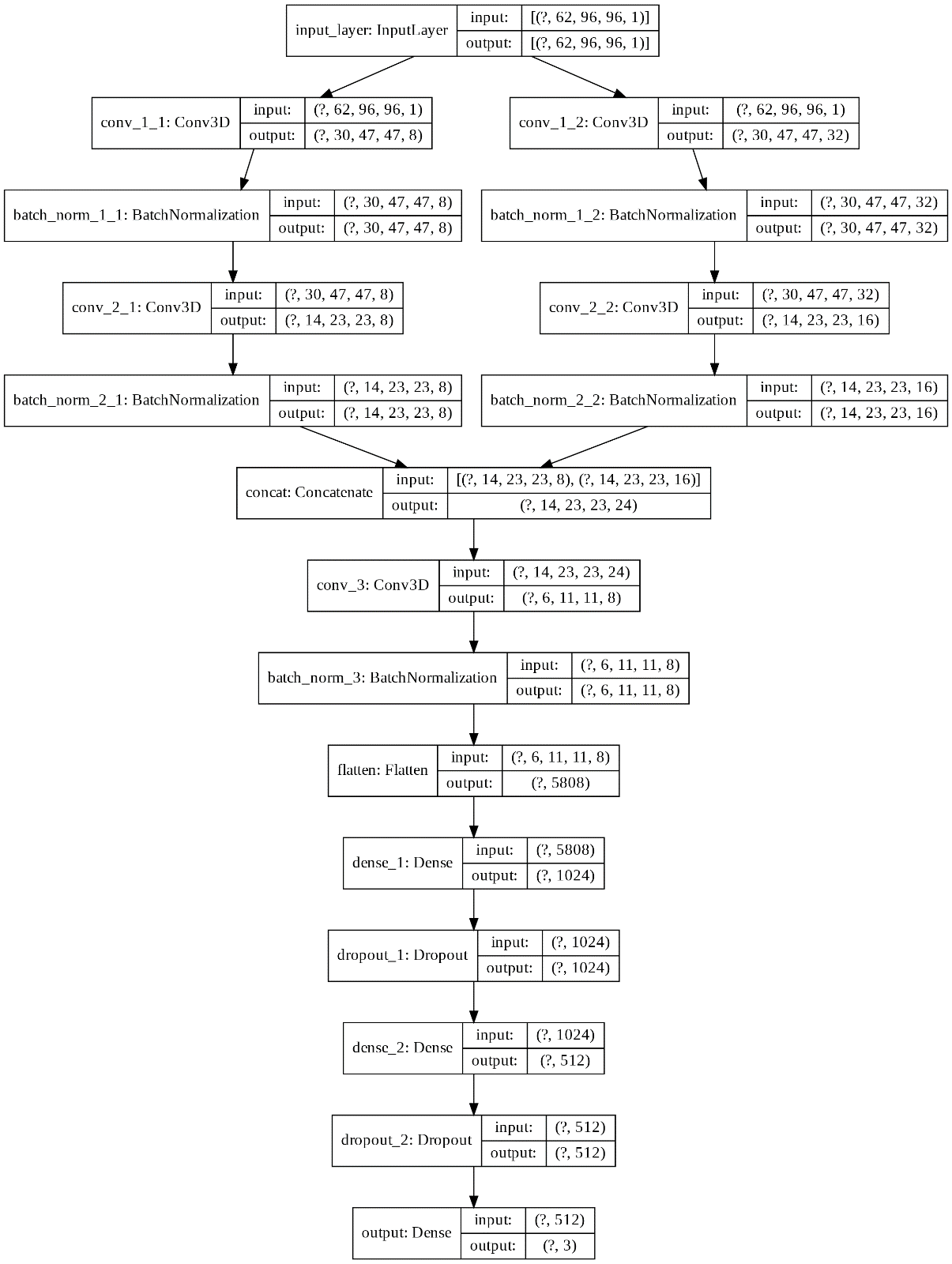


Figure 7.6 Parallel model

The scores for the parallel model are as follows,

We can see that the total accuracy of the model has increased by 1.4% in this parallel model. We can also see that the model was able to detect NC with almost close to 90% accuracy. This model detects NC patients better than our previous baseline model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Normal Control | Mild Cognitive Impairment | Alzheimer’s | Average |
| Accuracy | 92.5% | 61.2% | 74.2% | 76% |
| F1 Scores | 84.24% | 67.14% | 72.5% | 74.6% |
| Recall | 92.45% | 60% | 73.8% | 74.6% |

**7.5 Mixed Input model:**

This architecture uses a mixed input model.

From figure 7.7 we can observe that the architecture starts with 62 x 96 x 96 x 1 input layer nodes and reduces to 14 x 23 x 23 x 16 after 2nd batch normalization. During this stage, a second network is introduced starting with one layer and 2 nodes. After some iterations, the two branches concatenate and merge the layers together. Finally, after some more intermediate layers, the final output we get is in the form of 3 nodes corresponding to CN, MCI, and AD.

For more details about the architecture, please go through Figure 7.7 which shows a detailed design of the layers and nodes.

A close up of text on a white background

Description automatically generated

Figure 7.7 Mixed Input model

The scores for the mixed input model are as follows,

We can see that the total accuracy of the model is almost like the previous model, but this model is able to predict patients with AD better than the previous model. We can also see that the accuracy of this model for the Normal Control class is lower than that of the previous model. Overall, the total accuracy is close to the previous model, but we can find variations within the classes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Normal Control | Mild Cognitive Impairment | Alzheimer’s | Average |
| Accuracy | 86.8% | 59.2% | 80.4% | 75.87% |
| F1 Scores | 82.13% | 64.53% | 75.92% | 74.1% |
| Recall | 86.79% | 57.96% | 80.3% | 75.04% |

**7. CONCLUSION**

Since the percentage of subjects with no dementia (NC) is quite higher than that of other classes i.e. NC is majority class in the training dataset, any model built on this imbalanced data would be biased towards the NC class so did our models. We got more than 85% accuracy for NC class with all the three models and accuracies for the other two classes have not crossed 81%. We selected the best model based on the F1 score, which is robust to the class-imbalance problem. The mean F1 scores for the Baseline, Parallel, and Mixed input models are 74.1, 74.6, and 74.1, respectively. Due to the small difference in the F1 Scores, we considered the average accuracy scores: 74.66, 76, and 75.87. The Parallel model, which has two branches of CNNs, is the best model in terms of both mean F1 score and mean accuracy.

The mediocre results of our models can be attributed to CNN’s vulnerability to overfit and the MRI scans. Alternatively, the most successful approach has been to use a 3D Auto-Encoder as mentioned in the paper12. One can also try using 3D Convolutional Auto-Encoders, in which each convolutional layer is an Auto Encoder.

There is another approach called “Transfer Learning” in which the multi-classification problems are converted into a binary classification problem and built models on that and increase the neurons in the output layer later. This approach has shown good promise7.

Another approach is to use “Hybrid” Deep learning in which Deep learning is only used for feature extraction. Traditional machine learning models like SVM, are used for the classification. This approach has proven to be very effective in case of small datasets9.

The research paper6 has a summary of all the models used by several research papers. According to it, the models that used a fusion of more than one neuroimaging data (like MRI+PET, and MRI+PET+CSF) gave more accuracy than the ones that used only one neuroimaging data like us. This might be due to the better features obtained by combining the multiple neuroimaging data.

**8. List of Packages downloaded for the Project**

*Downloaded Packages:*

TensorFlow

*We used the following packages for the project:*

OS

Numpy

Pandas

Matplotlib

Sklearn

TensorFlow

**9. BIBLIOGRAPHY**

1. <https://www.alz.org/alzheimers-dementia/facts-figures#quickFacts>
2. Alzheimer’s Association: [2019 Alzheimer’s Disease Facts and Figures](https://www.alz.org/alzheimers-dementia/facts-figures" \t "_blank)
3. [Auto-Detection of Alzheimer's Disease Using Deep Convolutional Neural Networks](https://ieeexplore.ieee.org/document/8687207)
4. DaoqiangZhang, DinggangShena, and The Alzheimer's Disease Neuroimaging Initiative, “Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease, ” Neuroimage, Volume 59, Issue 2, Pages 895-907,16 January 2012.
5. Manhua Liu, Daoqiang Zhang, Dinggang Shen, and The Alzheimer's Disease Neuroimaging Initiative, “Ensemble sparse classification of Alzheimer's disease,” Neuroimage, vol. 60, 1106–1116, January 2012.
6. [Deep Learning in Alzheimer’s Disease: Diagnostic Classification and Prognostic Prediction Using Neuroimaging Data](https://www.frontiersin.org/articles/10.3389/fnagi.2019.00220/full)
7. [End-To-End Alzheimer’s Disease Diagnosis and Biomarker Identification - Soheil Esmaeilzadeh, Dimitrios Ioannis Belivanis, Kilian M. Pohl, and Ehsan Adeli](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjU77-UxZPpAhVaCs0KHY4EABIQFjAAegQIBhAB&url=https%3A%2F%2Fweb.stanford.edu%2F~eadeli%2Fpublications%2FMICCAI_DeepAD_Final.pdf&usg=AOvVaw2U7Hssm5i4NqA8r3uqECXl)
8. [Early diagnosis of Alzheimer's disease with deep learning](https://ieeexplore.ieee.org/document/6868045)
9. [A Robust Deep Model for Improved Classification of AD/MCI Patients](https://ieeexplore.ieee.org/document/7101222)
10. [3D Convolutional Neural Networks - Towards Data Science](https://towardsdatascience.com/step-by-step-implementation-3d-convolutional-neural-network-in-keras-12efbdd7b130)
11. [Classifying Alzheimer’s MRI scans using 2D and 3D convolutional neural networks - Bouke Regnerus and Raoul Fasel](https://github.com/regnerus/keras-alzheimers-3d-conv/files/2111208/main.pdf)
12. [Predicting Alzheimer’s disease: a neuroimaging study with 3D convolutional neural networks](https://arxiv.org/pdf/1502.02506.pdf)