**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. Project 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. Deep Learning Models

6.1 Details of Architecture

6.2 3D Convolutional Neural Networks

6.3 3D Deep Convolutional Neural Network (Baseline)

6.4 Parallel Model

6.5 Mixed input model

1. Conclusions
2. Bibliography

**1. PROJECT 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 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 put 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**

Our dataset primarily contains 2D brain segments stored as numpy array (.npy) files and some basic demographics information available in .csv format.

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 train, validation and test to train the models. The train dataset has 2109 rows, the validation dataset has 435 rows while the test dataset has 469 rows of data.

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

Table : Dataset division for data modelling and testing

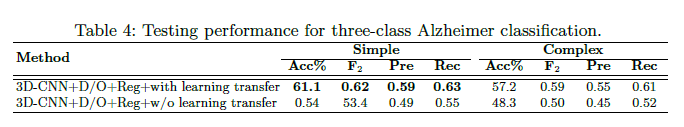
**4. LITERATURE REVIEW**

**4.1 Related Work:**

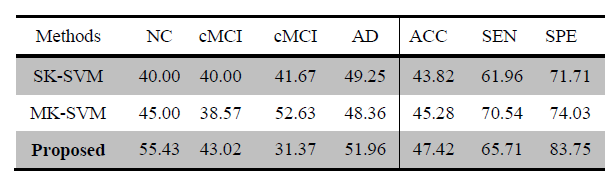
We divided the methods used by the researchers into two types: Pure Deep Learning, and Hybrid Learning. In the Pure Deep Learning, deep learning was used for both feature extraction and classification whereas in the Hybrid one, traditional machine learning models, like SVM, Logistic Regression and Multi-Kernel SVM, were used for classification. The former method 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 trained the models on only two classes: AD and NC.

Then after training the classifier with two classes, 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 the 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.

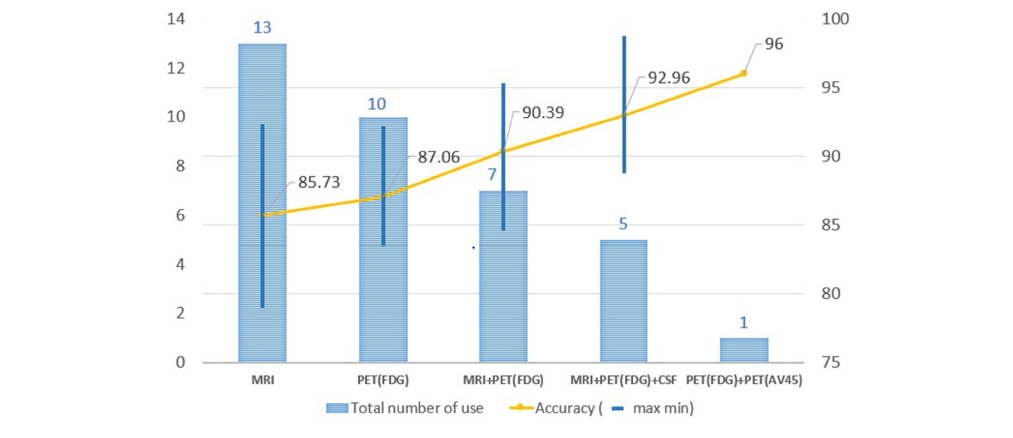
7

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.

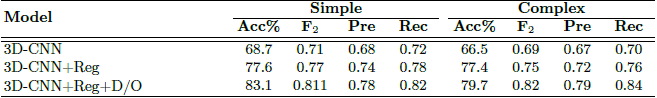
Two or more multimodal neuroimaging data types (MRI+PET, MRI+PET+CSF) resulted in better accuracy6.



Some researchers used 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 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 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. ANALYSIS AND DATA PROCESSING**

**5.1 Data Analysis – Demographics data:**

The dataset taken has three classifications which are CN (Cognitively Normal), MCI (Mild-Cognitive Impairment) and AD (Alzheimer’s disease). The dataset is divided into train, validation and test with respective labels to identify them. We have performed analysis on the data from train and validation sets.

The MRI dataset of all the subjects is in the form of npy (NumPy Arrays) files and the other demographics like age, gender etc. are included in a separate file with the classification labels.

The total dataset has a total of 3013 rows of information where some subjects were scanned multiple times. These 3013 rows are again divided into train, validation and test to train the models. The train dataset has 2109 rows while the validation set has 435 rows with almost equal ratio of labels.

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

1. Age at Scan

2. Sex

Using the idea from [11] we’ve implemented few plots as follows. The number of male and female subjects in the train and validation datasets are as follows:

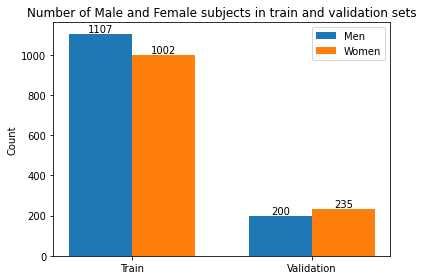


Figure 5.1: Distribution of Male and Females subjects in train and validation datasets.

|  |  |  |
| --- | --- | --- |
| ***GENDER*** | ***TRAIN*** | ***VALIDATION*** |
| ***MEN*** | 1107 | 200 |
| ***WOMEN*** | 1002 | 235 |

We have also conducted some analysis on the age of the subjects among the train and validation sets and they are as follows,

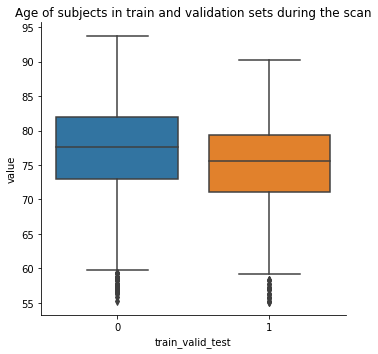


Figure 5.2: Train and Validation dataset – Distribution of subjects in train and validation sets with respect to their ages.

The highest age of the subject in train dataset is 94 and 90 for the validation set while 55 was the lowest age for both the datasets. Even though the plots show ages below 60 and 59 as outliers, we will not consider this as ages between 55 and 59 are still valid numbers.

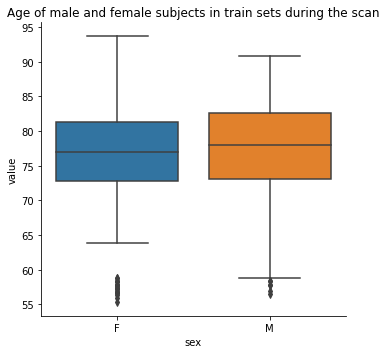
Further if we investigate the distribution on ages with respect to gender, the data is as follows shown in figure 5.3 and 5.4.

Figure 5.3: Train dataset – Distribution of Male and Females subjects in train dataset with respect to their age.

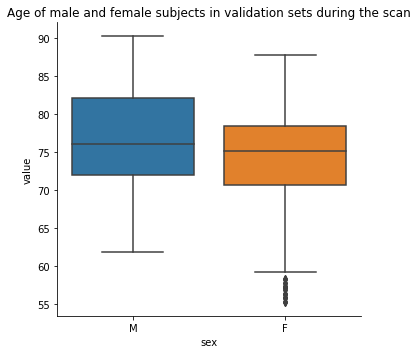


Figure 5.4: Validation dataset – Distribution of Male and Females subjects in validation dataset with respect to their age.

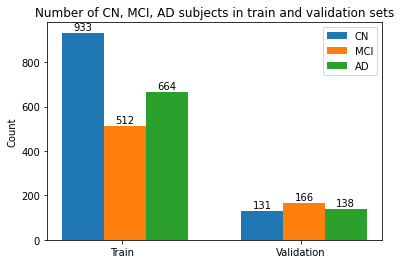
We then conducted some analyis to see the distribution of the subjects according to the labels i.e., CN, MCI and AD. The results were as follows shown in figure 5.5.

Figure5.5 Distribution of the 3 class labels among the train and validation sets.

Even though the distribution is not exactly the same among the classes in both the sets, the validation set has equal amount of the three classes which can be helpful for our model building.

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

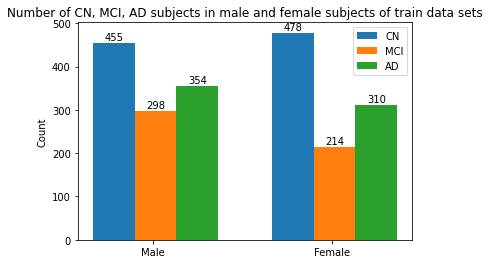


Figure 5.6: Distribution of 3 class labels among the

train dataset with respect to gender.

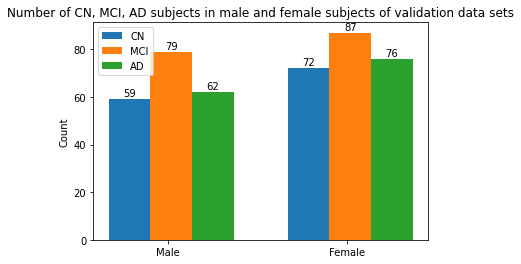


Figure 5.7: Distribution of 3 class

labels among the validation dataset with respect to gender.

Our data analysis on demographics gave us a good idea about how the disease was majorly divided among the people related to parameters like age and sex.

**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).

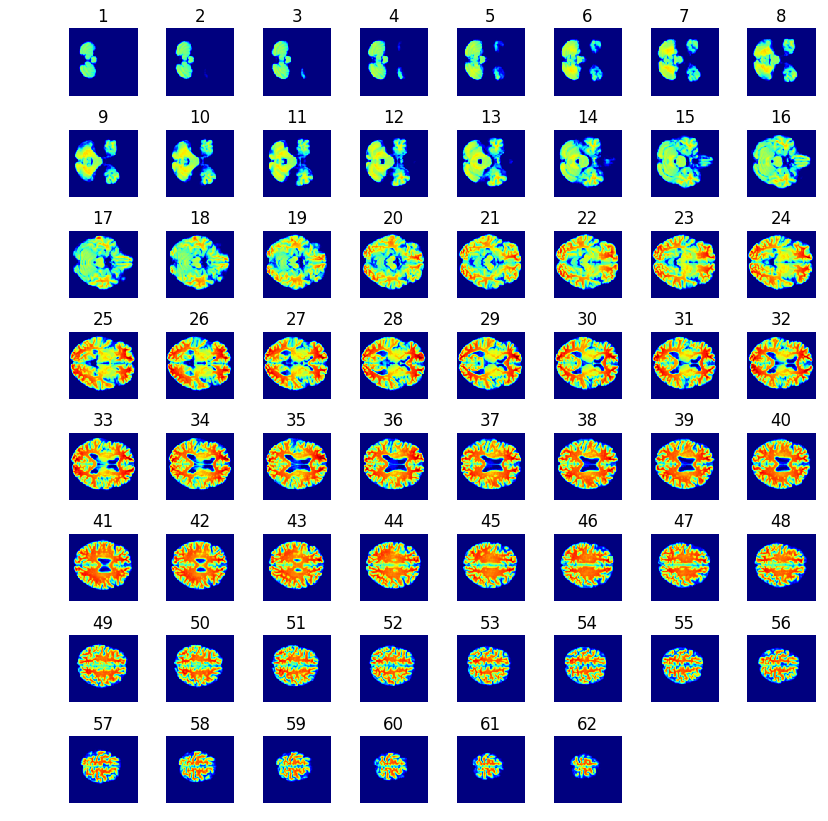
The MRI images are 3D images and these MRI images are represented in the form of 2D slices. So, each subject has 62 2D brain slices corresponding to one MRI image. The MRI images of each subject are received from npy files that is as “NumPy Arrays”. Consecutive set of 0 to 62 images are considered at a time as one 3D image. So, similarly every set of 62 elements of npy files are mapped to respective subject in the same order.

Figure 5.8: 2D slices of 3D MRI image of subject[0].

Each slice represents a section of the brain image cut horizontally.

Figure 5.8 shows the 62 2D brain slices of 3D MRI image corresponding to subject[0] i.e., the first subject in our data set which gives more detail about how the data is seggregated.

From figure 5.8 we can identify that each 2D slice is a part of 3D MRI image which is sectioned horizontally. This way it provides are clear view of every part in the image which makes it easier to feed the data to a neural network. Also, from the analysis we found out that the images from 1 to 4 and images from 59 to 62 do not explain the critical regions and they can be eliminated to reduce the data set if needed.

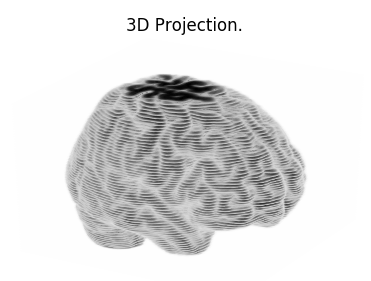
The 3D projection of the same MRI 2D slices for the same subject[0] can be viewed as the

Figure 5.9: 3D projection image of subject[0]

**6. DEEP LEARNING MODELS**

There are several pretrained 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 in spite of their simplicity relative to the models discussed above. In the forthcoming subsections, we discuss about our models in greater detail.

**6.1 Details of Architecture:**

Before we discuss about the models, we would like to go over some core ideas and some “what”s and “why”s of the building blocks of our models.

**6.2 3D Convolutional Neural Network:**

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

Naturally, there are 3D convolutions. They 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, channel of the image). At each position, the element-wise multiplication and addition produce 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 a 3D data.

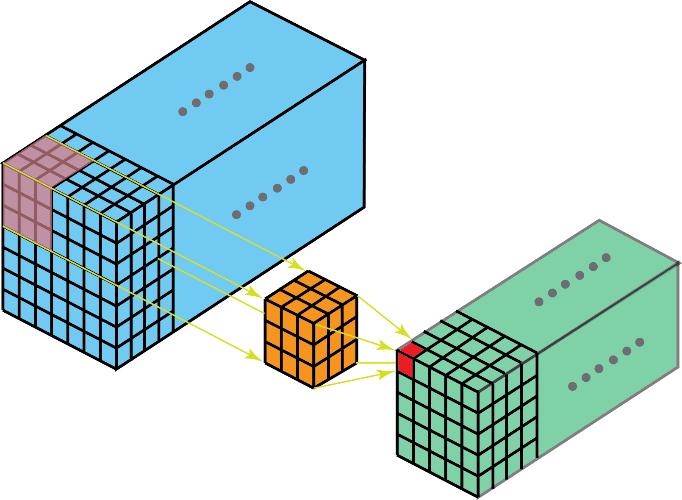


Figure 6.1: In 3D convolution, a 3D filter can move in all 3-direction (height, width, channel of the image). At each position, the element-wise multiplication and addition provide one number. Since the filter slides through a 3D space, the output numbers are arranged in a 3D space as well. The output is then a 3D data.

Like 2D convolutions which encode spatial relationships of objects in a 2D domain, 3D convolutions can describe the spatial relationships of objects in the 3D space. Such 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. Blow are the basic building blocks.

**Input block***:*

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

**3D Convolutional Blocks**:

Originally a 2d Convolution Layer[10] is an entry per entry multiplication between the input and the different filters, where filters and inputs are 2d matrices.

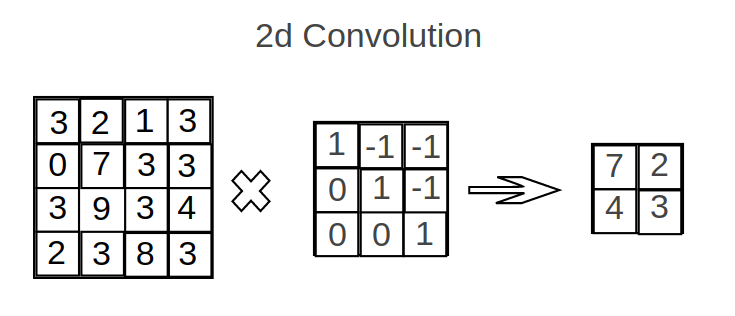
In a 3d Convolution Layer, the same operations are used. We do these operations on multiple pairs of 2d matrices. Padding options and slides step options work the same way.

Figure 6.2 2D Convolution

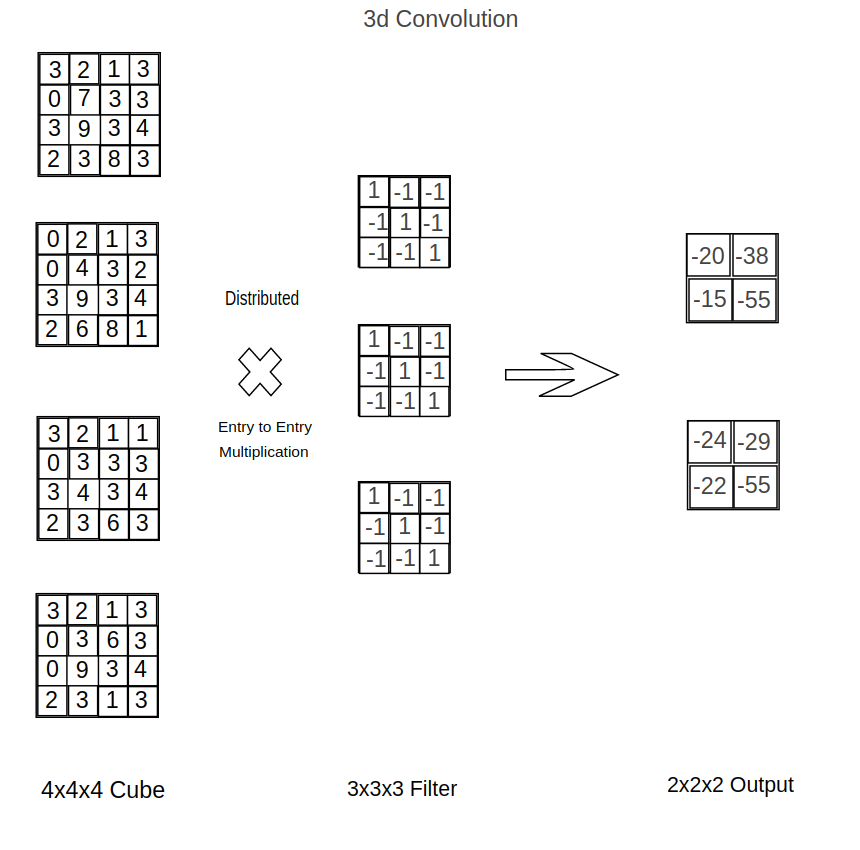


Figure 6.3 3D Convolution

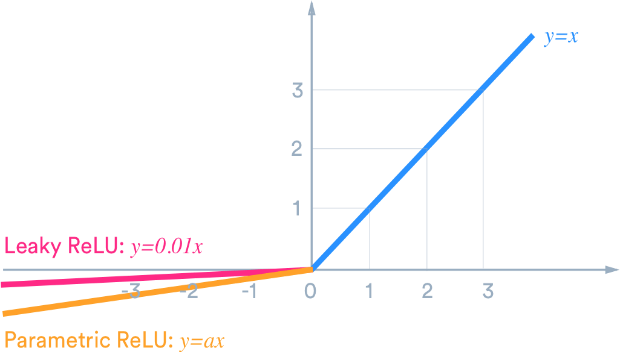
The convolutional blocks in out models contain ‘n’ filters of size 3x3 that perform three dimensional 2-strided convolutions on the input. The value for ‘n’ varies with each model. Having 2-strided convolutions allowed us to not include pooling blocks like MaxPool and AveragePool that are not capable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

**BatchNormalization***:*

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

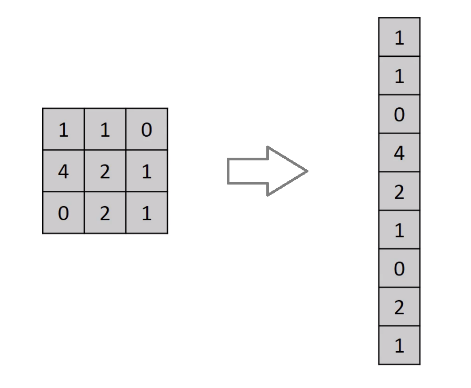
**Activations***:*

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.



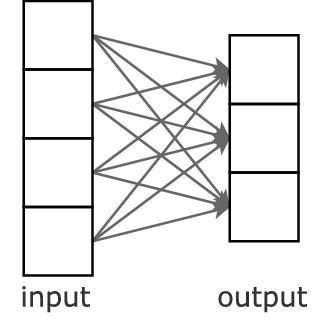
**Flatten Blocks***:*

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



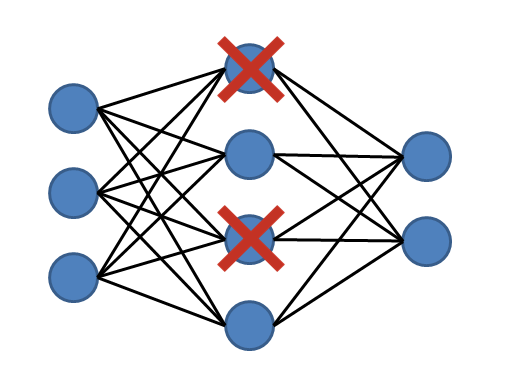
**Dense blocks***:*

Captures details from 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.

**6.3** **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 6.5.

From figure 6.5 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 the Figure 6.5 which shows a detailed design of the layers and nodes.

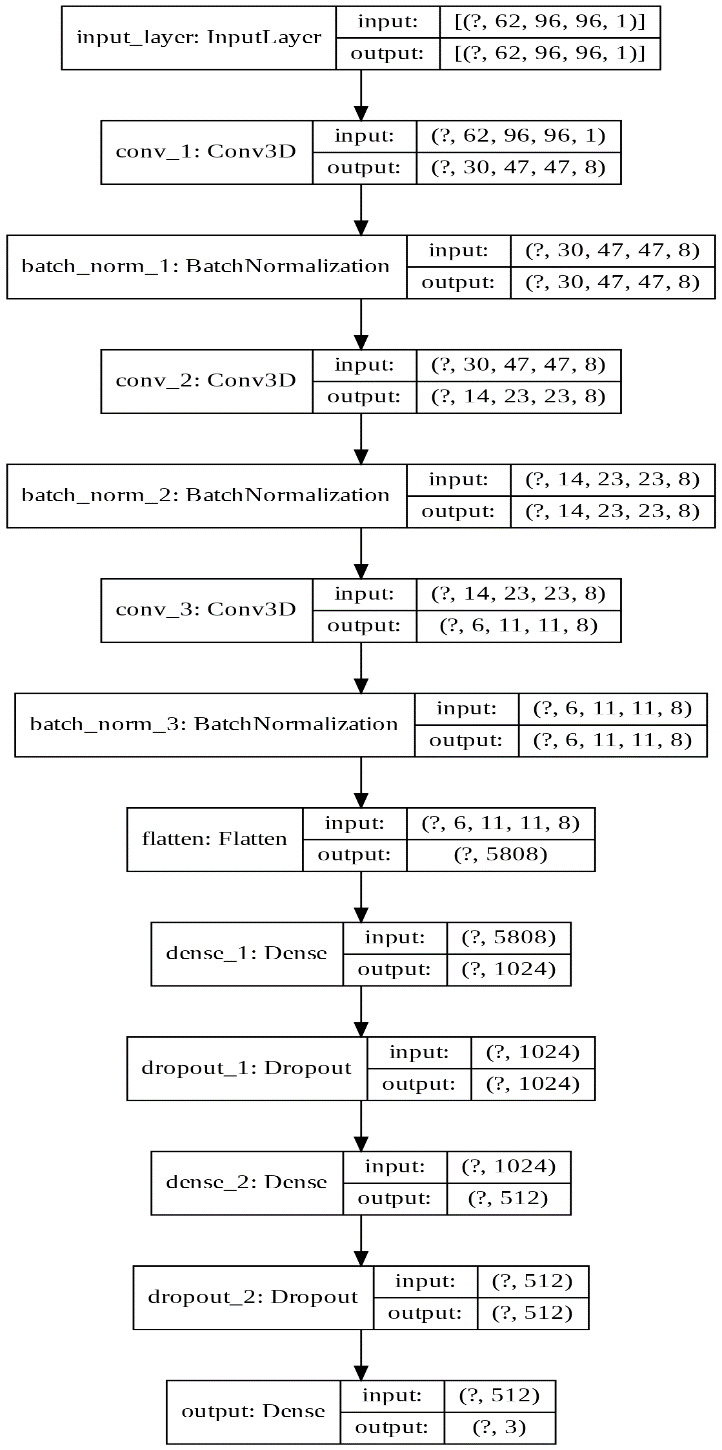


Figure6.5 3D Deep Convolutional Neural Network – Baseline model

The scores for the 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 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% |

**6.4 Parallel model:**

This architecture uses parallel Convolutional neural networks.

From figure 6.6 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 the Figure 6.6 which shows a detailed design of the layers and nodes.

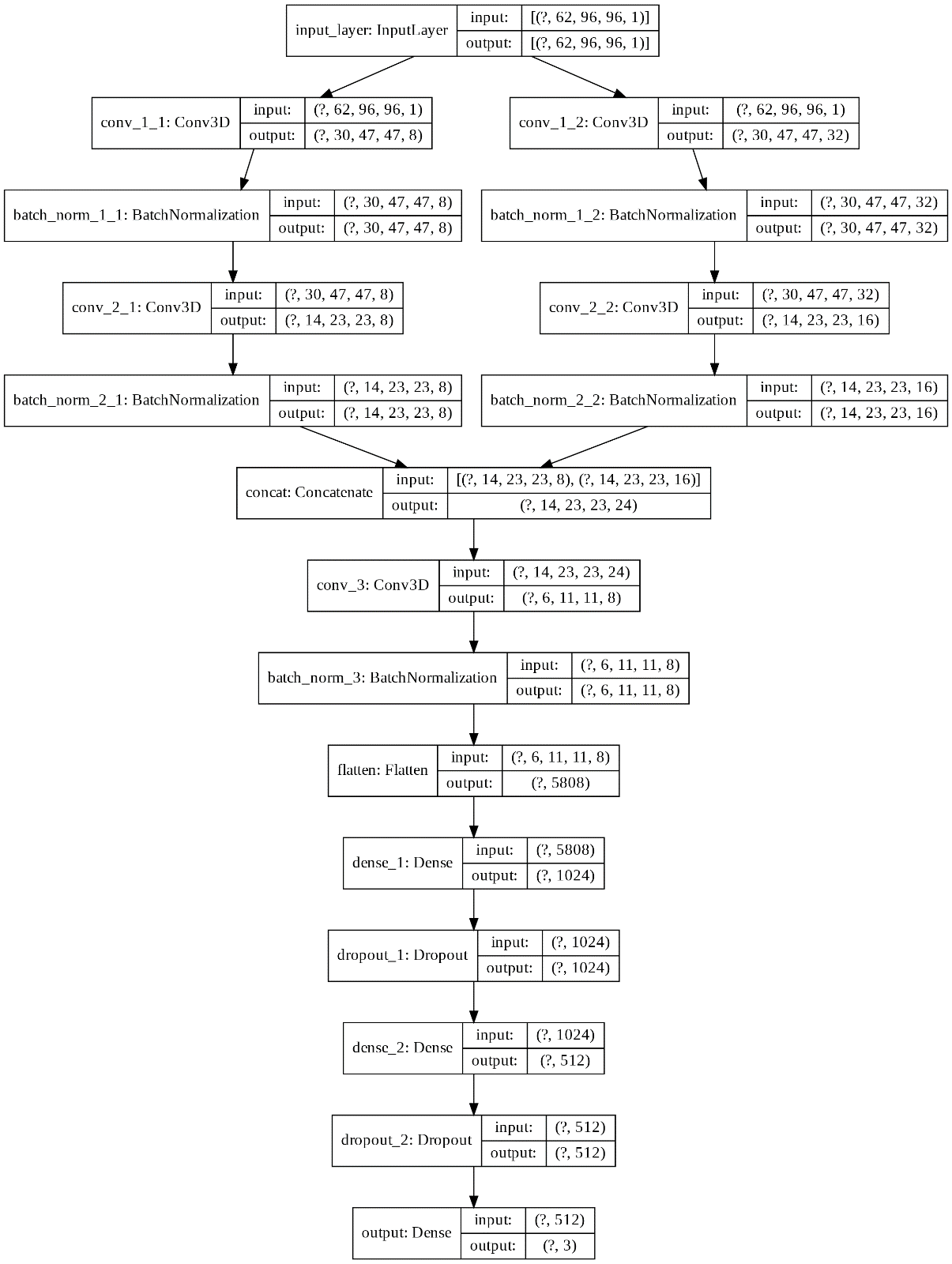


Figure 6.6 Parallel model

The scores for 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 the 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% |

**6.5 Mixed Input model:**

This architecture uses mixed input model.

From figure 6.7 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 the Figure 6.7 which shows a detailed design of the layers and nodes.

A close up of text on a white background

Description automatically generated

Figure 6.7 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 similar to the previous model, but this model is able to predict patients with AD better than the previous model. We can also see that this model does not predict NC patients better. 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 | 83.13% | 64.53% | 75.92% | 74.6% |
| Recall | 86.79% | 57.96% | 80.3% | 75.04% |

**7. CONCLUSIONS**

**8. BIBILIOGRAPHY**

1. <https://www.alz.org/alzheimers-dementia/facts-figures#quickFacts>
2. Alzheimer’s Association: [2017 Alzheimer’s Disease Facts and Figures](https://www.alz.org/documents_custom/2017-facts-and-figures.pdf)
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. [Bar charts in Matplotlib - Grouped Bar charts with labels](https://matplotlib.org/3.2.1/gallery/lines_bars_and_markers/barchart.html%23sphx-glr-gallery-lines-bars-and-markers-barchart-py)