Alzheimer's disease classification based on ANN, CNN, and Transfer Learning using magnetic resonance imaging

Md Younus Ahamed
Lane Department of Computer Science and Electrical Engineering
West Virginia University, Morgantown, USA

***Abstract*—Alzheimer's disease is a brain disorder that slowly demolishes memory function, ultimately fatal. The major problem here is only medical experts can interpret these. Human-made classification or detection of these diseases is tedious and remarkably prone to error. There is a limited number of medical experts, and the results may differ from person to person based on experience. We can use Machine Learning (ML) or Deep Learning (DL) to overcome this problem. In this report, the performance of different deep learning approaches for classifying Alzheimer's is shown. CNN outperformed ANN. The custom CNN surpassed the pre-trained networks (ex: VGG16, Inception V3, etc.). The ANN gave 81.09% test accuracy, whereas the CNN reached 92.93% test accuracy. The pre-trained model VGG16, Inception v3, and ResNet50 showed 87.89% 81.17% 75.31% test accuracy.**

***Index Terms*—Alzheimer's, Deep Learning, MLP, CNN, VGG16, Inception V3, ResNet50, Transfer Learning.**

I. INTRODUCTION

The Healthcare sector is different from other industries. It is a high-priority sector, and people expect the highest care and services regardless of cost [18]. The World Health Organization recognizes Alzheimer's disease (A.D.) as a global public health priority [3]. It is the U.S.'s sixth leading cause of death [15]. Alzheimer's disease (A.D.) is the most common form of dementia, and its prevalence is increasing [7]. Detection of this disease is very subjective. Medical experts can only do it. It is a critical problem, especially in rural areas where finding an expert pathologist is challenging. Nowadays, a vast amount of image data is available. Deep learning, a state-of-the-art machine learning approach, has shown outstanding performance over traditional machine learning in identifying intricate structures in complex high-dimensional data, especially in computer vision. The application of deep learning to early detection and automated classification of Alzheimer's disease (A.D.) has recently gained considerable attention [11]. CNN is best for image classification; it gives very high accuracy that sometimes surpasses human-level accuracy. In this work, a comparison of the performance among Multilayer perceptron (MLP), Convolutional Neural Networks (CNN), and Transfer Learning were made. For transfer learning, VGG16, Inception v3, and ResNet50 were used. The Dataset was publicly available, and it was collected from Kaggle. The Dataset contained four imbalanced classes. An oversampling technique called SMOTE was used to solve the imbalance problem. Then an MLP was trained with the oversampled images. The performance of a deep neural network depends on many hyperparameters. For this MLP, the list of significant hyperparameters was the number of hidden layers, nodes in each layer, number of epochs, and batch sizes. A hyperparameter tuning was done to find the optimal hyperparameters. Then a deep CNN was trained with the MRI images. Besides the number of hidden layers, nodes in each layer, number of epochs, and batch size, deep CNN had two important hyperparameters: the number of kernels and the kernel size. A hyperparameter tuning was also done for this deep CNN. In transfer learning, a pre-trained model was reused as starting point for a model on a new task. A pre-trained model like VGG16 was trained on the ImageNet Dataset. The pre-trained layers were used in transfer learning, but the output layer was changed. All the weight for the pre-trained network was used for the new classification task. The same job was done using two more pre-trained networks, Inception v3 and ResNet50. Then the performance of these approaches was compared to each other. Loss, accuracy, and AUC (area under the curve) were used for comparison.

II. RELATED WORKS

Numerous recent studies have been conducted to understand Alzheimer's disease detection in a better way. Alzheimer's disease is one of the most common neurological dementia, resulting in various memory-related neurological disorders. According to the 2015 World Alzheimer's Report, around 50 million people worldwide have dementia, with Alzheimer's disease accounting for 70–80% of cases [16]. Alzheimer's disease will impact 131.5 million people globally by the year 2050, according to projections [17]. A 3D CNN is time-consuming and computationally expensive. Xiang et al. proposed a 2D CNN architecture for 3D MRI image Alzheimer's disease classification. The proposed system achieved 92% accuracy with a 91% f-1 score which is 9.5% better than the baseline 3D model [20]. Rizwan et al. proposed a method that significantly handled data insufficiency problems by augmentation and managed to classify 2D images with an accuracy of 98.9% with an F1 score of 96.3% [14]. Ammarah et al. implemented a 4-way classifier with a prediction accuracy of 98.8% [8]. Bijen et al. proposed that the misclassification was higher if only classified using the CNN network. This work presented feature selection with feature ranking algorithms like Mutinffs, ReliefF, Laplacian and UDFS. The CNN network they used was like Alexnet but had some parametric changes. The performance of the result with classification accuracy was around 98% to 99% [12]. Choi et al. used a convolutional neural network (CNN) to classify Alzheimer's disease (A.D.), mild cognitive impairment (MCI), and normal control (N.C.) subjects based on MRI. The prediction model achieved an accuracy of 92.3% for AD/NC, 85.6% for AD/MCI, and 78.1% for MCI/NC [5]. Bijen et al. employed scratched-trained CNN or pre-trained Alexnet CNN as a generic feature extractor of the 2D image, which dimensions were reduced using PCA+TSNE, and finally classifying using simple Machine learning algorithm like KNN, Navies Bayes Classifier [13]. Shaik et al. applied both binary and multiclass classification and achieved 100% accuracy, 100% sensitivity and 100% specificity in the case of AD-CN. 96.2% of accuracy, 93% Sensitivity and 100% Specificity in the case of AD-MCI. 98.0% of accuracy, 96% of sensitivity, and 100 specificities in the case of CN-MCI. 86.7% accuracy, 89.6% sensitivity, and 86.61% specificity in the case of AD-MCI-CN [2]. Rachna et al. proposed a mathematical model based on transfer learning. A CNN architecture, VGG-16, trained on the ImageNet Dataset, is used as a feature extractor for a 3-way classification task. The proposed method achieved 95.73% accuracy for the validation set [10]. Yousry et al. proposed a CNN-based end-to-end framework for A.D. classification. The proposed framework achieved 99.6%, 99.8%, and 97.8% classification accuracies onbADNI Dataset for the binary classification of A.D. and C.N. In multi-classification experiments, the proposed framework achieved 97.5% classification accuracy on the ADNI dataset [1].

III. Methodology

A. Data Preprocessing

The Dataset used in the study was collected from Kaggle. The Dataset consisted of Preprocessed MRI (Magnetic Resonance Imaging) Images. The Dataset had a total of 6400 images divided into four classes. The four classes and the number of images in each category are presented in Table I. The size of the image was 128*128. Fig 2 depicted images from each class in the dataset.

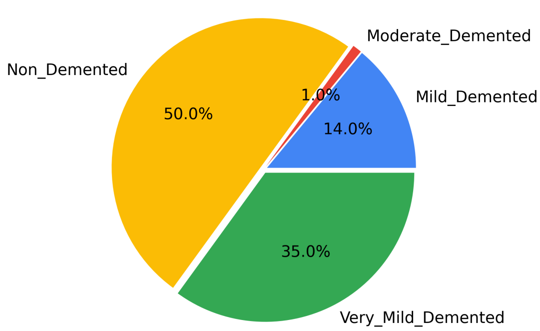

Fig. 1: Class distribution

TABLE I: Classes in Dataset

Class	Number of images	Percentage
Non Demented	3200	50%
Mild Demented	896	14%
Very Mild Demented	2240	35%
Moderate Demented	64	1%

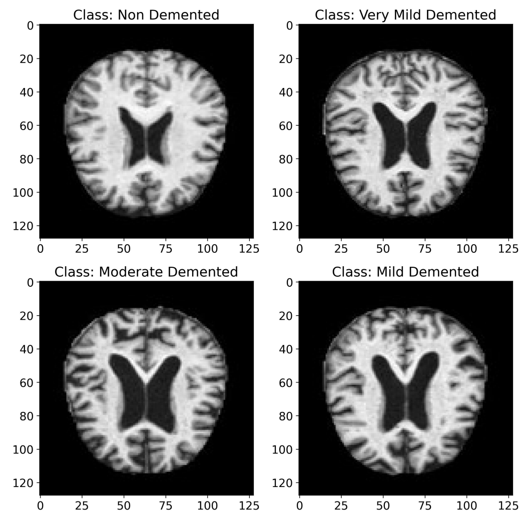

Fig. 2: Sample image from each class

1) Image to CSV for MLP: I found it difficult to feed the images into the MLP directly from the directory due to the input shape of the MLP. A python function was designed that converted the image dataset into CSV. In this case, the image size was (128*128), and there were four classes. So there will be 16384 feature columns in the CSV file and a target column where each row of the Dataset represents a single image.

2) SMOTE: From Fig. 1, we can see the Moderate Demented class has only 1% image of the total size, and the Mild Demented class has only 14% of the entire Dataset. It was a clear case of an imbalance classification problem. A balanced multiclass classification was done by oversampling the minority class. A popular over-sampling technique called SMOTE(Synthetic Minority Over-sampling Technique) was used to solve the imbalance problem. Synthetic samples were generated from the minority class using SMOTE. This oversampling caused a new data sample but did not give any new variation or information in the model. After using SMOTE, the performance improved remarkably. Before oversampling, the accuracy of MLP and CNN was around 40-65%. After using SMOTE, we got an accuracy of 81-93%.

B. Train validation test split

The Dataset was split into a train test set. There was 80% data in the train set and 20% in the test set. Then another split was done on the train set to create the validation data. Again there were 80% data in the train set, and 20% was in the validation set. The model was trained using the train set and validated using the validation set. Then the trained model was tested on an entirely new test set. And a comparison among train, validation and test accuracy was made. For train test validation split, a function called train_test_split() from Python's open-source sci-kit-learn library was used [4].

IV. Multilayer perception (MLP)

Artificial neural networks (ANNs), non-linear models inspired by the brain's neural architecture, were developed to model the learning capacity of biological neural systems [19]. The Multilayer Perceptron is a typical Artificial Neural Network (ANN). MLP consists of one input layer, one or more hidden layers and one output layer. The input layer takes the feature of the Dataset and distributes it to the first hidden layer. The other hidden layer's input comes from the previous layer's output. In each layer, the inputs are multiplied by random weights then a bias term is added with this multiplication. Then this result is gone through a non-linear activation function to generate the output. In this work, the designed MLP consists of one input layer, six hidden layers with different numbers of nodes, and one output layer. Each fully connected layer was followed by batch normalization and a dropout layer. Output from each layer was normalized by batch normalization. Then the normalized output was multiplied by an arbitrary parameter then added another random parameter to the resulting product. This calculation set a new standard deviation and mean for the data. This process helped the weight so that they could not become imbalanced by some very high or low values. This batch normalization increased the speed of training. Each dropout layer had a constant dropout rate of 0.5 which will randomly turn off 50% of nodes in each iteration. The number of nodes for each dense layer was taken as multiples of 2. The first hidden layers had 512 nodes; the second one had 256; the rest had 128, 64, 32, and 16 nodes. The rectified linear activation unit (ReLU) was used in each layer. It is a popular activation function in the deep learning domain. It is an activation function that returns the value itself if it's positive or returns 0 if the input is negative.

$$f(x) = max(0, x) \tag{1}$$

Eq.(1) represented the ReLU activation function. ReLU is more computationally efficient than another popular activation function, sigmoid. It helps the model to prevent the vanishing gradient problem. This function only activates some of the hidden units at the same time. It is a multiclass classification. The output layer had four nodes with the softmax activation. The softmax activation function converted the neural network output into a vector of probabilities.

$$\sigma(x) = \frac{1}{1 + e^{-z}} \tag{2}$$

Eq.(2) represented the softmax activation function. The MLP was compiled with Adam optimizer, categorical cross-entropy loss, and performance metrics (accuracy and AUC score). Adam is a stochastic gradient descent method to find the optimal weights during the backpropagation and the optimal learning rate. The loss function for this MLP was categorical cross entropy, also known as softmax loss. The target value was one hot encoded for using this loss function. After building and compiling the model, it was trained on the training dataset and validated using the validation dataset. The number of epochs was 100, and the batch size was 32. The result of this MLP can be found in the result section of this article. A sample fully connected layer was depicted in Fig. 3

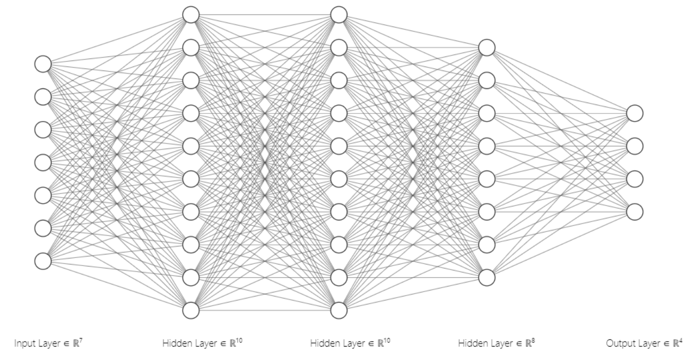

Fig. 3: A sample fully connected network.

V. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) have dominated machine vision in recent years [9]. A sample CNN architecture was depicted in Fig. 4. Just like the MLP, CNN also consists of one input layer, one or more hidden layers, and one output layer. But in CNN, the hidden layers are convolutional layers instead of fully connected layers. Convolutional layers are the primary building block of CNN. This work used a combination of convolutional and separable convolutional layers. Separable convolution is a variation of traditional convolution. It's more efficient and computes faster than conventional convolution. It performs a depthwise spatial convolution followed by a pointwise convolution. A max pooling layer and batch normalization followed most convolutional and separable layers.

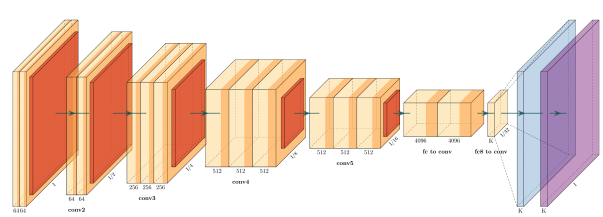

Fig. 4: A sample CNN architecture.

A dropout layer with a random dropout rate followed some. The proposed architecture of the CNN can be mainly divided into two sections. One section is for feature learning. This section consists of convolutional, separable convolutional layers, and max pooling. The second section is the fully connected part that was responsible for classification. In this network, there were two convolutional layers followed by a max pooling layer; there were three groups of separable convolutional layers each group had two separable convolutional layers. A max-pooling layer followed these groups. The pool size for each max pooling layer was (2*2). The first hidden convolutional layer had 16 kernels of size (3*3) the next convolutional layer had 32 kernels of size (2*2). The first group of separable convolutional layers had 64 kernels, the next group had 128 kernels, and the final group had 256 kernels. The size of the kernels for these layers was (3*3). Each filter convolved with the image and created feature maps. Max pooling layers selected the maximum component from the area of feature maps. After convolution, there were three fully connected hidden layers with 512,128 and 64 nodes. Each dense layer was followed by batch normalization and dropout with a random dropout rate. The ReLU activation function was used in each hidden layer. And the output layers had a softmax activation function. The CNN was compiled with Adam optimizer, categorical cross-entropy loss, and performance metrics. After building and compiling the model, it was trained on the training dataset and validated using the validation dataset. The number of epochs was 100.

A. *Hyperparameter tuning*

A deep learning model has some learnable parameters like bias and weights. A model can learn their parameters by training. But some parameters are not learnable, called hyperparameters. The performance of a model immensely depends on hyperparameter tuning. In this work, the MLP and the CNN were trained multiple times with different hyperparameters to find the optimal number of hyperparameters. The ANN model was trained with varying numbers of hidden layers and nodes in each layer, different activation functions (ex., tanh, sigmoid, and relu etc.), different optimizers (Adam, AdaDelta, Adagrad, and RMSprop)batch sizes, and the number of epochs, besides these hyperparameters, trained the CNN multiple times with a different number of kernel and kernel sizes. A list of hyperparameters for the MLP and CNN has been shown in Table II.

TABLE II: List of hyperparameters

Hyperparameters	MLP	CNN
Optimizer	Adam	Adam
Batch size	32	-
Epoch	100	100
Loss function	Categorical cross entropy	Categorical cross entropy
Kernel size	-	(2*2), (3*3)

B. *Precaution for overfitting and vanishing gradient problem*

To prevent overfitting, a callback was used for training the models. This callback monitored the validation loss. The training was stopped if the validation loss was not improving. The dropout layers with random dropout rates also helped the model from being overfitted by turning off some percentage of hidden nodes. Another callback was used to adjust the learning rate. This callback also monitored the validation loss. The learning rate was readjusted if the validation loss needed to be improved. In deep neural networks vanishing gradient is a common problem. The ReLU activation function can take care of the vanishing gradient problem. But on top of that, used batch normalization and Xavier weight initialization.

VI. Transfer Learning, VGG16, Inception v3, ResNet50

In transfer learning, a pre-trained mode is reused for a new task. The knowledge gained from a job is used to solve a different task. The pre-trained model is used as a starting point for a model on another task. First, a pre-trained model was chosen (VGG16, Inception v3, and ResNet40). Two options were available when working with transfer learning: download the network weights or train the model from scratch. In this work, the weights were downloaded. VGG16, Inception v3, and ResNet 50 had more neurons in the final output layer than required. The final output layer was removed and used as an output layer required for this 4-class classification. The starting layers of the pre-trained model were frozen. VGG16, Inception v3, and ResNet 50 model was trained on the ImageNet Dataset for 1000 classes. However, this classification problem had only four categories.

VII. Results

This proposed system aims to detect Alzheimer's disease with MLP, CNN and Transfer Learning and compare the performance of each approach. Fig. 5 represented the learning curve of training vs validation accuracy for MLP. Here an early stopping callback occurred. The training was stopped after over 35 epochs to prevent over-fitting. Training vs validation AUC score form MLP was shown in Fig. 6.

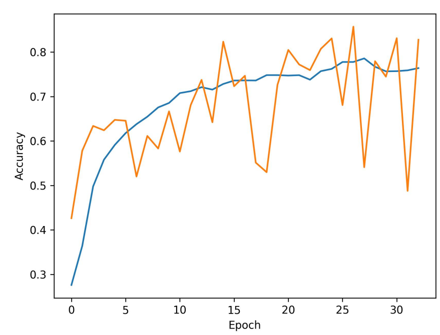

Fig. 5: Accuracy score of MLP.

Training vs validation loss of MLP was shown in Fig. 7. The experimental environment was developed using Keras, a Python Deep Learning library [6]. This Alzheimer's detection was multiclass classification. Fig. 8, Fig. 9, and Fig. 10 represented the training vs validation accuracy score, AUC score and loss of CNN, respectively. The Dataset contained four classes. The proposed system was trained, tested and validated on 6400 MRI images. The MLP for this classification task achieved 81.09% test accuracy with a 97.14% AUC score. The testing loss for this MLP classifier was 0.39. CNN is best suited for image classification. The CNN here gave a much better performance than MLP. The CNN not only surpassed

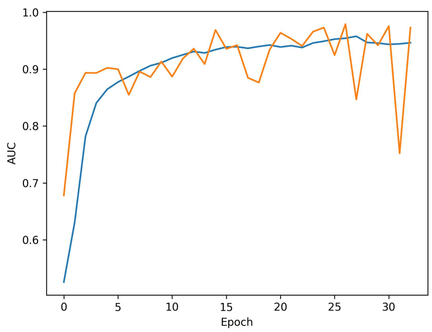

Fig. 6: AUC score of MLP.

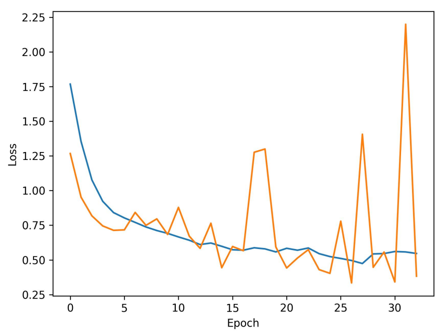

Fig. 7: Loss score of MLP.

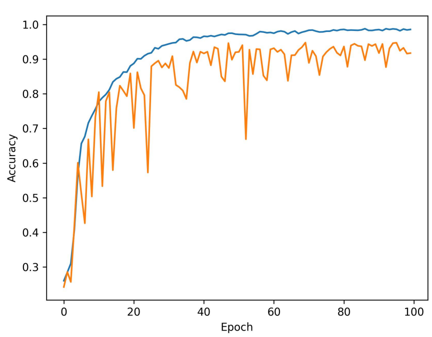

Fig. 8: Accuracy score of CNN.

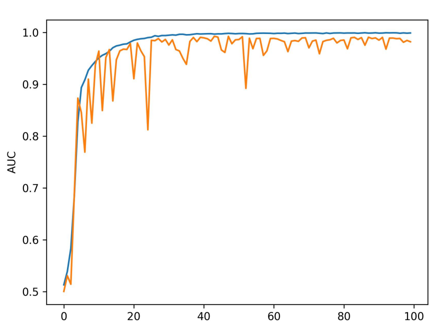

Fig. 9: AUC score of CNN.

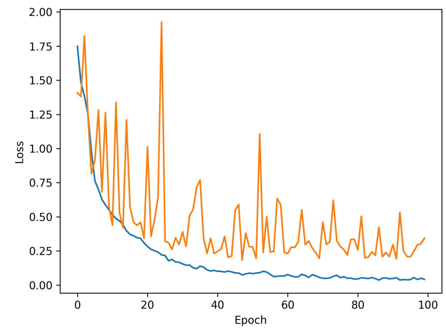

Fig. 10: Loss score of CNN.

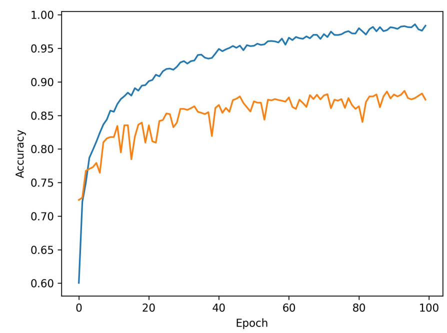

Fig. 11: Accuracy score of VGG16.

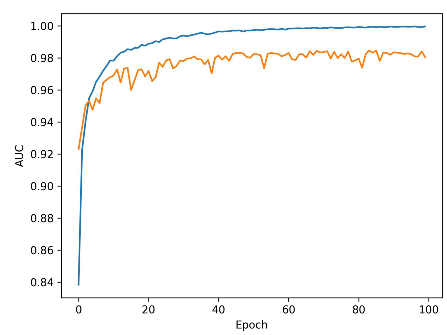

Fig. 12: AUC score of VGG16.

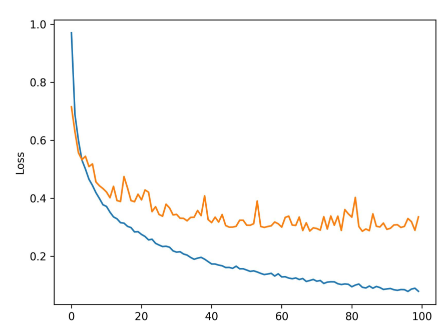

Fig. 13: Loss score of VGG16.

the MLP performance but also exceeded the performance of transfer learning.

Fig. 11, Fig. 12, and Fig. 13 represented the training vs validation accuracy score, AUC score and loss of VGG16 model, respectively. The CNN achieved 92.93% test accuracy with a 98.62% AUC score. The loss for the CNN classifier was 0.27. The pre-trained VGG16 model reached 87.89% accuracy with a 98.03% AUC score and a loss of 0.33. The accuracy and AUC score for Inception v3 and ResNet50 was 81.17%, 93.04%, 75.31% and 93.92%, respectively. And these two networks had 1.09 and 0.58, respectively. Table 3. represents the performance of all the classifiers.

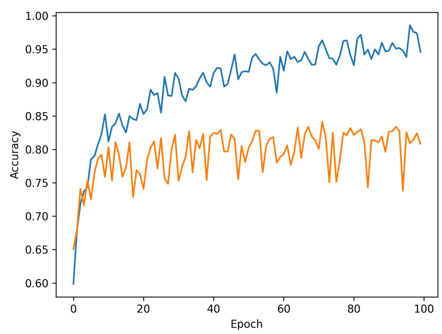

Fig. 14: Accuracy score of Inception v3.

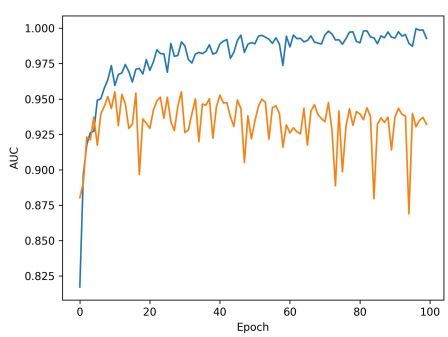

Fig. 15: AUC score of Inception v3.

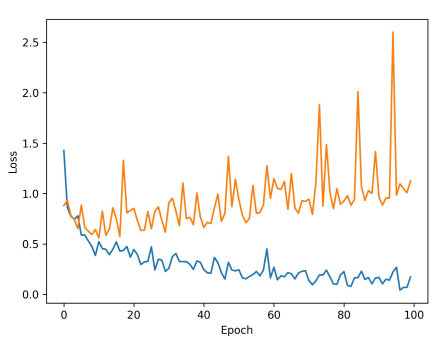

Fig. 16: Loss score of Inception v3.

Fig. 14, Fig. 15, and Fig. 16 represented the training vs validation accuracy score, AUC score and loss of Inception v3 model, respectively. Fig. 17, Fig. 18, and Fig. 19 represented the training vs validation accuracy score, AUC score and loss of ResNet50 model, respectively.

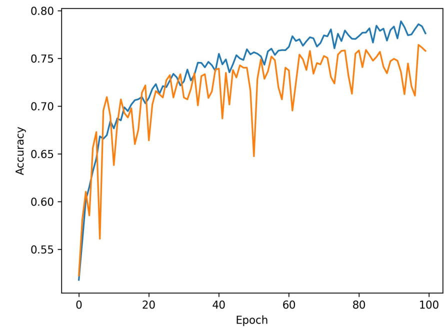

Fig. 17: Accuracy score of ResNet50.

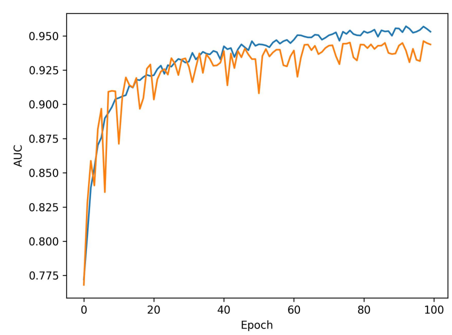

Fig. 18: AUC score of ResNet50.

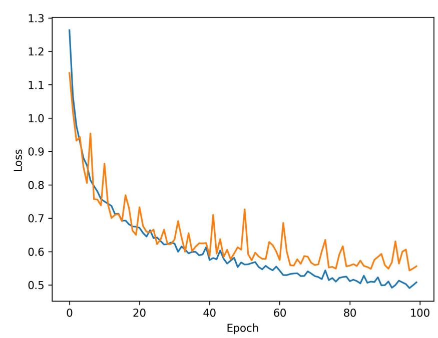

Fig. 19: Loss score of ResNet50.

VIII. Conclusion and Future work

I have started my deep learning journey by doing this project. I learned how to build Multi-layer perceptrons, Convolution Neural Networks, and Transfer learning models. I had exposure to many topics for building MLP and CNN. Activation function, kernel, convolution, hyperparameter tuning, vanishing and exploding gradient problem, dropout and batch normalization, and many more. This was the first time I was

TABLE III: Comparison of MLP, CNN, and Transfer Learning

Classifier	Loss	Accuracy	AUC Score
MLP	0.39	81.09%	97.14%
CNN	0.27	92.93%	98.62%
VGG16	0.33	87.89%	98.03%
Inception v3	1.09	81.17%	93.04%
ResNet50	0.58	75.31%	93.92%

writing the entire report in LaTeX. I learned how to write a report in LaTeX.
The number of samples is crucial for deep learning. In future work, we will train the model with a large dataset. Here in this work, we used the weights of the pre-trained networks, which may be why the pre-trained models did not work better than the custom CNN. In future work, we will build those models (ex., VGG16) from scratch for this specific task.

REFERENCES

[1] Yousry AbdulAzeem, Waleed M Bahgat, and Mahmoud Badawy. A cnn based framework for classification of alzheimer's disease. *Neural Computing and Applications*, 33(16):10415–10428, 2021.

[2] Shaik Basheera and M Satya Sai Ram. A novel cnn based alzheimer's disease classification using hybrid enhanced ica segmented gray matter of mri. *Computerized Medical Imaging and Graphics*, 81:101713, 2020.

[3] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3034–3042, 2016.

[4] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[5] Boo-Kyeong Choi, Nuwan Madusanka, Heung-Kook Choi, Jae-Hong So, Cho-Hee Kim, Hyeon-Gyun Park, Subrata Bhattacharjee, and Deekshitha Prakash. Convolutional neural network-based mr image analysis for alzheimer's disease classification. *Current Medical Imaging*, 16(1):27–35, 2020.

[6] François Chollet et al. Keras. https://keras.io, 2015.

[7] Jeffrey L Cummings, Richard S Isaacson, Frederick A Schmitt, and Drew M Velting. A practical algorithm for managing alzheimer's disease: what, when, and why? *Annals of clinical and translational neurology*, 2(3):307–323, 2015.

[8] Ammarah Farooq, SyedMuhammad Anwar, Muhammad Awais, and Saad Rehman. A deep cnn based multi-class classification of alzheimer's disease using mri. In *2017 IEEE International Conference on Imaging systems and techniques (IST)*, pages 1–6. IEEE, 2017.

[9] Mahbub Hussain, Jordan J Bird, and Diego R Faria. A study on cnn transfer learning for image classification. In *UK Workshop on computational Intelligence*, pages 191–202. Springer, 2018.

[10] Rachna Jain, Nikita Jain, Akshay Aggarwal, and D Jude Hemanth. Convolutional neural network based alzheimer's disease classification from magnetic resonance brain images. *Cognitive Systems Research*, 57:147–159, 2019.

[11] Taeho Jo, Kwangsik Nho, and Andrew J Saykin. Deep learning in alzheimer's disease: diagnostic classification and prognostic prediction using neuroimaging data. *Frontiers in aging neuroscience*, 11:220, 2019.

[12] Bijen Khagi, Goo-Rak Kwon, and Ramesh Lama. Comparative analysis of alzheimer's disease classification by cdr level using cnn, feature selection, and machine-learning techniques. *International Journal of Imaging Systems and Technology*, 29(3):297–310, 2019.

[13] Bijen Khagi, Chung Ghiu Lee, and Goo-Rak Kwon. Alzheimer's disease classification from brain mri based on transfer learning from cnn. In *2018 11th biomedical engineering international conference (BMEiCON)*, pages 1–4. IEEE, 2018.

[14] Rizwan Khan, Zahid Hussain Qaisar, Atif Mehmood, Ghulam Ali, Tamim Alkhalifah, Fahad Alturise, and Lingna Wang. A practical multiclass classification network for the diagnosis of alzheimer's disease. *Applied Sciences*, 12(13):6507, 2022.

[15] Christopher A Lane, John Hardy, and Jonathan M Schott. Alzheimer's disease. *European journal of neurology*, 25(1):59–70, 2018.

[16] Martin James Prince, Anders Wimo, Maelenn Mari Guerchet, Gemma Claire Ali, Yu-Tzu Wu, and Matthew Prina. World alzheimer report 2015-the global impact of dementia: An analysis of prevalence, incidence, cost and trends. 2015.

[17] Farheen Ramzan, Muhammad Usman Ghani Khan, Asim Rehmat, Sajid Iqbal, Tanzila Saba, Amjad Rehman, and Zahid Mehmood. A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks. *Journal of medical systems*, 44(2):1–16, 2020.

[18] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps*, pages 323–350, 2018.

[19] Linda M Salchenberger, E Mine Cinar, and Nicholas A Lash. Neural networks: A new tool for predicting thrift failures. *Decision Sciences*, 23(4):899–916, 1992.

[20] Xin Xing, Gongbo Liang, Hunter Blanton, Muhammad Usman Rafique, Chris Wang, Ai-Ling Lin, and Nathan Jacobs. Dynamic image for 3d mri image alzheimer's disease classification. In *European Conference on Computer Vision*, pages 355–364. Springer, 2020.