Alzheimer MRI Diagnosis and Reproduction with CNN and GAN

Gong Zimu, Li Yifan, Wang Ming
Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai, China

ABSTRACT

Begin the abstract two lines below author names and addresses. The abstract summarizes key findings in the paper. It is a paragraph of 250 words or less. For the **keywords**, select up to 8 key terms for a search on your manuscript's subject.

Keywords: Machine Learning, Alzheimer, CNN, GAN

1. INTRODUCTION

1.1 Alzheimer's Disease and Diagnosis

Alzheimer's Disease is the main type of dementia, with symptoms including losing memory in the beginning and even the ability of conversation in final stages. Most people with the disease are aged 65 or above, which is alarming as many developing and developed countries are facing the challenge of population aging. In the U.S., around 5.8 million people are living with the disease, while in China, there are more than 15 million [1][2].

Considering the seriousness of the disease, it is important to be able to diagnosis the disease at an early stage so that actions can be taken not only to provide remedy, but also to cover the daily care for the diseased. The current diagnosis methods include conversations with patients and their close relatives, tests of memory and problem-solving abilities, combined with blood, urine and other tests [3].

Magnetic Resonance Imaging (MRI) is one of the most crucial diagnosis methods for Alzheimer's disease due to its ability to provide bio-marker. Structural MRI scans often display a decrease of hippocampal and ERC volumes accompanied by other symptoms. By checking the images, experienced doctors can directly judge whether the patient is diseased and also the stage of the disease [4].

2. RELATED WORK AND PROJECT PURPOSE

2.1 Current Application of Machine Learning in Alzheimer Diagnosis

Effective as MRI technology in diagnosing Alzheimer's disease, the final diagnosis still relies heavily on experienced doctors. However, there is an insufficiency of doctors in many developing countries like China. Despite the number of licensed doctors reaches 4.29 million in 2021 [5], it is still not enough compared with the population of 1.4 billion.

Machine learning algorithms can be applied to help the doctors with Alzheimer's disease diagnosis. Researches on this field has been done by many scholars. Using multiple MRI images from different angles as inputs, the diagnosis with CNN has reached an accuracy of over 96% [6].

2.2 Project Purpose

Despite the effectiveness of machine learning algorithm in diagnosis, in China, the lack of medical resources also led to deficiency of Alzheimer's disease database. Combined with the reluctance of patients and their families to donate their medical data for scientific use, researchers may find it hard to collect a large enough database to support the training process of machine learning algorithms.

Generative Adversarial Nets (GAN) is a framework able to reconstruct similar images after learning from a set of image inputs. Training GAN with current Alzheimer MRI datasets and extracting reconstructed images from the net has great potential in aiding the training of CNN networks for diagnosis.

The project aims to compare multiple classification methods including KNN and CNN network in Alzheimer's disease diagnosis, reaching higher accuracy with only single-angled MRI dataset and use GAN network to reconstruct brain MRI photos of Alzheimer's disease patients, which would then be fed back to the classification network to test its result.

3. METHODS

3.1 Classification Methods and CNN

The following are the methods we have used in the project for comparison.

1. Naive Bayes takes the input data to train the posterior repeatedly to obtain a prior parameter to classify future data.
2. Decision Tree gives a tree as the classification rule generated by each feature of those labeled data and then classify test data according to the decision tree.
3. Support Vector Machine (SVM) finds the boundary between data with different labels and future data can be classified by their location related to different boundaries.
4. K-Nearest Neighbor (KNN) determines the label by examining the labels of the k nearest points in the training data.
5. Logistic Regression finds the linear formula that can best represent the distribution of data with the same label and then calculate the indicator of each label on the testing data to determine their label.
6. Random Forest uses multiple decision trees to comprehensively determine the label of test data.

Other than the methods above, Convolutional Neural Network (CNN) has become popular in recent years. Unlike traditional methods like SVM and KNN, which require rigid assumption in the distribution of labeled data and definition of the kernel, in CNN, the filter can self-develop to adapt to different environment.

Common CNN consists of 3 components: Convolutional layer, Pooling layer and Fully Connected layer. Convolutional layers use filters to convolute through the input to extract features. The adjustable parameters include padding mode, stride and number and size of filter. Polling layers help decrease the size of output. Finally, fully connected layers are introduced to generate the prediction label. Activation functions like ReLU are also included between several layers to enable the model to fit randomly shaped non-linear functions.

After deciding the structure of neural network, it must be trained with predefined loss function according to the needs. The designing of CNN has lots of variants with additional components such as residual block in ResNet [7].

3.2 GAN

In 2014, professor Ian Goodfellow raised the idea of Generative Adversarial Network (GAN). Given a training set, GAN is able to generate new data that resembles the same class. The algorithm is based on two models: a discriminative model, and a generative model. There are many ways of gradient-based learning to update the parameter in these two neural networks, and the most common momentum ones is applied in the project. The ratio of discriminator and generator training in one iteration (k2) is set to be 1 since it is found that due to the size and number of training data, even if k2 is set to be larger values, it is still very likely for the loss of the generator to diverse during the training process, which results in an unstable GAN model.

The global optimum is that generator will become closer to the actual distribution of true data and the discriminator can only do a random guess for any input data, which also indicates that the two neural networks reach a balanced stage [8]. Thus, it is desired to let the generator approximate the hidden distribution of the data set. To prevent the generator from imitating the data set too much, data should be selected from the dataset randomly to ensure that not too much percent of the data is shown to the generator at once.

GAN is versatile in application. For instance, the face ID uses GANs to generate models of faces by examining some photos in different angles. After that, the GAN will realize the users' face by comparing with the stored model [9].

In recent years, GANs develops rapidly and there are lots of variants including multiple players with more complicated rules for the game, cycle GAN is an example that can transform one certain kind of photo into another kind of photo (like transforming a horse into a zebra). The algorithm below is taken from the original essay of GAN by Goodfellow that describes its working mechanism [8].

Algorithm 1 : Generative Adversarial Network[8]

Input : a data generator X, iteration time k
Output: the generator and discriminator after training

Function `Generative Adversarial Network Training`(X, $k1$, $k2$):
  Build two neural network Y and Z;
  for *i from 0 to k1-1* **do**
    for *j from 0 to k2-1* **do**
      Sample n data y_i from Y;
      Sample n data x_i from X;
      Update the parameter in Z by $\nabla_{\theta_d} \frac{1}{n} \sum_{i=1}^{n} \left[\log D\left(\mathbf{x}^{(i)}\right) + \log\left(1 - D\left(\mathbf{y}^{(i)}\right)\right) \right]$
    end for
    Sample n data y_i from Y;
    Update the parameter in Y by $\nabla_{\theta_g} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 - D\left(\mathbf{y}^{(i)}\right)\right)$
  end for
  return Y and Z;
end

Algorithm 1. Original GAN model

4. EXPERIMENTS

4.1 Dataset Introduction

The dataset used in this project comes from Kaggle, a comprehensive machine learning and data science community. The full name of the data set is “Alzheimer MRI Preprocessed Dataset” [9], contributed by Dr. Sourabh Shastri from University of Jammu. It collects 6400 MRI images of suspected Alzheimer’s disease patients. The images are classified into 4 types: 3200 Nondemented, 2240 Very Mid Demented, 896 Mild Demented and 64 Moderate Demented. The images have also been preprocessed to 128x128 pixels each, so that it is easier to fit in the machine learning model within affordable computational power. The following are 4 samples taken from each class in the dataset. A trend of demented tissues can be spotted in the middle part of the MRI figure with the increase of disease intensity.

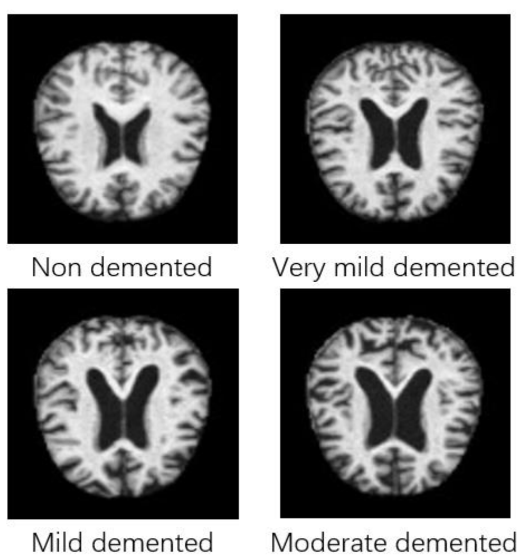

Figure 1. 4 samples collected from the dataset of each class

4.2 Non-CNN Models for Classification

The first goal is to find a method for accurate classification of Alzheimer's disease MRI images. First, we tried non-CNN models and listed the accuracies in the following chart.

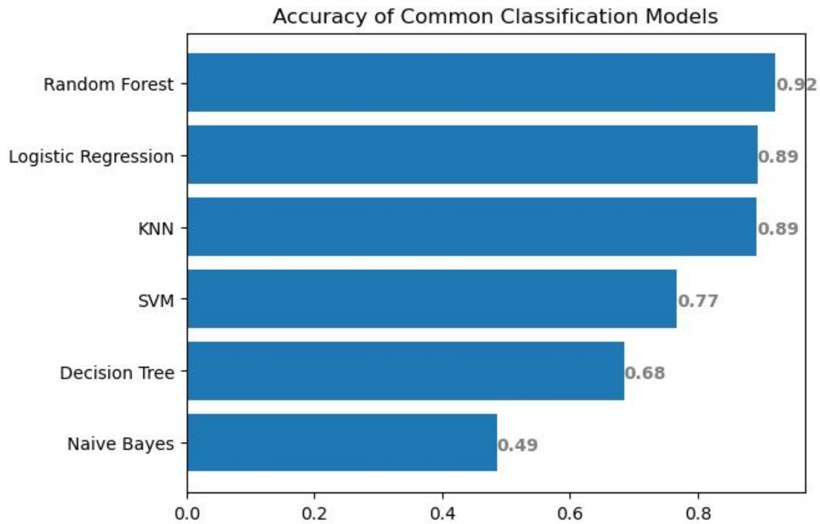

Figure 2. Accuracy of Common Classification Models

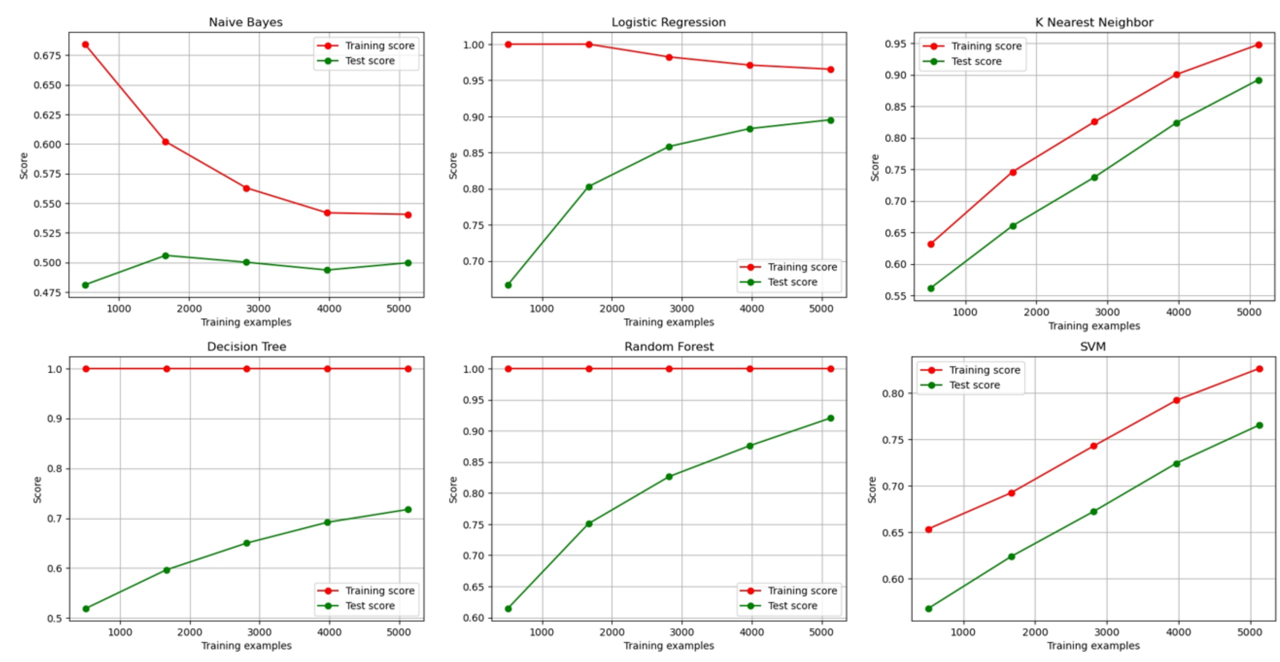

Figure 3. Learning Curve of Common Classification Models

The models we use for classification are Naïve Bayes, Logistic Regression, K Nearest Neighbor, Decision Tree, Random Forest and SVM (with RBF kernel) respectively. From the figure, we spot that in terms of the degree of fitting to the training set, Logistic Regression, Decision Tree and Random Forest all fit the training data quite well, and both Logistic Regression and Random Forest have a quite high accuracy on the test set, which are around 90%. While for Decision Tree, it is likely that it somehow suffers from overfitting. With an around 100% fitting to the training set, it only presents around 70% accuracy on the test set.

For the remaining algorithms, KNN actually doesn't have the process of fitting the training set, but it also has quite high accuracy on test set, outperforming SVM which only has around 75% accuracy. Naïve Bayes presents the worst performance. From the figure, it neither fits the training set well nor shows high test accuracy, thus it seems that Naïve Bayes isn't a good method to classify this dataset.

We thus conclude that classification models like Naïve Bayes, Decision Tree and SVM gave unsatisfactory results while KNN, Random Forest and Logistic Regression performed better, reaching an accuracy of around 90%. However, this value may not be adequate for medical applications. On the other hand, the amount of time taken for these methods would increase significantly with the growth of dataset, which is against our goal. What's more, these algorithms take in all the features to learn and predict, which is likely to lead to overfitting problem, and there isn't a good way to select useful features in these models. Therefore, we turned our focus to CNN, in pursuit of doing better feature selection and reaching higher accuracy.

4.3 CNN Model for classification

With convolutional network, we can select essential features and increase the predict accuracy to a higher level. To create a train dataset and a test dataset for the CNN model, a random split is applied to put the data into 80% and 20% training and testing sets. The CNN model goes through multiple rounds of optimization and the final model is depicted in the following figure.

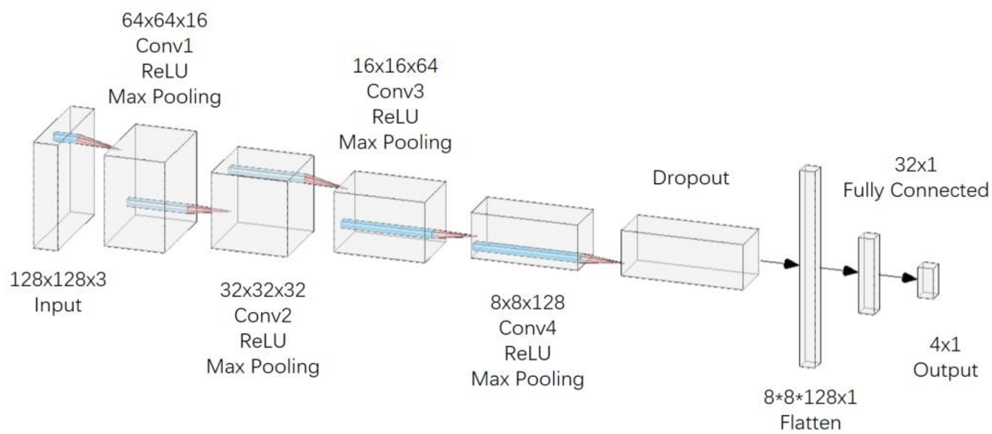

Figure 4. Graphical model of CNN network

The optimum network structure includes:

1. An input of picture with128x128 pixels.
2. Four convolutional layers, selected to have a kernel size of 3, with padding equals to 1. They serve to detect the features of input images.
3. A ReLU layer and a max pooling layer after each of the convolutional layers. The ReLU layer serves as the activation function, while the 2x2 max pooling layers serve to reduce the size of the image for 4 times, which helps to detect more features on a larger scale.
4. A dropout layer after all convolutional layers. It serves to turn off some layers randomly during training process so that the model is less likely to get over fitted. The layer is not used during the testing process.
5. A flatten layer, which flattens the 3-d image into 1-d so that it can be put in a fully connected classification network.
6. Two fully connected layers with another ReLU layer between them. They serve to do the job of classification.
7. A Softmax layer to derive a probability function for classification. In testing, the maximum probability of the Softmax result is taken as the prediction result.

Fifty epochs are trained so that the accuracy and loss level of the model become steady. The data is fed into the model with a batch size of 128 for optimum performance. One round of testing with the test set is applied after each epoch of training. When updating the convolutional network, we observed the losses and accuracy curves, then either added or deleted layers to find if there are any differences. Training iterations are graphically recorded with tensorboard plugin and could be seen with the following figures:

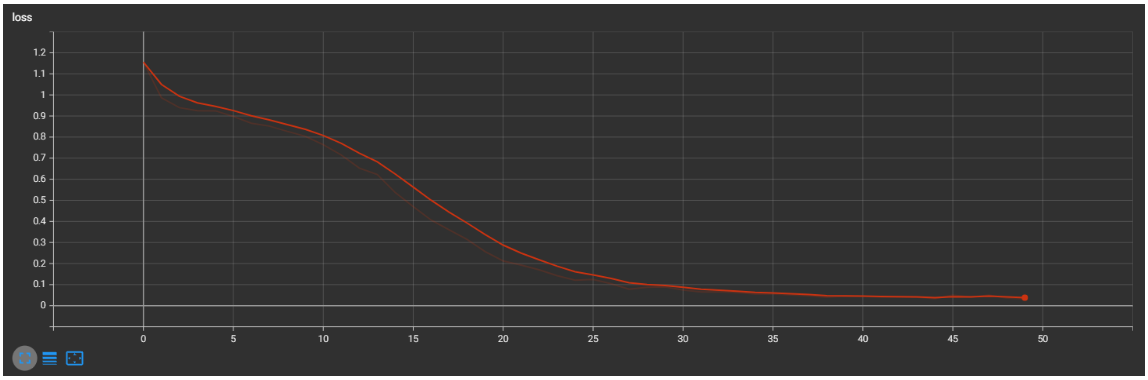

Figure 5. Loss of the CNN model in 50 epochs

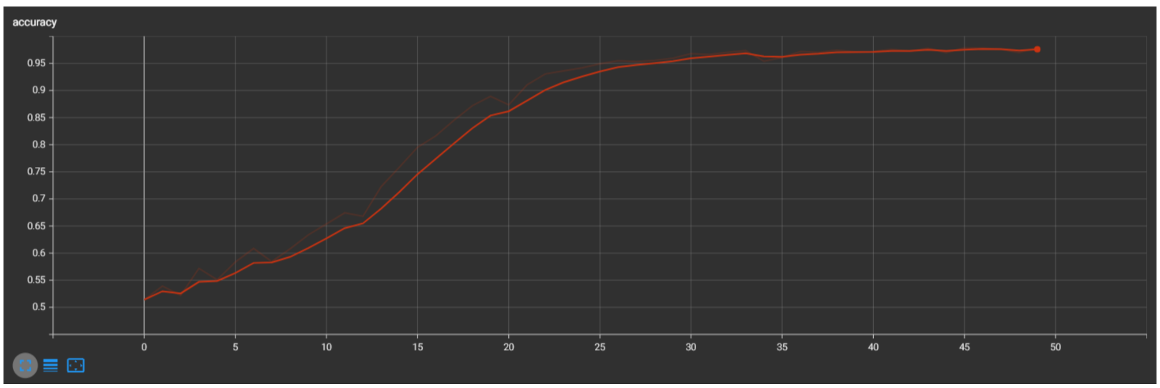

Figure 6. Accuracy of the CNN model in 50 epochs

From the figure, it can be observed that the accuracy increases rapidly in the first 25 epochs, then becomes mild for the following 25 epochs. In the end, the testing accuracy reaches around 98%, which is significantly higher than that of other methods. Besides, we also figure out the confusion matrix for the classification result of CNN, as shown below.

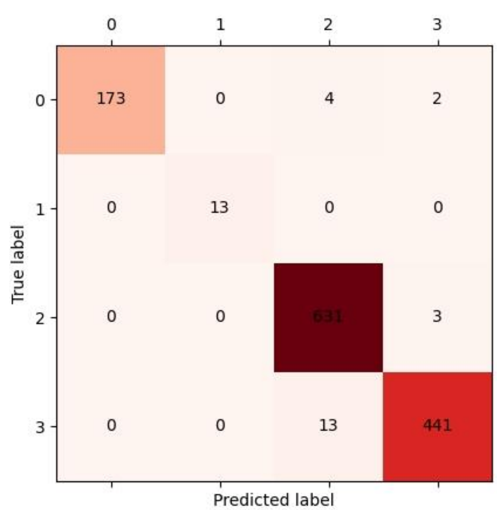

Figure 7. Confusion matrix of CNN classification result

We also calculate the precision score, recall score and F1 score for CNN result. It turns out that the precision rate is around 98.30%, recall rate is around 98.28% and F1 score reaches 98.30%, as listed in the following table.

Score Type	Value
Precision score	98.30%
Recall score	98.28%
F1 score	98.30%

Table 1. Precision score, recall score and F1 score of CNN

The high values of these scores suggest that the CNN model has a quite small possibility of suffering from Type-I and Type-II error, and thus it has the potential to be used in medical diagnosis of Alzheimer. Our result indicates the accuracy and performance advantage of CNN network to some extent.

4.4 GAN Model in Reproducing Pictures

After constructing the model for classification, a GAN model is designed to reproduce new images from original ones. The GAN model consists of the following components: discriminator net, generator net and training model.

The discriminator net is derived from the CNN net from the last section. It consists of 4 convolutional layers with ReLU and max-pooling by 2x2 and 2 fully connected layers. Different from the diagnosis network, the discriminator has the final output processed with sigmoid function so that it can decide whether the picture is real or is generated by the generator. The graphical model is shown below.

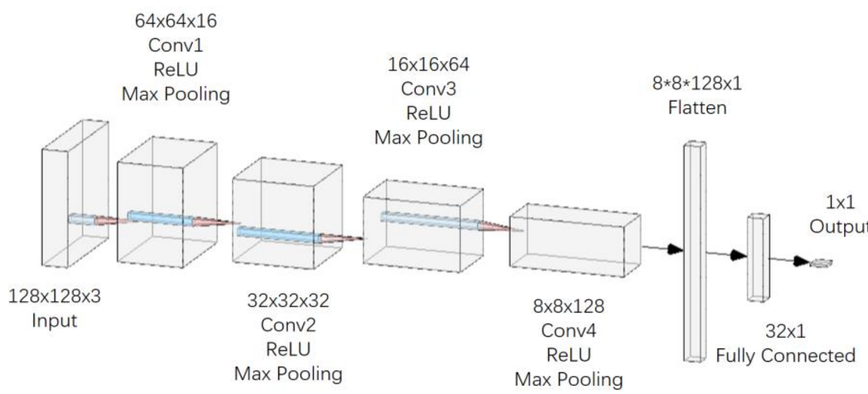

Figure 8. Graphical model of discriminator network

The generator net is a reversed CNN network for feature learning. It takes 100 random parameters as input, and goes through 5 transpose convolutional layers to finally generate an output of 128x128x3 image, which is in the same dimension as the input for the discriminator net. The graphical model is shown below.

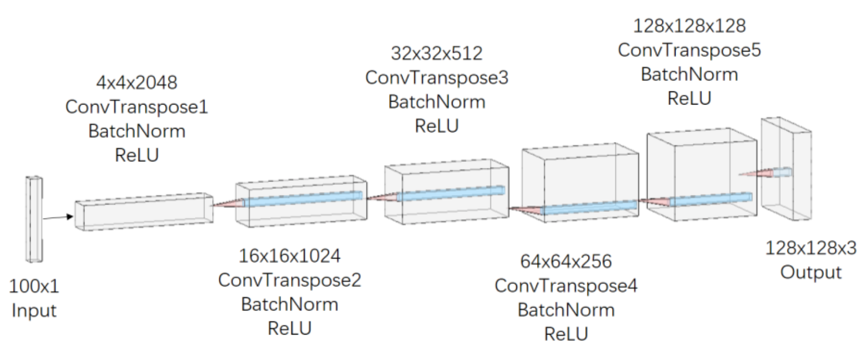

Figure 9. Graphical model of generator network

The training model trains discriminator and generator network alternatively to allow them compete with each other and finally reach a balanced stage. In one epoch, a batch of real images are first fed into the discriminator net (D-net), then a batch of fake image from the generator net (G-net) is also fed into the D-net. The D-net then gets optimized with the combined BCE Loss of both real and fake inputs. After that, the G-net will get optimized by the updated D-net with the same BCE Loss function.

The loss for the D-net and the G-net are recorded and are shown in the following figures.

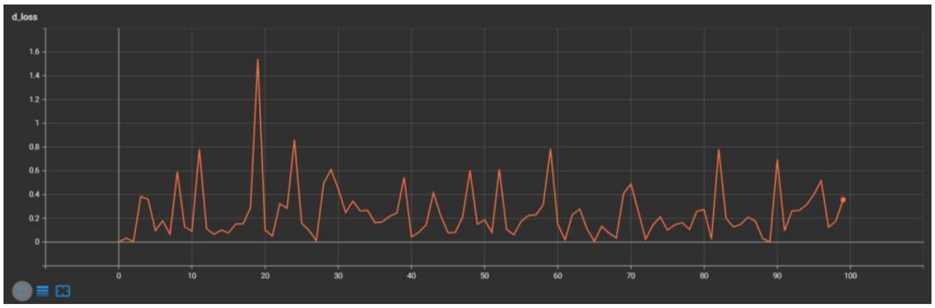

Figure 10. D-Loss for the 100 epochs

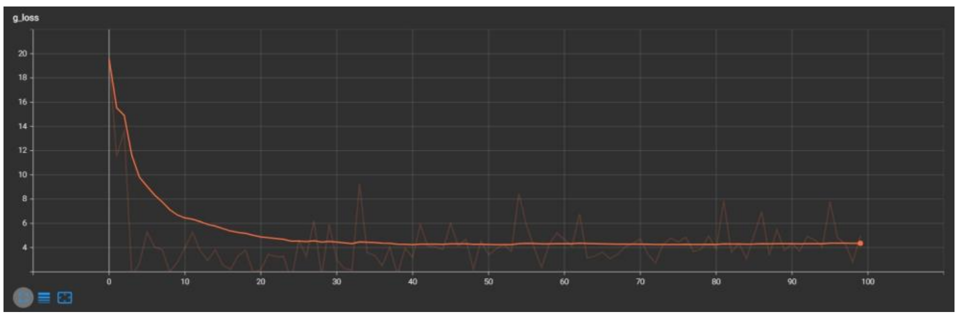

Figure 11. G-Loss for the 100 epochs (smoothed)

By observation, we can tell that the loss of D-net is rather consistent throughout the training process. It can be explained by the rather small amount of data we fed into it. On the other hand, after smoothening, the decrease trend of G-net loss is observable. We can tell that the model reached a rather stable status after 100 epochs.

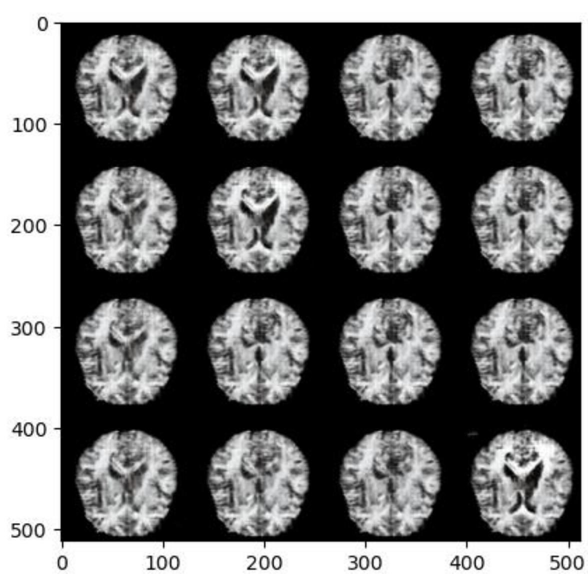

Figure 12. Matrix of generated GAN pictures

Then the 8x8 matrix of generated pictures in the final epoch can be taken as the final result and shown in the figure above. The following picture compares one of the MRI images from the dataset from image generated from our GAN model.

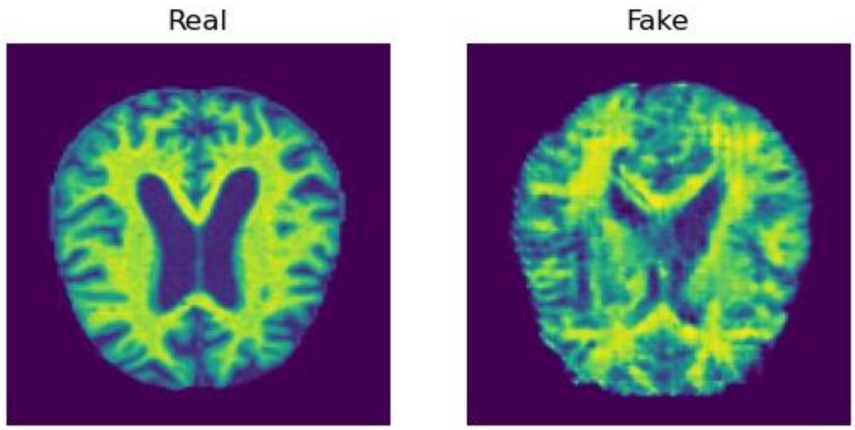

Figure 13. Comparison between a random sample in the dataset and a random GAN result

In both pictures, the outer shape of the brain is easy to tell, and the shape of the middle part is also clear. The GAN reproduction also managed to reproduce a similar neuro-like structure. However, it is admitted that the level of noise in the GAN reproduction remains high, which require further improvement. The model output is also unstable and can change after every epoch.

Finally, a set of generated figures are put into the original CNN classification model again as test data. The results shown that the CNN model is able to classify the reproduced data well. The following figure shows three samples generated by GAN and classified by CNN. One non-demented, one very mild demented and one mild demented. Due to the limitation of moderate demented dataset, we could not reconstruct any moderate demented image with MRI, which is promising to be done with more training data for GAN.

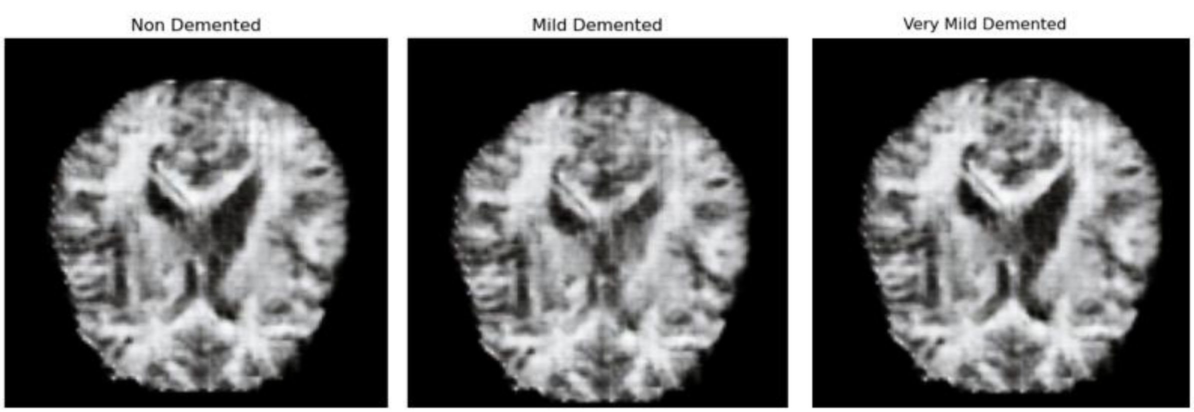

Figure 14. Sample GAN results with corresponding predicted labels

5. CONCLUSIONS

5.1 Conclusion of Project

The project uses “Alzheimer MRI Preprocessed Dataset” with 6400 MRI images of suspected Alzheimer’s disease patients to first find a method for accurate classification of Alzheimer’s disease MRI images with non-CNN and CNN methods. It can be concluded that classification models like KNN, Random Forest and Logistic Regression reached a relatively high accuracy of around 90%, but are inadequate for medical applications with larger data scale. Therefore, we turned our focus to CNN.

Experimental results show that CNN achieved a higher prediction accuracy of 98% after 50 epochs. 4 convolutional layers with ReLU and max-pooling layers are applied in the layer with 2 fully-connected layers. It is proven that CNN has better prediction potential.

Finally, a GAN network is created with CNN discriminator and generator. The two nets reached a balanced state after 100 epochs. Then samples are collected from the generative pictures and compared with original pictures. We can tell that the model did generate picture quite resembling original ones. However, the resolution of the pictures can be enhanced since there are still lots of noises. After putting them back into the CNN network, it gave a satisfactory diagnosis result.

To sum up, the project coped with the problem of Alzheimer’s Disease diagnosis with CNN and managed to establish a simple GAN model to reconstruct pictures to help enhance CNN diagnosis network.

5.2 Space for Improvement

Unlike many similar works done in the field by distinguished experts and scholars, our project only includes a rather small dataset due to limitation of computational power. We also cannot expand the number of parameters in our CNN and GAN networks arbitrarily. Thus, our learning results may be less accurate and less powerful in predicting with a larger dataset, and the pictures generated by our GAN model may also not be so authentic. Given more data for training and greater computational power, the GAN network is promising to reach a better performance and output more realistic medical images. Nevertheless, all team members have worked hard in the project and had a sense of applying the knowledge of the course to solve a real-life problem. All three project members would like to thank the instructor Dr. Zhou Weimin and the teaching assistant Chen Wentao for their guidance.

REFERENCES

[1] Alzheimer's Disease and Healthy Aging, CDC, www.cdc.gov, What is Alzheimer's Disease? | CDC

[2] China Scientific Research Center on Aging -- The prevalence rate of senile dementia in China is nearly 6.0%, and the Need for care services for the elderly with cognitive disease needs to be actively addressed (crca.cn). http://www.crca.cn/index.php/13-agednews/191-6-0.html.

[3] How Is Alzheimer's Disease Diagnosed?, NIH How Is Alzheimer's Disease Diagnosed? | National Institute on Aging (nih.gov)

[4] Chandra, A. et al, Magnetic resonance imaging in Alzheimer's disease and mild cognitive impairment, National Library of Medicine, Jun, 2019, doi: 10.1007/s00415-018-9016-3. Epub 2018 Aug 17.

[5] Zhang, W. Number of licensed doctors in China since 2011-2021, Statistica, Dec 14, 2022

[6] AlSaeed D, Omar SF. Brain MRI Analysis for Alzheimer's Disease Diagnosis Using CNN-Based Feature Extraction and Machine Learning. Sensors (Basel). 2022 Apr 11;22(8):2911. doi: 10.3390/s22082911. PMID: 35458896; PMCID: PMC9025443.

[7] ECE4450J-Lecture18-22, Zhou, W. UM-SJTU JI, 2022

[8] Goodfellow, I, et al, Generative Adversarial Networks, Cornell University, 10, Jun, 2014, https://doi.org/10.48550/arXiv.1406.2661

[9] Yujun Shen etc. "FaceID-GAN: Learning a Symmetry Three-Player GAN for Identity-Preserving Face Synthesis". In: CVPR (2018), pp. 821–830.

[10] Fukushima, Kunihiko (1980). "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position" (PDF). Biological Cybernetics. 36 (4): 193–202. doi:10.1007/BF00344251. PMID 7370364. S2CID 206775608. Retrieved 16 November 2013.

[11] Shastri, S. Alzheimer MRI Preprocessed Dataset, Kaggle, 10.34740/kaggle/dsv/3364939

APPENDIX

In appendix, we attach all the codes used in the project in printed ipynb format.