Final Report

Machine Learning: Detecting Brain tumors in Magnetic Resonance Imaging (MRI)

By:

Zongzhu Li

Maliha Aziz

ML6202- by Professor Amir Jafari

The George Washington University

Table of Contents

PROJECT PROPOSAL

- **What problem did you select, and why did you select it?**

  To sharpen our machine learning skills and understand real-world data , our group decided to select the topic of detecting brain tumors in Magnetic Resonance Imaging from a Kaggle competition: https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri

  The objective of this project is trying to classify the tumor types after training all the MRI images.

  The reason we select it is because automated classification techniques using Machine Learning(ML) and Artificial Intelligence(AI) has consistently shown higher accuracy than manual classification. Hence, proposing a system performing detection and classification by using Deep Learning Algorithms using Artificial Neural Network (ANN) would be helpful to doctors all around the world.

- **What database will you use? Is it large enough to train a machine learning network or different algorithms?**

  We would use the dataset provided by Kaggle. This dataset is collected in the form of images through the scans which is the best technique to detect brain tumors. The dataset needs to be read out into matrix data. After that, we will have lots of data which is large enough to train a machine learning network.

- **What neural network will you use? Will it be a standard form of the network, or will you have to customize it? What algorithms will you use?**

  The ANN classification algorithm will be used in this project according to the objects. Some of the parameters, like layer number, learning rate, will be modified to make a comparison.

- **What software will you use to implement the neural network or different algorithms? Why?**

  Python will be adopted here to implement the neural network because it is easy to use and many packages are plug-in-use.

- **What reference materials will you use to obtain sufficient background on applying the chosen network or algorithm to the specific problem that you selected?**

  In this project, EDA, preprocessing, image reading out, model building and evaluation strategy, will be applied, so all the knowledge and reference materials related to the above topic will be referred.

- **How will you judge the performance of the network? What metrics will you use?**

  After building a model with the 80% randomly selected data from the training dataset, we will apply the model to the 20% left dataset to evaluate the model performance and fitness. The evaluation metrics will be the accuracy score.

- **Provide a rough schedule for completing the project.**

  June 15 - 16: proposal writing and data preparation;

  June 17 - 20: code for model building and evaluation strategy;

  June 21 - 23: presentation preparation and final report.

INTRODUCTION

A typical brain tumor diagnosis requires an expert neuro-oncologist, a pathologist and series of scans and surgeries. The process is time consuming and causes distress to the patient while they wait for the results. On the other hand the oncologist has to wade through a series of MRI scans and make expert judgment on whether they identify a tumor or not. The recent groundbreaking research [13] utilizing stimulated Raman histology (SRH) - an optical imaging method in conjunction with artificial intelligence (AI) motivated our team to explore similar ideas with a different dataset i.e. MRI scans.

DESCRIPTION OF THE DATASET

Brain tumors are named according to the region of the brain they occur in (Figure 1). We downloaded a training set of MRI scans for brain tumors from Kaggle [12]. The tumors were classified into categories detailed in Table 1.

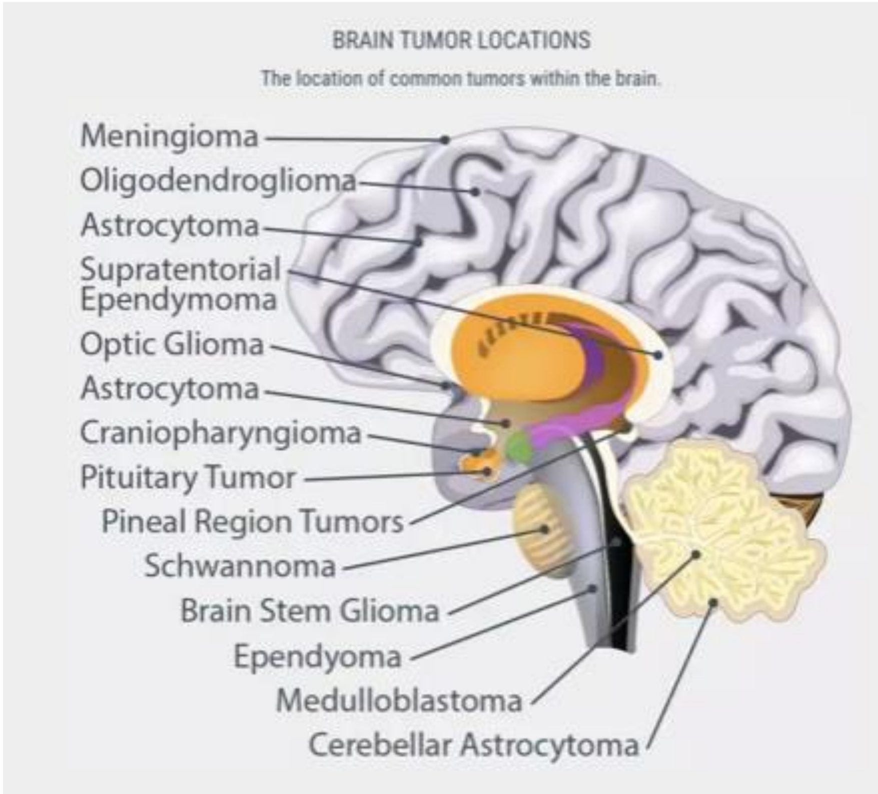


Figure 1. Types of Brain Tumors [9]

Tumor Type	Dataset Type	Image Number
No Tumor	Training	395
Glioma	Training	826
Meningioma	Training	822
Pituitary	Training	827

Table 1. Classes of Brain Tumor in Kaggle training set

MACHINE LEARNING MODELS

We explored various machine learning methods that ranged from classical to deep learning techniques.

Support Vector Machine (SVM)

SVM is a supervised machine learning model that uses classification algorithms for classification problems. Compared to newer algorithms like neural networks, they have two main advantages: higher speed and better performance with a limited number of samples (in the thousands)[13]. It depends on three working principles:

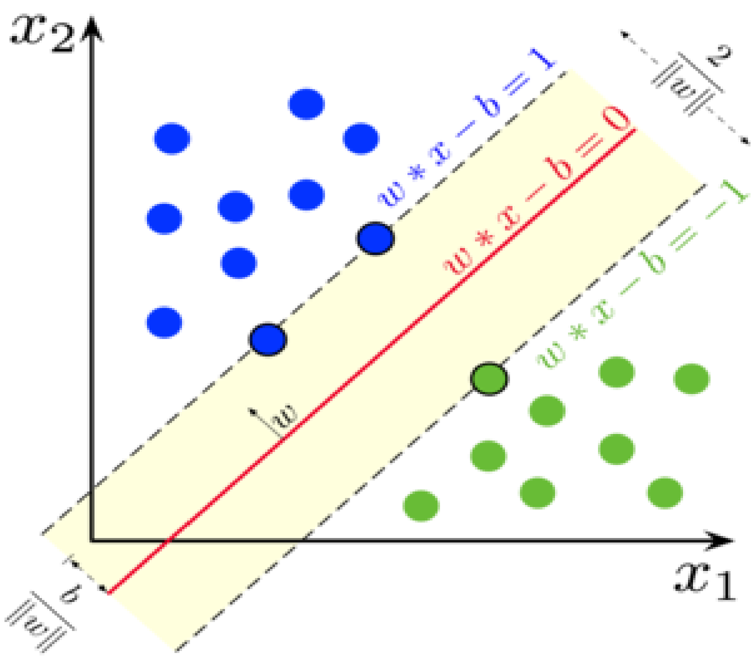


Figure 2. SVM (from Wikipedia)

I. Maximum margin classifier

The maximum margin classifier considers a hyperplane with maximum separation width to classify the data.

II. Support Vector Classifiers

This type of classifier can be regarded as an extended version of the maximum margin classifier which also deals with the non-separable cases. The kernel trick to make SVM work on non-linear separatable problems.

III. Support Vector Machine

The support vector machine approach is considered during a non-linear decision and the data is not separable by a support vector classifier irrespective of the cost function.

k-nearest neighbors (KNN)

KNN is one of the simplest Machine Learning algorithms based on Supervised Learning technique. KNN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. KNN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using KNN algorithm. The KNN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems[14].

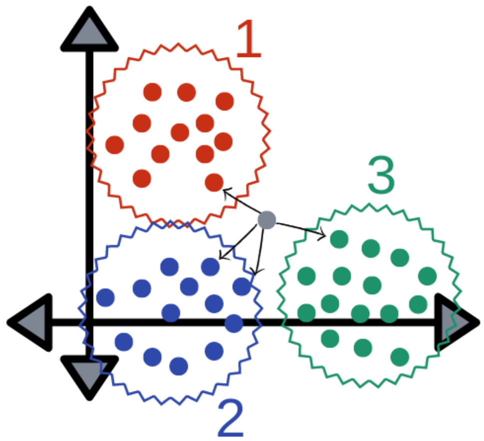


Figure 3. KNN (from Wikipedia)

Artificial Neural Network (ANN)

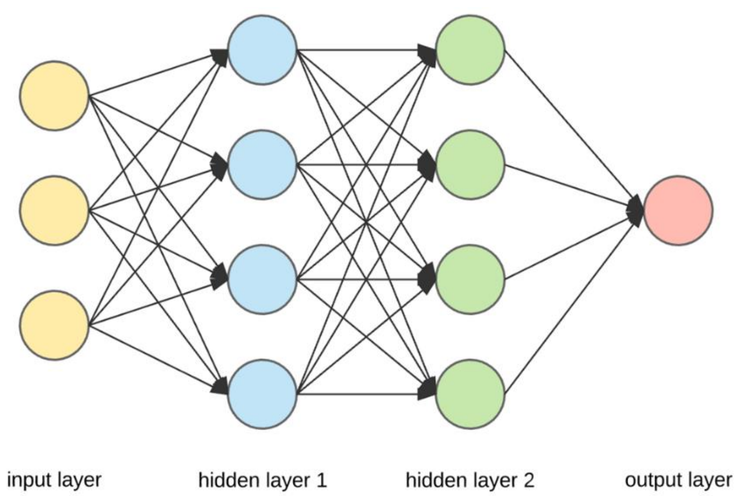


Figure 4. ANN (from Wikipedia)

ANN is typically organized in layers. Layers are being made up of many interconnected 'nodes' which contain an 'activation function'. A neural network may contain the following 3 layers:

1. Input layer to receive as input the values of the explanatory attributes for each observation.

2.Hidden layer to transform the input values to activation.

3.Output layer which receives connections from hidden layers or from input layer.

It returns an output value that corresponds to the prediction of the response variable. In classification problems, there is usually only one output node. The active nodes of the output layer combine and change the data to produce the output values.

ANNs are considered as simple mathematical models to enhance existing data analysis technologies. Although it is not comparable with the power of the human brain, still it is the basic building block of the Artificial intelligence [15].

Model Evaluation Statistics

I. Accuracy

The accuracy is the ratios of correctly predicted observations to the total number of observations. Here, TP means true positives, TN means true negative, FP means false positives, and FN means false negatives.

$$Accuracy = (TP+TN)/(TP+TN+FP+FN)$$

II. Precision

Precision is the ratio of correctly predicted positive samples to the total of predicted positive observations

$$Precision = TP/(TP+FP)$$

III. Recall

Recall is the ratio of TP to all the observations in actual area.

$$Recall = TP/(TP+FN)$$

IV. F1-Score

F-1 score is used to evaluate the classification model based on precision and recall.

$$F1=2*precision*recall/(precision+recall)$$

EXPERIMENTS DESIGN

1. Data Visualization

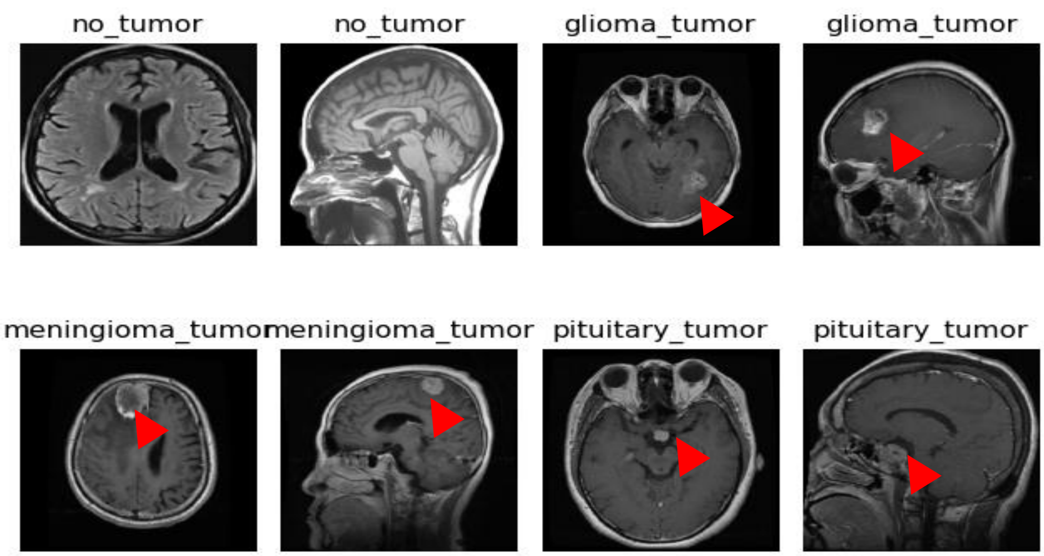


Figure 5. Overview of Dataset for 3 Types of Brain Tumors

The presentative images for no_tumor, glioma_tumor, meningioma_tumor and pituitary_tumor were picked from the files to show the phenotypes (the tumors were pointed out with red arrow head).

2. Data Preprocessing

The MRI scans were put through a series of preprocessing steps before being fed into the deep learning layer. The images were of varying sizes therefore they were resized to a standard 128 by 128. Next we performed data augmentation to include variations of the image to improve our prediction accuracy. The images were then transformed to an array, normalized by dividing the pixels by 255 and finally reshaped from a 3D to 2D.

```
# Data augmentation
X_datagenerator = ImageDataGenerator(rotation_range=20,
                                     width_shift_range=0.2,
                                     height_shift_range=0.2,
                                     zoom_range=0.2,
                                     horizontal_flip=True)
X_datagenerator.fit(X)

# reshape from 3D to 2D
X_train_flatten =
X_train.reshape(X_train.shape[0],X_train.shape[1]*X_train.shape[2]*X_trai
n.shape[3])
X_test_flatten =
X_test.reshape(X_test.shape[0],X_test.shape[1]*X_test.shape[2]*X_test.sh
ape[3])
```

3. Classical Modeling, Prediction and Evaluation – SVM, KNN and Naïve Bayes

The packages that have been used for the modelling is Scikit Learn and we have imported Support Vector Classifier, KNeighborsClassifier and GaussianNB library to perform the functions of the algorithm

```
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB

# creating the classifier object
clf = SVC(kernel="poly")
clf = KNeighborsClassifier(n_neighbors=4)
clf = GaussianNB()
```

After training the models with X_train (splitted from X dataset) and testing the models with X_test(splitted from X dataset), The performance will be evaluated and presented with confusion matrix, classification report and accuracy score.

```
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print("Accuracy : ", accuracy_score(y_test, y_pred) * 100)
```

4. Deep learning: ANN Parameters Optimization

Optimized parameters in ANN are important for the final model's performance. Here we select the following parameters to do modify for our trials:

a. Hidden layers and nodes

b. Iteration numbers

c. Activation with "sigmoid" or "relu"

d. Early-stopping

e. Solver with Adam/ SGD/ RMSprop/ Adagra/ Adadelta/ Adamax

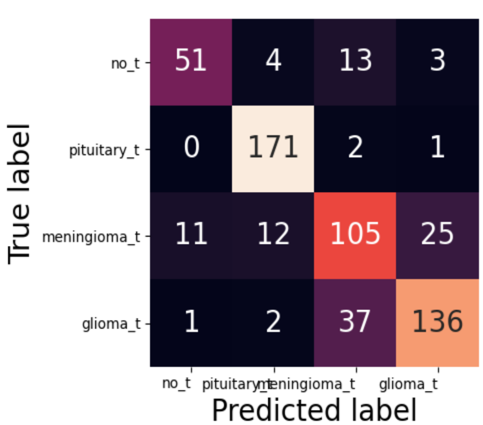
The combination of parameters is listed in the table as following:

Table 2. Experiment Design for ANN

ANN	Layers & Nodes	Iteration	Activation	Early-stopping	Solver
01	(20, 20)	500	“sigmoid”	False	Adam
02	(20, 20)	500	“relu”	False	Adam
03	(20, 20)	500	“sigmoid”, ”relu”	False	Adam
04	(500, 500)	500	“sigmoid”	True	Adam
05	(20, 20, 20,20)	500	“sigmoid”	False	Adam
06	(20, 20)	2000	“sigmoid”	False	Adam
07	(20,20)	500	“sigmoid”	False	SGD
08	(20, 20)	500	“sigmoid”	False	RMSprop
09	(20, 20)	500	“sigmoid”	False	Adagrad
10	(20, 20)	500	“sigmoid”	False	Adadelta
11	(20,20)	500	“sigmoid”	False	Adamax
ANN	(500, 500, 20)	500	“sigmoid”	False	RMSprop

RESULTS

Modeling, Prediction and Evaluation – SVM



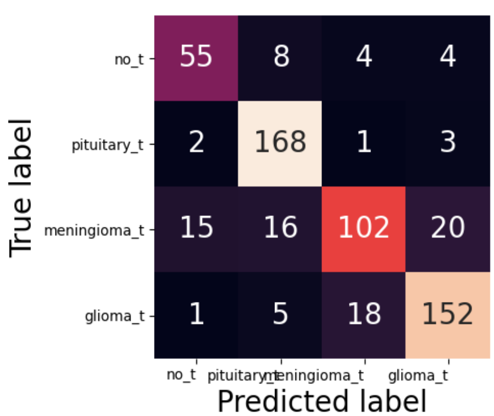

```
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.72      0.76        71
           1       0.90      0.98      0.94       174
           2       0.67      0.69      0.68       153
           3       0.82      0.77      0.80       176

    accuracy                           0.81       574
   macro avg       0.80      0.79      0.79       574
weighted avg       0.81      0.81      0.80       574
```

Figure 6. Confusion matrix and classification for SVM

From the confusion matrix shown in figure.6, we could find that there are 51 true positives for no_tumors, 171 true positives for pituitary_tumors, 105 true positives for meningioma_tumors and 136 true positives for glioma_tumors. From the classification report graph, we can see the accuracy of SVM model is 81%(f1 score for 0's = 0.76 and f1 score for 1's = 0.94, f1 score for 2's = 0.68 and f1 score for 3's = 0.80). It means, for the given test data set, the ratio of the number of samples correctly classified by the classifier to the total number of samples is 81%. It is calculated by (TP+TN)/(TP+FP+FN+TN). Here, TP represents the true positive, and the FP represents the false positive, and the FN represents the false negative, and TN means the true negative. We could get those result from confusion matrix above.

Overall, SVM model shows high classification rate only for pituitary_tumors, we cannot say it is a great model. So, next we will try KNN.

Modeling, Prediction and Evaluation – KNN



Classification Report:

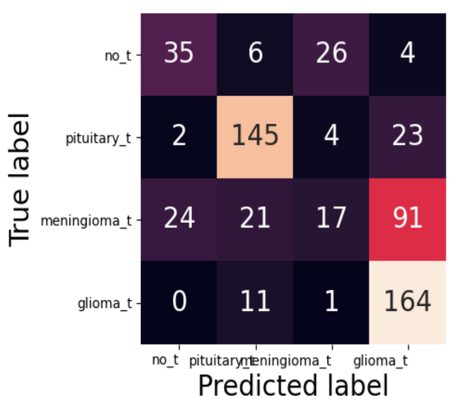
	precision	recall	f1-score	support
0	0.75	0.77	0.76	71
1	0.85	0.97	0.91	174
2	0.82	0.67	0.73	153
3	0.85	0.86	0.86	176
accuracy			0.83	574
macro avg	0.82	0.82	0.81	574
weighted avg	0.83	0.83	0.83	574

Figure 7. Confusion matrix and classification for KNN

From the confusion matrix shown in figure.7, we could find that there are 55 true positives for no_tumors, 168 true positives for pituitary_tumors, 102 true positives for meningioma_tumors and 152 true positives for glioma_tumors. From the classification report graph, we can see the accuracy of KNN model is 83%(f1 score for 0's = 0.76 and f1 score for 1's = 0.91, f1 score for 2's = 0.73 and f1 score for 3's = 0.86).

Overall, KNN model shows better performance on classification rate for all tumor types, however, the accuracy is still not high enough. So, this model is acceptable but not the ideal one.

Modeling, Prediction and Evaluation – NB




```
Classification Report:
              precision    recall  f1-score   support

           0       0.57      0.49      0.53        71
           1       0.79      0.83      0.81       174
           2       0.35      0.11      0.17       153
           3       0.58      0.93      0.72       176

    accuracy                           0.63       574
   macro avg       0.58      0.59      0.56       574
weighted avg       0.58      0.63      0.58       574
```

Figure 8. Confusion matrix and classification for Naïve Bayes

We also tried Naïve Bayes, from the confusion matrix shown in figure.8, we could find that there are 35 true positives for no_tumors, 145 true positives for pituitary_tumors, 17 true positives for meningioma_tumors and 164 true positives for glioma_tumors. From the classification report graph, we can see the accuracy of NB model is only 63%(f1 score for 0's = 0.53 and f1 score for 1's = 0.81, f1 score for 2's = 0.17 and f1 score for 3's = 0.72), which is much lower than SVM and KNN.

Overall, NB model shows a much poor performance on classification rate for all tumor types, So, we will leave it behind.

ANN Model Parameters Optimization

ANN is a model can be used to work on the complex issues. For this image classification problem. We got it involved and tried many different combination of parameters.

From the accuracy shown in Table 3, we could find that

I. activation with “sigmoid” shows a higher accuracy than activation with “relu”(84.5 vs 79.4);
II. Model with more nodes (500,500) for hidden layers gets a higher accuracy than model with less nodes (20,20);
III. Model with more hidden layers (20,20,20,20) does not show better performance in our case, the accuracy is even lower than model with less hidden layers(20,20)(80.0 vs 84.5).
IV. Models with 6 different solver were compared. The highest accuracy is from solver with “RMSprop”(84.8) which is almost same with solver with “Adam”(84.5).

The accuracy and loss curves for all the experiments are shown in figure.9.

ANN	Layers & Nodes	Iteration	Activation	Early-stopping	Solver	Accuracy (%)
01	(20, 20)	500	“sigmoid”	False	Adam	84.5
02	(20, 20)	500	“relu”	False	Adam	79.4
03	(20, 20)	500	“sigmoid”, ”relu”	False	Adam	82.1
04	(500, 500)	500	“sigmoid”	True	Adam	85.2
05	(20, 20, 20,20)	500	“sigmoid”	False	Adam	80.8
06	(20, 20)	2000	“sigmoid”	False	Adam	82.2
07	(20,20)	500	“sigmoid”	False	SGD	81.2
08	(20, 20)	500	“sigmoid”	False	RMSprop	84.8
09	(20, 20)	500	“sigmoid”	False	Adagrad	81.0
10	(20, 20)	500	“sigmoid”	False	Adadelta	78.6
11	(20,20)	500	“sigmoid”	False	Adamax	73.3
ANN	(500, 500, 20)	500	“sigmoid”	False	RMSprop	87.5

Table 3. The accuracy for ANN with different parameters

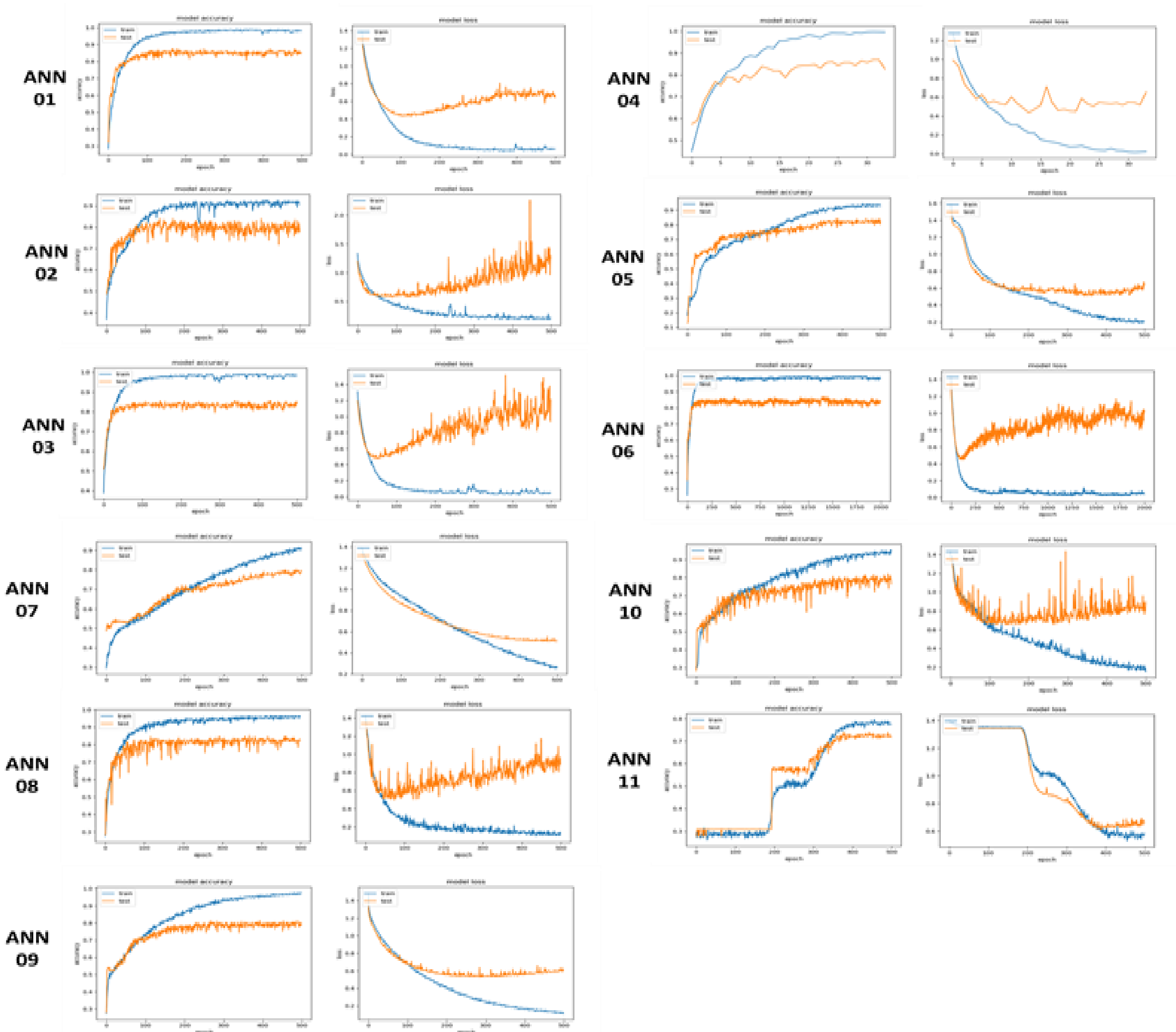
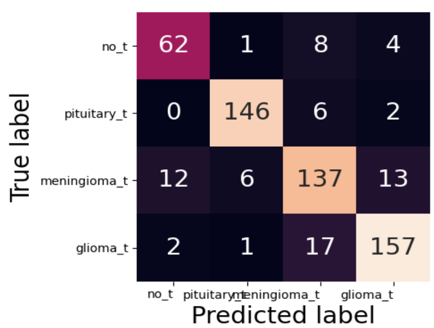


Figure 9. The accuracy and loss curves for ANN with different parameters

Modeling, Prediction and Evaluation – ANN




```
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.83      0.82        75
           1       0.95      0.95      0.95       154
           2       0.82      0.82      0.82       168
           3       0.89      0.89      0.89       177

    accuracy                           0.87       574
   macro avg       0.87      0.87      0.87       574
weighted avg       0.87      0.87      0.87       574
```

Figure 10. Confusion matrix and classification for ANN

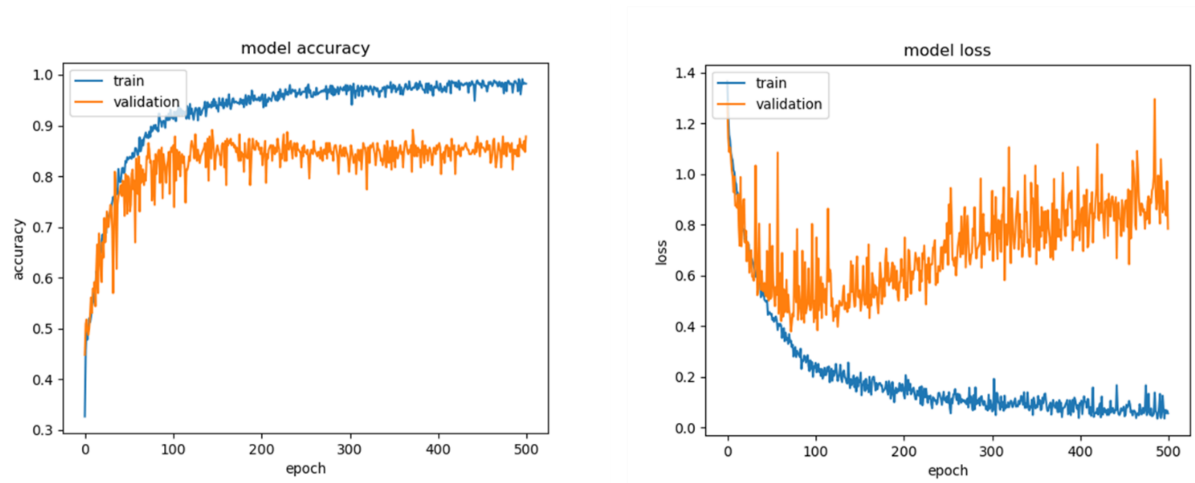


Figure 11. The accuracy and loss curves for ANN

Based on the parameter optimization in ANN model above, an ANN was designed with the following parameters: hidden layers and nodes (500.500.20); iteration (500); activation with “sigmoid” and solver with “RMSprop”.

From the confusion matrix shown in figure 10, we could find that there are 62 true positives for no_tumors, 146 true positives for pituitary_tumors, 137 true positives for meningioma_tumors and 157 true positives for glioma_tumors. From the classification report graph, we can see the accuracy of ANN model is 87%(f1 score for 0’s = 0.82 and f1 score for 1’s = 0.95, f1 score for 2’s = 0.82 and f1 score for 3’s = 0.89), which is higher than SVM and KNN.

Overall, ANN model shows a much better performance on classification rate for all tumor types.

SUMMARY AND DISCUSSION

It feels like technology has come a full circle. A method that was adapted from the functioning of the brain is being used to detect its aberrations and help heal it. We applied various machine learning and artificial neural net methods on the MRI scans of the brain tumor dataset. The successful network gave us a f1-score of 87% where the precision and recall were 87%. We observed that activation with “sigmoid” shows better performance and higher accuracy than activation with “relu” and “tanh”. Furthermore, the RMSprop solver showed higher prediction accuracy with the brain cancer dataset.

Our next step will be to apply more advanced methods such as convolution and maxpool to obtain higher accuracy.

REFERENCES

1. Aldape, K., Brindle, K.M., Chesler, L. et al. Challenges to curing primary brain tumours. Nat Rev Clin Oncol 16, 509–520 (2019). https://doi.org/10.1038/s41571-019-0177-5

2. Cao, H., Erson-Omay, E.Z., Li, X. et al. A quantitative model based on clinically relevant MRI features differentiates lower grade gliomas and glioblastoma. Eur Radiol 30, 3073–3082 (2020). https://doi.org/10.1007/s00330-019-06632-8

3. Nadeem, M. W., Ghamdi, M., Hussain, M., Khan, M. A., Khan, K. M., Almotiri, S. H., & Butt, S. A. (2020). Brain Tumor Analysis Empowered with Deep Learning: A Review, Taxonomy, and Future Challenges. Brain sciences, 10(2), 118. https://doi.org/10.3390/brainsci10020118

4. Hollon, T.C., Pandian, B., Adapa, A.R. et al. Near real-time intraoperative brain tumor diagnosis using stimulated Raman histology and deep neural networks. Nat Med 26, 52–58 (2020). https://doi.org/10.1038/s41591-019-0715-9

5. Kingma, D.P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. CoRR, abs/1412.6980.

6. https://ml-cheatsheet.readthedocs.io/en/latest/index.html

7. https://ruder.io/optimizing-gradient-descent/index.html#adamax

8. https://miamineurosciencecenter.com/en/conditions/brain-tumors/types/

9. https://mlfromscratch.com/optimizers-explained/#/

10. https://en.wikipedia.org/wiki/Support-vector_machine

11. https://www.datasciencecentral.com/profiles/blogs/artificial-neural-network-ann-in-machine-learning

12. https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri

13. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

14. https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26

15. https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/

APPENDIX I

Technical Considerations:

1. Installation of Python 3.5 & above, Pycharm and Anaconda are necessary
2. Packages like numpy, pandas, matplotlib, seaborn, sklearn, tensorflow and keras needs to be installed in the computer in order to execute the application

GitHub Repo Link:

All the documents except data files(too large to upload) related to this project are included in the following repo link:

https://github.com/andrew120606/ML-Final-Project-Group5

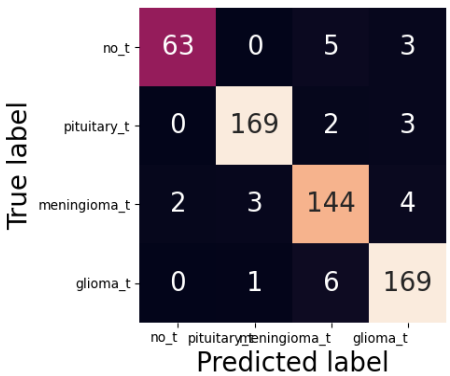
The repo has following folders and file:

1. README.md – Defines the structure of the repo
2. Final_Group5_Project_Proposal
3. Group5_Code – This folder contains the code.
4. Final _Group5 _Presentation – This folder has the PDF version of group project presentation
5. Final_Group5_Project_Report – This folder includes the complete report of the project in PDF format.
6. Data – This folder contains the data source used in this project.

The data files can be found in the following kaggle link:
https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri?select=Training

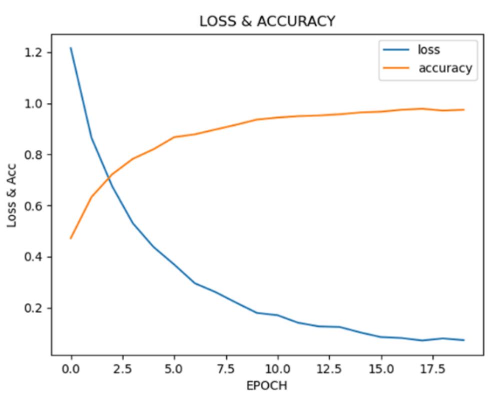
APPENDIX II

Modeling, Prediction and Evaluation – CNN




```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.89      0.93        71
           1       0.98      0.97      0.97       174
           2       0.92      0.94      0.93       153
           3       0.94      0.96      0.95       176

    accuracy                           0.95       574
   macro avg       0.95      0.94      0.95       574
weighted avg       0.95      0.95      0.95       574
```

Fig. Confusion matrix and classification for CNN




```
Epoch 9/50
72/72 [==============================] - 339s 5s/step - loss: 0.2201 - accuracy: 0.9155
Epoch 10/50
72/72 [==============================] - 315s 4s/step - loss: 0.1801 - accuracy: 0.9355
Epoch 11/50
72/72 [==============================] - 310s 4s/step - loss: 0.1710 - accuracy: 0.9434
Epoch 12/50
72/72 [==============================] - 312s 4s/step - loss: 0.1416 - accuracy: 0.9490
Epoch 13/50
72/72 [==============================] - 318s 4s/step - loss: 0.1273 - accuracy: 0.9517
Epoch 14/50
72/72 [==============================] - 315s 4s/step - loss: 0.1252 - accuracy: 0.9564
Epoch 15/50
72/72 [==============================] - 306s 4s/step - loss: 0.1038 - accuracy: 0.9634
Epoch 16/50
72/72 [==============================] - 305s 4s/step - loss: 0.0855 - accuracy: 0.9665
Epoch 17/50
72/72 [==============================] - 323s 4s/step - loss: 0.0819 - accuracy: 0.9739
Epoch 18/50
72/72 [==============================] - 317s 4s/step - loss: 0.0722 - accuracy: 0.9778
Epoch 19/50
72/72 [==============================] - 455s 6s/step - loss: 0.0803 - accuracy: 0.9708
Epoch 20/50
72/72 [==============================] - 391s 5s/step - loss: 0.0737 - accuracy: 0.9739
```

Fig. The accuracy and loss curves for CNN

Report for MLP-1

```
[[ 52   2  10   7]
 [  0 168   1   5]
 [ 14   7 111  21]
 [  1   1  33 141]]
              precision    recall  f1-score   support

           0       0.78      0.73      0.75        71
           1       0.94      0.97      0.95       174
           2       0.72      0.73      0.72       153
           3       0.81      0.80      0.81       176

    accuracy                           0.82       574
   macro avg       0.81      0.81      0.81       574
weighted avg       0.82      0.82      0.82       574
```

Report for MLP-2

```
[[ 48   6  11   6]
 [  0 168   1   5]
 [ 13   7 114  19]
 [  1   1  27 147]]
              precision    recall  f1-score   support

           0       0.77      0.68      0.72        71
           1       0.92      0.97      0.94       174
           2       0.75      0.75      0.75       153
           3       0.83      0.84      0.83       176

    accuracy                           0.83       574
   macro avg       0.82      0.81      0.81       574
weighted avg       0.83      0.83      0.83       574

Accuracy :  83.10104529616724
```

Report for MLP-3

```
[[ 49   5  15   2]
 [  0 167   2   5]
 [ 11  10 114  18]
 [  0   5  29 142]]
              precision    recall  f1-score   support

           0       0.82      0.69      0.75        71
           1       0.89      0.96      0.93       174
           2       0.71      0.75      0.73       153
           3       0.85      0.81      0.83       176

    accuracy                           0.82       574
   macro avg       0.82      0.80      0.81       574
weighted avg       0.82      0.82      0.82       574

Accuracy :  82.22996515679442
```

Report for MLP-4

```
[[ 48   5  12   6]
 [  0 167   1   6]
 [ 15   8 110  20]
 [  2   3  34 137]]
              precision    recall  f1-score   support

           0       0.74      0.68      0.71        71
           1       0.91      0.96      0.94       174
           2       0.70      0.72      0.71       153
           3       0.81      0.78      0.79       176

    accuracy                           0.80       574
   macro avg       0.79      0.78      0.79       574
weighted avg       0.80      0.80      0.80       574

Accuracy :  80.48780487804879
```

Report for MLP-5

```
[[ 57   1   9   4]
 [  1 163   6   4]
 [ 20   8  35  90]
 [  2   0   6 168]]
              precision    recall  f1-score   support

           0       0.71      0.80      0.75        71
           1       0.95      0.94      0.94       174
           2       0.62      0.23      0.33       153
           3       0.63      0.95      0.76       176

    accuracy                           0.74       574
   macro avg       0.73      0.73      0.70       574
weighted avg       0.74      0.74      0.70       574

Accuracy :  73.69337979094077
```

Report for MLP-6

```
[[ 44   4  21   2]
 [  0 169   1   4]
 [ 14  12 108  19]
 [  3   7  36 130]]
              precision    recall  f1-score   support

           0       0.72      0.62      0.67        71
           1       0.88      0.97      0.92       174
           2       0.65      0.71      0.68       153
           3       0.84      0.74      0.79       176

    accuracy                           0.79       574
   macro avg       0.77      0.76      0.76       574
weighted avg       0.79      0.79      0.78       574

Accuracy :  78.57142857142857
```

Report for MLP-7

```
[[ 48   4  13   6]
 [  0 166   2   6]
 [ 17  12 102  22]
 [  1   4  36 135]]
              precision    recall  f1-score   support

           0       0.73      0.68      0.70        71
           1       0.89      0.95      0.92       174
           2       0.67      0.67      0.67       153
           3       0.80      0.77      0.78       176

    accuracy                           0.79       574
   macro avg       0.77      0.77      0.77       574
weighted avg       0.78      0.79      0.78       574

Accuracy :  78.57142857142857
```

Report for MLP-8

```
[[ 52   2  14   3]
 [  0 167   2   5]
 [ 12   2 119  20]
 [  0   1  28 147]]
              precision    recall  f1-score   support

           0       0.81      0.73      0.77        71
           1       0.97      0.96      0.97       174
           2       0.73      0.78      0.75       153
           3       0.84      0.84      0.84       176

    accuracy                           0.84       574
   macro avg       0.84      0.83      0.83       574
weighted avg       0.85      0.84      0.85       574

Accuracy :  84.49477351916377
```

Report for MLP-9

```
[[ 54   2  12   3]
 [  4 164   2   4]
 [ 22   2 107  22]
 [  2   4  24 146]]
              precision    recall  f1-score   support

           0       0.66      0.76      0.71        71
           1       0.95      0.94      0.95       174
           2       0.74      0.70      0.72       153
           3       0.83      0.83      0.83       176

    accuracy                           0.82       574
   macro avg       0.80      0.81      0.80       574
weighted avg       0.82      0.82      0.82       574

Accuracy :  82.05574912891987
```