MINI PROJECT SYNOPSIS

ON

BRAIN.MRI

A CNN based approach to detecting Alzheimer's Disease using Brain MRI scans

(CSE VI Semester Mini Project)

2021-2022

Submitted to:	**Submitted by:**
Mr. Avnish Panwar	Aytijha Chakraborty
(CC-CSE-K-VI-Sem)	Roll. No.: 1914009
	CSE-K-VI-Sem

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

i. ABOUT PROJECT

This project aims at detecting Alzheimer's Disease using Brain MRI scans and classifying its stage, using Deep Learning. This is done using a variety of tools, libraries and modules. This project was written and compiled on Visual Studio using iPython Notebook on Windows 11 OS.

ii. PRE-REQUISITES OF PROJECT

Tested support for OS: Windows 11, Mac 10.15.7, Ubuntu 20.04 LTS

Softwares/Frameworks Used:

- Python 3.10
- Visual Studio Code
- HTML 5
- CSS 3
- Bootstrap 5.2
- JavaScript ES2015
- iPython 7.12.0
- Flask 2.1.2 (Python)
- Werkzeug 2.1.2 (Python)
- NumPy 1.22.0 (Python)
- TensorFlow 2.8.0 (Python)
- Matplotlib 3.5.1 (Python)
- OpenCV 4.5.5 (Python)
- Imbalanced-learn 0.9.1 (Python)
- Scikit-Learn 1.1.1 (Python)
- Seaborn 0.11.2 (Python)

Softwares/Frameworks Required To Run:

- iPython 7.12.0
- Flask 2.1.2 (Python)
- Werkzeug 2.1.2 (Python)
- NumPy 1.22.0 (Python)
- TensorFlow 2.8.0 (Python)

iii. MODULES OF PROJECT

The project comprises of 3 modules:

Module 1. Data Acquiring and Preprocessing:

i. Data acquired from Kaggle user Sarvesh Dubey, who provided a dataset consisting of 6000+ MRI images belonging to four classes of images both in training as well as a testing set:
   a. Mild Demented,
   b. Moderate Demented,
   c. Non-Demented,
   d. Very Mild Demented.

ii. Used ImageDataGenerator (Tensorflow) to augment the images from the Training Dataset and form new training samples in real-time while the model is still training.

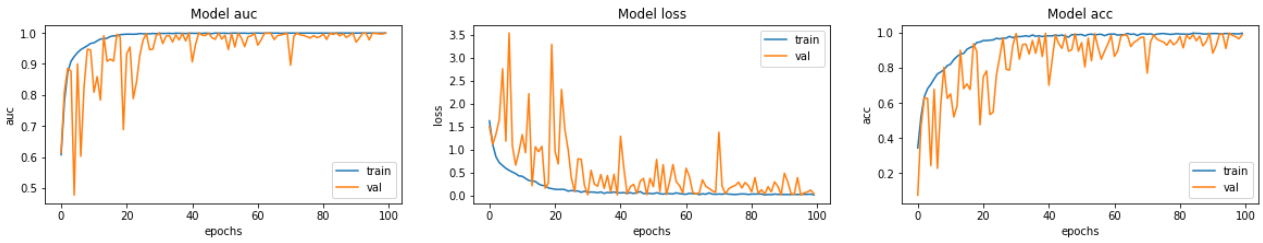
iii. Performed over-sampling of the data using SMOTE to eliminate the class imbalance in the Training dataset.

Module 2. Model Training and Evaluation:

i. Model architecture:

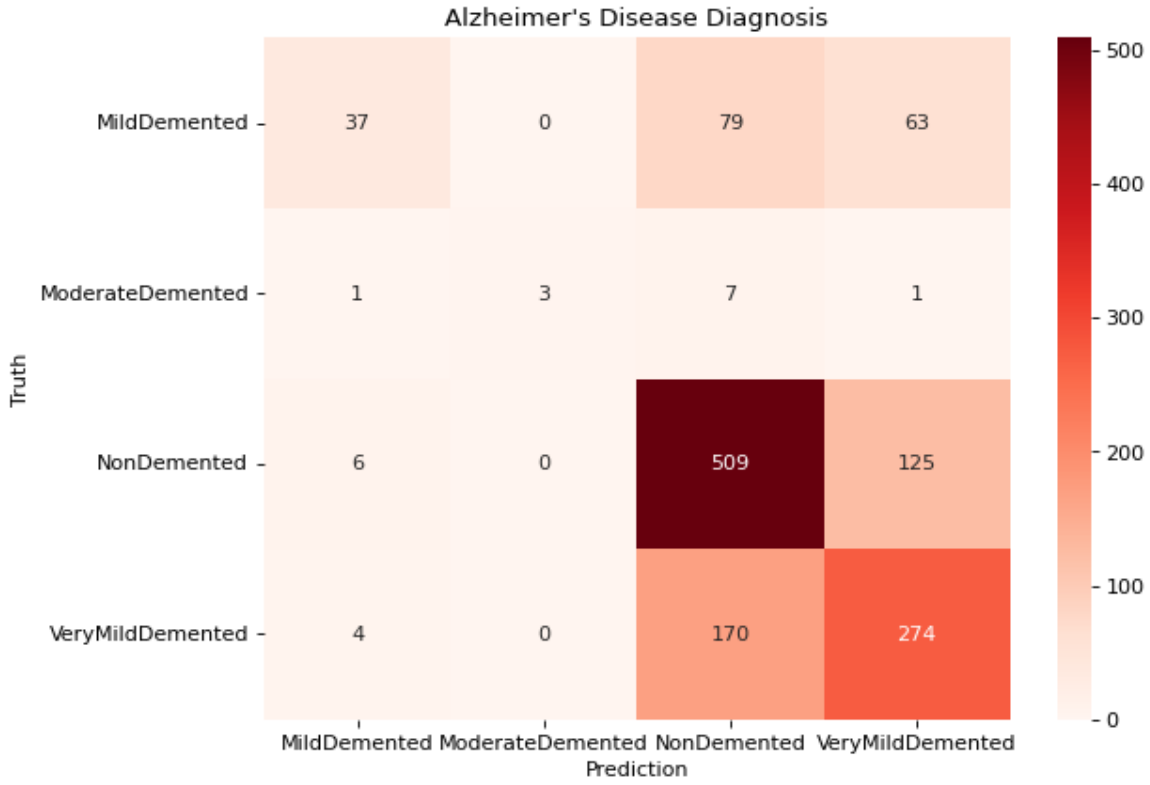
```
Model: "cnn_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_10 (Conv2D)          (None, 176, 176, 16)      448

 conv2d_11 (Conv2D)          (None, 176, 176, 16)      2320

 max_pooling2d_5 (MaxPooling  (None, 88, 88, 16)       0
 2D)

 sequential_7 (Sequential)   (None, 44, 44, 32)        14016

 sequential_8 (Sequential)   (None, 22, 22, 64)        55680

 sequential_9 (Sequential)   (None, 11, 11, 128)       221952

 dropout_5 (Dropout)         (None, 11, 11, 128)       0

 sequential_10 (Sequential)  (None, 5, 5, 256)         886272

 dropout_6 (Dropout)         (None, 5, 5, 256)         0

 flatten_1 (Flatten)         (None, 6400)              0

 sequential_11 (Sequential)  (None, 512)               3279360

 sequential_12 (Sequential)  (None, 128)               66176

 sequential_13 (Sequential)  (None, 64)                8512

 dense_7 (Dense)             (None, 4)                 260

=================================================================
Total params: 4,534,996
Trainable params: 4,532,628
Non-trainable params: 2,368
_________________________________________________________________
```

ii. Model trained for 100 epochs, with a validation split of 20%, resulting in:



iii. When tested on previously unseen data (~1500 images), the results were: loss: 1.9561, acc: 0.6435, auc: 0.8288

iv. It was observed from the confusion matrix that the model was a little confused with Very Mildly Demented and Non-demented samples.



Module 3. Flask App:

i. An HTML page is rendered through Flask to retrieve User-MRI file through a form.

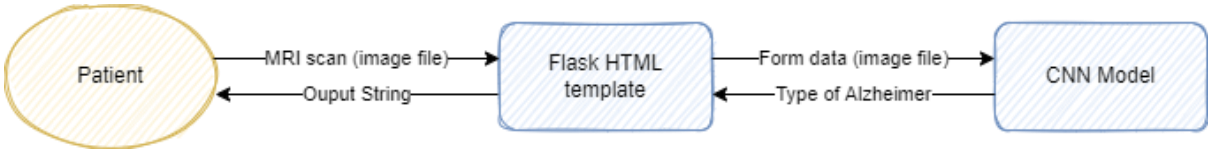
ii. The HTML template is adorned using CSS, Bootstrap, Google Fonts, icons from Icon8, background video from Pexels, and JavaScript.

iii. The MRI image file is converted to an image array of suitable dimensions for the model to work upon.

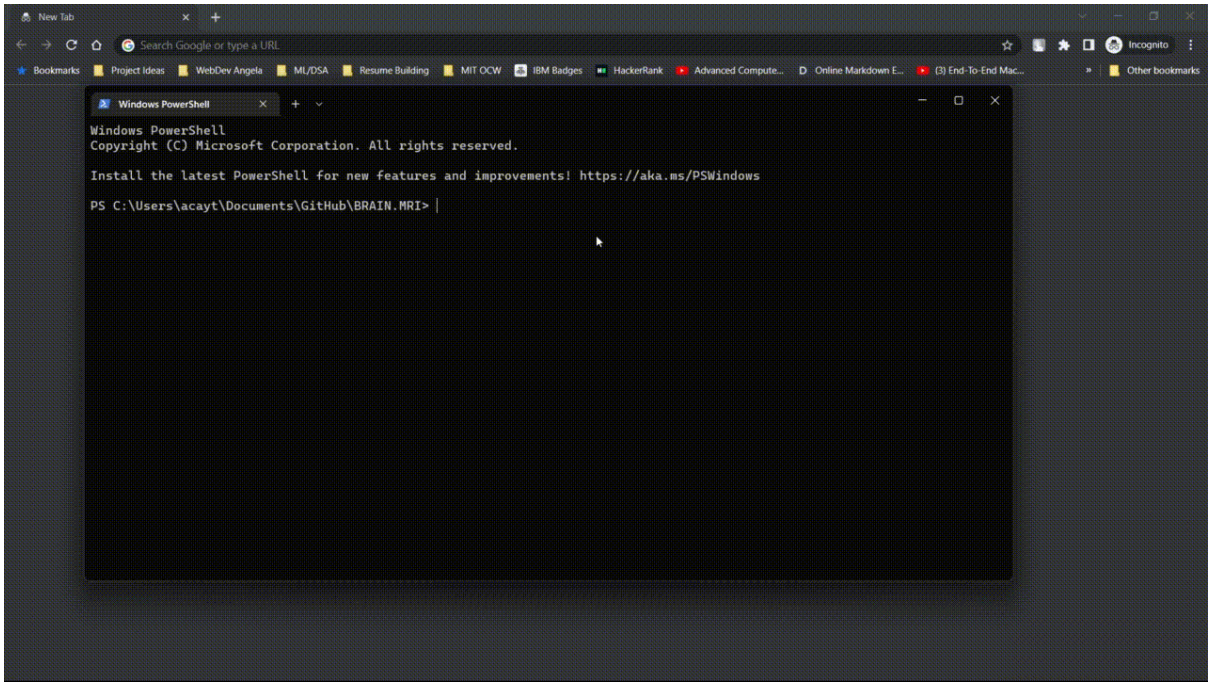
iv. The model classifies a type of Dementia (from the previously mentioned 4 classes) and the output is printed both on the webpage and the terminal.

v. The image file is also saved to the device for future uses.

iv. DATA FLOW DIAGRAM



v. OUTPUT GIF



vi. REFERENCES

- https://aytijha.github.io/BRAIN.MRI/
- https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator
- https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html