Analyzing Brain Scan Images for the Early Detection and Diagnosis of Alzheimer's Disease

Toronto Canada Chapter

Project Lead: **Kausthab Dutta Phukan**

Table of contents :

INTRODUCTION

The Problem :

- Alzheimer’s disease is a brain condition that causes a progressive decline in memory, thinking, learning and organizing skills. It eventually affects a person’s ability to carry out basic daily activities.

- As per the World Health Organization, neurological disorders are responsible for 9% of all deaths globally, and Alzheimer's and other dementias alone are among the top ten leading causes of death worldwide

- Despite significant advances in medical technology, early detection and accurate diagnosis of these conditions remain challenging.

Who does Alzheimer's disease affect?

Alzheimer's disease mainly affects people over age 65. The higher your age over 65, the more likely you'll develop Alzheimer's. Some people develop Alzheimer's disease before age 65 — typically in their 40s or 50s. This is called early-onset Alzheimer's disease. It's rare. Less than 10% of AD cases are early-onset.

How common is Alzheimer's disease?

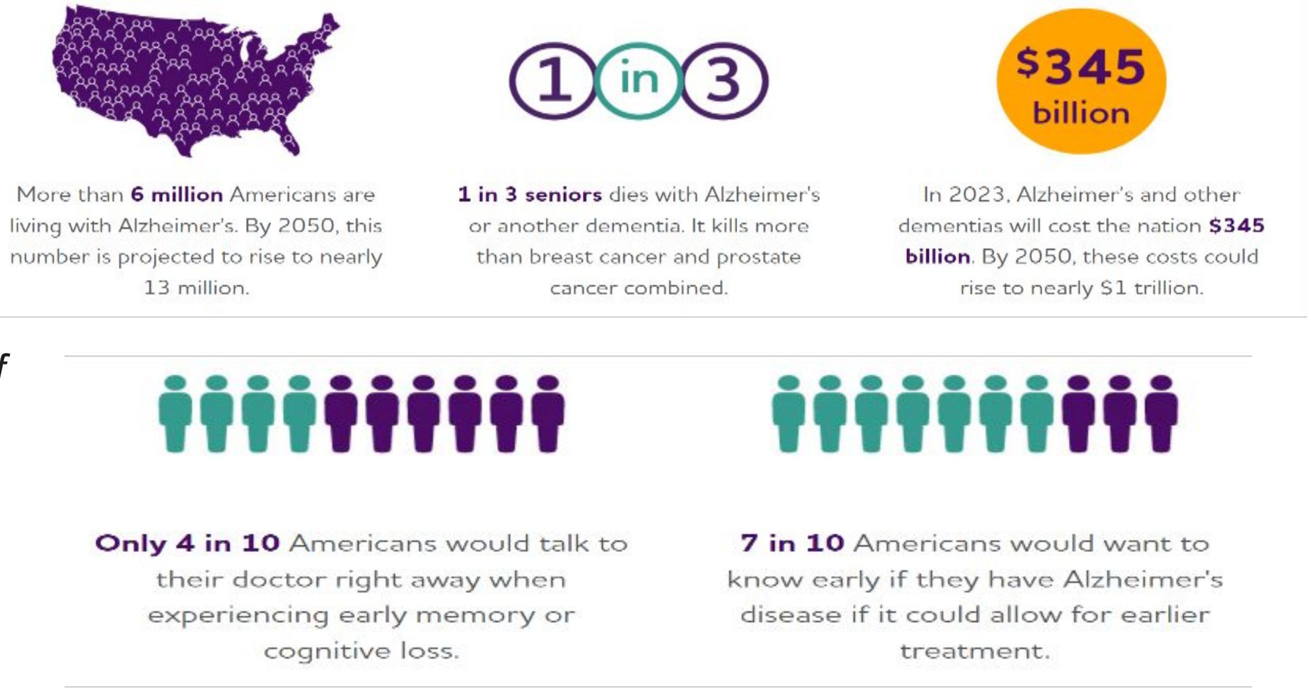
Alzheimer's disease is common. It affects approximately 24 million people across the world. One in 10 people older than 65 and nearly a third of people older than 85 have the condition.

Alarming Statistics of Alzheimer

"In 2006, the worldwide prevalence of Alzheimer's disease was 26.6 million. By 2050, the prevalence will quadruple, by which time 1 in 85 persons worldwide will be living with the disease. We estimate about 43% of prevalent cases need a high level of care, equivalent to that of a nursing home."

Ron Brookmeyer, Elizabeth Johnson, Kathryn Ziegler-Graham, H. Michael Arrighi

[*Forecasting the global burden of Alzheimer's disease – ScienceDirect*]



Importance of early detection

An early diagnosis of Alzheimer's provides a range of benefits for the individuals who are diagnosed.

- **<u>Medical Benefits:</u>**
    - Access to treatment options.
    - An opportunity to participate in clinical trials.
    - A chance to prioritize your health.

- **<u>Emotional and social benefits:</u>**
    - May help lessen anxieties about why you are experiencing symptoms.
    - You and your family also have the opportunity to maximize your time together and access resources and support programs.

Importance of early detection

- **<u>More time to plan for the future:</u>**
  - Planning ahead allows you to express your wishes about legal, financial and end-of-life decisions.

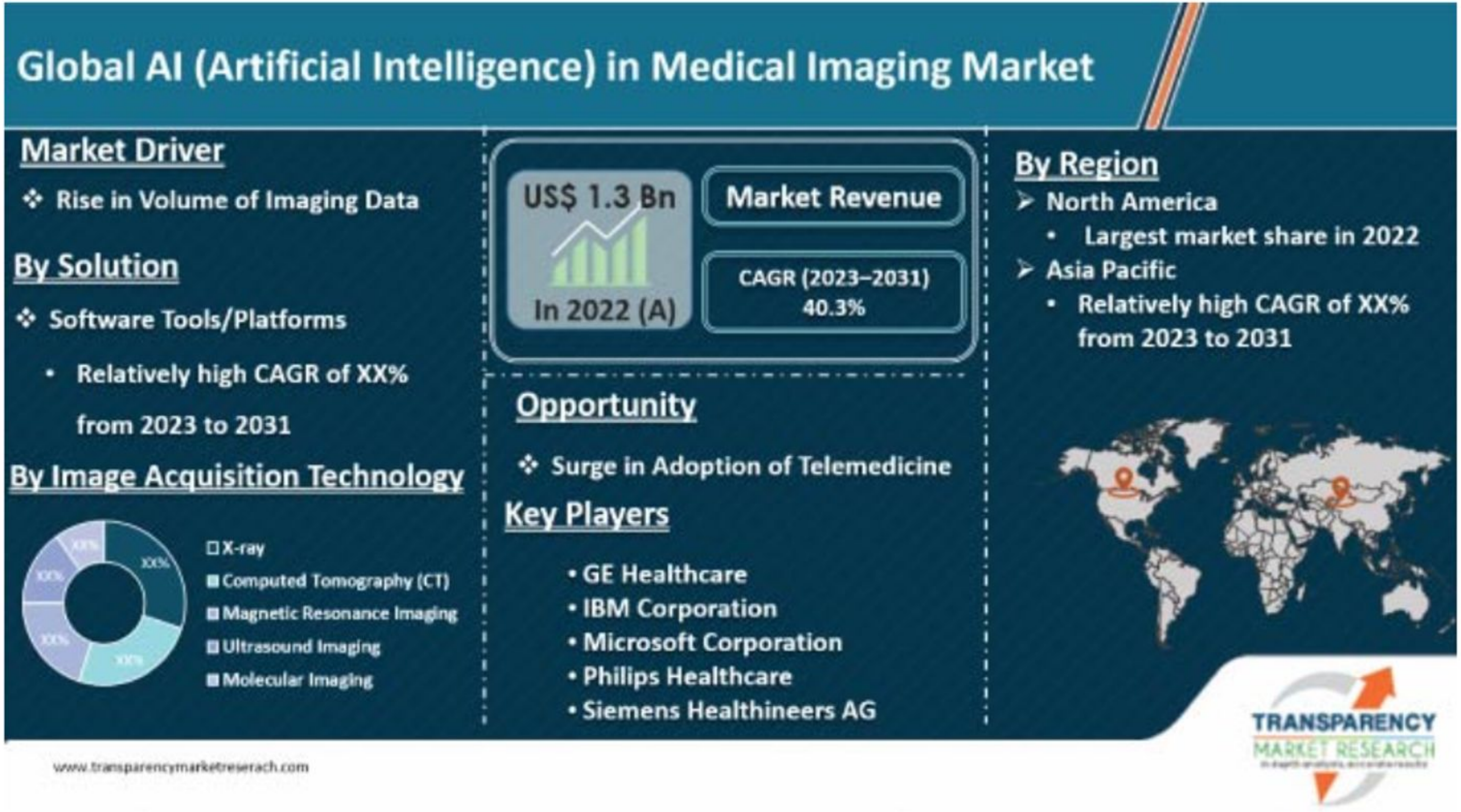
- **<u>Cost savings:</u>**
  - Early diagnosis saves costs of medical and long-term care for both families and the U.S. government.
  - Among all Americans alive today, if those who will get Alzheimer's disease were diagnosed when they had mild cognitive impairment, before dementia, it would collectively save approximately $7 trillion* in health and long-term care costs.

Use of AI in medical imaging

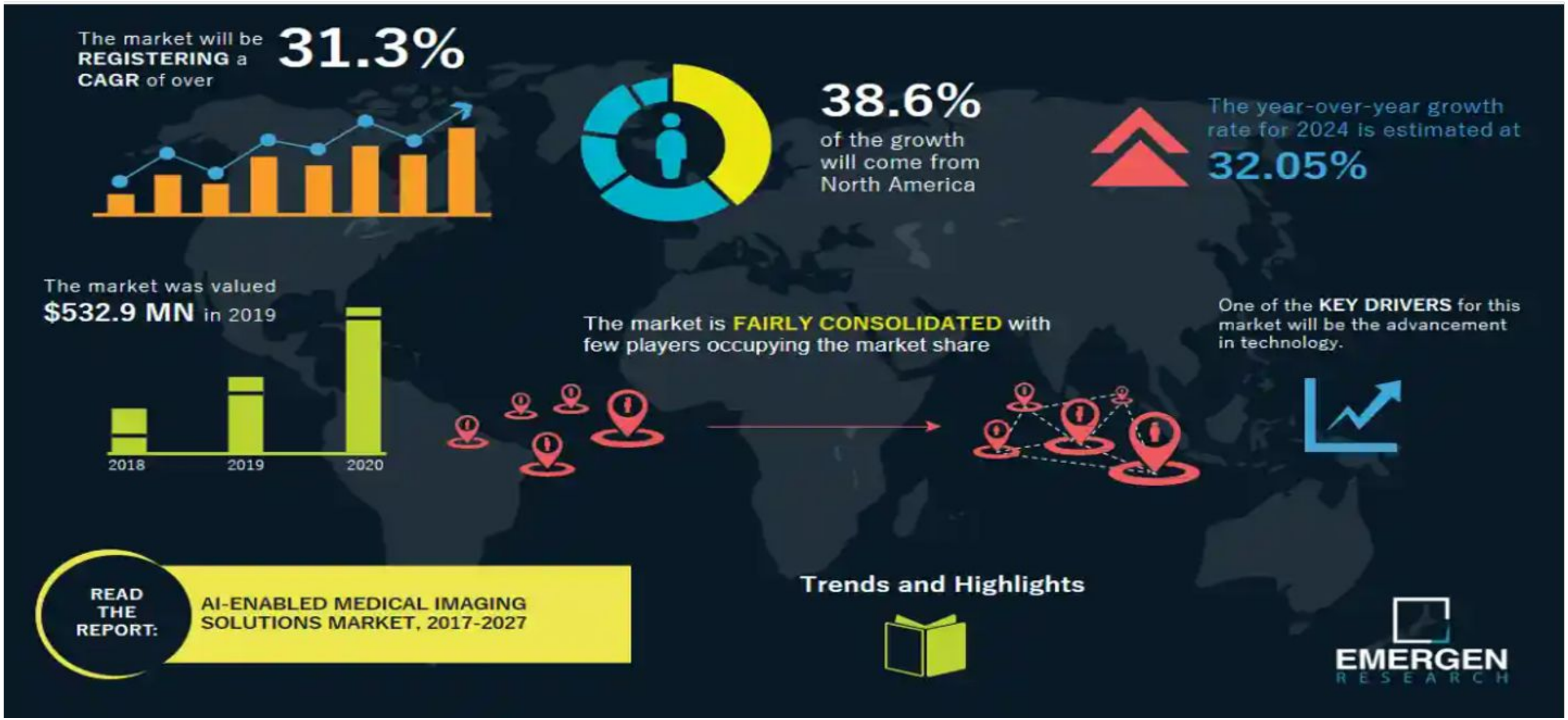
"AI will lead to increased use of quantitative imaging and structured reports, but he noted that it's a far cry from replacing the radiologist. Instead, deep-learning algorithms will be tools for the radiologist. While computers can do things humans cannot, that argument works both ways; it also makes the case for computers working hand in hand with humans"

Dr Eliot Siegel , M.D., RadSite's Chief Technology Officer and Standards Committee Chair, Podcast 21st Feb 2017. [(gotowebinar.com)]

AI in medical imaging-Industry review



AI in medical imaging-Industry review



Project Goals :

- The goal of this project is to leverage the power of **artificial intelligence**, specifically **machine learning and computer vision techniques**, to analyze brain scan images for the early detection and diagnosis of Alzheimer's disease.

- Our aim is to create an AI model that can **analyze these images**, **identify patterns** that may be indicative of these disorders, and **make predictions** with high accuracy.

- The expectation is that such a tool could **supplement existing diagnostic practices**, providing a more objective and potentially earlier indication of these diseases.

Our Approach :

Project Timeline

1. Data Acquisition
2. Data Preprocessing & Exploration
3. Model Selection & Baseline Development
4. Optimizing the Model
5. Validation & Model Deployment
6. Documentation & Presentation

Tools Used :

1. **Discussions and version-controlling:**
    - Slack
    - Dagshub
    - Google Workspaces
2. **Data Pre-processing & Manipulation**
    - Nibabel
    - OpenCV
    - Scikit-Image
    - Keras
3. **Data Visualization**
    - Matplotlib
    - Seaborn
4. **Augmentation**
    - imgaug
5. **Modelling**
    - Tensorflow
    - Pytorch
    - CNN Transfer learning
6. **Model Testing and Evaluation**
    - DeepChecks
7. **Deployment**
    - Streamlit
    - HuggingFace Spaces
    - Docker and FastAPI

Data Sources & Data Sets :

DATASET	DESCRIPTION	DATA SOURCE
Alzheimer_s Dataset	The data consists of MRI images. The data has four classes of images both in training as well as a testing set: 1. **Mild Demented** 2.**Moderate Demented** 3. **Non Demented** 4. **Very Mild Demented** There are a total of 6400 images.	KAGGLE : https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images
ADNI Baseline 3T-Pre-processed	Collections of uniformly pre-processed images of 300 patients in NIFTI(.nii) format belonging to 3 classes : **AD(Alzheimer's), CN(Control Normal)** and **MCI(Mild Cognitive Impairment)**	ADNI : https://ida.loni.usc.edu/login.jsp?project=ADNI

EDA, PRE-PROCESSING AND AUGMENTATION

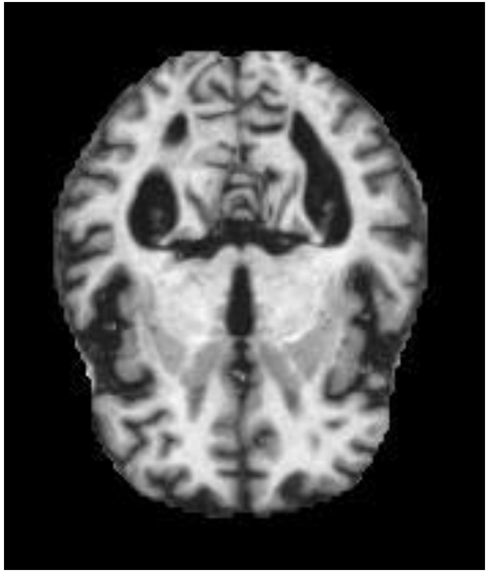
KAGGLE data Exploration

- Kaggle data has 4 classes **MildDemented**, **ModerateDemented**, **NonDemented** , **VeryMildDemented**.

- Dataset has a total of 6400 grayscale images of dimension **(208, 176).**

- These are pre-processed MRI images. Pixel intensity varies between **(0,255)**

- This dataset is finalised for analysis since there are many images to work with, compared to ADNI where we have less number of Subjects.

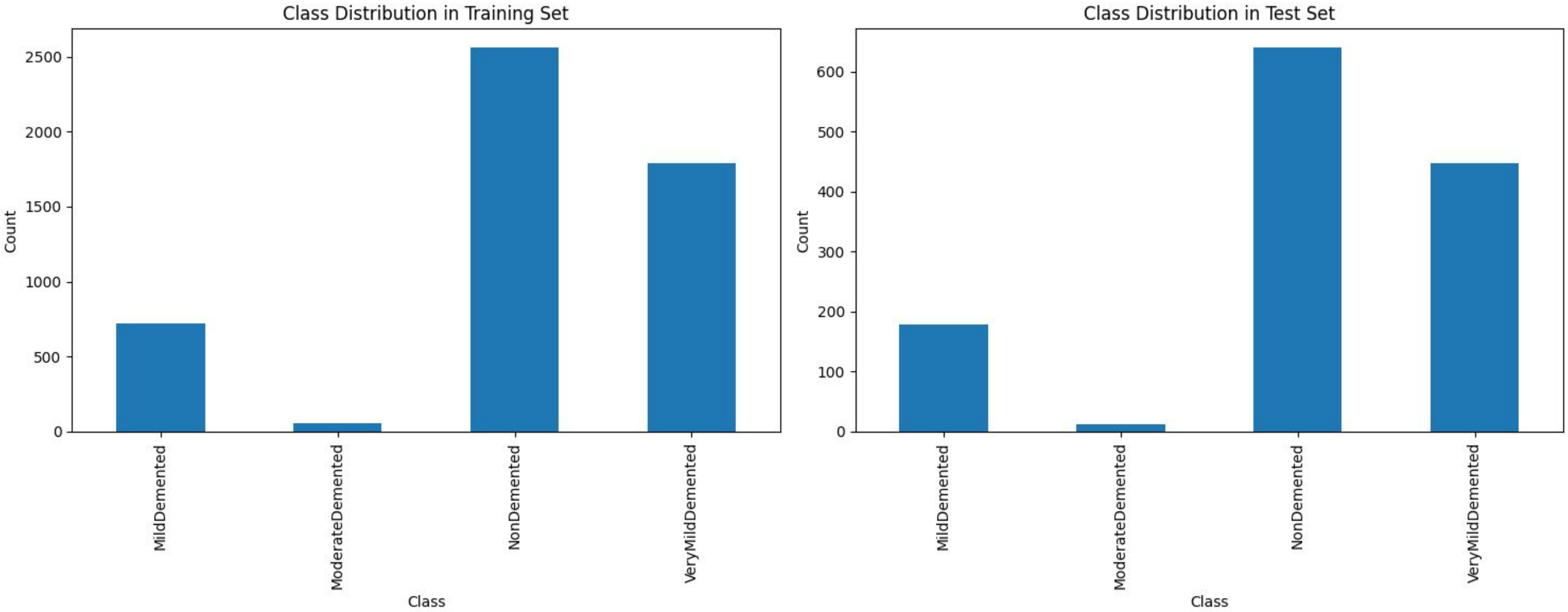


A ModerateDemented Scan

KAGGLE data Exploration

The dataset is highly **class imbalanced** which can lead to biased results and model interpretation. Thus, there is a need for **data augmentation** or **weighted loss function**. The distribution of classes in training and testing folders is shown below.

	Training Data	Testing Data	Class Totals
NonDemented	2560	640	3200 **(50 %)**
VeryMildDemented	1792	448	2240 **(35 %)**
MildDemented	717	179	896 **(14 %)**
ModerateDemented	52	12	64 **(1 %)**
Train Test totals	5121 **(80 %)**	1279 **(20 %)**	6400



KAGGLE data class imbalance problem

ADNI data Exploration

- Three classes are most widely used for the classification task:
  - **AD**: Alzheimer’s disease.
  - **MCI**: Mild-Cognitive Impairment.
  - **CN**: Control Normal.

- CN are patients with no evidence of Alzheimer’s present , MCI patients show mild symptoms and have a chance to convert to AD with age or disease. Progression.

- The sMRI are collected in a 3D manner such that we get images along the 3-axes (**Axial**, **Saggital**, **Coronal**). Each axis is a **Voxel** and the .nii file dimension are **(256,256,170)** , this represents the number of slices in each voxel.

- All the information on the subjects and data collected is present in the .csv file. **Image Data ID** is unique while **Subject** can repeat, depending on the category of scans like **MPRAGE**, **MPRAGE-repe**, **Survey**, etc.

ITK-SNAP is used to visualise a patients NIFTI(.nii) file. It shows the volume slices across the 3D planes **Axial, Sagittal** and **Coronal.**

Image Data ID	Subject	Group	Sex	Age	Visit	Modality	Description	Type	Acq Date	Format	Downloaded
[redacted]	[redacted]	AD	M	80	bl	MRI	MPRAGE repe	Original	5/30/2006	DCM	11/29/2023
[redacted]	[redacted]	AD	M	80	bl	MRI	SURVEY	Original	5/30/2006	DCM	11/29/2023
[redacted]	[redacted]	AD	M	80	bl	MRI	MPRAGE 3dtf	Original	5/30/2006	DCM	11/29/2023
[redacted]	[redacted]	AD	M	80	bl	MRI	SURVEY	Original	5/30/2006	DCM	11/29/2023
[redacted]	[redacted]	AD	M	80	bl	MRI	SURVEY	Original	5/30/2006	DCM	11/29/2023
[redacted]	[redacted]	AD	M	57	bl	MRI	SURVEY	Original	6/07/2006	DCM	11/30/2023
[redacted]	[redacted]	AD	M	57	bl	MRI	MPRAGE repe	Original	6/07/2006	DCM	11/30/2023
[redacted]	[redacted]	AD	M	57	bl	MRI	SURVEY	Original	6/07/2006	DCM	11/30/2023
[redacted]	[redacted]	AD	M	57	bl	MRI	SURVEY	Original	6/07/2006	DCM	11/30/2023
[redacted]	[redacted]	AD	M	57	bl	MRI	SURVEY	Original	6/07/2006	DCM	11/30/2023

.csv file showing the patients metadata and scan information.

KAGGLE data Pre-processing

The pre-processing steps included:

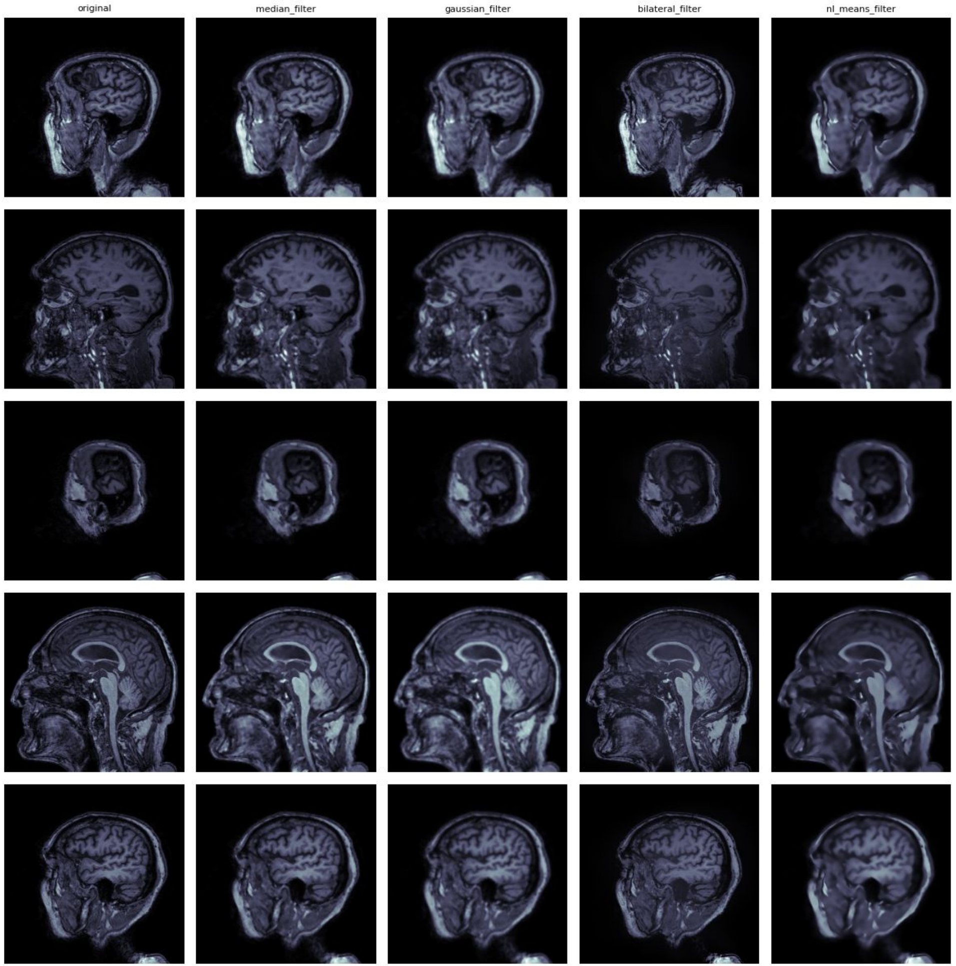
- Conversion of images from DICOM to PNG.
- Resampling and Noise Reduction filtering.
- Intensity Normalization
- Feature Extraction
- Classical Segmentation
- Image Augmentation

Image Conversion

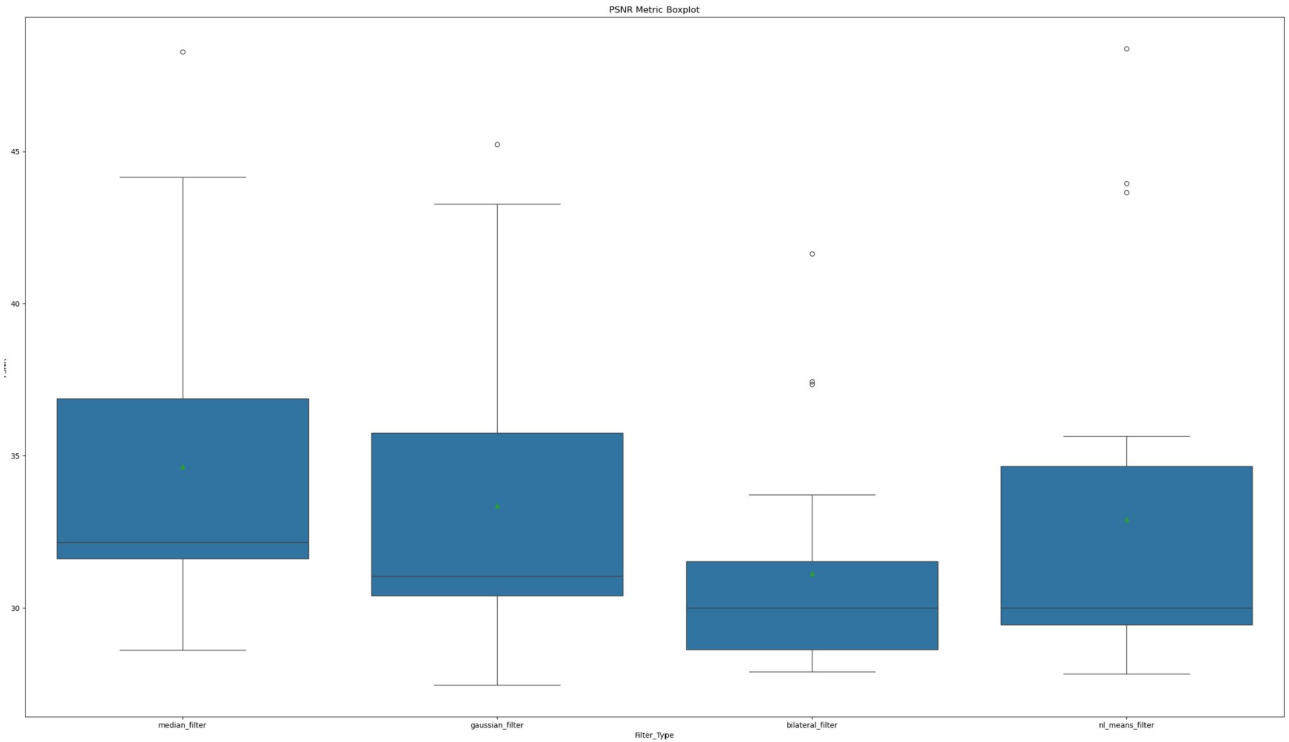
- This tasks helps combine data from various sources into one file/folder format.
- Function `dicom_to_png` is defined to convert DICOM images to PNG.
- The function converts the .dcm images to **normalized** images with intensities **(0,255)** and converts them to **uint8** format.
- Creates PNG output directories corresponding to each DICOM input directory. Output PNG files are saved in the respective directories.

Resampling and Noise Reduction

- **simpleITK** is used for resampling images to a consistent voxel size.
- Noise in medical images can compromise diagnostic accuracy and impede feature extraction. Various filters are employed to reduce noise and enhance image quality. The filters used are:
  - **Median Filter**
  - **Gaussian Filter**
  - **Bilateral Filter**
  - **Non-local Means Filter**
- **Peak Signal-to-Noise Ratio (PSNR)** is used as a metric quantifying the quality of an image by measuring the ratio of signal strength to noise.



Comparison of different filtering **Median**, **Gaussian**, **bilateral** and **nl_means** and the original image on a sample .dcm images.



PSNR metric boxplots of different filtering **Median**, **Gaussian**, **bilateral** and **nl_means.**

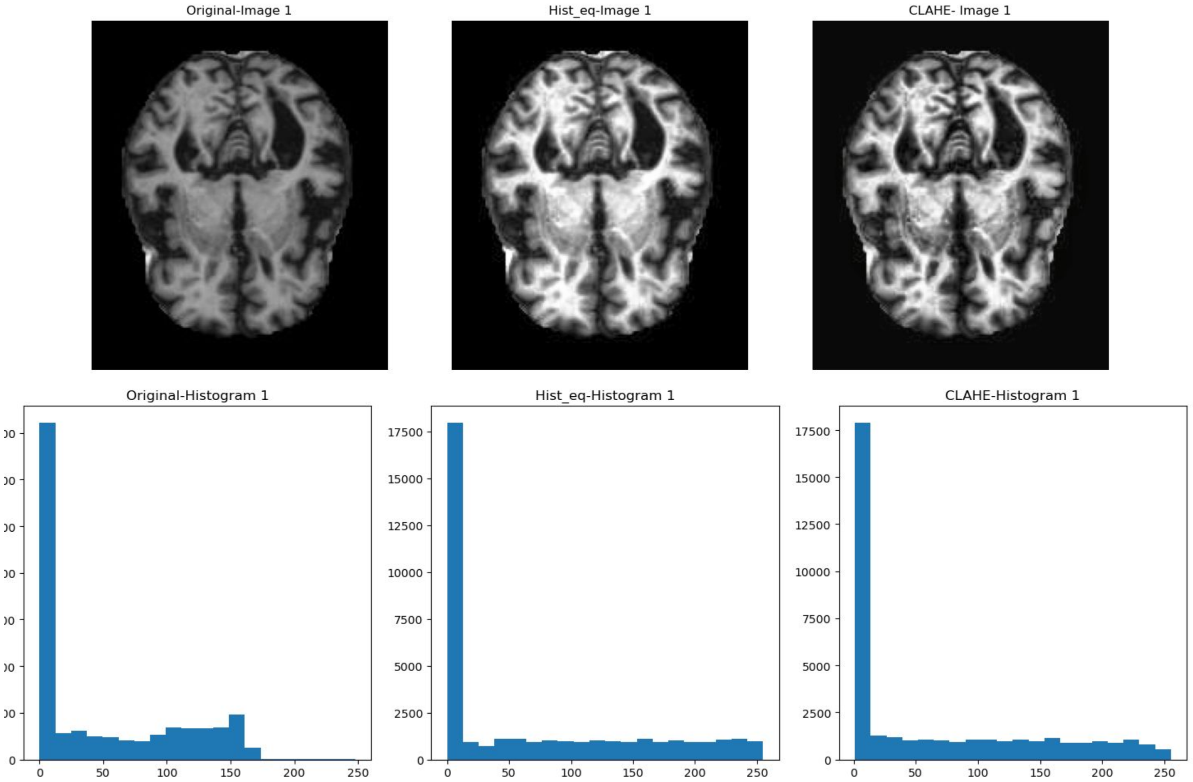
Intensity Normalization

- **Histogram Equalization and CLAHE (**Contrast Limiting Adaptive Histogram Equalization**):** Histogram equalization stretches the **pixel intensity histogram of an image to the full range**, enhancing contrast. CLAHE performs histogram equalization in **local contrast ranges**, preventing image washout and preserving features.

- **Z-score normalization** subtracts the mean and divides by the standard deviation, standardizing pixel intensities to a normal distribution.

- **Zero-One Normalization:** Zero-one normalization subtracts the minimum pixel intensity and divides by the intensity range, scaling pixel values to the range 0 to 1.

- **Percentile Normalization:**Percentile normalization subtracts a percentile value and divides by the difference between percentiles, scaling pixel values to the range 0 to 1.



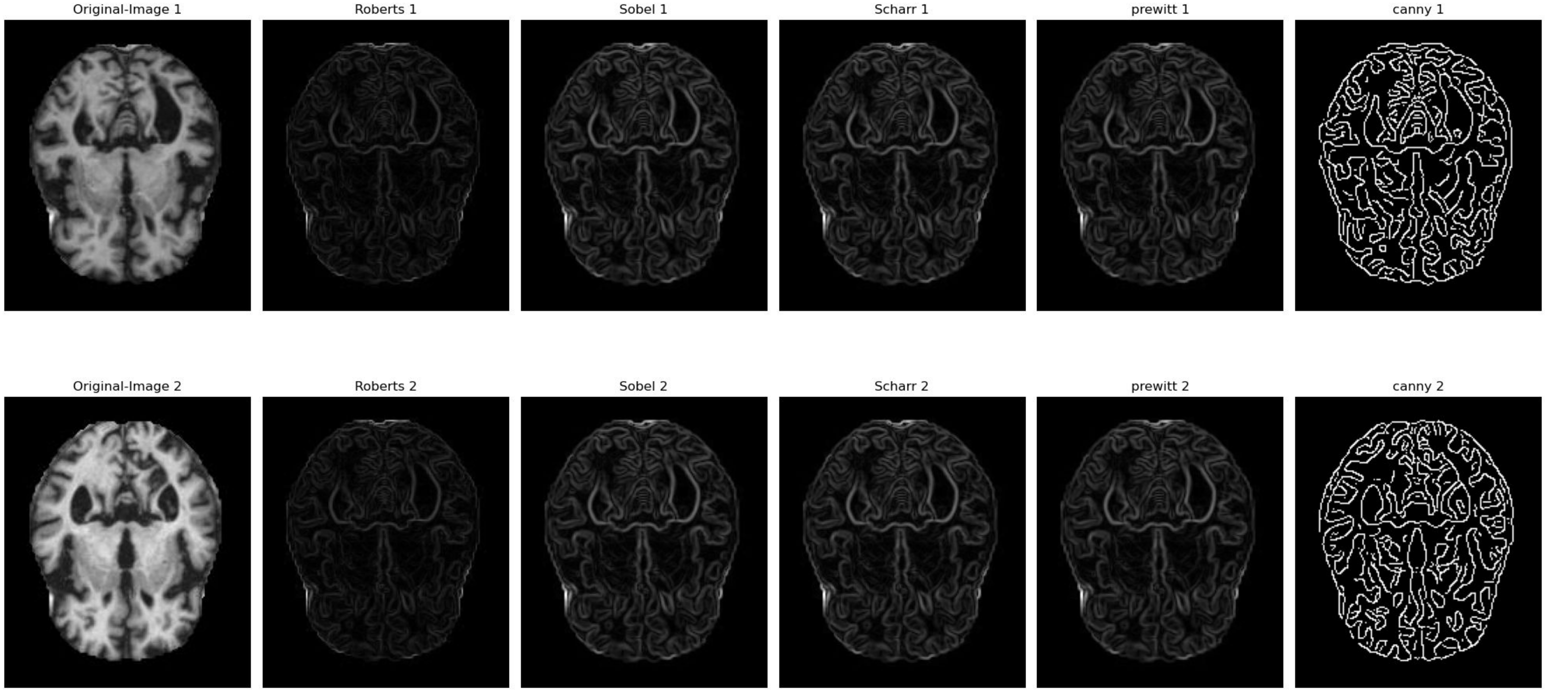
Histogram Equalization

Note how histogram equalization and CLAHE methods stretch the intensity histograms making the overall image sharper and having high contrast.

Feature Extraction

Edge Detection

- **Roberts**: Detects edges by highlighting areas with rapid changes in pixel intensity.
- **Sobel**: Highlights areas with high first-order derivatives in both horizontal and vertical directions.
- **Scharr**: Provides better edge detection results, especially for images with high-frequency content.
- **Prewitt**: Similar to Sobel but with a slightly different kernel.
- **Canny**: Multi-stage algorithm for high-quality edge maps with minimal noise. It results in binary edge map.



Feature extraction with edge detection

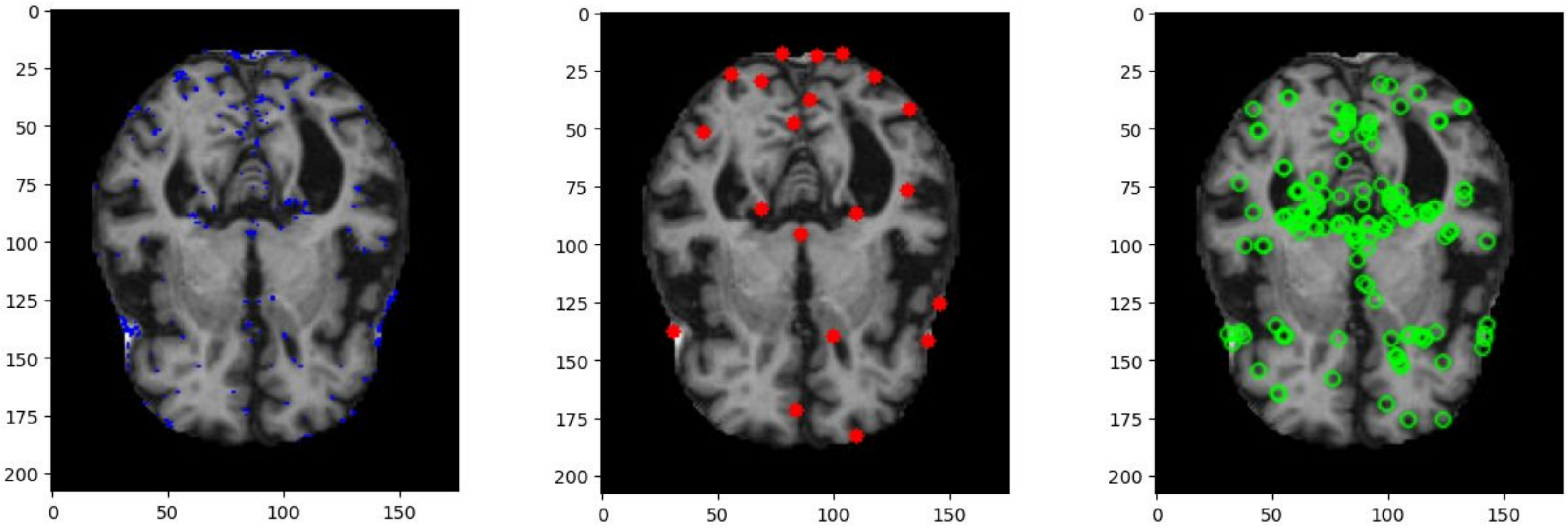
Feature Extraction (contd.)

Corner Detection

- **Harris Corner Detection**: Identifies corners by detecting large variations in intensity using a sliding window approach.
- **Shi-Tomasi Corner Detection**: Similar to Harris, with a different computation for the score value R, capable of finding the best corners in an image.

Keypoint Detection

- **SIFT (Scale-Invariant Feature Transform)**: Widely used for detecting corners, blobs, circles, and scaling images.
- **ORB (Oriented FAST and Rotated BRIEF)**: Utilizes FAST and BRIEF algorithms for one-shot facial recognition and keypoint matching based on intensity variations.



Feature extraction with Harris corner detection (left), Shi-Tomasi (middle) and ORB keypoint detector (right)

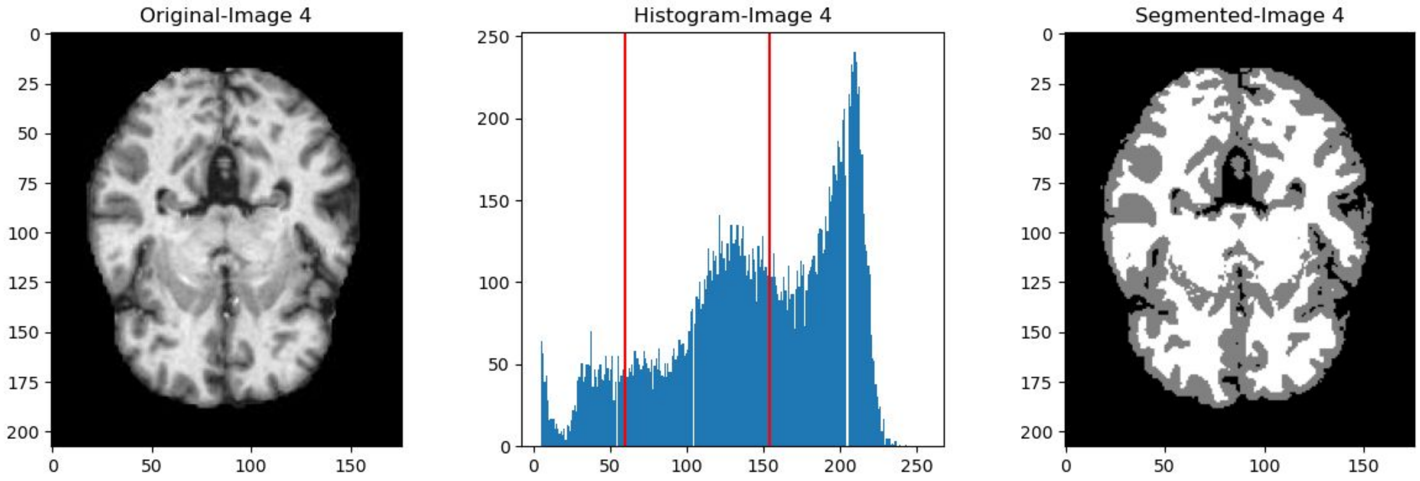
Classical Segmentation

Image segmentation helps us identifying **regions of interests** in the scans.The methods used are:
Multi-Otsu Thresholding:

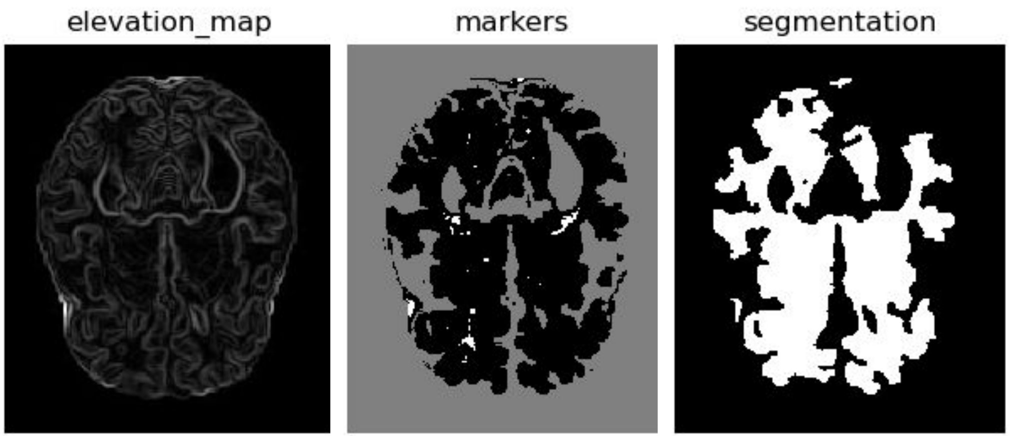
- Algorithm for segmenting pixels into **multiple classes** based on intensity. It determines several **thresholds**(based on input) to classify pixels into different intensity levels and returns threshold values, typically represented by a red line in the histogram.

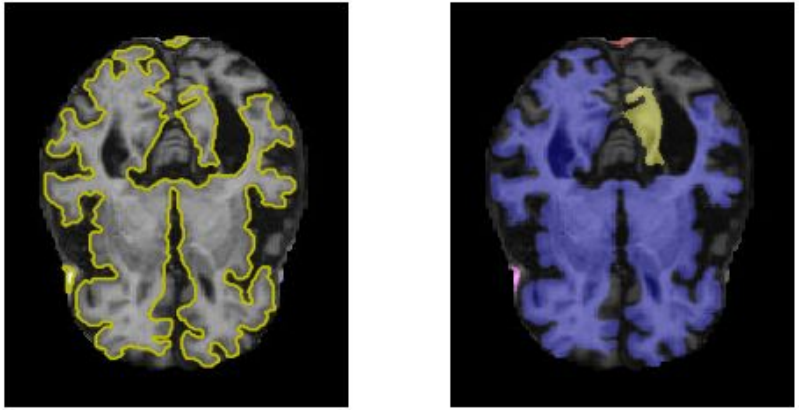
Region-based Segmentation:

- Sobel filter generates **elevation map** highlighting regions. **Markers** are placed strategically to guide segmentation. **Watershed algorithm** segments image based on marker guidance. **Fills holes** in regions and labels connected components for comprehensive segmentation.



Multi-Otsu Thresholding with three classes





Region based Segmentation

Showing the **elevation map**, **markers** and **segmentation binary** (top) and the final segmentation image (bottom).

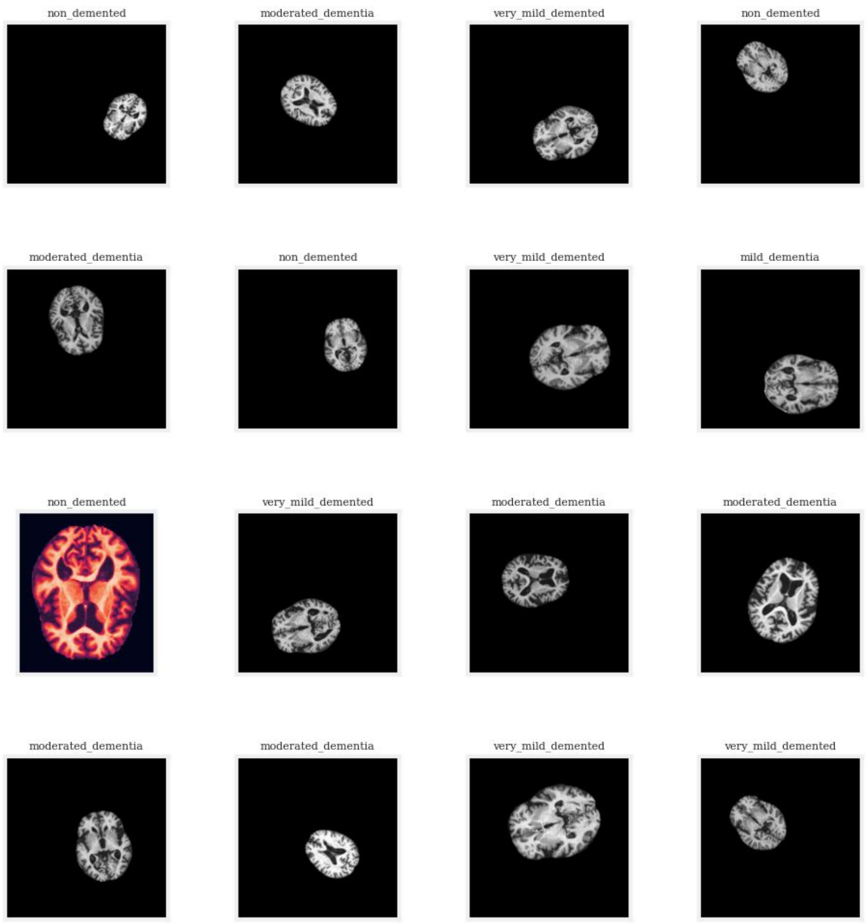
Image Augmentation

The image augmentation methods used are:

- **Keras Image Generator Augmentation:** Used during model training to address class imbalance by **scaling** and **geometric transformations**.

- **Cut, Paste, and Learn Synthesis:** Used for **synthetic data generation** to increase dataset diversity and reduce class imbalance problem.

- **Combined Augmentation Techniques:** Utilized **imgaug package** for a mix of augmentation methods to enhance dataset variability and model robustness. The methods included **Horizontal and vertical flipping** , **Scaling** and **translation** , **Brightness** and **contrast adjustment** , **Gaussian blurring**, **Additive Gaussian noise**, **Saturation adjustment**, **Shear**, **Contrast-limited adaptive histogram equalization (CLAHE).**



Omdena

Example augmented dataset using various geometric transformation and scaling methods.

MODEL DEVELOPMENT

Model Architecture

- A custom **Convolutional Neural Network(CNN)** architecture was used with two **convolution** layers each of which is followed by a **max pooling** layer, **batch normalization** layer and a **dropout** layer, used for feature extraction and later followed by dense layers.

- **ReLU (Rectified Linear Unit)** activation function was used to **introduce non-linearity**, aiding in capturing complex patterns.

- **Softmax** activation function is used in the last dense layer with 4 neurons representing the 4 classes of the dataset needed for **multi-class classification**.

Model Architecture

- **Max pooling** layers helps **downsample** the spatial dimensions of the input, retaining important features while reducing computational complexity.

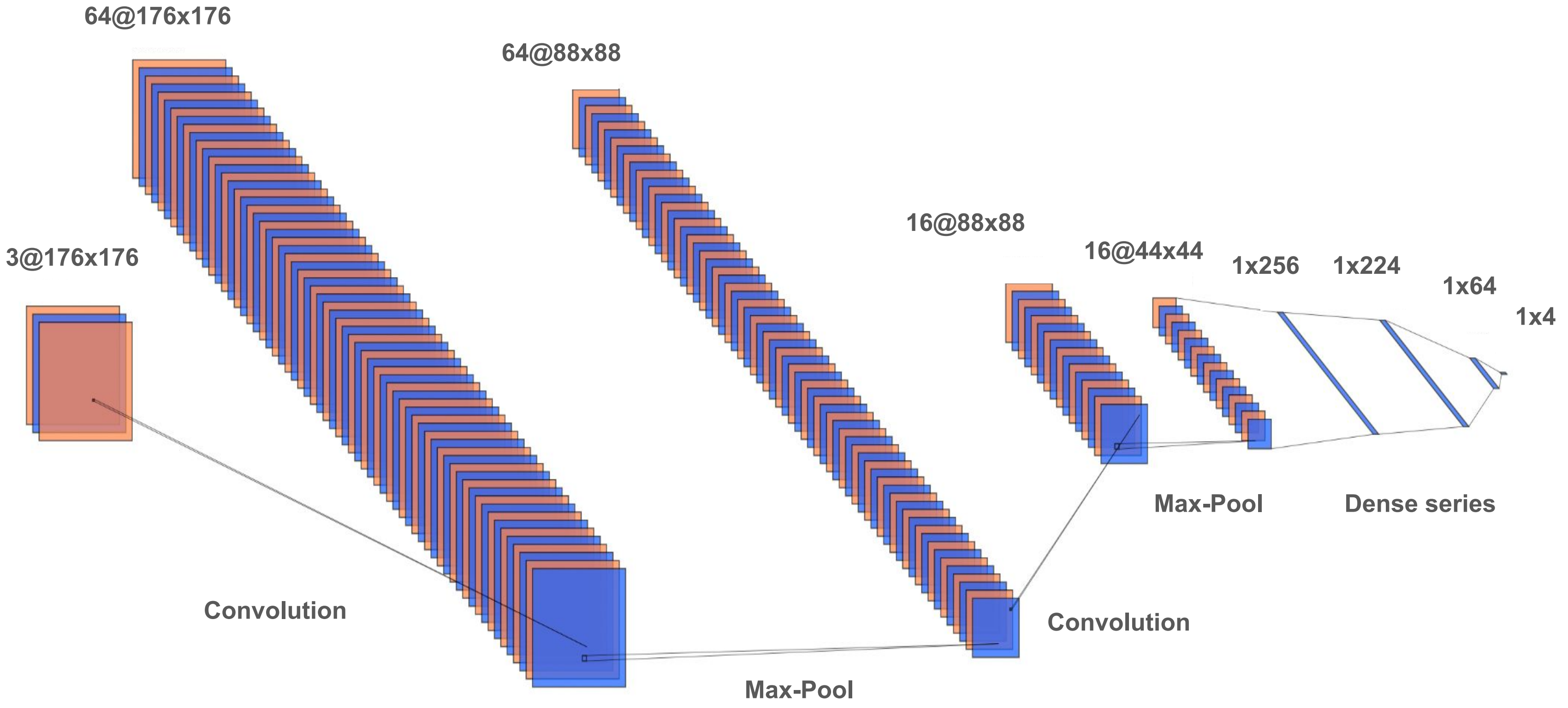
- **Batch normalization** layers normalizes activations within each layer, reducing **internal covariate shift** and accelerating training convergence.

- **Dropout** layers are used as **regularization measures** to prevent overfitting by randomly dropping a fraction of connections during training.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 176, 176, 64)      1792

 max_pooling2d (MaxPooling2  (None, 88, 88, 64)        0
 D)

 batch_normalization (Batch  (None, 88, 88, 64)        256
 Normalization)

 dropout (Dropout)           (None, 88, 88, 64)        0

 conv2d_1 (Conv2D)           (None, 88, 88, 16)        25616

 max_pooling2d_1 (MaxPoolin  (None, 44, 44, 16)        0
 g2D)

 dropout_1 (Dropout)         (None, 44, 44, 16)        0

 flatten (Flatten)           (None, 30976)             0

 dense (Dense)               (None, 256)               7930112

 batch_normalization_1 (Bat  (None, 256)               1024
 chNormalization)

 dense_1 (Dense)             (None, 224)               57568

 dropout_2 (Dropout)         (None, 224)               0

 dense_2 (Dense)             (None, 64)                14400

 batch_normalization_2 (Bat  (None, 64)                256
 chNormalization)

 dropout_3 (Dropout)         (None, 64)                0

 dense_3 (Dense)             (None, 4)                 260

=================================================================
Total params: 8031284 (30.64 MB)
Trainable params: 8030516 (30.63 MB)
Non-trainable params: 768 (3.00 KB)
_________________________________________________________________
```

Model Summary

This summary shows the two convolution layers. The first layer has **64** filters with a kernel size of **(3 x 3)** with padding parameter set to same.
The second convolutions step has **16** filters with a filter size of **(5 x 5)** again with same padding.
This summary shows the number of parameters for each layer.



Schematic diagram of the CNN architecture

(Generated using [NN-SVG](NN-SVG))

Data Imbalance and SMOTE

- **Synthetic Minority Over-sampling Technique (SMOTE)** is a method to address the class imbalance problem.

- The SMOTE algorithm synthesizes new minority class instances by **interpolating between existing minority class instances**. This helps balance the class distribution.

- SMOTE helps prevent the model from being **biased towards the majority class** by generating synthetic examples of the minority class.

Hyperparameter Optimization using KerasTuner

- **Keras Tuner**, a **hyperparameter optimization tool**, is used to efficiently search for the optimal set of hyperparameters for the convolutional neural network architecture.

- Keras Tuner employs **Bayesian optimization**, a probabilistic model-based approach, to iteratively explore the hyperparameter space and identify configurations that maximize the validation accuracy of the model.

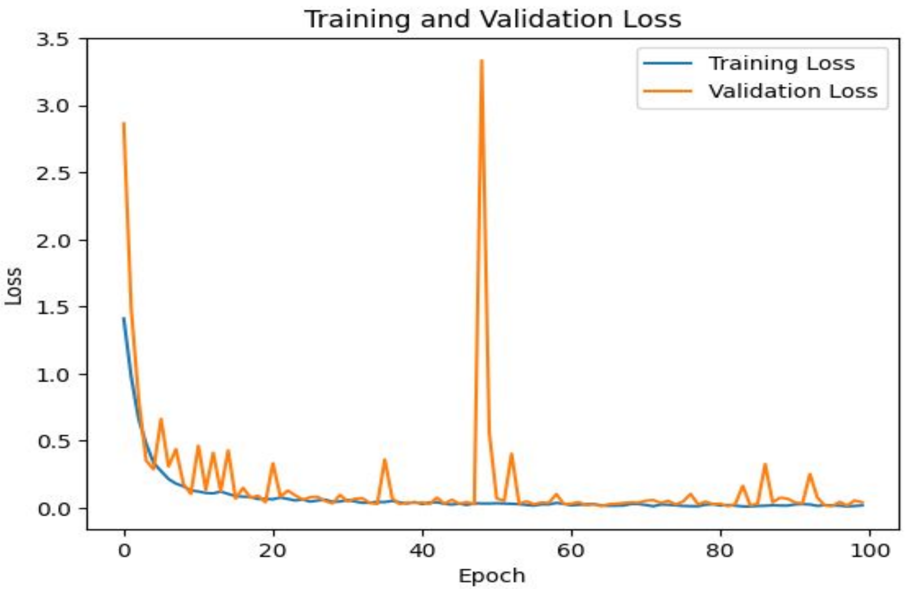
- Some of the parameter space used to iterate over are : **Number of Convolutional layer (range 1 to 15)** , **Number of filters for each convolutional layer (16 to 128)** with **filter size ranging from ((3 x3) to (5 x5))** etc. Similarly the **dropout rate**, **number of dense layer neurons etc.** are also varied to find the optimal parameters.

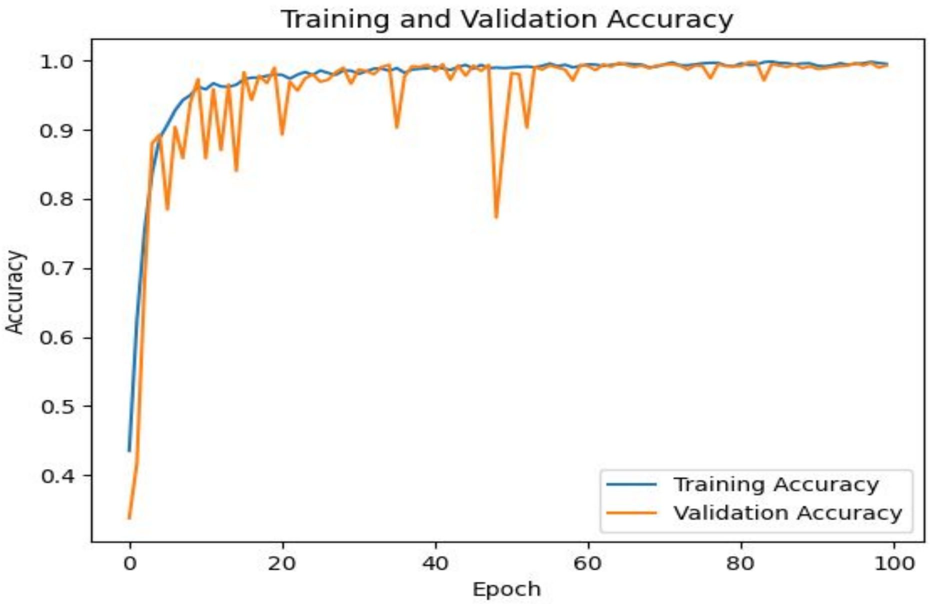
- The model is then trained for **10 epochs** and **20 max iterations** of the combinations to find the best model.

Model Training

- The dataset is initially split into **train** data(80%) and **test** data(20%). The train data is again split to create the **validation** data(20% of training data)
- After data imbalance problem was solved, the model was trained with 100 epochs and with a batch size of 16.
- **KerasTuner** is used for hyperparameter tuning in this model.
- The **learning rate** is set to 0.002 and **Adam** optimiser is used to train the model.
- **Categorical Cross-entropy** loss function is used as the suitable loss function for multi-class classification problem.
- The **accuracy** metric of the training and validation datasets were monitored.

Model Training





The decrease in loss function and increase in accuracy is plotted against epochs for both the training and validation datasets.

Other Approaches

Besides the final model a few more approaches were tried:

- **Transfer learning** using pre-trained models like **VGG19** and **EfficientNetV2.**
- Using **fast.ai** library and using the pre-trained models: **resnet18**, **convnext_tiny_in22k**, **vgg16**, and **regnetx_080.**
- Use of **Error Level Analysis(ELA)** for image classification and **ResNet50** pre-trained model.
- Use of other loss functions like **weighted cross-entropy loss** to tackle class imbalance.

MODEL TESTING AND EVALUATION

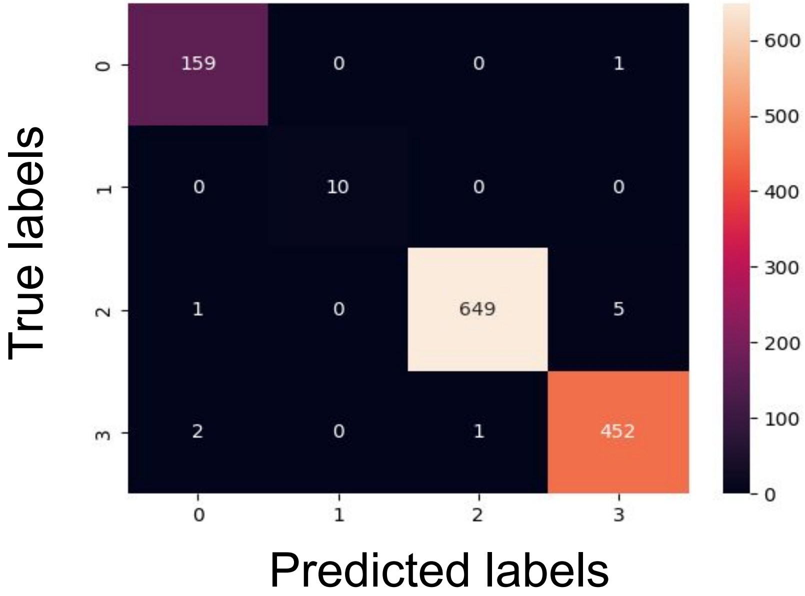
After the model is trained for 100 epochs it performed relatively well on the training and validation data as can be inferred from the metrics and loss values.

	Training Data	Validation Data
Loss	0.0175	0.0404
Accuracy	0.9946	0.9922

To check how well the model generalizes to unseen data or if the model overfit the training data we need to compute the metrics on the test dataset.

Model Testing

Upon testing the model on our test data, we get an accuracy of **0.9922.**



Confusion Matrix

The confusion matrix shows that the model predicts most of the labels correctly.
Also we note that the minority class has been well classified.

Classification Report

The classification report helps evaluate the fitted model using the following metrics :

- **Precision:** Reflects the accuracy of positive predictions, indicating the proportion of correctly predicted instances among all instances predicted as positive.
- **Recall:** Represents the sensitivity or true positive rate, showing the proportion of correctly predicted instances among all actual positive instances.
- **F1-score:** Harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives.
- **Support:** The number of instances in each class in the test set.
- **Accuracy:** Overall accuracy of the model across all classes.
- **Macro Avg:** Average of precision, recall, and F1-score calculated for each class independently and then averaged, treating all classes equally.
- **Weighted Avg:** Similar to macro average but considers the number of instances in each class, providing a weighted average that accounts for class imbalances.

Classification Report

	precision	recall	f1-score	support
0	0.99	0.98	0.99	162
1	1.00	1.00	1.00	10
2	0.99	1.00	0.99	650
3	0.99	0.99	0.99	458
accuracy			0.99	1280
macro avg	0.99	0.99	0.99	1280
weighted avg	0.99	0.99	0.99	1280

Classification Report on the test dataset

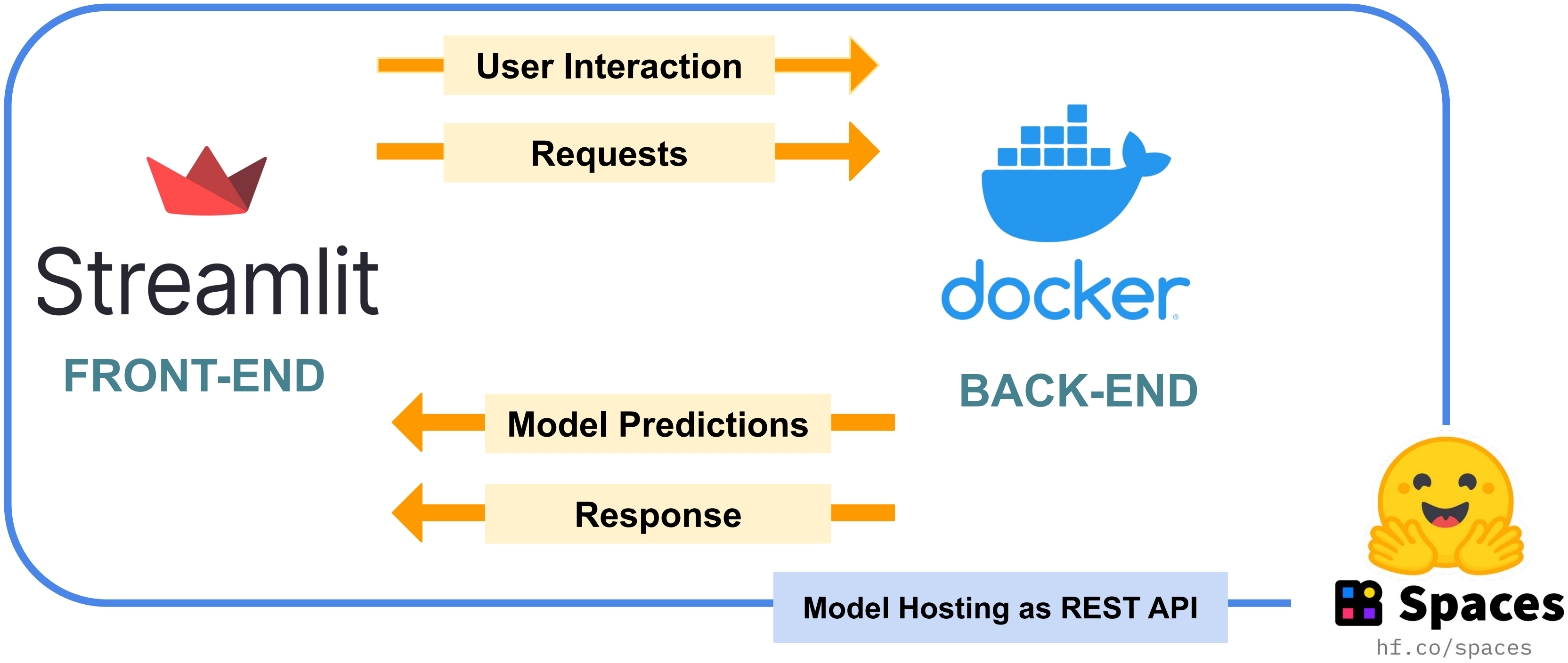
The numbers in the report suggest excellent performance for each class and an overall high accuracy of the model in the multi-class classification task.
The high precision, recall, and F1-score values indicate the model's ability to make accurate predictions across different classes.

DEPLOYMENT

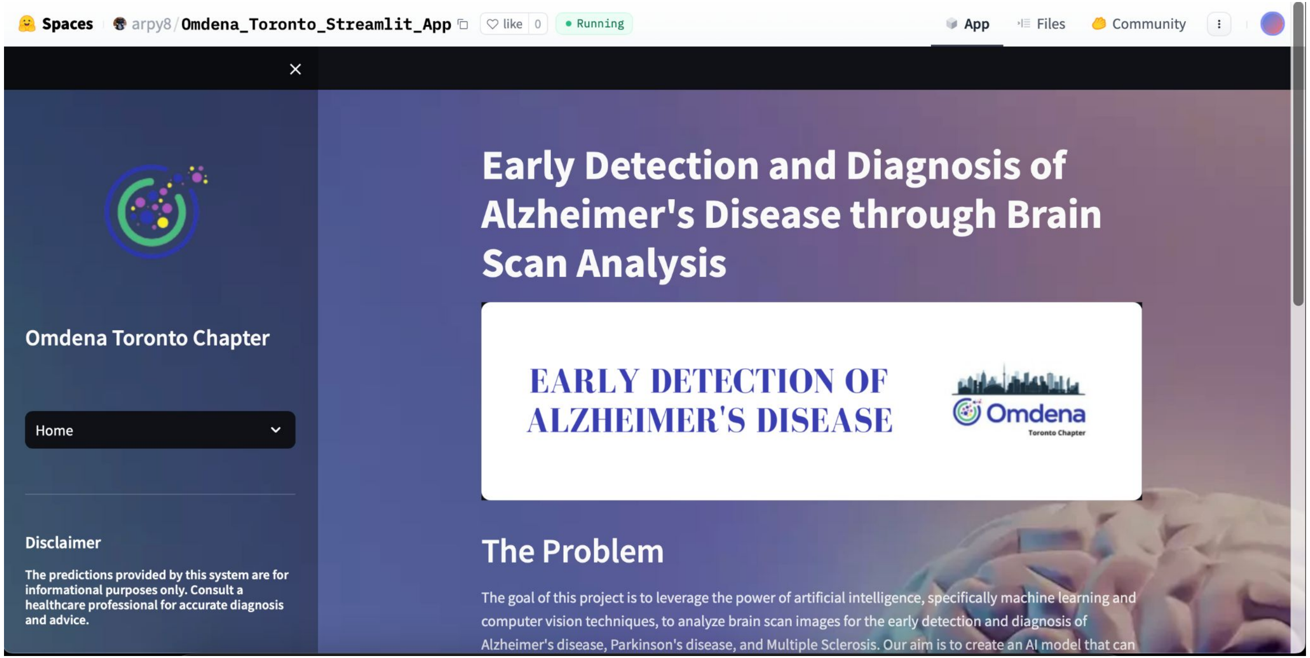
Deployment Architecture

The deployment architecture involves a separation of the front-end and back-end components, ensuring a modular and scalable structure.

- **Front-End:** Utilizes the **Streamlit** framework for creating an interactive and visually appealing user interface.
- **Back-End:** Utilizes a **Docker container** to encapsulate the machine learning model and serve it as a **REST API**. It abstracts away the complexities of environment setup and dependency management, making deployment more streamlined and efficient.
- **Deployment:** The deployment of the application is done on **Hugging Face Spaces**, a platform that offers streamlined hosting and sharing of machine learning models and applications as **RESTful APIs**.


User Interaction
Requests
Streamlit
FRONT-END
docker
BACK-END
Model Predictions
Response
Model Hosting as REST API
Spaces
hf.co/spaces

Hugging Face Spaces deployment of Streamlit App



Deployment(Link):

One can see the model prediction by going to the **Model** section and uploading an image of the **MRI Scan** in **jpg, png** or **jpeg** format. The model responds with the **class** of Alzheimer's with the **prediction probability**.

CONCLUSIONS

Summary :

- **Data Collection**: For the Alzheimer's classification task, data is collected from various sources like **Kaggle**, **ADNI** etc.. The Kaggle dataset is used for further analysis due to its large number of well annotated images. The data had 6400 images belonging to 4 classes and was **shuffled** then **split** into train, test and validation sets.

- **Data Pre-processing** : Various methods were applied to enhance the images such as using **filters** to sharpen the images and contrast and intensity improving methods. Classical **feature extraction** and **segmentation** methods were used to find prominent sections before model building.

- **Data Augmentation:** To tackle class imbalance problem various data augmentation methods were tried out like **Keras image generator** augmentation during model training, **Cut, Paste and Learn** synthesis methods or using a combination of methods like **flipping (horizontal and vertical), scaling and translation, brightness and contrast adjustment, Gaussian blurring, additive Gaussian noise, saturation adjustment, shear, and contrast-limited adaptive histogram equalization (CLAHE).** The keras method was used to get the final results.

Summary contd. :

- **Model Development:** Custom **Convolutional Neural Network (CNN)** architecture was used for the classification task. Various other methods such as **pre-trained models** were tried but the best results were obtained for the custom model. **SMOTE augmentation** method was tried yet model performed slightly better without use of SMOTE. **Keras Tuner** helped in hyperparameter tuning. **Accuracy** metric was used for monitoring the training.

- **Model Testing and Evaluation:** The final model produced good results on both the **validation** and **testing** dataset. Moreover, the classification report shows good metrics. **Accuracy** for test data is **99.22%**, along with good **recall** , **F1-score** and **precision**. This shows the model is not overfitted and also classifies most of the images correctly as can be inferred from the **Confusion matrix.**

- **Model Deployment:** The best fit model is then deployed to a **Hugging Face Space**, using a **Streamlit app** for front-end and **docker container** as back-end using **RESTful API**.

Problems Faced and Recommendations

Some problems faced and potential recommendations to enhance the prediction of the model are as follows:

- **Data Diversity:** Although the Kaggle data is well annotated and has a relatively small size with large number of images, use of **ADNI** data could shed more light on the real nature of the problem. As, the ADNI data is **clinically approved** and has relatively high quality **3D images** of various **modalities** the scope of the project could be further defined by using that dataset.

- **Data Leakage:** Since the Kaggle data is shuffled , the model may suffer from unwanted data leakage problem. To solve this issue we recommend either using the **ADNI data** or going with better **pre-processing** and **augmentation** steps to avoid such an issue and make the model better **generalizable to unseen data**.

- **Assistance of Subject experts:** Lastly, having assistance from **medical experts** in this domain can helps significantly understand the **problem statement** better and look for targeted solutions using better **feature-extraction** and **well known modelling methods** in this domain.

Project Resources

RESOURCES	LINKS
Dagshub Repository	https://dagshub.com/Omdena/TorontoCanadaChapter_BrainScanImages
Deployed App	https://huggingface.co/spaces/arpy8/Omdena_Toronto_Streamlit_App
Dataset Source	https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images

ACKNOWLEDGEMENT

We extend our sincere appreciation to our project lead **Kausthab Dutta Phukan** for his outstanding leadership and unwavering support throughout the duration of this project. His guidance has been instrumental in shaping our efforts and ensuring the project's success.

Additionally, we express our gratitude to all the **task leaders**, who diligently oversaw specific tasks with dedication and expertise, contributing significantly to our achievements.

Lastly, we extend our heartfelt thanks to **all the contributors** who dedicated their time, skills, and efforts to the project. Your collective contributions have been invaluable and have played a vital role in bringing this project to fruition.

THANK YOU