# Brain Tumour Classification through MRI image analysis

**Alkesh Shukla**(S20210020252), **N Rohit** (S20210020297), **Gaurav Anand** (S20210020273)

PR Semester Project

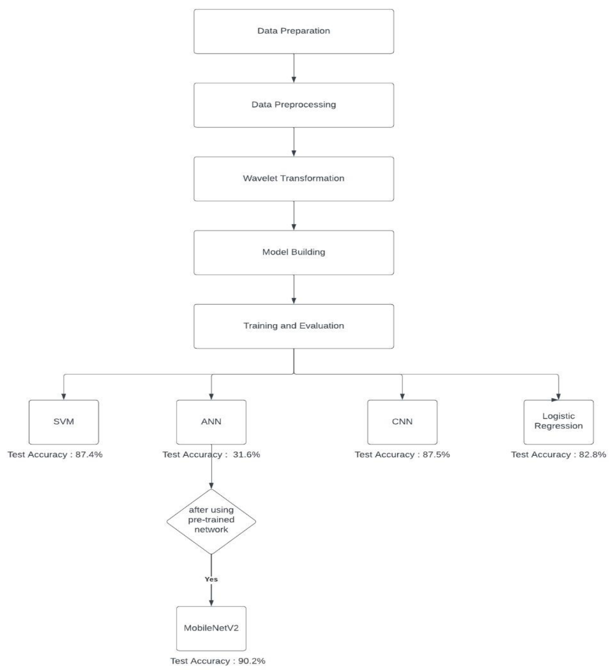
***Abstract*: This project employs deep learning and image processing to classify brain tumours. Extracting archives, building models (ANN, CNN, SVM, Logistic Regression), and incorporating wavelet transformations, it achieves accurate classification, providing a comprehensive solution for brain tumour image analysis.**

***Keywords***: Brain tumour, Wavelet Transformation, Artificial Neural Networks(ANN), Convolutional neural network(CNN), Support vector machines(SVM), Logistic Regression, MobileNetV2.

## I. Introduction

Brain tumours are regarded as a fatal condition that impacts the lives of so many people worldwide . The kind, location, and size of a brain tumour all affect how it will be treated. There are several forms of brain tumours, some of which are benign (non-cancerous), while others are malignant (cancerous). A benign tumour commonly referred to as a low-grade tumour does not significantly harm surrounding healthy tissues. On the other hand, a benign tumour is the contrary of a malignant tumour; in this case, the tumour cells directly cause the person's death and they can readily disseminate across the surrounding brain tissues. They are also known as high-grade tumours. One of the most prevalent tumour forms that develop in the brain and have the greatest fatality rate is glioma, which accounts for around 33% of all brain tumours. Gliomas can be classified into several categories based on their propensity for development and level of severity. This project endeavours to revolutionize brain tumour detection by leveraging a robust set of technologies and methodologies. It begins with meticulous data handling, segmenting MRI images into distinct training and testing sets. Preprocessing techniques, including grayscale conversion and standardizing image dimensions (240x240 pixels), ensure data uniformity for subsequent analysis. The implementation harnesses the power of Logistic Regression and Support Vector Machines (SVM), prominent machine learning models, for precise brain tumour classification. Rigorous evaluation through accuracy metrics scrutinizes the models' performance and their ability to generalize to new data. Visual representations vividly showcase the models' predictions, aiding in comprehension of their decision-making processes. Further analysis includes dissecting misclassified samples, unravelling insights into model limitations. This comprehensive approach aims to contribute significantly to the field of medical diagnostics, specifically in enhancing brain tumour detection through innovative technology and computational methods.

## II. Block Diagram



## III. METHODOLOGY

### 1. Data Preprocessing

Data preprocessing involves rescaling and normalization of images using the ImageDataGenerator in TensorFlow. Rescaling is achieved through dividing pixel values by 255. This process ensures uniformity in pixel scales, facilitating effective model training. Additionally, the code employs techniques like rotation, brightness adjustment, and horizontal flipping, enhancing the dataset through data augmentation.

### 2. Data Augmentation

The ImageDataGenerator is utilized for data augmentation in the code. It introduces variability into the training dataset by applying transformations like rotation, brightness adjustment, and horizontal flipping. These augmented images contribute to a more diverse dataset, preventing overfitting and improving the model's ability to generalize to unseen data.

### 3. Feature Extraction

Wavelet analysis is a powerful method for feature extraction in image processing. In the provided code, the 'bior1.3' wavelet is applied after converting the image to grayscale. This process decomposes the image into four components, capturing both low and high-frequency details in horizontal and vertical directions. These components serve as distinctive features for tasks like image compression and pattern recognition. The grayscale conversion is essential to focus on structural details rather than color.

### 4. Data Visualisation

The code incorporates data visualization techniques using pie charts and bar graphs to illustrate the distribution of different tumour types in the training and testing datasets. This exploratory data analysis aids in understanding class imbalances and guides decisions related to data sampling and representation.

### 5. Artificial Neural Network

The code implements an Artificial Neural Network (ANN) using the Sequential API from Keras. The architecture consists of flattened input layers, dense hidden layers with ReLU activation functions, and an output layer with SoftMax activation for multiclass classification. The model is compiled with the Adam optimizer and categorical cross-entropy loss function, suitable for the provided multi-class tumour classification task. Training involves multiple epochs with early stopping based on loss.

### 6. Convolutional Neural Networks

The CNN model is constructed using the Sequential API, incorporating convolutional layers, max-pooling layers, and fully connected layers. The model architecture is designed to capture hierarchical spatial features in image data. Training involves multiple epochs, and the model's performance is monitored using callbacks, including early stopping.

### 7. Support Vector Machines

SVMs are versatile classifiers. In the context of image classification, they can be implemented using libraries like scikit-learn. SVMs excel in handling non-linear relationships in feature space and can be a valuable alternative or complement to deep learning approaches.

### 8. Logistic Regression

logistic regression serves as a foundational classification algorithm. It is interpretable, computationally efficient, and suitable for binary or multi-class problems. Logistic regression can serve as a benchmark model for comparison with more complex architectures.

### 9. MobileNetV2 (Pertained Network)

The code demonstrates the integration of MobileNetV2, a lightweight ANN architecture designed for mobile and edge devices. The model is pre-trained on ImageNet and fine-tuned for the specific brain tumour classification task. MobileNetV2 offers a balance between computational efficiency and accuracy, making it suitable for real-time applications.

## IV. Results

The experiments encompassed various machine-learning models for brain tumour classification. Notably, the MobileNetV2 pre-trained model with ANN surpassed all others, demonstrating exceptional accuracy on both training and testing datasets, achieving an impressive score of **90.2%.** Followed by MobileNetV2, the Convolutional Neural Network (CNN) model showcased commendable performance, obtaining an accuracy score of **87.5%.** However, the Support Vector Machine (SVM) and Logistic Regression models trailed behind, demonstrating considerable accuracies with scores of **87.4%** and **82.85%**, respectively. Conversely, the artificial neural network (ANN) showed a notably lower accuracy of **31.64%.** To boost its accuracy and lower errors, we teamed it up with MobileNetV2, a pretrained network that already knows a lot about images. This collaboration helps the ANN get better at recognising things. A detailed examination of misclassified samples offered insights into the models' limitations. MobileNetV2 emerged as the frontrunner, showcasing superior capabilities in accurately classifying brain tumour types, surpassing CNN, SVM, Logistic Regression, and the deliberately lower-performing ANN. But give good Accuracy after using a ANN with a pertained network MobileNetV2.

| **S.No** | **Training Model** | **Testing Accuracy** |
|---|---|---|
| 1. | Artificial Neural Network | **31.64%** |
| 2. | Convolutional Neural Networks | **87.5%** |
| 3. | Support Vector Machines | **87.4%** |
| 4. | Logistic Regression | **82.85%** |
| 5. | ANN+MobileNetV2 | **90.2%** |

## V. Conclusion

Exploring different machine learning models for brain tumor classification has revealed important findings. Using pre-trained models like MobileNetV2 has proven to be a game-changer, especially in using prior knowledge for better medical image analysis. This goes beyond just looking at accuracy scores and emphasizes the importance of pre-trained models in handling complex tasks with limited data.

Despite competitive performances from models like CNN, SVM, and Logistic Regression, there are still challenges in understanding subtle details in brain tumor images. This journey has shown how AI in healthcare is always changing and improving. It highlights the need for ongoing innovation in how models are built and how easily they can be understood.

In the end, what we've learned from this project is the close relationship between what we already know, how algorithms are getting better, and the constant effort to improve accuracy in analyzing medical images.

This whole experience stresses the importance of continuously enhancing computer methods in healthcare. It's not just about the numbers; it's about understanding how these models function and finding better ways to assist doctors in making accurate diagnoses from medical pictures.

# Team Information and planned contributions

**Team Member 1.** Alkesh Shukla (S20210020252)

**Data Augmentation**:

Configuring and applying the ImageDataGenerator for data augmentation.

Visualizing the augmented data using show_images_ImageDataGenerator.

**Model Architecture - CNN:**

Defining the architecture of the Convolutional Neural Network (CNN).

Configuring convolutional layers, max-pooling layers, and fully connected layers.

Setting up and compiling the CNN model.

**Model Training - CNN:**

Training the CNN model on the preprocessed data.

Implementing early stopping callbacks for efficient training.

Visualizing and saving the architecture of the CNN model.
Made the Accuracy and loss plots on training and validation data.
Implemented Confusion Matrix.

## Extension as suggested (Use a Pretained Network)

Used Pretrained **MobileNet Version 2** Model for increasing the accuracy of ANN.
Made the Accuracy and loss plots on training and validation data.

**Team Member 2.** Rohit N (S20210020297)

**Data Preprocessing**:

Extracting and organizing the dataset.

Implementing the get_data_labels function for data import.

Setting up file paths for training and testing.

Performing initial data visualization to understand the distribution.

**Wavelet Transformation:**

Implementing the Apply_wavelet function for wavelet transformation.

Applying wavelet transformation to sample images for feature extraction.

Contributing insights into how wavelet transformation impacts image features.

**Model Architecture - ANN:**

Defining the architecture of the Artificial Neural Network (ANN).

Configuring flattened input layers, dense hidden layers, and output layers.

Setting up and compiling the ANN model.

Visualizing and saving the architecture of the ANN model.
Made the Accuracy and loss plots on training and validation data.
Implemented Confusion Matrix.

**Model Training - ANN:**

Training the ANN model on the preprocessed data.

Implementing early stopping callbacks for efficient training.

Visualizing and saving the architecture of the ANN model.

**Team Member 3.** Gaurav Anand(S20210020273)

**Data Visualization:**

Visual Representation: Our code uses pie charts and bar graphs.

Tumor Type Distribution: Shows how different tumor types are distributed in training and testing datasets.

Data Exploration: Helps explore the data visually.

Informed Decisions: Guides decisions on how to sample and represent the data.

Enhanced Accuracy: The visual approach improves the accuracy of our analysis.

**Model Architecture - SVM:**

SVMs (Support Vector Machines) are flexible classifiers.

Can be easily implemented in image classification using libraries like scikit-learn.

Excels in handling non-linear relationships in feature space.

**Model Architecture - Logistic Regression:**

Logistic regression is implemented using inbuilt functions.

Can be easily implemented in image classification using libraries like scikit-learn.

**Metrics and Analysis:**

Implementing functions for **metric analysis** (CM and calculate_metrics).

Evaluating and analyzing model performance using precision, recall, F1-score, and accuracy.

Contributing to the overall assessment of model effectiveness.

This division of responsibilities allows each team member to focus on specific aspects of the code, ensuring a balanced and collaborative effort in building the brain tumor detection system.