**PROJECT REPORT:**

**CLASSIFYING BRAIN MRI’S:**

**Introduction:**

Brain tumors are classified by biopsy, which can only be performed through definitive brain surgery. Computational intelligence-oriented techniques can help physicians identify and classify brain tumors. Herein, we proposed two deep learning methods and several machine learning approaches for diagnosing three types of tumor, i.e., glioma, meningioma, and pituitary gland tumors, as well as healthy brains without tumors, using magnetic resonance brain images to enable physicians to detect with high accuracy tumors in early stages.

**Project Objectives:**

* Categorizing Brain Tumors as meningioma, glioma and pituitary tumors based on the location of the tumor
* Staging Dementia: Classifying dementia stages into very mild, mild and moderate
* Supporting Early Detection and Diagnosis : Assisting medical professionals with reliable predictions for early interventions
* This can be further expanded to diagnose other diseases and conditions from MRI scans of different parts

**Model Design:**

* **Data Augmentation and Preprocessing**
* **Class Balancing**
* **Model Architecture**
* **Fine Tuning**

**Data Processing:**

* + We used around **60,000 images** from **7 different categories/classes**. to train our model .
  + **Training Data**: Images were normalized and augmented to account for the variance in real-world data. (40,974 images used)
  + **Validation Data**: Normalized and used during training to evaluate the model. (8782 images used)
  + **Testing Data**: Used for the final evaluation of the model's performance, it uses data which the model has not seen before. (8787 images used).

**Class Balancing:**

1. **Extract Class Labels:**

* The code retrieves class labels from train\_generator using class\_indices.keys().

1. **Count Class Instances:**

* It counts the number of samples (files) in each class directory to determine class distributions.

1. **Use compute\_class\_weight:**

* Balances class weights by calculating weights inversely proportional to the class frequencies, ensuring underrepresented classes are given higher importance.

1. **Create a Weight Dictionary:**

* Converts the computed weights into a dictionary mapping class indices to their respective weights, which can be directly passed to the fit method.

1. **Purpose of Class Balancing:**

* Ensures the model does not favor overrepresented classes, improving fairness and accuracy across all classes.

**Model Architecture:**

**Base Model:**

* We have used **DenseNet121** as the base model for the Transfer Learning process
* DenseNet121 is a deep convolutional neural network (CNN) architecture introduced as part of the DenseNet (Dense Convolutional Network) family. It is specifically designed to address challenges like vanishing gradients, parameter inefficiency, and redundancy in traditional CNNs.
* It has **121 layers** including convolutional layers, pooling layers and fully connected layers.
* Every layer in DenseNet is connected to every other layer in a **feed-forward manner.**

**Custom Layers:**

* + **Global Average Pooling Layer** for dimensionality reduction.
  + Fully connected Dense layers with **ReLU** activation and Dropout for regularization and to prevent overfitting.
  + Output layer with **Softmax** activation on multi-class classification.

**Training:**

* + **Early Stopping:** This monitors the **validation loss** during training, if the validation loss does not improve for 5 consecutive epochs, training stops early to **prevent overfitting**.
  + We have used the Optimizer **Adam** for efficient **gradient descent.**
  + Loss Function we have used in our project is **Categorical** **CrossEntropy.**

**Fine-Tuning:**

**Layer Unfreezing:**

* The fine-tuning process began by gradually unfreezing the layers of the DenseNet121 model. Instead of unlocking all layers at once, a systematic approach was adopted to ensure that the pre-trained features were not disrupted.
* Initially, the deeper layers, closer to the classification head, were unfrozen as they capture task-specific features. Gradual unfreezing allowed the model to adapt to the specific dataset without forgetting the generalized knowledge from ImageNet.

**Learning Rate Scheduling:**

* A reduced learning rate was used during fine-tuning to ensure stable and incremental updates to the model weights. This prevented drastic changes that could lead to performance degradation or loss of pre-trained knowledge.
* Learning rate scheduling techniques, such as gradual decay, helped optimize the learning process, balancing between fine-tuning efficiency and model stability.

**Implementation:**

**Training:**

* Conducted initial training with frozen base layers and class weights to address imbalance.
* Used EarlyStopping to halt training when validation loss plateaued

**Fine-Tuning:**

* Unfroze selected layers of DenseNet121 to improve feature learning.
* Trained with a reduced learning rate for stability.

**Evaluation:**

* Assessed model performance on validation and testing datasets using metrics like accuracy, precision, recall, and F1-score.

**Results:**

1. **Initial Training Performance:**
   * Achieved high accuracy (≥90%) in classifying normal and abnormal MRIs.
2. **Fine-Tuning Results:**
   * Further improved classification metrics, particularly for underrepresented classes.
3. **Testing Evaluation:**
   * Demonstrated robust generalization with consistent results on unseen data.

**Conclusions:**

1. **Model Effectiveness:**
   * Successfully classified brain MRIs into categories with high accuracy, making it a reliable diagnostic tool.
2. **Scalability:**
   * The modular design allows for future extensions to other diseases and body parts

**Real-Life Impact of this Project:**

1. **This model can assist Radiologists and help reduce diagnostic errors.**
2. **It can help with early detection of neurological conditions**
3. **It gives the patients the option of Early and accessible diagnosis**

**Future Work**

1. **Dataset Expansion:**
   * Include MRIs of other organs to build a multi-purpose diagnostic tool.
2. **Integration with Clinical Data**:
   * Combine MRI-based predictions with other medical data for holistic diagnoses.
3. **Cloud Deployment:**
   * Enable remote diagnostics for underserved regions.
4. **Explainable AI:**
   * Develop interpretable models to provide detailed insights to doctors.