Journal of Advanced Engineering Trends
ISSN : 2682 - 2091

Vol.43, No.2. July 2024

http://jaet.journals.ekb.eg

CNN-MR Tumor Classifier: Brain Tumors Classification System Based on CNN Transfer Learning Models combined with Distributed computing process

Hend Fat'hy Khalil[1], **Hassan I. Sayed Ahmed**[2], **Ashraf Mahrouss Said**[3], **Hesham F. A. Hamed**[4, 5], **Eman Mohammed Mahmoud**[1]

[1] *Communication and Electronics Department, Modern Academy for Engineering and Technology, Cairo, Egypt*

[2] *Electronics Research Institute, Cairo, Egypt*

[3] *Biomedical Departments, Faculty of Engineering, Banha University, Banha, Egypt*

[4] *Faculty of Engineering, Minia University, Minia, Egypt*

[5] *Faculty of Engineering, Egyptian - Russian University, Cairo, Egypt*

**corresponding author E-mail:* eng.hend2025@gmail.com

ABSTRACT:

Preserving human health and life is of utmost importance in the development of automatic detection methods for early brain tumor diagnosis, considering the severe neurological impairments and potential fatality associated with the disease. Computational efficiency plays a critical role in brain tumor classification for real-time decision-making, treatment planning, and overall healthcare system optimization. While convolutional neural networks (CNNs) are widely used for brain tumor detection due to their exceptional accuracy, their high computational demands present significant challenges. To address the challenge at hand, a hybrid model is employed, integrating a pre-trained convolutional neural network (CNN) transfer learning model and the distributed computing programming paradigm. The primary objective involves two stages: In the first stage, InceptionV3 and VGG19 CNN transfer learning models are deployed on GPUs for detecting brain malignancies. Performance metrics, including accuracy, precision, recall, and F1-Score, are assessed, along with a comparative analysis of computational time on CPUs and GPUs. Results show InceptionV3 achieving a higher accuracy rate (approximately 98.83%) than VGG19 (77.65%), with superior computational speed on both CPU and GPU platforms. GPU execution significantly reduces computational time by up to 90%, attributed to the efficient architecture of InceptionV3. In the second stage, real-time classification is conducted using distributed computing process with previously trained CNN models for gliomas, meningiomas, and pituitary tumors, respectively. This integrated approach offers an efficient solution for real-time classification of large-scale brain tumor datasets.

Revised:25 September , 2023, Accepted:19 April , 2024

Keywords: CNN transfer learning, inceptionv3, VGG19, GPU, real time brain tumor classification, MapReduce programming model

1. INTRODUCTION:

Brain tumors are abnormal growths of cells that can occur within the brain or its surrounding tissues. These tumors may pose significant health risks [1]. Benign and malignant brain tumors are broadly categorized groups of brain tumors. Benign brain tumors are non-cancerous tumors and have a slow growth rate. Malignant tumors are malignant growths that have quickly encroached on nearby tissues [2].

The malignancy tumors have grades that evaluate their aggressive behavior of it in human health[2].

The three most common forms of brain tumors that can be labeled as benign or malignant are gliomas, meningiomas, and pituitary tumors. The glial cells that give rise to gliomas. The meninges and the barrier membranes covering the brain are where meningiomas develop. The pituitary gland, a key player in hormone regulation, can develop a pituitary tumor[1].

The efficient imaging techniques for brain tumors are computed tomography (CT), positron emission tomography (PET), and magnetic resonance imaging (MRI). [3]. CT and MRI are among the most commonly used imaging techniques. MRI provides detailed anatomical information with high resolution, enabling precise visualization of the tumor and its surroundings. It begets multiple plane images, which allows comprehensive evaluation and precise localization of the tumor within the brain[4]. Unlike CT scans, MRI uses non-ionizing radiation, making it safer for repeat imaging or long-term monitoring. Advanced functional and physiological assessment methods are also available with MRI, including functional magnetic resonance imaging (fMRI), perfusion-weighted imaging, and diffusion-weighted imaging (DWI), which offer useful insights into tumor features and brain activity[5]. Additionally, MRI can be enhanced with contrast agents, aiding in tumor detection and differentiation from normal tissues[6][7].

Advancements in computer-aided diagnoses (CAD) have led to the availability of both traditional machine-learning algorithms and deep neural networks for diagnosing brain tumors. There are fundamental differences in the approach and capabilities between traditional machine learning algorithms and deep neural networks. Traditional machine learning algorithms typically rely on handcrafted feature engineering, where domain experts manually select and design relevant features to represent the data. These algorithms then learn patterns and make predictions based on these engineered features using statistical methods like decision trees or support vector machines. In contrast, deep neural networks, such as CNNs, automatically learn features from raw input data through a hierarchical architecture of interconnected layers. This ability to learn hierarchical representations enables CNNs to extract complex features and capture intricate patterns from large-scale data, making them particularly effective for tasks involving images, videos, and sequential data[8]. Additionally, CNNs often employ convolutional and pooling layers that exploit spatial relationships and reduce the dimensionality of the data, allowing for more efficient processing.

The computational time required for training and inference is a known disadvantage of Convolutional Neural Networks (CNNs). Due to their deep and complex architecture, CNNs often have a substantial number of parameters, which increases the computational demands during training and inference. However, one way to mitigate this drawback is by utilizing transfer learning models. Transfer learning leverages pre-trained CNN models on large-scale datasets, allowing for the reuse of learned features and reducing the need for extensive training from scratch. By employing transfer learning, the computational time for training and inference can be significantly decreased, as the model benefits from the pre-existing knowledge embedded in the pre-trained network[9]. To further improve computational time and achieve better performance, running transfer learning on a Graphics Processing Unit (GPU) instead of a primary phases—Map and Reduce—the model involves assessing the accuracy of the Map Reduce programming model for each specific type of brain tumor.

partitions the input dataset into smaller segments during the Map phase. An autonomous "mapping" function is concurrently applied to each segment in parallel, facilitating simultaneous execution across distinct portions of the dataset. The subsequent Reduce phase involves the application of a "reducing" function to the key-value pairs generated in the Map phase. This function aggregates, combines, or analyzes values associated with each key, yielding a final set of output results. The Map Reduce programming model's Central Processing Unit (CPU) can be beneficial. GPUs are highly parallel processors that excel at performing complex computations required by deep learning algorithms like CNNs[10]. They have multiple cores designed to handle numerous calculations simultaneously, enabling faster training and inference times compared to CPUs. By leveraging the parallel processing power of GPUs, the computational workload of transfer learning can be distributed across multiple cores, resulting in significant speed-ups.

In the realm of real-time classification, handling the substantial volume of brain tumor data necessitates a strategy beyond exclusive reliance on pre-trained transfer learning models for swift classification. Consequently, diverse algorithms have emerged to tackle the complexities presented by big data in real-time classification, with notable methodologies including distributed computing process. The MapReduce programming model , originating from Google, serves as a distributed computing process tailored for parallel and scalable processing and analysis of extensive datasets[11]. Comprising two distinctive division of computation into these interconnected phases offers an efficient approach for large-scale data processing in distributed environments, contributing to enhancements in computational time and overall performance [12].

In this study, we employed the Inception V3 and VGG19 architectures as transfer learning models for Convolutional Neural Networks (CNNs) to categorize brain tumors in the pituitary, meningioma, and glioma using the Kaggle Brain Tumor Classification MRI database. The accuracy, precision, recall, and F1-Scoremetricsand elapsed time on Graphics Processing Unit (GPU) were assessed to determine the acceleration of the computational efficiency of the training process, providing insights into optimizing the performance of the CNN models for brain tumor diagnosis. The comparative evaluation of computational time entailed contrasting the execution of models on a high-performance Graphics Processing Unit (GPU) against the utilization of a Central Processing Unit (CPU).To achieve real-time classification, the previously designed CNN transfer learning model is employed to train a dataset. Subsequently, this trained dataset serves as input to the MapReduce programming model, facilitating real-time classification.

Our proposed studies make several significant contributions:

- Several deep convolutional transfer learning model architectures, including VGG19 and Inceptionv3, are investigated to assess how well they perform when classifying brain tumors from MRI data.

- Investigating the effects of freezing certain layers in previously trained models and then swapping out the final layers for the categorization of various types of brain cancers. With this method, we may use the pre-training information while still customizing the model for the categorization of brain tumors.

- We introduce an efficient hybrid model for real-time classification of diverse brain tumors. This model integrates a pre-trained CNN transfer learning model with the distributed computing process, specifically employing the MapReduce programming paradigm.

The structure of this paper adheres to a systematic organization. Section 1 serves as

The introduction provides an overview of the research topic. Section 2 delves into the related work, examining previous studies and approaches in the field. In Section 3, the proposed methodology is presented, outlining the novel techniques and methods employed in this research also, describes the evaluation metrics employed to assess the effectiveness of the transfer learning models used in the study. Section 4 is dedicated to the discussion of the obtained results, where the findings are analyzed and interpreted in detail. Section 5 provides a comparison between the proposed method and related work. Finally, the paper concludes with a comprehensive summary and concluding remarks in the final section.

2. RELATED WORK:

Researchers have put forth a variety of techniques over the years for spotting brain malignancies in MRI scans. Both traditional machine learning algorithms and deep learning models are included in this group of methods, which include a wide range of methodologies. We review the relevant research on detecting brain tumors in this part.

Kaur, T., & Gandhi, T. K. (2019)[13] developed an automated brain image classification system using the VGG-16 architecture and transfer learning techniques. The authors utilized a dataset comprising brain MRI images for four different classes: glioma, meningioma, pituitary, and normal brain tissue. They applied transfer learning by utilizing the pre-trained VGG-16 model. The study involved several stages, including data preprocessing, model training, and evaluation. During the preprocessing phase, the MRI images of the brain underwent a filtering process to improve their quality and eliminate any potential noise or artifacts present in the images. The preprocessed dataset was then used to train the VGG-16 model, with the weights of the previously trained layers frozen to preserve the generic image representation capabilities discovered by ImageNet. Using a variety of criteria, including accuracy, precision, recall, and F1-score, the authors assessed the effectiveness of their categorization method. The outcomes revealed that their method classified brain images into the four classes with a high accuracy of 100%, confirming the efficiency of transfer learning and the VGG-16 architecture for this task.

Deepak, S., & Ameer, P. M. (2019)[14] presented deep transfer learning and made use of a Google Net architecture that had already been trained to extract complex characteristics from brain MRI images. The framework seamlessly incorporates reputable classifier models to efficiently classify the retrieved information. The experimental evaluation uses a patient-level five-fold cross-validation strategy to ensure robustness and generalization, utilizing an MRI dataset obtained from Figshare that has been meticulously edited and adjusted to adhere to the highest scientific standards. These modifications include preprocessing techniques such as normalization and resizing. Remarkably, this pioneering system demonstrates a superior mean classification accuracy of 98%, surpassing all state-of-the-art methodologies.

Rehman, A., Naz, S., et al. (2020)[15] used three alternative convolutional neural network (CNN) architectures, AlexNet, GoogleNet, and VGGNet, to develop a paradigm for classifying brain cancers. Using MRI slices of brain tumor datasets downloaded from Figshare (www.figshare.com), the framework investigated transfer learning approaches, notably fine-tuning and freezing. To enhance the generalization of results and prevent overfitting, data augmentation techniques were applied to the MRI slices, increasing the dataset size. The three studies conducted within the framework revealed that fine-tuning the VGG16 architecture yielded the best results, with classification and detection accuracy up to 98.69%.

Mehrotra, R., Ansari, M. A. (2020)[16] presented a study utilizing artificial intelligence (AI) and deep learning models, namely AlexNet, GoogleNet, ResNet50, and ResNet101, for the classification of brain tumors (BTs) based on their types. The researchers utilized publicly available datasets from The Cancer Imaging Archive (TCIA) consisting of 696 T1-weighted brain images for testing purposes. Before applying TL models, various preprocessing techniques were applied to enhance the data for the network's training. These techniques include image augmentation methods such as flipping, mirroring, and rotation. The purpose of these techniques is to generate a larger set of data for the network, which is commonly employed to prevent overfitting. The system undergoes multiple rounds of training, where the existing pre-trained networks are modified through various renowned optimization techniques. These methods include adaptive moment estimation (Adam), root mean square propagation (RMSprop), and stochastic gradient descent with momentum (SGDM). By employing these advanced optimizers, the system aims to achieve the most optimal and finely-tuned trained network configuration for enhanced performance. Through the implementation of these AI algorithms, the researchers achieved remarkable results in accurately classifying BTs as either malignant or benign.

Saxena, P., Maheshwari, A., & Maheshwari, S. (2020)[17] established a new convolutional neural network (CNN) architecture to carry out transfer learning for dividing brain MRI scans into benign and malignant categories. This approach utilizes three pre-existing and well-established models, capitalizing on their knowledge and expertise. The proposed methodology in this study is structured into three distinct phases. The initial dataset used for brain tumor detection consists of 253 images from brain MRI scans. In the first phase, the brain MRI images undergo preprocessing to prepare them for further analysis and evaluation of the developed model. The second phase involves the application of data augmentation techniques to the dataset, as it is relatively small. The third stage involves employing transfer learning to train several pre-trained convolutional neural network (CNN) models. The comparative analysis of these models demonstrates their respective efficacy. Empirical findings reveal that the ResNet-50 model attains an outstanding accuracy rate of 95%, positioning it at the peak of performance. Succeeding the VGG-16 and InceptionV3 models exhibit commendable accuracies of 90% and 55%, respectively.

Choudhury, C. L., Mahanty, C., et al. (2020) [18], developed a novel approach to identify brain MRI images as either benign or suggestive of malignancies using convolutional neural networks (CNN). The proposed model has exhibited an impressive accuracy rate of 96.08%, accompanied by a remarkable F-score of 97.3%. Utilizing a precisely crafted CNN architecture comprising three layers, the approach requires minimal preprocessing steps and achieves optimal results within 35 training epochs.

Srinivas, Chetana, et al. (2022) [19], presented a comparison of the abilities of transfer learning-based CNN models VGG-16, ResNet-50, and Inception-v3 to detect brain tumor cells automatically. The pre-trained models were evaluated on a dataset consisting of 233 MRI brain tumor images, with the aim of accurately locating brain tumors using the VGG-16 pre-trained CNN model. The utilization of data preprocessing and data augmentation techniques, coupled with hyperparameter tuning, was employed to assess the performance of pre-trained Convolutional Neural Network (CNN) architectures for the detection of brain tumor cells. The study evaluated the performance of VGG-16, Inception-v3, and ResNet50 models, aiming to identify the architecture that achieves the highest accuracy for both training and validation datasets. Overfitting was identified as a potential issue, whereby the model becomes limited to the specific training data and cannot make accurate predictions on new datasets. According to the study's findings, all architectures achieved training accuracy above 90%, with the highest validation accuracy reaching 88.26%. According to the analysis, VGG-16 provided the maximum accuracy on the trained and tested dataset, with validation accuracy being more closely related to accuracy with less loss and validation loss.

Arbane, M., Benlamri, et al. (2021)[20], proposed employing convolutional neural performance; transfer learning techniques are employed. The model aims to categorize brain MRI images into two classes: those depicting tumors and those without tumors. To improve the model's efficacy, various preprocessing techniques such as image augmentation, normalization, cropping, and resizing are utilized. The dataset consists of 253 MRI images, comprising 155 samples with tumors and 98 samples without tumors. Within the implemented system, multiple CNN architectures are explored, including ResNet, Xception, and MobileNet-V2. Among these architectures, MobileNet-V2 demonstrates the most favorable outcomes, achieving an accuracy of 98.24% and an F1-score of 98.42%.

Sadad, T., Rehman, A., et al. (2021)[21], developed a unique strategy for brain tumor segmentation using ResNet50 and UNet architecture as the foundational model. The researchers achieved a remarkable intersection over the union (IOU) level of 0.9504 using the Figshare dataset, demonstrating accurate segmentation results.

They introduced preprocessing methods and data augmentation techniques to increase the classification rate. The researchers also investigated the multi-classification of brain tumors, utilizing evolutionary algorithms and reinforcement learning via transfer learning. They employed a variety of deep learning techniques, including MobileNetV2, InceptionV3, ResNet50, and DenseNet201. The results indicated that the suggested research framework outperformed current state-of-the-art approaches. For tumor classification, various CNN models including MobileNetV2, InceptionV3, ResNet50, DenseNet201, and NASNet were employed, achieving accuracy percentages of 91.8%, 92.8%, 92.9%, and 93.1%, respectively.

Ullah, Naeem, et al. (2022) [22], used pre-trained transfer learning (TL) techniques to recognize and detect brain malignancies such as pituitary, meningioma, and glioma. The goal was to evaluate the abilities of nine pre-trained TL classifiers, including InceptionResNetV2, InceptionV3, Xception, ResNet18, ResNet50, ResNet101, ShuffleNet, DenseNet201, and MobileNetV2, to detect and automatically identify brain cancers using fine-grained classification. A baseline brain tumor classification (MRI) dataset provided by Kaggle was used to test the TL techniques, and all deep learning (DL) models were adjusted using their default parameters. The InceptionResNetV2 model, achieving an accuracy of 98.71% in the categorization of brain tumors, was demonstrated to be the most successful. This research illustrates the superiority of InceptionResNetV2 over other hybrid techniques using DL models for deep feature extraction and classification of brain tumors in terms of reliability and accuracy.

3. PROPOSED METHODOLOGY:

The proposed approach encompasses a hybrid system amalgamating a transfer learning framework utilizing Convolutional Neural Networks (CNNs) with MapReduce distributed computing methodology. The proposed methodology consists of two consecutive stages. In the first stage, transfer learning models are employed for the training and classification of data. Following this, the second stage leverages the distributed computing process, employing the MapReduce programming paradigm. This involves utilizing the trained data from the initial stage to expedite real-time brain tumor classification through the parallelized processing capabilities inherent in MapReduce. The block diagram for the proposed methodology is shown in Figure (1). The first stage comprises four distinct steps. Firstly, the images are preprocessed to enhance their quality and prepare them for analysis. In the second step, transfer learning models, such as InceptionV3 and VGG19, are utilized to extract significant features from preprocessed images and classify them. In the third step, the performance of the transfer learning models is assessed by evaluating the accuracy of their predictions.

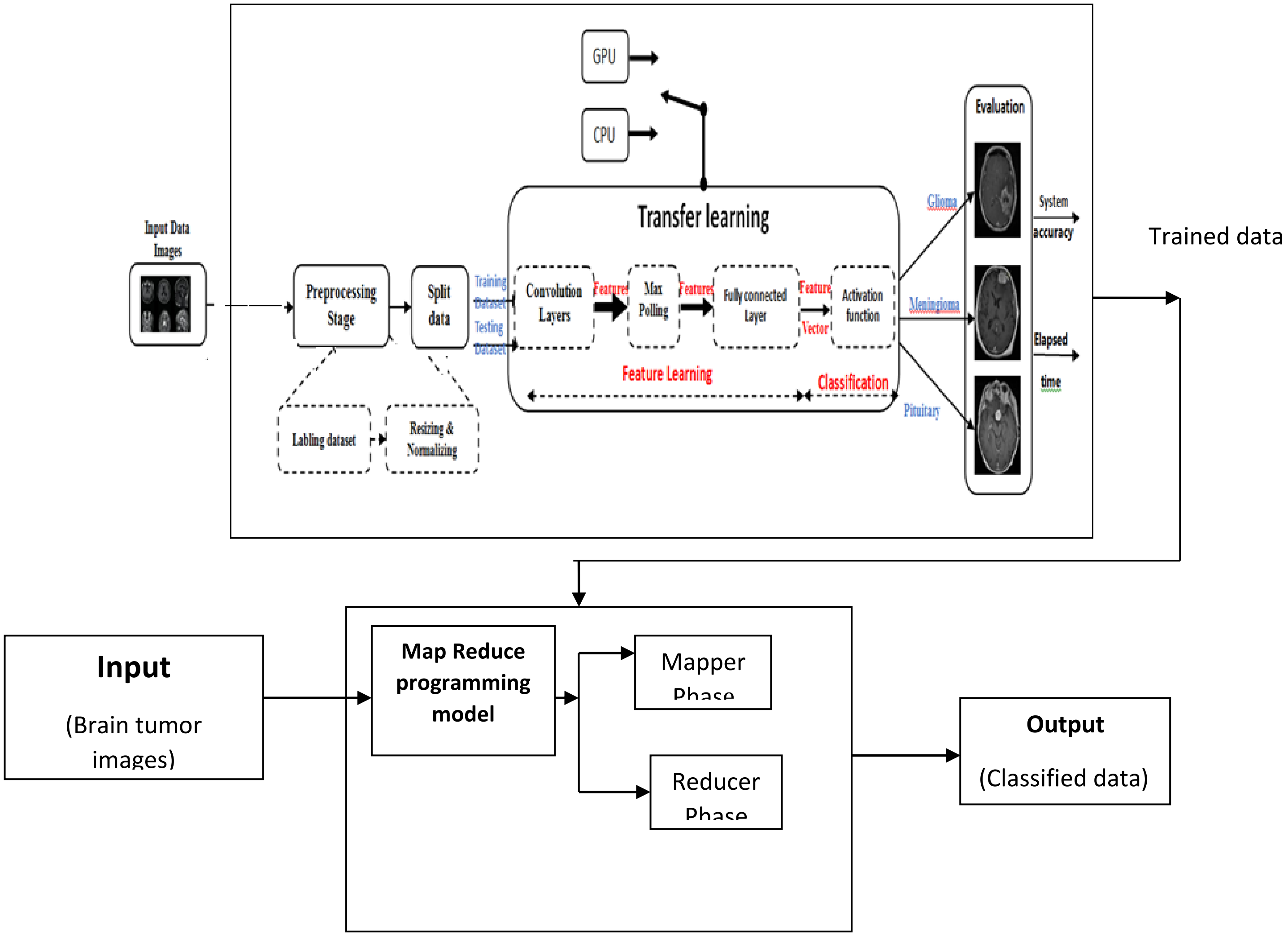

Figure (1) the block diagram for the proposed methodology

Initially, magnetic resonance imaging (MRI) images were acquired from Kaggle, followed by the labeling of the dataset and preprocessing, which included resizing based on a transfer learning algorithm. Subsequently, the data were partitioned into training and testing sets. Before implementing the transfer learning model, a crucial decision was made regarding whether to execute the model on a Central Processing Unit (CPU) or Graphics Processing Unit (GPU). The transfer learning model integrated convolutional layers for feature extraction, max-pooling layers for dimensionality reduction, fully connected layers for pattern recognition, and an activation function to introduce non-linearity. Evaluation of the model involved metrics such as accuracy, precision, recall, F1-Score, and elapsed time.

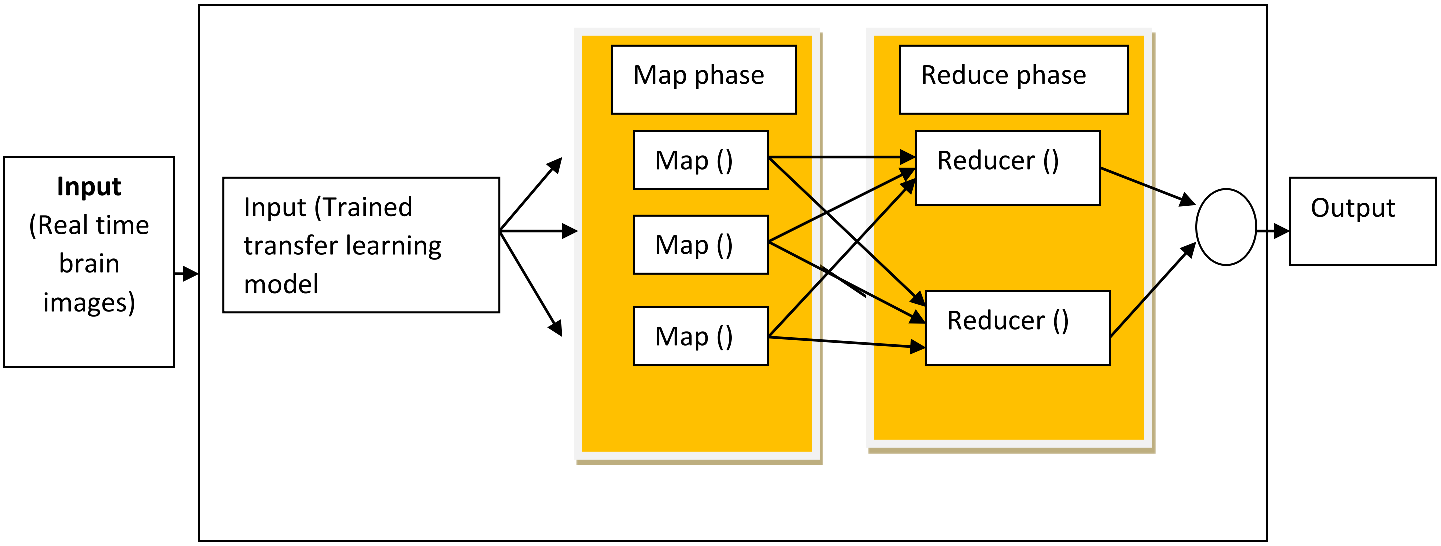

Figure (2) the second stage of the proposed methodology

The second stage comprises two steps: the Map phase and the Reduce phase. In the Map phase, the testing dataset is partitioned into smaller, manageable segments for distribution. Each segment undergoes independent processing by a mapper function, applying the pre-trained model to generate predictions for individual images. Subsequently, in the Reduce phase, the results from the Map phase are aggregated and synthesized to yield the final output.

3.1 The proposed model for the processing task based on Transfer learning models of CNN:

3.1.1 The acquisition and preprocessing of datasets:

A publicly accessible dataset, available on Kaggle (www.kaggle.com), encompasses a variety of categories of brain tumor MRI images. Four different types of brain tumor MRI images are included in the brain tumor MRI image dataset: no tumor, meningioma tumors, pituitary tumors, and glioma tumors. These images are divided into testing and training collections. However, the study was restricted to the MRI scans of the meningioma, pituitary, and glioma tumors. The training folder of the dataset currently contains 826 MRI images of glioma brain tumors, 822 MRI images of meningiomas, and 827 MRI images of pituitary tumors. 115 MRI images of meningiomas, 72 MRI images of pituitary tumors, and 100 MRI images of glioma brain tumors are included in the testing folder. Images from both folders were pooled, and 80% of the data were utilized for training, leaving 20% for testing. The 256x256 pixel grayscale images are offered in the JPG format and come in a variety of resolutions. Table (1) illustrates the brain tumor dataset.

Table 1The brain tumor dataset

Brain tumor Type	**Number of training images**	**Number of testing images**	**Total number of images**	**Type**	**Format**
Pituitary	827	72	899	Grayscale	JPG
Glioma	826	100	926		
Meningioma	822	115	937		

The transfer learning model resizes and normalizes the images during the preprocessing phase. The block diagram of the preprocessing of MRI brain images is shown in Figure (3).

Figure (3) the block diagram of the preprocessing of MRI brain images

Normalizing brain MRI images is a crucial step in the pre-processing of medical images, as it helps to reduce variations in image intensity, contrast, and brightness, which can negatively impact image analysis and interpretation. The MRI images are normalized by scaling the pixel values to fit within a predetermined range of values, often between 0 and 1 or -1 and 1. This process helps to standardize the image intensities across different datasets, making it easier to compare and analyze images. The equation commonly used for normalizing brain MRI images is:

$$I_{norm} = (I - I_{min}) / (I_{max} - I_{min}) \quad (1)$$

Where I_{norm} is the normalized pixel value, I is the original image's pixel value, I_{min} is the image's lowest pixel value, and I_{max} is the highest pixel value.

The resizing image adjustments are implemented to satisfy the specific prerequisites of each model and obtain the best possible outcomes. The dimensions for each model are predetermined and Specifically, the VGG19 model resizes the images to dimensions of 224x224, while the InceptionV3 model resizes them to 299x299

3.1.2 Convolutional Neural Network (CNN):

Convolutional Neural Network, or CNN, is a deep learning algorithm. It is frequently employed in computer vision applications like picture segmentation, object detection, and classification [23]. The layers of a CNN are its main structural components, including the input layer, convolutional

optimized based on the architecture and parameters of the models themselves.

layers, pooling layers, and fully connected layers. The convolutional layers are in charge of learning and extracting information and features via convolution operations from the input images. The feature maps' dimensionality is decreased and the most crucial data is captured by the pooling layers. The classification or regression tasks are then carried out by the fully connected layers using the learned features. The building of CNN is depicted in Figure(4).

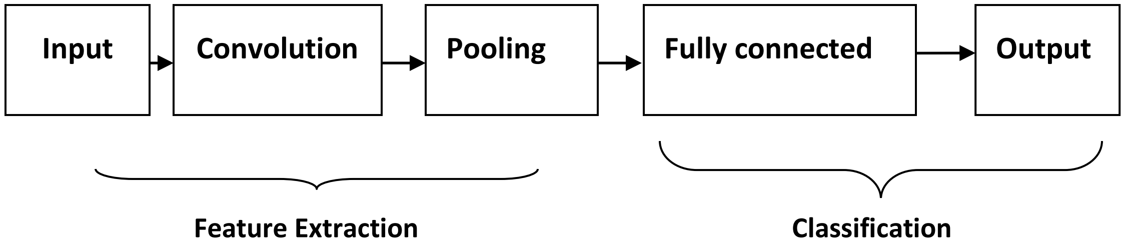

Figure (4) the building of CNN

In the realm of deep learning, transfer learning is a potent method, especially when used with convolutional neural networks (CNNs). CNNs are frequently employed for applications including object identification, recognition, and image categorization [10]. Transfer learning uses previously learned information from models that have been used to train on massive datasets like ImageNet and apply it to new tasks using smaller datasets. In transfer learning for CNNs, the convolutional layers of the trained model serve as feature extractors [24]. These convolutional layers capture low-level features like edges, textures, and shapes. The model can be fine-tuned to suit the particular task at hand, such as identifying different objects or classifying different categories, by freezing these layers and adding additional layers on top. Since the pre-trained model has already learnt the necessary characteristics, this method conserves computational resources and training time. When there is not enough labeled data to train a model from scratch, CNNs' transfer learning models can be useful. The pre-trained models give the model a solid foundation from which to learn rapidly and effectively [25]. Additionally, transfer learning facilitates generalization and improves performance by avoiding over fitting on small datasets. Overall, transfer learning models of CNNs have revolutionized the field by enabling effective results even with limited data. Figure (5) shows the Overview of pretrained transfer learning model of proposed methodology. The extraction of significant features is facilitated by employing two strategies of transfer learning: freezing layers and fine-tuning. This process involves transferring pre-trained architecture weights from a source database, such as ImageNet, to a target database, exemplified by our migration from ImageNet to Kaggle. Fine-tuning within the framework of transfer learning is utilized to augment the efficacy of Convolutional Neural Network (CNN) architectures by selectively adjusting the final layers of the pre-trained model. In essence, this entails substituting the 1,000 categories of ImageNet with the three categories of brain tumor delineated in the Kaggle dataset. Popular transfer learning models include VGG, ResNet, MobileNet, and Inception [26]. In numerous image recognition applications, these models have been widely applied and shown to be successful. In Our study, the inceptionv3 and VGG19 are used.

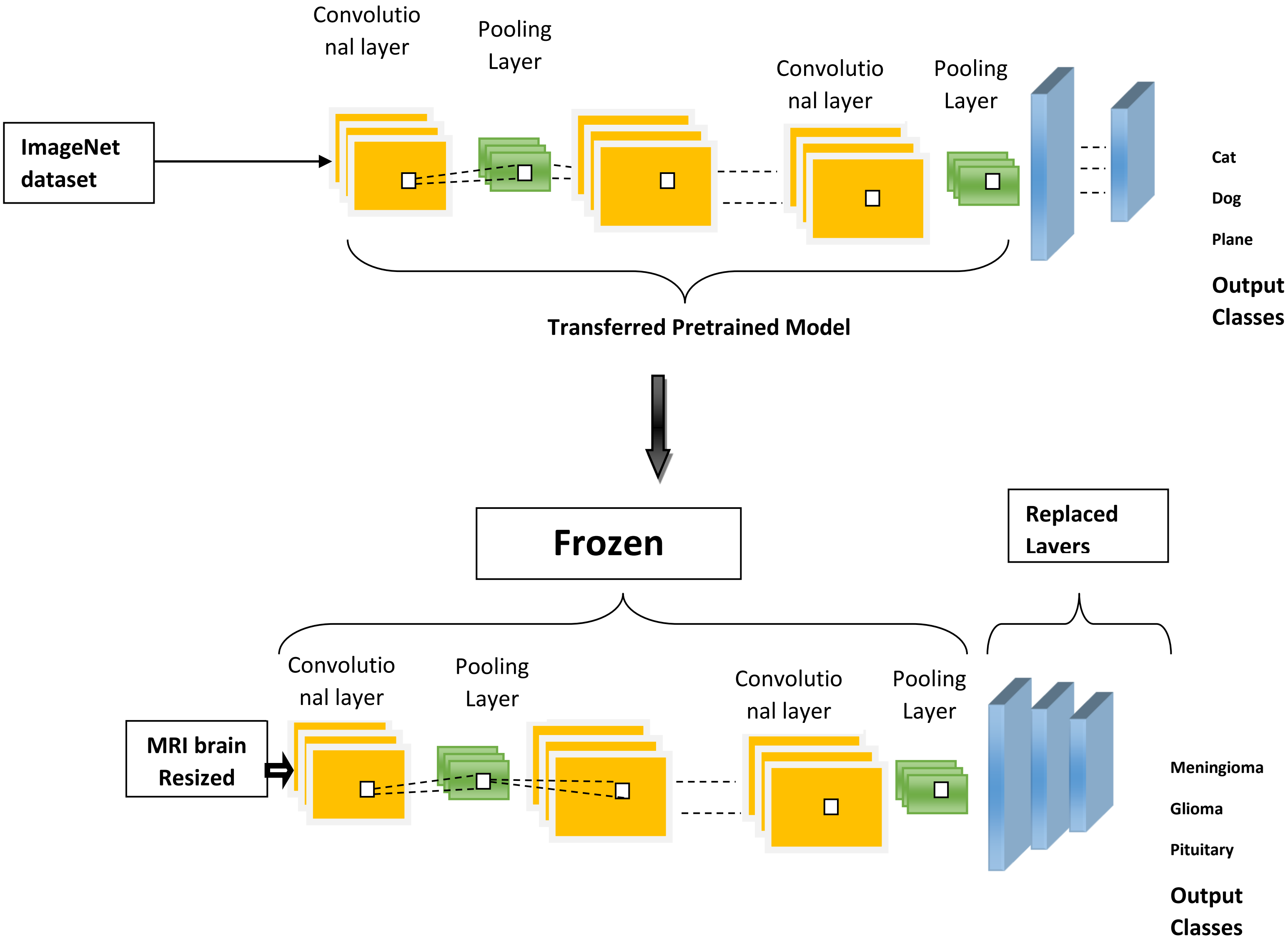

Figure (5) the Overview of pretrained transfer learning model of proposed methodology

3.1.2.1 InceptionV3 Transfer Learning Model:

The use of "inception modules," which are intended to collect characteristics at various scales and minimize the amount of parameters in the model, is a hallmark of the Inception v3 architecture. The subsequent structure of the network is composed of a number of convolutional layers that extract features from the input image. To extract information from the input image, these layers employ a series of filters. The inception modules serve as the fundamental building blocks of the Inception v3 architecture and are intended to collect information at various scales while minimizing the amount of model parameters. Each inception module is comprised of multiple convolutional layers of various sizes and pooling layers. These inception modules consist of several parallel convolutional layers of different sizes that are concatenated, allowing the network to gather features at multiple scales, a crucial aspect for identifying objects at various granularities. The network's next layer receives the output from each inception module. The fully connected layers in the Inception v3 model are used to map the features that the convolutional layers extracted to the output classes. These layers combine a set of weights with the output of the convolutional layers to produce the final output. The network's final output, which is a probability distribution over the output classes, is produced by the output layer.

In total, there are 48 layers, including 11 inception modules, in the Inception v3 model as a whole. The model contains over 23 million parameters, making it a sizable model. It can be tailored for particular image classification tasks with only a small amount of data, though, because it was pre-trained on a large dataset of images [27]. Additionally, the architecture includes additional components like batch normalization and dropout for improved performance. The Inception v3 model's input layer accepts images with a resolution of 299x299 pixels.

3.1.2.2 VGG19Transfer Learning Model:

The Visual Geometry Group (VGG) at the University of Oxford developed the convolutional neural network (CNN) architecture known as the VGG19 model for use in image recognition and classification applications. It is a pre-trained model honed for specific image classification tasks using a vast dataset of images.

The VGG19 architecture can capture fine-grained information in the input images because it employs a sequence of convolutional layers, each with a small filter size of 3x3. Max pooling layers and fully connected layers are also used in the architecture to boost performance. The input layer of the VGG19 architecture receives images with a resolution of 224x224 pixels. The subsequent layers in the network consist of 16 convolutional layers, 5 max pooling layers, and 3 fully connected layers. The pool size for the maximum pooling layers is 2x2, while the filter size for each convolutional layer is 3x3.

The feature extraction process on the input image is carried out by the convolutional layers. Each layer employs a collection of filters to extract features from the input image. The spatial dimensions of the feature maps are reduced by the max pooling layers, which also provide the network with some translational invariance. The fully connected layers map the output classes to the features extracted by the convolutional layers.

The designation VGG19 refers to the model's total of 19 layers. The first 16 layers are convolutional layers, followed by 5 max pooling layers and 3 fully connected layers.

3.1.3 EVOLUTION METRICS:

The transfer learning model was evaluated using accuracy, precision, F1-score metrics, and elapsed time, which is the amount of time required by the model to complete the training process.

3.1.3.1 Accuracy:

The effectiveness of the integrated transfer learning model was evaluated using accuracy as the main assessment criterion in the study.

Accuracy= (TP+TN)/Ts (2)

Where (TS) is the total number of samples and (TP) and (TN) are the true positive and true negative, respectively.

3.1.3.2Precision:

Precision measures the accuracy of positive predictions made by a model. It is calculated as the number of true positive predictions divided by the sum of true positives and false positives:

Precision=TP/ (TP+FP) (3)

In simpler terms, precision evaluates the reliability of a model when it predicts a positive outcome, indicating how many of those predictions are correct.

3.1.3.3 Recall:

Recall measures the ability of a model to correctly identify all relevant instances of a particular class. It is also known as sensitivity or true positive rate:

Recall=TP/(TP+FN) (4)

A higher recall indicates better performance. Sensitivity or true positive rate refers to the proportion of actual positive cases that were correctly identified by the model.

3.1.3.4 F1-Score:

The F1-Score is a statistical measure of a classification test's accuracy that balances both precision and recall. It is defined as:

F1Score=2((precision*Recall)/(precision+Recall)) (5)

3.1.3.5 Elapsed time:

The time taken to train and assess a transfer learning model is referred to as the model's elapsed time. This comprises the time needed for data preprocessing, model training, and performance testing utilizing one or more metrics. The complexity of the model, the quantity of the training dataset, the technology utilized for training, and the optimization strategies used can all impact how long it takes. Given that longer elapsed intervals can lead to higher computational costs and longer development cycles, it is a crucial factor to take into account in machine learning.

3.2 The proposed model based on distributed computing process:

MapReduce serves as a distributed computing process, functioning as a programming model and processing framework tailored for large-scale data processing tasks. Widely utilized in distributed computing environments like Apache Hadoop, MapReduce efficiently processes extensive datasets in parallel across a cluster of computers [29]. The MapReduce paradigm is typically used in distributed computing to process large data. In the context of brain tumor classification, the MapReduce methodology can be implemented as follows:

3.2 .1. Mapper Phase:

The Map phase in the context of brain tumor classification based on MapReduce involves loading a pre-trained deep learning model and distributing the testing dataset into smaller, manageable chunks. Each chunk is processed independently by a mapper function, which applies the pre-trained model to make predictions for individual images. The mapper handles potential errors, such as empty or corrupted images, and produces intermediate key-value pairs. These pairs consist of unique identifiers (e.g., filenames) as keys and the corresponding prediction results as values. The intermediate output is logged, facilitating debugging and troubleshooting. This parallelized approach enables efficient prediction across the dataset, paving the way for subsequent aggregation and analysis in the Reduce phase. The pseudo code for the task of the Map phase is as follows:

Mapper Phase Algorithm:

```
1. Input:
     dataChunk: Chunk of data with image points.
     CNNmodel: Trained CNN-based transfer learning model.
2. Initialization:
     predictions list.
3. Loop:
     For each dataPoint:
         Try:
             Preprocess the brain tumor image.
             Get prediction using the CNN model.
             Emit filename and prediction.
         Catch exceptions: Emit filename and error.
4. Output:
     predictions list with imageclass-prediction pairs.
5. Overall Process:
     Load test data from brain tumor images.
     Split data into chunks.
     mappedPredictions list.
6. Map Phase:
     For each chunk:
         Apply map function using CNN: Preprocess, predict,
         and emit.
         Append results to mappedPredictions.
```

3.2 .2. Reducer Phase:

In the Reduce phase of the MapReduce paradigm for brain tumor classification, the grouped and sorted key-value pairs from the Shuffle and Sort Phase are processed to consolidate predictions and derive final outcomes. Each unique identifier (e.g., filename) is identified as a key, and the associated values encompass predictions from various mappers. The reducer function is designed to aggregate these predictions, often involving statistical operations. The parallelized nature of this phase enables simultaneous processing of multiple key groups across different reducers, enhancing overall efficiency. The Reduce phase concludes with the generation of the final output, providing insights into brain tumor classification based on the predictions made by the pre-trained deep learning model across the testing dataset. The pseudo code for the task of the Reduce phase is as follows:

```
Reducer Phase algorithm:
   1. Initialize an empty dictionary named combinedPredictions.
   2. Loop through each set of predictions in the input (mappedPredictions).
   3. For each (imageclass, prediction) pair in the predictions set:

          Check if the filename is not already a key in combinedPredictions.
          If not, add the filename as a key with an associated empty list.
          Append the prediction to the list associated with the filename.
   4. After processing all predictions, combinedPredictionscontains filenames as
      keys, and each key has a list of associated predictions.
   5. Iterate through combinedPredictions to calculate the final prediction for
      each filename by aggregating the predictions.
   6. Return the combinedPredictionsdictionary as the result of the reduce
      operation.
```

3.2. 3. Post-Processing:

The Post-Processing phase for the MapReduce model based on brain tumor classification involves the aggregation and analysis of the grouped predictions produced by the Shuffle and Sort Phase. The following step is the organization of data based on unique identifiers, after which the Reduce phase begins. In this phase, reducer functions are applied to the grouped data to generate overall predictions. This can include tasks such as combining probabilities from multiple mappers, determining the most likely class for each unique identifier, or calculating aggregate statistics. The Post-Processing phase aims to consolidate distributed predictions into cohesive and labeled output. It often includes logging and debugging mechanisms to assist in identifying any issues that may arise during the aggregation and analysis process. The efficiency of the Post-Processing phase is crucial for obtaining accurate and actionable insights from the distributed predictions generated by the parallelized Map phase.

Let's compare this with a traditional (non-distributed) approach. In a traditional approach, the elapsed time (ET_{trad}) might be expressed as:

$$ET_{trad}=D/S_{trad} \quad (6)$$

Where S_{trad} is the speed of processing in a traditional, non-parallelized setting.D is the size of the input data. The key comparison is the effect of parallelization in the Map Reduce approach, and then ET in Map Reduce can be considerably lower than ET_{trad} as S_{trad} is higher in distributed processing (parallelization strategy).

4. RESULT:

The dataset is available on Kaggle for free. Glioma, meningioma, and pituitary tumors are three different forms of brain tumors shown in Figure (6).

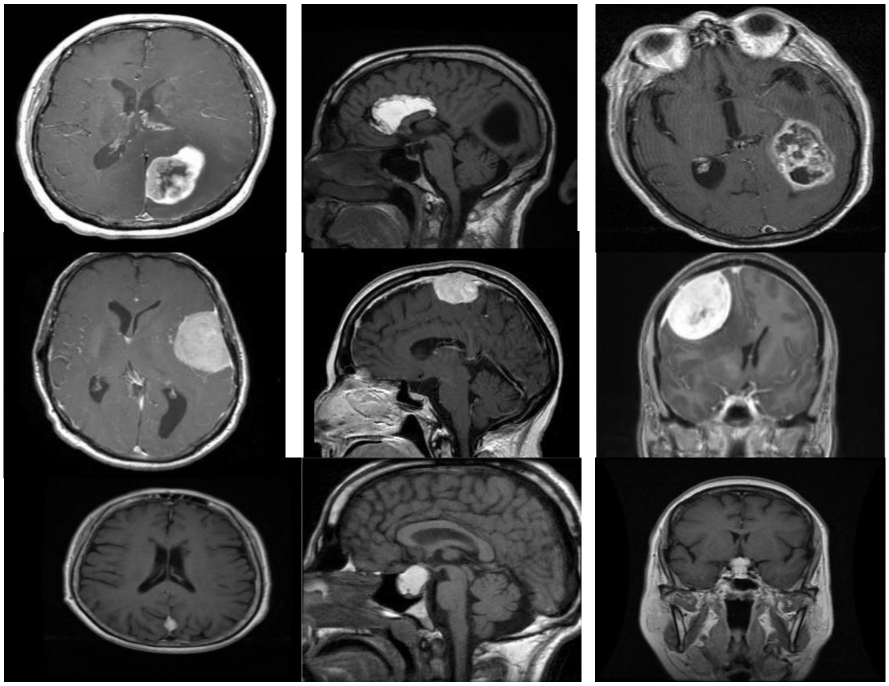

Figure (6) the first row is a glioma brain tumor, the second row is a meningioma brain tumor, and the third row is a pituitary brain tumor

InceptionV3 and VGG19 are two of the pre-trained transfer learning (TL) network classifiers used in this study. Together, they have the capacity to classify images into 1000 different item classes. 1.28 million Images from the ImageNet database were used to train these classifiers on a sizable dataset. However, a more particular task, the three-class classification of brain tumors, is the main focus of this work. We used Google Colab platforms to implement our suggested model, which was written in the Python programming language. The Adam optimizer was used to train the transfer learning models, using a batch size of 16 images and a learning rate of 0.01. Each deep learning model also completed 10 training epochs. The characteristics of the transfer learning architecture are shown in Table(2)

Table 2 the parameters of transfer learning architecture

Parameters	**Value**
algorithm for optimization	Adam
Loss	binary_crossentropy
Batch size	16
Learning rate	0.01
Maximum Epochs	10
Train size	0.8
Test size	0.2
Random state	42

A comprehensive analysis and evaluation of transfer learning models was carried out using accuracy, precision, and F1-Score metrics. The findings demonstrate that the InceptionV3 transfer learning model reached an impressive average classification accuracy of 98.83%. Conversely, the VGG19 transfer learning model attained a moderate average classification accuracy of 77.65%. InceptionV3 exhibits a precision of 97.43%, denoting its high accuracy in positively identifying relevant instances. Meanwhile, VGG19 demonstrates a precision of 76.07%, reflecting a comparatively lower precision rate. Additionally, the F1-score, a harmonic mean of precision and recall, is 96.34% for InceptionV3 and 72.65% for VGG19, providing a comprehensive evaluation of their classification performance. Table (3) illustrates the average accuracy classification and elapsed time on GPU and CPU.

Table 3 the average accuracy classification and elapsed time

Model	Accuracy	precision	Recall	F1-Score	Elapsed Time on GPU	Elapsed Time on CPU
Inceptionv3	98.83%	97.43%	95.15%	96.34%	3min,40sec	42min,43sec
VGG 19	77.65%	76.07%	75.44%	72.65%	5min,54sec	71min,6sec

The elapsed time is the duration it takes for the model to accurately categorizes all images within the dataset. The elapsed time varies based on factors such as the volume of data utilized and the intricacy of the model's architecture. According to the data presented in Table (3), it is evident that the utilization of GPU for execution leads to higher efficiency compared to CPU execution. Consequently, leveraging GPU computing enables a reduction in computational time and enhancement of performance for transfer learning models, thereby optimizing their overall efficacy. Training and validation accuracy and loss charts for InceptionV3 are displayed in Figure (7), while Figure (8) displays the charts for VGG19. In both graphical representations, the x-axis denotes the number of epochs, while the y-axis depicts the levels of loss and accuracy. The observed reduction in loss during early epochs signifies the efficacy of the transfer learning model. Additionally, an increase in the number of epochs results in the stabilization of the transfer learning model.

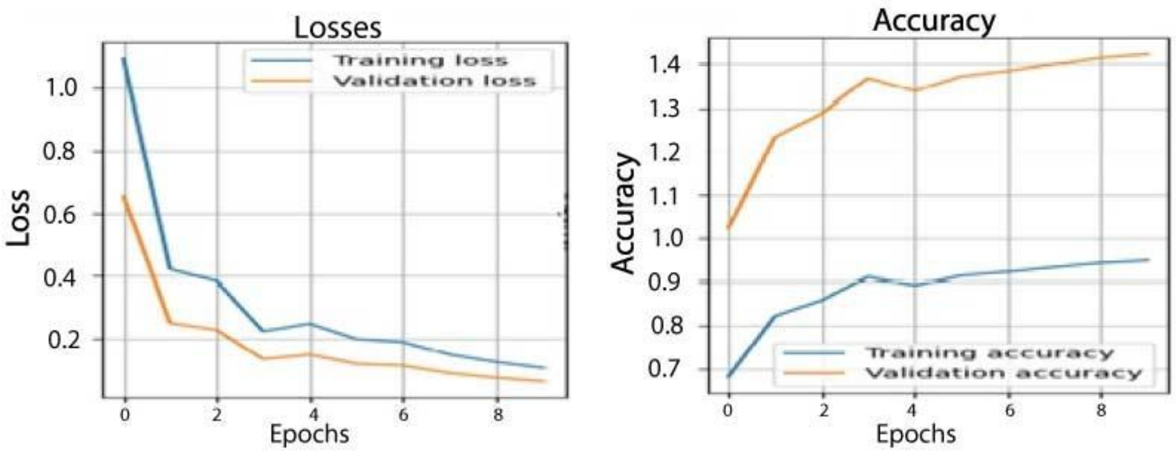

Figure (7) Inceptionv3 training and validation accuracy and loss plots

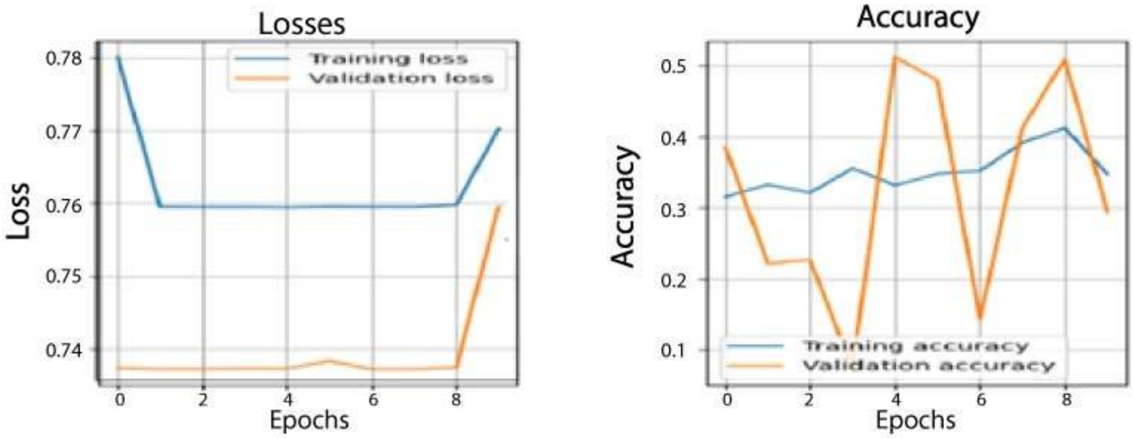

Figure (8) VGG19 training and validation accuracy and loss plots

The usage of inception modules, which are designed to capture features at various sizes while simultaneously reducing the number of parameters required, has led to the InceptionV3 model performing better than VGG19. Utilizing these multi-scale characteristics, InceptionV3 can achieve high accuracy while maintaining computational efficiency, resulting in a faster training process. In contrast, VGG19 has a more straightforward architecture, which can lead to a higher number of parameters and a longer training time.

MapReduce itself is a programming model and framework for processing and generating large datasets, and it is not inherently designed for machine learning model evaluation. However, in the context of brain tumor classification using MapReduce and deep learning, the evaluation metrics typically involve assessing the performance of the trained model based on its predictions. A common evaluation metric for brain tumor classification is accuracy. The result of evaluation based on the accuracy metric for different types of brain tumors is shown in Figure (9).

.

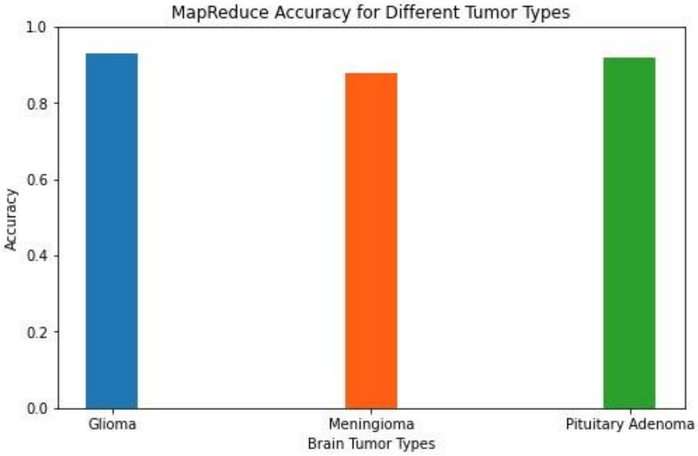

Figure (9) Accuracy of Map Reduce model for different types of brain tumor

The figure shows accuracy (93%, 88%, and 92%) for different types of brain tumor classifiers based on MapReduce and deep learning.

5. COMPARISON WITH RELATED WORK:

We evaluated how well the top CNN-based transfer learning model, InceptionV3, classified brain cancers in comparison to other approaches. In particular, we contrasted the proposed work with transfer learning methods [20]. Table 4 presents a comparative analysis between the proposed method and related works based on accuracy. The superior performance of the proposed approach is attributed to its utilization of a greater number of epochs compared to the related work under consideration. Additionally, the implementation of a diverse optimization algorithm further contributes to the improved results achieved by the proposed method.

.

Table 4 the comparison between proposed methods with related work

Work	transfer learning used	algorithm for optimization	Maximum Epochs	Accuracy
Ullah, Naeem[22]	Inception v3	SGDM	14	94.48%
The proposed method		Adam	16	98.8%

6. Conclusion and future work:

This research presents an automated method for classifying various brain cancers, including gliomas, meningiomas, and pituitary tumors. Utilizing InceptionV3 and VGG19 transfer learning models, the study achieves notable success, with InceptionV3 exhibiting a high average classification accuracy of 98.8%. Additionally, the paper thoroughly compares the computational time between GPU and CPU, demonstrating a significant advantage in favor of GPU execution. The study also introduces MapReduce as a crucial parallel processing framework for real-time classification in deep learning tasks, particularly for handling extensive datasets. This framework enables distributed processing, markedly reducing time for tasks like data preprocessing, feature extraction, and model evaluation by parallelizing smaller, independent data segments. These insights contribute to the advancement of brain tumor classification methodologies, emphasizing the benefits of MapReduce for real-time classification. Future work will explore alternative approaches for large brain tumor datasets and conduct a comparative analysis to identify the most efficient approach.

REFRENCES:

[1] American Brain Tumor Association, “Types of Brain Tumors,” 2021. https://www.abta.org/tumor_types/brain-tumor-types/

[2] A. B. T. Association, “Malignant Brain Tumors,” 2018. https://www.abta.org/tumor_types/glioblastoma/

[3] J. Y. Rhee *et al.*, “Palliative care in Africa: a scoping review from 2005–16,” *Lancet Oncol.*, vol. 18, no. 9, pp. e522–e531, 2017, doi: 10.1016/S1470-2045(17)30420-5.

[4] B. Hakyemez, N. Yıldırım, G. Gokalp, C. Erdogan, and M. Parlak, “The contribution of diffusion-weighted MR imaging to distinguishing typical from atypical meningiomas,” *Neuroradiology*, vol. 48, no. 8, pp. 513–520, 2006, doi: 10.1007/s00234-006-0094-z.

[5] C. M. Zylak, J. R. Standen, G. R. Barnes, and C. J. Zylak, “Pneumomediastinum revisited,” *Radiographics*, vol. 20, no. 4, pp. 1043–1057, 2000, doi: 10.1148/radiographics.20.4.g00jl131043.

[6] A. Iglesias, M. Arias, P. Santiago, M. Rodríguez, J. Mañas, and C. Saborido, “Benign Breast Lesions that Simulate Malignancy: Magnetic Resonance Imaging with Radiologic-Pathologic Correlation,” *Curr. Probl. Diagn. Radiol.*, vol. 36, no. 2, pp. 66–82, 2007, doi: 10.1067/j.cpradiol.2006.12.001.

[7] A. R. Padhani *et al.*, “Diffusion-weighted magnetic resonance imaging as a cancer biomarker: Consensus and recommendations,” *Neoplasia*, vol. 11, no. 2, pp. 102–125, 2009, doi: 10.1593/neo.81328.

[8] G. Goos *et al.*, *ECCV Computer Vision 2014*. 2014.

[9] J. P. Lai, Y. M. Chang, C. H. Chen, and P. F. Pai, “A survey of machine learning models in renewable energy predictions,” *Appl. Sci.*, vol. 10, no. 17, 2020, doi: 10.3390/app10175975.

[10] X. Shi, L. Xu, P. Wang, Y. Gao, H. Jian, and W. Liu, “Beyond the Attention: Distinguish the Discriminative and Confusable Features for Fine-grained Image Classification,” *MM 2020 - Proc. 28th ACM Int. Conf. Multimed.*, pp. 601–609, 2020, doi: 10.1145/3394171.3413883.

[11] J. Li, Y. Liu, J. Pan, P. Zhang, W. Chen, and L. Wang, “Map-Balance-Reduce: An improved parallel programming model for load balancing of MapReduce,” *Futur.*

Gener. Comput. Syst., vol. 105, pp. 993–1001, 2020, doi: 10.1016/j.future.2017.03.013.

[12] R. Lämmel, "Google's MapReduce programming model - Revisited," *Sci. Comput. Program.*, vol. 68, no. 3, pp. 208–237, 2007, doi: 10.1016/j.scico.2007.07.001.

[13] T. Kaur and T. K. Gandhi, "Automated brain image classification based on VGG-16 and transfer learning," *Proc. - 2019 Int. Conf. Inf. Technol. ICIT 2019*, pp. 94–98, 2019, doi: 10.1109/ICIT48102.2019.00023.

[14] S. Deepak and P. M. Ameer, "Brain tumor classification using deep CNN features via transfer learning," *Comput. Biol. Med.*, vol. 111, no. June, p. 103345, 2019, doi: 10.1016/j.compbiomed.2019.103345.

[15] A. Rehman, S. Naz, M. I. Razzak, F. Akram, and M. Imran, "A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning," *Circuits, Syst. Signal Process.*, vol. 39, no. 2, pp. 757–775, 2020, doi: 10.1007/s00034-019-01246-3.

[16] R. Mehrotra, M. A. Ansari, R. Agrawal, and R. S. Anand, "A Transfer Learning approach for AI-based classification of brain tumors," *Mach. Learn. with Appl.*, vol. 2, no. October, p. 100003, 2020, doi: 10.1016/j.mlwa.2020.100003.

[17] P. Saxena, A. Maheshwari, and S. Maheshwari, "Predictive Modeling of Brain Tumor: A Deep Learning Approach," *Adv. Intell. Syst. Comput.*, vol. 1189, pp. 275–285, 2021, doi: 10.1007/978-981-15-6067-5_30.

[18] S. Manjunath, M. B. Sanjay Pande, B. N. Raveesh, and G. K. Madhusudhan, "Brain tumor detection and classification using convolution neural network," *Int. J. Recent Technol. Eng.*, vol. 8, no. 1, pp. 34–40, 2019, doi: 10.2139/ssrn.3507904.

[19] N. Nandhagopal, C. Jaichander, and R. Ponniwalavan, "Image Classification using MRI Images in Brain Tumor," *Asian J. Res. Soc. Sci. Humanit.*, vol. 6, no. cs1, p. 422, 2016, doi: 10.5958/2249-7315.2016.00974.6.

[20] M. Arbane, R. Benlamri, Y. Brik, and M. Djerioui, "Transfer Learning for Automatic Brain Tumor Classification Using MRI Images," *2020 2nd Int. Work. Human-Centric Smart Environ. Heal. Well-Being, IHSH 2020*, pp. 210–214, 2021, doi: 10.1109/IHSH51661.2021.9378739.

[21] T. Sadad *et al.*, "Brain tumor detection and multi-classification using advanced deep learning techniques," *Microsc. Res. Tech.*, vol. 84, no. 6, pp. 1296–1308, 2021, doi: 10.1002/jemt.23688.

[22] N. Ullah *et al.*, "An Effective Approach to Detect and Identify Brain Tumors Using Transfer Learning," *Appl. Sci.*, vol. 12, no. 11, 2022, doi: 10.3390/app12115645.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[24] T. D. Jeff Donahue∗ , Yangqing Jia∗ , Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, "A Deep Convolutional Activation Feature.," vol. 32, 2014.

[25] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, *A Survey on Deep Transfer Learning BT -*

Artificial Neural Networks and Machine Learning. 2018.

[26] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017, [Online]. Available: http://arxiv.org/abs/1704.04861

[27] D. Kumar, P. Samui, D. Kim, and A. Singh, “A Novel Methodology to Classify Soil Liquefaction Using Deep Learning,” *Geotech. Geol. Eng.*, vol. 39, no. 2, pp. 1049–1058, 2021, doi: 10.1007/s10706-020-01544-7.

[28] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[29] M. Ramachandran, R. Patan, A. Kumar, S. Hosseini, and A. H. Gandomi, “Mutual Informative MapReduce and Minimum Quadrangle Classification for Brain Tumor Big Data,” *IEEE Trans. Eng. Manag.*, vol. 70, no. 8, pp. 2644–2655, 2023, doi: 10.1109/TEM.2021.3073018.