Brain Tumor Detection Using Deep Learning

* Indrajit Chakraborty
Dept. of CSE
Ahsanullah University of Science & Technology
indrajit.cse.200104135@aust.edu

* Jubayer Ahmed Nayem
Dept. of CSE
Ahsanullah University of Science & Technology
jubayer.cse.200104137@aust.edu

* Tanvir Rahman
Dept. of CSE
Ahsanullah University of Science & Technology
tanvir.cse.200104136@aust.edu

* Sabbir Ahmed Salman
Dept. of CSE
Ahsanullah University of Science & Technology
sabbir.cse.200104146@aust.edu

***Abstract*—This study presents an approach for the classification of brain tumors using convolutional neural networks (CNNs) and transfer learning techniques. We have examined various pre-trained CNN architectures, including VGG, EfficientNet, Xception, and MobileNet, to distinguish between glioma, meningioma, pituitary, and non-tumor brain MRI images. Through meticulous experimentation, our CNN model has achieved an accuracy of 90.46%, which outperforms other architectures. While VGG is 87.40% and EfficientNet demonstrate accuracies of 88.55%, MobileNet achieves 88.55%. However, Xception lags behind with an accuracy of 53.14%. The findings underscore the effectiveness of CNN-based approaches in accurately categorizing brain tumors, with the CNN model showcasing superior performance. This research contributes to advancing medical image analysis, offering a robust framework for automated brain tumor classification. It has practical implications for healthcare professionals in diagnosis and treatment planning.**

***Index Terms*—Convolutional neural networks (CNNs), VGG, EfficientNet, Xception, MobileNet, MRI, Glioma, Meningioma, Pituitary tumor**

I. Introduction

The prevalence of brain tumors presents a formidable healthcare challenge, necessitating precise and prompt diagnosis for effective treatment planning. Manual interpretation of medical imaging, particularly Magnetic Resonance Imaging (MRI) scans, is laborious and prone to human error. In response, this research proposes the development of a robust Convolutional Neural Network (CNN)-based model to automatically detect and classify brain tumors in medical images. By leveraging pre-trained models such as VGG, EfficientNet, Xception, and MobileNet, this approach aims to streamline the diagnostic process, providing healthcare professionals with a reliable and efficient tool for early detection.

Early detection holds paramount importance in brain tumor cases, significantly enhancing prognosis and treatment outcomes. However, existing detection methods often rely on manual interpretation, leading to delays and potential inaccuracies. Deep learning, particularly CNNs, has demonstrated remarkable success in various image analysis tasks, including medical imaging. By specifically addressing the challenges associated with brain tumor detection, this research seeks to improve the speed and accuracy of diagnosis, ultimately advancing patient care and outcomes.

The proposed approach capitalizes on the capability of CNNs to automatically learn and extract relevant features from medical images. Pretrained models like VGG, EfficientNet, Xception, and MobileNet provide a strong foundation for this task, enabling the model to generalize and accurately identify tumors in brain MRI scans encompassing Glioma Tumor, Meningioma Tumor, No Tumor, and Pituitary Tumor cases. The convolutional layers of the network will capture hierarchical features, while subsequent fully connected layers will facilitate classification.

This research will involve fine-tuning and validation through rigorous experimentation to ensure the model's reliability and effectiveness in real-world scenarios. The anticipated outcomes of this research hold promise for significantly enhancing the efficiency and accuracy of brain tumor detection, thereby contributing to advancements in medical diagnostics.

II. Literrature Review

Paul et al. [1] Provide two neural network models: fully connected and convolutional, and use a dataset that has three classes divided into three distinct planes for classification. To prevent the model from being confused between the three different planes, the authors test the model by choosing only the axial plane for performance accuracy. They note that a basic model like the one put out can outperform and perform better than specialist methods, and they specify that the CNN performs better with an accuracy of 91.43%.Abiwinanda et al.

Abiwinanda et al.[2] investigate a basic CNN model without making any changes; instead, they work with CNN and alter its many layers by adding or removing layers. They construct seven distinct CNN architectures, varying in the number of layers for each, and find that the second architecture—which has two layers for each convolution, activation, and max-pooling—is the most effective of the bunch, yielding a training accuracy of 98.51%.

Ghassemi et al.[3] propose a pretraining-focused model and then use CNN to implement it. The pretraining of the model with several publically available datasets is the primary focus.

Following training, the model is applied to CNN, where the softmax in the main model takes the role of the fully connected layer. The resulting model achieves a 95.6% accuracy rate when tested on the main dataset (T1), which comprises three distinct kinds of tumors.

III. Dataset Review

A. Training Dataset

The **"Training"** dataset consists of a total of 2870 images, categorized into the following classes:

1) Gpaul et al. [1] Provide two neural network models: fully connected and convolutional, and use a dataset that has three classes divided into three distinct planes for classification. To prevent the model from being confused between the three different planes, the authors test the model by choosing only the axial plane for performance accuracy. They note that a basic model like the one put out can outperform and perform better than specialist methods, and they specify that the CNN performs better with an accuracy of 91.43
2) Meningioma Tumor
3) No Tumor
4) Pituitary Tumor

The images are provided in JPG format and have undergone preprocessing steps, including resizing to a standardized format. This subset serves as the foundation for training the deep learning model.

B. Testing Dataset

The **"Testing"** dataset is designed for assessing umor, meningioma tumor, no tumor, and pituitary tumor. These images, also in JPG format, have undergone the same preprocessing steps as the training set.

C. Data Preprocessing

The preprocessing steps applied to both the training and testing datasets are as follows:

- **Resizing:** All images have been resized to a uniform dimension to ensure compatibility with the deep learning model architecture.
- **Array Conversion:** The resized images have been converted into NumPy arrays for efficient handling and compatibility with deep learning frameworks.
- **Shuffling:** To enhance model training, the order of images in the training dataset has been shuffled, introducing randomness and preventing the model from learning spurious patterns based on the sequence of images.

D. Frequency Distribution

The dataset's key characteristics include a balanced representation of the four tumor classes, enabling the model to learn from diverse examples. The images exhibit variations in tumor size, location, and appearance, mirroring the challenges encountered in real-world brain tumor detection scenarios. This balanced and varied composition ensures that the curated dataset provides a comprehensive foundation for the development and evaluation of a deep-learning model for brain tumor detection. The inclusion of diverse tumor classes and robust preprocessing steps ensures that the model is trained and tested on a representative set of data, enhancing its potential for real-world application. This dataset review outlines the essential aspects of the dataset used in the study, emphasizing its importance in advancing the field of brain tumor detection using deep learning.

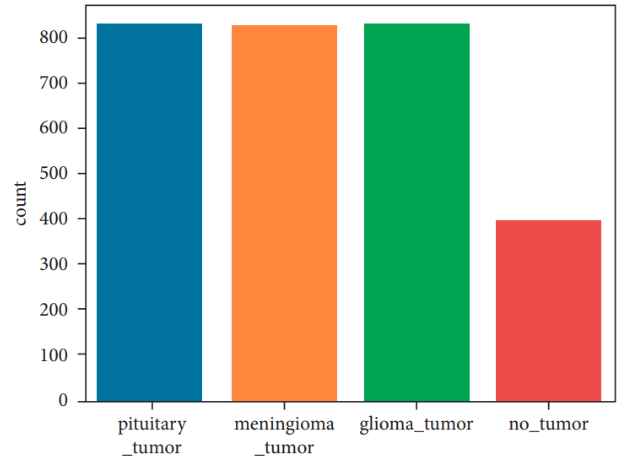

Fig. 1. Frequency Distribution of Dataset

IV. Methodology

This methodology flowchart outlines training and testing using CNN. First, we have labeled training images, preprocessed them, and choose a CNN architecture (like VGG or EfficientNet, MobileNet, Xception). Then, train the model on the images, evaluating its performance on a separate validation set. Finally, test the trained model on new images, assessing its generalization ability with metrics like accuracy. By iterating on these steps and adjusting hyperparameters, we have built a robust CNN model.

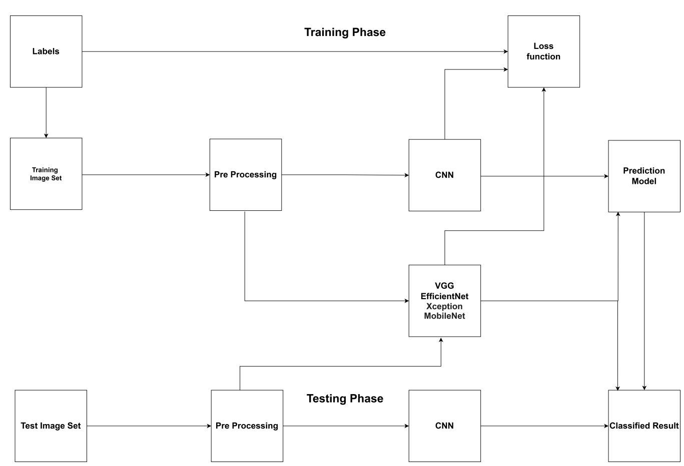

Fig. 2. Methodology Flowchart

V. Result Analysis

A. CNN

A convolutional neural network model showing significant improvement in accuracy, reaching around 90.46% on the test set after 30 epochs of training, indicating its effectiveness in image classification tasks.

TABLE I
CNN Results

Epoch	Train. Loss	Train. Acc.	Val. Loss	Val. Acc.
1	1.7378	0.2822	1.3684	0.2748
2	1.2379	0.4283	1.0615	0.5649
3	0.9907	0.5585	0.9530	0.6069
4	0.9061	0.5951	0.8386	0.6450
5	0.7804	0.6611	0.7803	0.6565
6	0.7125	0.6871	0.7706	0.6641
7	0.6581	0.7280	0.6898	0.6947
8	0.5784	0.7522	0.6993	0.7099
9	0.5118	0.7948	0.5297	0.7634
10	0.4233	0.8378	0.4705	0.7748
11	0.3490	0.8616	0.5372	0.7710
12	0.3477	0.8685	0.4455	0.8168
13	0.2866	0.8876	0.3543	0.8702
14	0.2395	0.9063	0.3678	0.8779
15	0.2759	0.8914	0.5231	0.7977
16	0.2060	0.9195	0.4798	0.8282
17	0.2220	0.9217	0.5361	0.8130
18	0.1948	0.9272	0.4491	0.8397
19	0.1746	0.9332	0.3894	0.8473
20	0.1536	0.9502	0.5098	0.8130
21	0.1307	0.9464	0.4650	0.8244
22	0.0936	0.9625	0.4256	0.8550
23	0.1453	0.9519	0.3774	0.8779
24	0.1029	0.9676	0.3715	0.8740
25	0.0843	0.9723	0.3282	0.9008
26	0.1091	0.9647	0.6025	0.8473
27	0.1401	0.9519	0.4518	0.8626
28	0.1081	0.9617	0.4933	0.8855
29	0.1091	0.9659	0.4321	0.8740
30	0.1291	0.9617	0.3215	0.9046

B. VGG

A VGG model architecture trained for image classification, achieving a high test accuracy of approximately 87.40%.

TABLE II
VGG Results

Epoch	Train. Loss	Train. Acc.	Val. Loss	Val. Acc.
1	4.8185	0.6471	1.2184	0.7939
2	1.1514	0.7718	0.5780	0.8130
3	0.6112	0.8140	0.6936	0.8321
4	0.4947	0.8506	0.8202	0.8015
5	0.5038	0.8561	0.6670	0.8359
6	0.4344	0.8782	0.5420	0.8550
7	0.3561	0.8987	0.4571	0.8702
8	0.3194	0.9106	0.6416	0.8588
9	0.3052	0.9204	0.5993	0.8626
10	0.3019	0.9149	0.4199	0.8664
11	0.2694	0.9276	0.4525	0.8931
12	0.2563	0.9302	0.6322	0.8740
13	0.2020	0.9336	0.4989	0.8893
14	0.2283	0.9408	0.4778	0.8931
15	0.2219	0.9370	0.6119	0.8855
16	0.1943	0.9438	0.7656	0.8664
17	0.2103	0.9417	0.6408	0.8893
18	0.2209	0.9425	0.5726	0.8893
19	0.2077	0.9413	0.4654	0.9046
20	0.2207	0.9408	0.7129	0.8740

C. EfficientNet

Utilizing the EfficientNet architecture, this model achieved a test accuracy of around 88.55%, demonstrating its effectiveness in image classification tasks.

TABLE III
Efficient net

Epoch	Train. Loss	Train. Acc.	Val. Loss	Val. Acc.
1	2.0498	0.6360	0.5048	0.8321
2	0.5901	0.7880	0.5272	0.8092
3	0.5071	0.8101	0.3969	0.8435
4	0.3855	0.8616	0.3962	0.8588
5	0.3690	0.8587	0.3869	0.8511
6	0.3266	0.8855	0.3620	0.8550
7	0.2942	0.8919	0.4262	0.8511
8	0.2526	0.9063	0.5108	0.8664
9	0.2585	0.9136	0.4144	0.8588
10	0.2167	0.9166	0.3208	0.8969
11	0.1918	0.9323	0.3615	0.8969
12	0.1923	0.9315	0.4023	0.8817
13	0.1900	0.9357	0.2986	0.9008
14	0.1738	0.9319	0.4107	0.8931
15	0.1603	0.9391	0.3717	0.8893
16	0.1676	0.9442	0.4055	0.8664
17	0.1665	0.9464	0.4137	0.9046
18	0.1618	0.9476	0.5721	0.8817
19	0.2152	0.9383	0.4633	0.8702
20	0.1580	0.9447	0.4670	0.8855

D. Xception

The Xception model, although showing some overfitting, demonstrated reasonable performance with a test accuracy of about 53.14%.

TABLE IV
XCEPTION RESULTS

Epoch	Train. Loss	Train. Acc.	Val. Loss	Val. Acc.
1	31.4759	0.3869	2.9360	0.4979
2	1.8324	0.4314	1.3542	0.3915
3	1.2492	0.4726	1.2183	0.4596
4	1.1877	0.4711	1.0561	0.5021
5	1.1191	0.4853	1.0419	0.4766
6	1.2195	0.4395	1.3689	0.4426
7	1.1873	0.4423	1.2713	0.5064
8	1.2110	0.4191	1.2126	0.3915
9	1.2290	0.3978	1.1640	0.4979
10	1.2013	0.4059	1.1391	0.4043
11	1.1862	0.4201	1.0721	0.4340
12	1.1319	0.4607	1.0464	0.4638
13	1.1623	0.4432	1.1502	0.4255
14	1.1635	0.4607	1.0996	0.5489
15	1.1112	0.4522	1.0659	0.4851
16	1.0968	0.4655	1.0256	0.4851
17	1.1157	0.4560	1.1257	0.5532
18	1.1343	0.4678	1.0508	0.4426
19	1.1151	0.4551	1.0946	0.5447
20	1.1020	0.4664	1.2228	0.5314

E. MobileNet

Employing the MobileNet architecture, this model exhibited good performance with a test accuracy of approximately 88.55%, showcasing its suitability for resource-constrained environments.

TABLE V
MOBILE NET

Epoch	Train. Loss	Train. Acc.	Val. Loss	Val. Acc.
1	3.5782	0.5121	0.7106	0.7214
2	0.7529	0.6909	0.6392	0.7328
3	0.6335	0.7322	0.6149	0.7672
4	0.5797	0.7590	0.6068	0.8206
5	0.5292	0.7914	0.5153	0.8130
6	0.5013	0.8118	0.5083	0.7824
7	0.4571	0.8097	0.4587	0.8435
8	0.4142	0.8306	0.4230	0.8740
9	0.4053	0.8284	0.4262	0.8626
10	0.3670	0.8472	0.4488	0.8817
11	0.3842	0.8382	0.3838	0.8664
12	0.3771	0.8382	0.4284	0.8550
13	0.3346	0.8565	0.4873	0.8626
14	0.2879	0.8736	0.4212	0.8588
15	0.3785	0.8378	0.4207	0.8664
16	0.4190	0.8272	0.5086	0.8359
17	0.4987	0.7803	0.4499	0.8435
18	0.3720	0.8416	0.4854	0.8511
19	0.3544	0.8625	0.4298	0.8740
20	0.3023	0.8714	0.3929	0.8855

In the comparison of CNN architectures for brain tumor detection, the CNN model achieved the highest accuracy at 90.46%, followed closely by VGG 87.40% and EfficientNet at 88.55%. MobileNet showed respectable performance at 88.55%, while Xception lagged behind at 53.14%. These findings highlight the CNN model's effectiveness, with VGG and EfficientNet also performing well. MobileNet offers efficiency, but Xception falls short in accuracy. These insights aid in selecting suitable architectures based on accuracy and computational needs in brain tumor classification tasks.

VI. CONCLUSION

The study evaluates CNN architectures for brain tumor detection, finding the CNN model achieved 90.46% accuracy, outperforming VGG, EfficientNet, Xception, and MobileNet. This underscores CNN's efficacy in brain tumor classification, with practical implications for healthcare. Future research can refine CNN models and explore new datasets for improved performance.

VII. REFERENCES

[1] J. S. Paul, A. J. Plassard, B. A. Landman, and D. Fabbri, "Deep learning for brain tumor classi- fication," Proceedings of SPIE, vol. 10137, pp. 1–16, 2017.

[2] N. Abiwinanda, M. Hanif, S. T. Hesaputra, A. Handayani, and T.R.Mengko, "Brain tumor classification using convolutional neural network," in World congress on Medical Physics and Biomedical Engineering 2018, pp. 183–189, Springer, Singapore, 2019.

[3] N. Ghassemi, A. Shoeibi, and M. Rouhani, "Deep neural network with generative adversar- ial networks pre-training for brain tumor classification based on MR images," Biomedical Signal Processing and Control, vol. 57, Article ID 101678, 2020.