

Brain Tumor Classification Using Machine Learning

Guide: Prof. Mukund Kulkarni, Group Members: Aaditya Deshpande (1), Ajinkya Walunj (5), Akash Bhandari(6),Aneesh Dighe(10),Anish Sagri(11)

Department of Computer Engineering
Vishwakarma Institute of Technology, Pune, 411037, Maharashtra, India

Abstract —A brain tumor, also known as an intracranial tumor, is an abnormal mass of tissue in which cells grow and multiply uncontrollably, seemingly unchecked by normal cell control mechanisms. More than 150 different types of brain tumors have been identified, but the two main types of brain tumors are classified as primary and metastatic. The classification of medical images has attracted a lot of attention recently, and the Convolutional Neural Network (CNN) is the most popular neural network model for this task. CNN uses a variety of building pieces, including convolution layers, pooling layers, and fully connected layers, to determine features adaptively through backpropagation. Creating a CNN model for diagnosing brain cancers in T1-weighted contrast-enhanced MRI images was the major goal of this study. There are two key steps in the suggested system. Utilize several image processing techniques to preprocess the photos, and then use CNN to classify the preprocessed images. The study uses a collection of 3264 photos that includes three different forms of brain tumors (glioma, meningioma, pituitary).

Keywords —Brain Tumor; Convolutional Neural Network, ReLu, Max Pooling

I. INTRODUCTION

A mass or collection of aberrant brain cells is known as a brain tumor. Your brain is housed inside a highly stiff skull. Any growth within such a constrained area can lead to issues.

Brain tumors can be malignant (cancerous) or benign (not cancerous). The pressure inside your skull may rise as benign or cancerous tumors enlarge. This has the potential to be fatal and can result in brain damage. According to the American Cancer Society's prediction, 23,820 malignant tumors of the brain or spinal cord (13,410 in men and 10,410 in women) will be identified in the United States in 2019. Benign (non-cancerous) tumors are not included in these estimates. They determined that there is a chance that 17,760 people (9,910 men and 7,850 women) will pass away from brain and spinal cord malignancies in 2019. The likelihood of surviving a brain tumor varies depending on the patient's age and the type of tumor. According to the Central Brain Tumor Registry of the United States (CBTRUS), the survival rate varies greatly depending on the patient's age. In this study, we suggested a CNN model that can correctly categorize brain tumors. As a result, the tumor's treatment might begin at an early stage. A neural network (artificial neuron network) is a computational model that mimics how nerve cells function in the human brain.

Artificial neural networks (ANNs) employ learning algorithms that can independently make adjustments - or learn - as new input is received. As a result, they are an extremely effective tool for non-linear statistical data modelling. Deep understanding ANNs are crucial in machine learning (ML) and the broader field of artificial intelligence (AI) technology. A convolutional neural network (CNN/ConvNet) is a type of deep neural network that is commonly used to analyses visual imagery in deep learning. When we think of neural networks, we typically think of matrix multiplications, but this is not the case with

ConvNet. It employs a technique known as Convolution. Convolution is a mathematical operation on two functions that yields a third function that expresses how the shape of one is modified by the other.

II. Literature Review

[1] The paper discusses the Deep Learning technique, which was used to classify four types of tumors: normal, glioblastoma, sarcoma, and metastatic bronchogenic carcinoma tumors. The proposed methodology trains the DNN classifier for brain tumor classification using a set of features extracted from segmented brain MRI images using the discrete wavelet transform (DWT) feature extraction technique.

The image was segmented using the Fuzzy C means clustering technique, which divides the image into five parts. After segmenting the Brain MR images into 5 sections, the segmented tumor's features are extracted using the discrete wavelet transform (DWT). As they provide localized time-frequency information of a signal using cascaded filter banks of high-pass and low-pass filters to extract features in a hierarchy manner, DWT has the advantage of extracting the most relevant features at different directions and scales. The proposed methodology's performance was measured in terms of average classification rate, average recall, average precision, average F-Measure, and average area under the ROC curve (AUC) of all four classes (normal, glioblastoma, sarcoma, and metastatic bronchogenic carcinoma tumors) and compared to the performance of other classifiers in the same terms.

[2] This study proposes a computerized system to differentiate between normal and abnormal MRI images and again classify the abnormal MRI's into HGG or LGG glioma tumors. The tumor is segmented by k-means. The main contribution of this study is the development of a computerized system for classifying abnormal MRI images into HGG or LGG glioma tumors using selective relevant features for the required procedure. Noise reduction, smoothing, feature extraction, feature reduction, and classification were the five stages of the proposed technique. The proposed system is intended for the accurate detection and

classification of normal and abnormal brain MRIs, as well as the classification of abnormal MRIs into HGG or LGG glioma tumors. The discrete wavelet transform is used for feature extraction (DWT). It is used to calculate the wavelet coefficient from MRI images. A subset of DWT coefficients captures all of the images' important features. SVM classifies between two classes by creating a hyperplane in high-dimensional feature space. Because the data has already been labelled, the term "supervised learning" was coined. It uses the training data to find the hyperplane with the greatest distance to the nearest training data point of any class.

III. METHODOLOGY/EXPERIMENTAL

Materials and Technology Used:

1. Technology- Jupyter Notebook, VS Code, Keras, Pandas, Matplotlib and TensorFlow

1. Jupyter Notebook- The original web application for creating and sharing computational documents is Jupyter Notebook. It provides a straightforward, streamlined, document-centric experience. Jupyter Notebook Advantages Jupyter Notebook, which was originally designed for data science applications written in Python, R, and Julia, is now useful for a wide range of projects: Visualizations of data. Most people are introduced to Jupyter Notebook through a data visualization, which is a shared notebook that includes a graphic rendering of some data set. Jupyter Notebook allows you to create visualizations, but also share them and make changes to the shared code and data set interactively. Sharing of code. Cloud services such as GitHub and Pastebin allow you to share code, but they are largely inactive. A Jupyter Notebook allows you to view code, execute it, and view the results in your web browser. Interactions with code in real time. The code in Jupyter Notebook is not static; it can be edited and re-run incrementally in real time, with feedback displayed directly in the browser. Notebooks Code samples are being documented. You could embed a Jupyter Notebook if you have a piece of code and want to explain it line by line, with live feedback all along the way. The best part is that the code will remain

fully functional—you can include interactivity alongside the explanation, showing and telling at the same time.

2. Visual Studio Code - Visual Studio Code is a lightweight but powerful source code editor for Windows, macOS, and Linux that runs on your desktop. It includes built-in support for JavaScript, TypeScript, and Node.js, as well as a robust ecosystem of extensions for other languages and runtimes (including C++, C#, Java, Python, PHP, Go, and .NET). Jupyter Notebooks are supported natively in Visual Studio Code, as well as through Python code files. This topic discusses the native support for Jupyter Notebooks and shows how to: Jupyter Notebooks can be created, opened, and saved. Use of Jupyter code cells.

1. View, inspect, and filter variables using the
2. Variable Explorer and Data Viewer
3. Connect to a Jupyter server.
4. Jupyter Notebook debugging

3. Tensorflow: TensorFlow's platform aids in the implementation of best practices for data automation, model tracking, performance monitoring, and model retraining. It is critical to use production-level tools to automate and track model training over the life of a product, service, or business process. TensorFlow enables developers to create dataflow graphs, which are structures that describe how data flows through a graph or series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a tensor, or multidimensional data array.

4. matplotlib-Matplotlib is a data visualization and graphical plotting library for Python and its numerical extension NumPy that is cross-platform. As such, it offers a viable open-source alternative to MATLAB. Developers can also embed plots in GUI applications by using matplotlib's APIs (Application Programming Interfaces). In most cases, a Python matplotlib script is structured so that only a few lines of code are required to generate a visual data plot. The matplotlib scripting layer is a layer that sits on top of two APIs:

1. The pyplot API is a set of Python code objects that is topped by matplotlib.
2. pyplot An OO (Object-Oriented) API collection of objects that can be assembled with greater

flexibility than pyplot. This API allows you to directly access Matplotlib's backend layers.

5. Keras-Keras is a Python-based deep learning API that runs on the TensorFlow machine learning platform. It was designed with the goal of allowing for quick experimentation. It is critical to be able to move from idea to result as quickly as possible when conducting research.

Keras is:

Simple, but not simplistic. Keras reduces developer cognitive load, allowing you to focus on the parts of the problem that really matter.

Keras follows the principle of progressive disclosure of complexity: simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds on what you've already learned.

Keras provides industry-leading performance and scalability: it is used by organizations and companies such as NASA, YouTube, and Waymo.

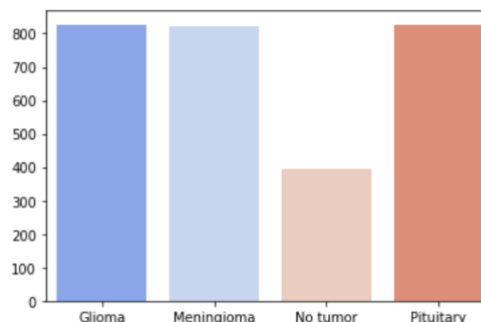
6. Pandas-Pandas is a popular open-source Python package for data science/data analysis and machine learning tasks. It is built on top of NumPy, another package that supports multidimensional arrays.

A. Method

1. Image Database:-

The dataset used was “Brain Tumor Classification (MRI)” from Kaggle. It contains Two directories labelled as Training and Testing which were divided into four parts:

Meningioma Tumor, Glioma tumor, pituitary tumor and no tumor.



2. Image Pre-Processing

The images were resized to 130 x 130 pixels and divided into batches of 16 using the image Data generator of the keras module.

3. Network Architecture

CNNs are unusually multi-layer neural networks. It is an effective recognition algorithm applied in pattern recognition and image processing. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that has the ability to take in an input image, rank various attributes and objects within the image, and distinguish between them. Comparatively speaking, a ConvNet requires substantially less pre-processing than other classification techniques. ConvNets have the capacity to learn these filters and properties, whereas in primitive techniques filters are hand-engineered. A ConvNet's architecture was influenced by how the Visual Cortex is organised and is similar to the connectivity network of neurons in the human brain. Only in this constrained area of the visual field, known as the Receptive Field, do individual neurons react to stimuli. The entire visual field is covered by a series of such fields that overlap. In our proposed network, a 2x2 kernel is used as filter with strides as 1, padding = 'same' is used. Stride defines how much the convolution filter matrix is moved at each step. Padding is same' Because the output size and the input size are equal (with stride=1). By using "SAME," it makes sure that all of the input's components are applied to the filter. This network has 3 subsampling layers and 3 Convolutional filter layers. By pooling, training time can be shortened by lowering the number of parameters. In first convolution (C1), the input image I with size 130 x 130 convolved with 32 filters with same padding and stride = 1 which produced 130x130 sized 32 convolved features. Hence, total parameters for this operation are 416. The algorithms will pick up these parameters throughout the training process. ReLU (Rectified Linear Unit) is employed as the activation function in each convolution layer. The images' non-linearity rises as a result. The C1 layer's generated features are passed into the first subsampling layer. Max pooling is utilised in

subsampling layer with a 22-window size. Then a layer with a 20% dropout rate is applied. Dropout is a regularisation technique that is used to reduce overfitting. According to the dropout rate, certain randomly chosen neurons are disregarded in dropout, which means that their ability to activate downstream neurons is shut down on the forward pass and that no weight updates are applied to the neurons on the backward pass. The second Convolutional layer was same as the first layer. 64 filters are employed in the third and final convolution layer (C3) to produce 64 feature maps, each measuring 32 by 32 pixels. Total parameters in this case are 8256. The output was then flattened using a second dropout layer with a 20% dropout rate. The model uses 3 Dense layers of 128, 32, 16 neurons respectively which is activated using the relu function. Finally, the output layers consist of 4 neurons which uses the SoftMax activation function.

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 130, 130, 32)	416
max_pooling2d_3 (MaxPooling 2D)	(None, 65, 65, 32)	0
conv2d_4 (Conv2D)	(None, 65, 65, 32)	4128
max_pooling2d_4 (MaxPooling 2D)	(None, 32, 32, 32)	0
dropout_2 (Dropout)	(None, 32, 32, 32)	0
conv2d_5 (Conv2D)	(None, 32, 32, 64)	8256
max_pooling2d_5 (MaxPooling 2D)	(None, 16, 16, 64)	0
dropout_3 (Dropout)	(None, 16, 16, 64)	0
flatten_1 (Flatten)	(None, 16384)	0
dense_4 (Dense)	(None, 128)	2097280
dense_5 (Dense)	(None, 32)	4128
dense_6 (Dense)	(None, 16)	528
dense_7 (Dense)	(None, 4)	68
Total params: 2,114,804		
Trainable params: 2,114,804		
Non-trainable params: 0		

IV. RESULTS AND DISCUSSIONS

The Adam optimizer is used to compile the model. It is a highly effective optimization approach used for deep neural network training. It combines the benefits of two optimization techniques, RMSProp (Root Mean Square Propagation) and AdaGrad (Adaptive Gradient Algorithm). A batch size of 16

and 50 epochs are utilised to fit the training data with the model. Our model gave an accuracy of 98.63% during training. The output is shown in the figure below.

```
from matplotlib.pyplot import imshow
from PIL import Image, ImageOps
data = np.ndarray(shape=(1, 130, 130, 3), dtype=np.float32)
image = Image.open('D:\\Students\\CAPSTONE PROJECT\\My Brain\\archive\\testing\\meningioma_tumor\\image(30).jpg')
size = (130, 130)

image = ImageOps.fit(image, size)
image_array = np.asarray(image)
image_array.shape
display(image)

data[0] = image

prediction = saved_model.predict(data)
print(prediction)
predict_index = np.argmax(prediction)
print(predict_index)
print(classes[predict_index])

1/1 [=====] - 0s 150ms/step
[[4.6323362e-11 9.9990571e-01 3.6132522e-05 5.8162561e-05]]
1
meningioma_tumor
```

V.CONCLUSION

The classification model based on CNN using Deep Learning was built successfully and the tumors were divided into four classes. CNN model is totally based on hit and trial. At every step tuning is required. Finally, the model worked well on test data as well.

VI. FUTURE SCOPE

- 1.Improve the model accuracy and try to integrate more types of tumors for classification.
- 2.Try different methods for feature extraction and classification and compare it with our model.

ACKNOWLEDGMENT

Hereby we would thank our faculty Mr. Mukund Kulkarni sir who helped us in developing the project. I would also like to express my gratitude to college dignitaries for giving us such a great opportunity of developing this project. We learnt many new concepts and further by using this we

will develop more useful and productive and projects required for societal needs in the future.

REFERENCES

- [1]T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of training convolutional neural networks," arXiv preprint arXiv:1506.01195, 2015.
- [2] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.
- [3].S. Das, O. F. M. R. R. Aranya and N. N. Labiba, "Brain Tumor Classification Using Convolutional Neural Network," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019, pp. 1-5, doi: 0.1109/ICASERT.2019.8934603.