Machine Learning Based Brain Tumor Prediction Using Decision Tree Algorithm

Saraswathi P[1], Srimathi R[2], Suriyalashmi V[3], Krithikashree J[4]

[1]Assistant Professor-III, Department of Information Technology, Velammal College of Engineering and Technology, Madurai

[2][3][4]UG Students, Department of Information Technology, Velammal College of Engineering and Technology, Madurai

---***---

Abstract: *A brain tumor is a collection, or mass, of abnormal cells in your brain. Your skull, which encloses your brain, is very rigid. Any growth inside such a restricted space can cause problems. Brain tumors can be cancerous (malignant) or noncancerous (benign). When benign or malignant tumors grow, they can cause the pressure inside your skull to increase. This can cause brain damage, and it can be life-threatening. This paper discuss detection/prediction of brain tumor using a machine learning technique "Decision tree" algorithm and Image Processing techniques.*

Keywords- Brain tumor, Decision tree, Magnetic Resonance Image(MRI), Image processing, Machine learning

1. INTRODUCTION

A brain tumor is a collection, or mass, of abnormal cells in your brain. Human skull, which encloses our brain, is very rigid. Any growth inside such a restricted space can cause problems. Brain tumors can be cancerous (malignant) or noncancerous (benign). When benign or malignant tumors grow, they can cause the pressure inside your skull to increase. This can cause brain damage, and it can be life threatening. Brain tumors are categorized as primary or secondary. A primary brain tumor originates in your brain. Many primary brain tumors are benign. A secondary brain tumor, also known as a metastatic brain tumor, occurs when cancer cells spread to your brain from another organ, such as your lung or breast. Symptoms of brain tumors depend on the location and size of the tumor. Some tumors cause direct damage by invading brain tissue and some tumors cause pressure on the surrounding brain. You'll have noticeable symptoms when a growing tumor is putting pressure on your brain tissue. If you have an MRI of your head, a special dye can be used to help your doctor detect tumors. An MRI is different from a CT scan because it doesn't use radiation, and it generally provides much more detailed pictures of the structures of the brain itself. The tumor is basically an uncontrolled growth of cancerous cells in any part of the body, whereas a brain tumor is an uncontrolled growth of cancerous cells in the brain. A brain tumor can be benign or malignant. The benign brain tumor has a uniformity in structure and does not contain active (cancer) cells, whereas malignant brain tumors have a non-uniformity (heterogeneous) in structure and contain active cells. The gliomas and meningiomas are the examples of low-grade tumors, classified as benign tumors and glioblastoma and astrocytomas are a class of high-grade tumors, classified as malignant tumors.

2. EXISTING SYSTEM

Brain tumor is one of disease type that attacks the brain in the form of clots. There is a way to see brain tumor in detail requires by an MRI image. There is difficulty in distinguishing brain tumor tissue from normal tissue because of the similar color. Brain tumor must be analyzed accurately. The solution for analyze brain tumor is doing segmentation. Brain tumor segmentation is done to separate brain tumor tissue from other tissues such as fat, edema, normal brain tissue and cerebrospinal fluid to overcome this difficulty, The MRI image must be maintained at the edge of the image first with the median filtering. Then the tumor segmentation process requires thresholding method which is then iterated to take the largest area. The brain segmentation is done by giving a mark on the area of the brain and areas outside the brain using watershed method then clearing skull with cropping method. In this study, 14 brain tumor MRI images are used. The segmentation results are compared brain tumors area and brain tissues area. This system obtained the calculation of tumor area has an average error of 10%.

3. PROPOSED SYSTEM

Predicting brain tumors based on machine learning technique "Decision tree algorithm". Segmenting, feature extraction and perimeter finding from MRI images of brain. Using feature extraction, perimeter finding and train the machine using Decision tree algorithm. Testing the accuracy of a disease using confusion matrix and classify the disease with above 95% accuracy.

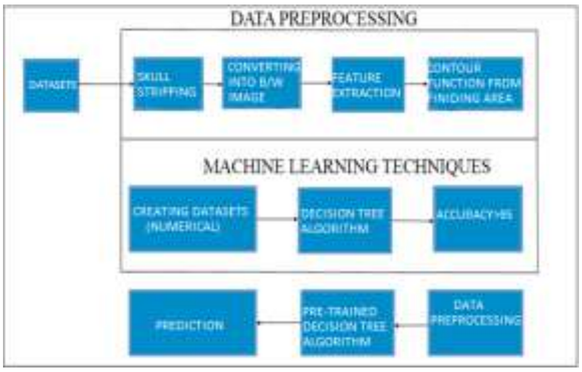
DATA PREPROCESSING
DATASETS
SKULL STRIPPING
CONVERTING INTO B/W IMAGE
FEATURE EXTRACTION
CONTOUR FUNCTION FROM FINDING AREA
MACHINE LEARNING TECHNIQUES
CREATING DATASETS (NUMERICAL)
DECISION TREE ALGORITHM
ACCURACY-95
PREDICTION
PRE-TRAINED DECISION TREE ALGORITHM
DATA PREPROCESSING

Fig-1: Block Diagram of Proposed System

4. SOFTWARES USED

4.1. Python IDLE

IDLE (Integrated Development and Learning Environment) is an integrated development environment (IDE) for Python. The Python installer for Windows contains the IDLE module by default. IDLE is not available by default in Python distributions for Linux. It needs to be installed using the respective package managers. IDLE can be used to execute a single statement just like Python Shell and also to create, modify and execute Python scripts. IDLE provides a fully-featured text editor to create Python scripts that includes features like syntax highlighting, auto completion and smart indent. It also has a debugger with stepping and breakpoints features.

4.2. Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Python is Interpreted − Python is processed at runtime by the interpreter. This is similar to PERL and PHP. Python is Interactive − We can actually sit at a Python prompt and interact with the interpreter directly to write our programs. Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It supports Object Oriented programming approach to develop applications. Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development. Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

5. LIBRARIES USED

5.1 OpenCV

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays.OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. In simple language it is library used for Image Processing. It is mainly used to do all the operation related to Images. OpenCV is available for free of cost.Since the OpenCV library is written in C/C++, so it is quit fast. Now it can be used with Python. It require less RAM to usage, it maybe of 60-70 MB. Computer Vision is portable as OpenCV and can run on any device that can run on C.

5.2 Scikit-Learn

Scikit-learn is probably the most useful library for machine learning in Python. It is on NumPy, SciPy and matplotlib, this library contains a lot of effiecient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. Scikit-learn is an open source Python library that has powerful tools for data analysis and data mining. Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy. Its name stems from the notion that it is a "SciKit" (SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy. The original codebase was later rewritten by other developers.

5.3 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

6. ALGORITHM USED

6.1. Decision tree classification

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. Decision trees can handle both categorical and numerical data.

6.2. Method

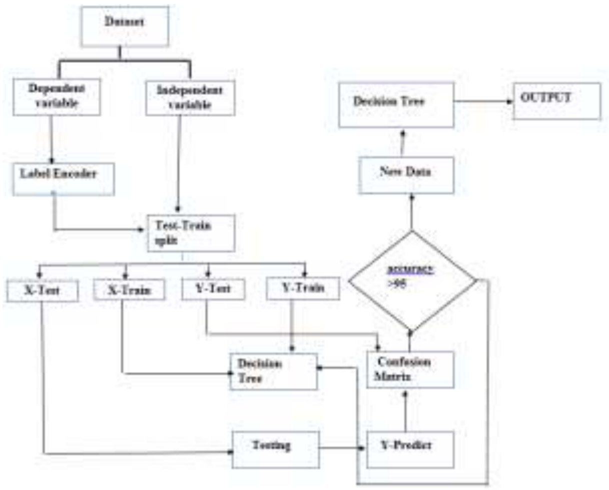

Fig-2 : Flow diagram of proposed algorithm

The workflow of brain tumor segmentation involves the following steps:

- Skull striping
- Feature Extraction
- Finding Perimeter by contours
- Dataset Preparation
- Import Libraries
- Import Dataset
- Encoding Categorical data
- Splitting the dataset
- Feature scaling

6.2.1. Skull Striping

Skull stripping is an important process in biomedical image analysis, and it is required for the effective examination of brain tumor from the MR images. Skull stripping is the process of eliminating all non-brain tissues in the brain images. By skull stripping, it is possible to remove additional cerebral tissues such as fat, skin, and skull in the brain images.

6.2.2. Feature Extraction

Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction.

When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vector). Determining a subset of the initial features is called feature selection.

6.2.3. Finding Perimeter by contours

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition.

For better accuracy, use binary images. So before finding contours, apply threshold or canny edge detection.

In OpenCV, finding contours is like finding white object from black background. So, object to be found should be white and background should be black.

6.2.4. Dataset Preparation

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable.

In statistics, data sets usually come from actual observations obtained by sampling a statistical population, and each row corresponds to the observations on one element of that population. Data sets may further be generated by algorithms for the purpose of testing certain kinds of software.

6.2.5. Import Libraries

First step is usually importing the libraries that will be needed in the program. A library is essentially a collection of modules that can be called and used. A lot of things in the programming world do not need to be written explicitly every time they are required. There are functions for them, which can simply be invoked.

```
import pandas as pd
import numpy as np
```

6.2.6. Import the dataset

A lot of datasets come in CSV formats. We will need to locate the directory of the CSV file at first (it's more efficient to keep the dataset in the same directory as your program) and read it using a method called *read_csv* which can be found in the library called *pandas*.

import pandas as pddataset = pd.read_csv('Medium.csv')

After inspecting our dataset carefully, we are going to create a matrix of features in our dataset (X) and create a dependent vector (Y) with their respective observations. To read the columns, we will use *iloc* of *pandas* (used to fix the indexes for selection) which takes two parameters — [row selection, column selection].

X = dataset.iloc[:, :-1].values

: as a parameter selects all. So the above piece of code selects all the rows. For columns we have :-1, which means all the columns except the last one.

6.2.7. Encoding Categorical Data

Sometimes our data is in qualitative form, that is we have texts as our data. We can find categories in text form. Now it gets complicated for machines to understand texts and process them, rather than numbers, since the models are based on mathematical equations and calculations. Therefore, we have to encode the categorical data.

we will import the scikit library. There's a class in the library called LabelEncoder which we will use for the task.

from sklearn.preprocessing import LabelEncoder

The next step is usually to create an object of that class. We will call our object labelencoder_X.

labelencoder_X = LabelEncoder()

6.2.8. Splitting the Dataset

Now we need to split our dataset into two sets:

- Training set
- Test set.

We will train our machine learning models on our training set, i.e our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to check how accurately it can predict. A general rule of the thumb is to allocate 80% of the dataset to training set and the remaining 20% to test set. For this task,we will import *test_train_split* from *model_selection* library of scikit.

from sklearn.model_selection import train_test_split

Now to build our training and test sets, we will create 4 sets

- X_train (training part of the matrix of features)
- X_test (test part of the matrix of features)
- Y_train (training part of the dependent variables associated with the X train sets, and therefore also the same indices)
- Y_test (test part of the dependent variables associated with the X test sets, and therefore also the same indices).

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2)

6.2.9. Feature Scaling

The final step of data preprocessing is to apply the very important feature scaling. It is a method used to standardize the range of independent variables or features of data.
from sklearn.preprocessing import StandardScaler sc_X = StandardScaler()

X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)

Now we will fit and transform our X_train set . That will transform all the data to a same standardized scale.

7. ACCURACY

The Accuracy of a system has been measured using Confusion matrix.

A confusion matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset.

Calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of errors it is making.

8. ADVANTAGES

- Extremely fast at classifying unknown records.
- Easy to interpret for small-sized trees
- Accuracy comparable to other classification techniques for many simple data sets.

9. CONCLUSION

We have tried to construct an ensemble to predict if a patient has brain tumor or not using features from brain

MRI images. After training and testing the model the accuracy we get is quite similar. Datasets is providing higher accuracy rate for predicting brain tumor.

The accuracy of brain tumor prediction is to be 98% using Decision Tree algorithm. The structure of our research has been built in such a way that with proper dataset and minor alternation it can work to classify the disease in any number of categories.

10. REFERENCES

[1] Sami S A, Gemal M Selim, Automated brain tumor detection and identification using image processing and probabilistic neural network techniques, International Journal of Image Processing and Visual Communication, ISSN:2319-1724, Volume 1, Issue 2, October 2012.

[2] K Bhima, A Jagan, Analysis of MRI based brain tumor identification using segmentation technique, International Conference on Communication and Signal Processing, Vol.2, Issue 4, April 2016.Hsin-Ying Wu, Understanding Customers Using Facebook Pages: Data Mining Users Feedback Using Text Analysis, IEEE 2014.

[3] Benson C C, Deepa V, Brain tumor segmentation from MR brain images using improved fuzzy-c means clustering and watershed algorithm", Conference on Advances in Computing, Communications and Informatics, Vol.1, Issue 2, September 2016.Rehab M. Duwairi, RUM Extractor: A Facebook Extractor for Data Analysis , IEEE 2015.

[4] Megha A Joshi, Prof D H Shah, Survey of brain tumor detection techniques through MRI images, American International Journal of Research in Formal, Applied and Natural Sciences, Vol-2, Page 9-12, June 2015.

[5] Swati Ghare, Nikita Gaikwad, Neha Kulkarni, Detection of possibility of Brain tumor using image segmentation, International Journal of Innovative Research in Computer and Communication Engineering, Vol-3, Issue 4, April 2015.