ABSTRACT

This paper present a model which is based on machine learning algorithms to detect brain tumours from magnetic resonance images with high accuracy. A Convolutional Neutral Network (CNN) has been used as the algorithm for feature extraction, and segmentation. The dataset used has been acquired from an internet website. The results show that this technique is promising and the accuracy of --- has been achieved

Keywords: Image segmentation, CNN, Augmentation, Image classification, MRI

Table of Content

List of Table

Abbreviations

MRI Magnetic Resonance Imaging

FCM Fuzzy Segmentation Method

DWT Discrete Wavelet Transform

DNN Deep Neural Network

KNN K-Nearest Neighbors Algorithm

LDA Linear Discriminant Analysis

SMO Sequential Minimal Optimization

LIPC Local Independent Projection

CA Cellular Automata

VOI Volume of Interest

MRF Margo Random Field Segmentation

SVM Support Vector Machine

RF Random Forest

MLP Multilayer Layer Preception

PCA Principle Component Analysis

BRATS Multimodal Brain Tumor Image SegmentationBenchmark

CNN Convolutional Neural Network

1. INTRODUCTION

This project has discussed the different stage of brain tumours. Cerebrum cancer division is a significant assignment in clinical picture handling. Early determination of mind cancers assumes a significant part in further developing therapy prospects and finding, from enormous measure of MRI pictures produced in clinical daily practice, is a troublesome and tedious errand. Additionally, mind growth analysis requires an acute level of precision, where a minor mistake in decision making may result in a calamity. Consequently, cerebrum cancer division is difficult for clinical purposes.

Among the right now proposed mind division strategies, cerebrum cancer division techniques dependent on conventional picture handling isn't sufficiently ideal. In customary strategy, an MRI is produced by utilizing attractive field radiation through which a two dimensional picture (predominantly dependent on a particular dark scale) is created and afterwards that picture is handled and inspected by a clinical expert. This makes a chance of human mistake and increments the general danger element of a clinical case which can at times prompt lamentable conceivable outcomes. That is the reason there is a requirement for programmed mind cancer picture division.

Current models that are based on deep learning algorithms are facing a big issue and that is their accuracy. And accuracy plays a crucial role in health care intelligent systems, hence for solving this issue, this model which is highly accurate has been developed.

2. LITRATURE REVIEW

In [1], the Fuzzy segmentation method (FCM) was applied to separate tumour and non-tumour regions of the brain. Wavelet features were alsoextracted using a multilevel discrete wavelet transform (DWT). Finally, deep neural networks (DNNs) were incorporated to classify brain tumours with high accuracy. This technique was compared with the methods of KNN classifier, Linear Discriminant Analysis (LDA) and Sequential Minimum Optimization (SMO). The accuracy rate was 96.97 in the DNN-based brain tumour classification analysis. But the complexity was very high and the performance was very poor. In [2], a new biomechanical model of tumour growth was presented for step-by-step analysis of patient tumour growth. It will be applied to gliomas and individual fringed solid tumours to capture a significant tumour effect. Discrete and continuous methods were combined to model tumour growth. The proposed scheme provides the possibility of implicit segmentation of atlas-based registry-based tumour-bearing brain images. This technique was mainly used to segment brain tissue. But the computation time was high.

Paper [3], exploited the new multi-feature feature (Multi FD) and improved the AdaBoost classification scheme used for brain tumour detection and segmentation. Structures of brain tumour tissue were extracted using the Multi FD feature extraction scheme. Advanced AdaBoost classification was used to determine whether the donated brain tissue is tumour tissue or non-tumour. Paper [4], explained a highly complex work that the Local Independent Projection (LIPC) based classifier was used to classify brain voxels. Also, the path function was extracted in this method.

In [5], a new method of segmenting granular tumours using the Cellular Automata (CA) technique was presented, which is compared with the histogram-based segmentation method. Seed selection and volume of interest (VOI) were calculated for efficient segmentation of brain tumours. Segmentation of tumour sections was also incorporated into this work. Thus, the complexity was less but the accuracy was also less. In [6], a brain tumour segmentation method, also known as multimodal brain tumour segmentation diagram was introduced. Also, it combined different segmentation algorithms to achieve high performance compared to the existing method. But the complexity was high.

In [7], studies on brain tumour segmentation were presented. It discussed different segmentation methods like Area Based Segmentation, Threshold Based Segmentation, C means Fuzzy Segmentation, Map-Based Segmentation, Margo Random Field Segmentation (MRF), Modelling formattable, geometry deformable model, Accuracy, robustness, validity, analysed for different types of models. In [8], Hybrid feature selection by ensemble classification was applied to the diagnostic process of brain tumours. It used the GANNIGMAC, decision tree, and bagging C-based wrapper approaches to get the decision rules. It also simplified decision rules with hybrid feature selection that includes a combination of (GANNIGMAC + MRMR C + Bagging C + Decision Tree).

Paper [9], used a Convolutional neural network for their algorithm and the data set used was BRATS 2015. The limitation was that the computation time was high. In [10], the algorithm used was KNN (k-nearest neighbour) but the accuracy achieved while using the proposed method was 62.07%. [11] used a Convolutional neural network for their algorithm and the data set used was BRATS 2015.

The limitation was that the computation time was high. In [12], the data set was acquired from the internet website GitHub. 2 different algorithms ANN (Artificial neural network) and CNN (Convolutional neural network) were used. The final result achieved was CNN is more accurate than ANN.

In [13], the dataset was acquired from sources University of Pavia, Kennedy space centre, Indian Pives. The algorithm used was CNN and the accuracy achieved was 88.75%. In [14], the dataset was acquired from The Cancer Imaging Archive" (TCIA). The algorithm used was KNN and classifiers are SVM, RF, LOG, MLP and PCA. The accuracy achieved by the proposed method was 83%. In [15], the dataset was acquired from Fig share Cheng. The algorithm used was CNN (Convolutional neural network) and the accuracy achieved was 84.19%.

3. MATERIAL REQUIRED

The dataset used in our model was obtained from Github and includes 000 MRI images. We categorized the images into two groups: tumour (representing unhealthy brains) and no tumour (representing healthy brains). The tumour group includes 000 images, while the no tumour group includes 000 images. The total dataset was trained on, with 000 images reserved for Preprocessing and testing. All MRI images in the dataset have the same dimensions. We chose to obtain the dataset from websites due to the difficulty of acquiring it from hospitals.

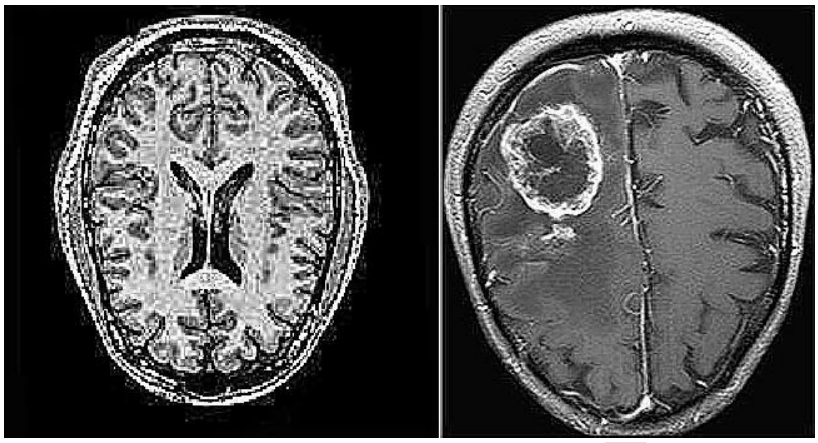


Figure 1. MRI images of the brain without tumour and with tumour

Number of images	Folder Directory
253	Total
98	No Tumour
155	Tumour

Table 1. Set of images

After a set of images are segregated with tumour and no tumour, the model has been pre-processed with

- Train Data
- Test Data

- Validation Data

4. METHODOLOGIES

This section covers the fundamental techniques utilized in constructing the model, including the use of Convolutional Neural Networks (CNN), Keras, and Tensor.

5. DATA ACQUISITION

The data collected had been separated into two categories as healthy and non- healthy ones. Further the model dataset is splitted into the category of train, test and validation model randomly. Further, the images are of different dimensions so they are converted into the same dimensions of 224*224 with RGB specification.

6. SYSTEM SPECIFICATION

The model is run and verified with output in Google Colab

- Keras version 2
- TensorFlow of version 2.x
- CNN model is used
- Python 3.9.16

7. PRE PROCESSING

The dataset consists of images with varying dimensions. To prepare the images for classification, they are randomly split into three subsets: train, test, and validation. The majority of images are assigned to the train set. The train and validation sets are shuffled using TensorFlow, while the test set is not shuffled as it is used to evaluate the performance of the model on unseen data. Using Keras, Convolution Neural Network (CNN) models are trained and optimized for accuracy over multiple epochs. The model is trained until it reaches a high level of accuracy, and then it is ready for object detection and classification.

8. SMOOTHING

The objective is to streamline images by retaining essential information, minimizing unwanted noise or details, and avoiding excessive distortion, all with the intention of facilitating subsequent analysis.

9. FEATURE EXTRACTION

Feature extraction refers to the method of extracting relevant information from an image. In this case, pixel-based feature extraction is employed to classify the image as either tumorous or non-tumorous. Then the size of the image is changed to 224*224 and then it is split to 1/255 for accuracy and deep view of diagnosing the image.

10. CLASSIFICATION

The classification of brain MRI images into tumorous or non-tumorous categories is accomplished through the utilization of Convolutional Neural Networks (CNN). The CNN itself serves as the classifier and is highly effective in analysing image-based datasets. This approach is particularly useful for accurately distinguishing between tumorous and non-tumorous MRIs with accuracy and validation details essentially.

11. PROPOSED WORK

The technique used for making this model work is CNN (Convolution Neural Network). The different stages by which CNN is applied on the dataset are

1. First, the required packages are imported.
2. Then the folder where the dataset is stored is imported.
3. Image reading is done after that it is labelled
(such as 0 non-tumour and 1 for tumour image) and then the images are stored in the data frame.
4. The size of the image is changed to 224*224
5. Image normalization is completed.
6. The data was split into 3 parts –
   - Training

- Validation
- Test

7. The model compilation is completed.
8. Then the model is applied to the training dataset and a validation dataset was used for evaluating the model.
9. Then the accuracy of the model is evaluated using the test images
10. The loss graph and the accuracy graph are plotted.

The entire implementation is done using python 3.9.16 and is executed on Google Collaboratory and the model is stored in Google Drive.

12. SAMPLE CODE

```
from google.colab import drive
drive.mount('/content/drive')
```

```
#Import the necessary libraries first

import tensorflow as tf
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from tensorflow.keras.preprocessing import image
from keras.preprocessing.image  import  ImageDataGenerator
from tensorflow.keras.metrics import categorical_crossentropy
from keras.models import Sequential, Model
from keras.layers import Conv2D, MaxPooling2D,GlobalAveragePooling2D
from keras.layers import Activation, Dropout, BatchNormalization, Flatt
en, Dense, AvgPool2D,MaxPool2D
from keras.models import Sequential, Model
from keras.optimizers import Adam
import cv2
```

```
data = '/content/drive/MyDrive/DATASET OF BRAIN'
No_brain_tumor = '/content/drive/MyDrive/DATASET OF BRAIN/DeepranjanG B
rain_Tumor_Detection main brain_tumor_dataset-no'
Yes_brain_tumor = '/content/drive/MyDrive/DATASET OF BRAIN/DeepranjanG
Brain Tumor Detection main brain tumor dataset-yes'
```

```
dirlist=[No_brain_tumor, Yes_brain_tumor]
classes=['No', 'Yes']
filepaths=[]
labels=[]
for i,j in zip(dirlist, classes):
    filelist=os.listdir(i)
    for f in filelist:
        filepath=os.path.join (i,f)
        filepaths.append(filepath)
        labels.append(j)
print ('filepaths: ', len(filepaths), '   labels: ', len(labels))
```

```
Files=pd.Series(filepaths, name='filepaths')
Label=pd.Series(labels, name='labels')
df=pd.concat([Files,Label], axis=1)
df=pd.DataFrame(np.array(df).reshape(253,2), columns = ['filepaths', 'l
abels'])
df.head()
print(df['labels'].value_counts())
```

```
#visualize brain tumor images

plt.figure(figsize=(12,8))
for i in range(15):
    random = np.random.randint(1,len(df))
    plt.subplot(3,5,i+1)
    plt.imshow(cv2.imread(df.loc[random,"filepaths"]))
    plt.title(df.loc[random, "labels"], size = 15, color = "white")
    plt.xticks([])
    plt.yticks([])

plt.show()
```

```
from sklearn.model_selection import train_test_split

train, test = train_test_split(df, train_size=0.95, random_state=0)
train_new, valid = train_test_split(train, train_size=0.90, random_stat
e=0)

print(f"train set shape: {train_new.shape}")
print(f"test set shape: {test.shape}")
print(f"validation set shape: {valid.shape}")
```

```
train_datagen = ImageDataGenerator(rescale = 1./255.,rotation_range = 4
0, width_shift_range = 0.2, height_shift_range = 0.2,
                                   shear_range = 0.2, zoom_range = 0.2,
 horizontal_flip = True, vertical_flip =True)
test_datagen = ImageDataGenerator(rescale = 1.0/255.)
```

```
train_gen = train_datagen.flow_from_dataframe(dataframe = train_new,
                                              x_col = 'filepaths', y_co
l ='labels',
                                              target_size = (224,224),
batch_size = 32,
                                              class_mode = 'binary', sh
uffle = True)
```

```
val_gen = train_datagen.flow_from_dataframe(valid,
                                            target_size=(224,224), x_co
l = 'filepaths', y_col ='labels',
                                            class_mode='binary',
                                            batch_size= 16, shuffle=Tru
e)
test_gen = test_datagen.flow_from_dataframe(test,
                                            target_size = (224,224), x_
col = 'filepaths', y_col ='labels',
                                             class_mode = 'binary',
                                            batch_size = 16, shuffle =
False)
```

```
train_gen.class_indices
val_gen.class_indices
```

```
test_gen.class_indices

from tensorflow import keras
base_model = keras.applications.ResNet50V2(
    weights="imagenet", # Load weights pre-trained on ImageNet.
    input_shape=(224, 224, 3),
    include_top=False,
)  # Do not include the ImageNet classifier at the top.

# Freeze the base_model base_model.trainable
= False

# Create new model on top
inputs = keras.Input(shape=(224, 224, 3))


# The base model contains batchnorm layers. We want to keep them in inf
erence mode
# when we unfreeze the base model for fine-
tuning, so we make sure that the
# base_model is running in inference mode here.x =
base_model(inputs, training=False)
x = keras.layers.GlobalAveragePooling2D()(x)
x = keras.layers.Dropout(0.2)(x) # Regularize with dropoutoutputs
= keras.layers.Dense(1, activation="sigmoid")(x) model =
```

```
keras.Model(inputs, outputs)

model.summary()
```

```
callbacks = [
    tf.keras.callbacks.ModelCheckpoint("Tumor_classifier_model.h5", sav
e_best_only=True, verbose = 0)
]
model.compile(loss='binary_crossentropy', optimizer=Adam(learning_rate=
 0.0001), metrics=['accuracy'])


history = model.fit(train_gen, validation_data = val_gen, epochs = 100,

                    callbacks = [callbacks], verbose = 1)
```

```
model.save("/content/drive/MyDrive/Colab_Notebook/Brain_Tumor_Detection
/Tumor classifier model.h5")
```

```
#model accuracy
from keras.models import load_model
model=load_model("/content/drive/MyDrive/Colab_Notebook/Brain_Tumor_Det
ection/Tumor_classifier_model.h5")
acc=model.evaluate_generator(train_gen)
print(f"the accuracy of our model is {max(acc*100)} %")
```

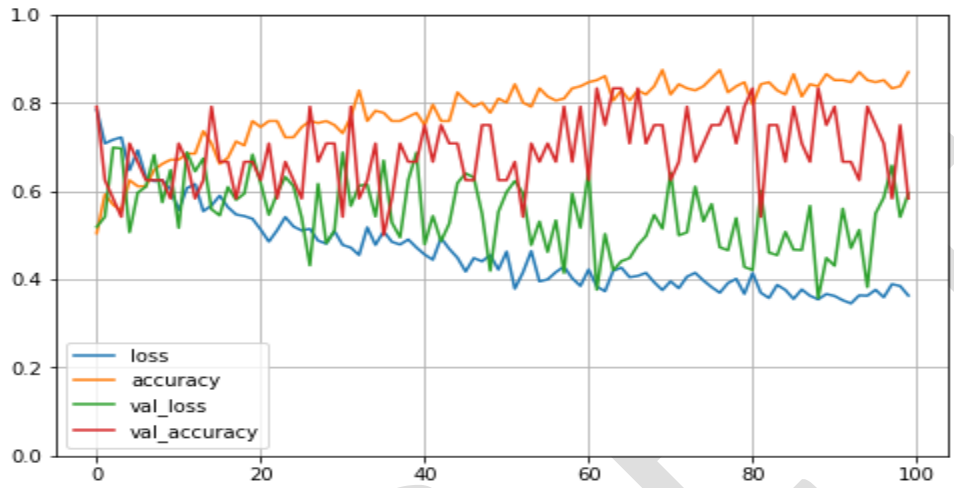
```
pd.DataFrame(history.history).plot(figsize=(8, 5))
plt.grid(True)
plt.gca().set_ylim(0, 1)
plt.show()
```

```
from PIL import Image
import cv2

model_path = "/content/drive/MyDrive/Colab_Notebook/Brain_Tumor_Detecti
on/Tumor_classifier_model.h5"
loaded_model = tf.keras.models.load_model(model_path)

import matplotlib.pyplot as plt
import numpy as np

image = cv2.imread(r"/content/drive/MyDrive/DATASET OF BRAIN/Deepranjan
G Brain_Tumor_Detection main brain_tumor_dataset-no/11 no.jpg")

image_fromarray = Image.fromarray(image, 'RGB')
resize_image = image_fromarray.resize((224,224))
expand_input = np.expand_dims(resize_image,axis=0)
input_data = np.array(expand_input)
```

```
input_data = input_data/255
pred = loaded_model.predict(input_data)
print(pred)
import matplotlib.pyplot as plt
image1= plt.imread("/content/drive/MyDrive/DATASET OF BRAIN/DeepranjanG
 Brain_Tumor_Detection main brain_tumor_dataset-no/11 no.jpg")
plt.imshow(image1)
plt.show()
perc=pred*100
if perc>=50:
  print("Person is having Brain Tumour")
else:
  print("Person is alright with Healthy Brain")
```

13. RESULT

When the model is applied to the testing data set for 100 epochs, an accuracy of 96.3% is obtained and the validation loss is also less**.**



CNN (Convolutional Neural Network) helps to predict just by reducing and resizing the image without losing any important information that will be used for predicting. The loss gradually starts decreasing with the increase in the number of epochs. The model loss is very less when applied to the training set whereas it is high when applied to the validation set.

14. REFERENCES

[1] D. Surya Prabha, J. Satheesh Kumar, Performance evaluation of image segmentation using objective methods, Indian J. Sci. Technol. 9 (8) (February 2016).

[2] Hany Kasban, Mohsen El-bendary, Dina Salama, A comparative study of medi-cal imaging techniques, Int. J. Inf. Sci. Intell. Syst. 4 (2015) 37–58; J. Clerk Maxwell, A Treatise on Electricity and Magnetism, vol. 2, 3rd ed.,Clarendon, Oxford, 1892, pp. 68–73.

[3] Brain Tumour: Statistics, Cancer.Net Editorial Board, 1/2021. (Accessed on January 2021).

[4] H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty and A.-B. M. Salem, "Classification using deep learning neural networks for brain tumors," Future Computing and Informatics Journal, pp. 68-71, 2018.

[5] S. Bauer, C. May, D. Dionysiou, G. Stamatakos, P. Buchler and M. Reyes, "Multiscale modeling for Image Analysis of Brain Tumor Studies," IEEE Transactions on Biomedical Engineering, vol. 59, no. 1, pp. 25-29, 2012.

[6] Islam, S. M. Reza and K. M. Iftekharuddin, "Multifractal texture estimation for detection and segmentation of brain tumors," IEEE Transactions on Biomedical Engineering, pp. 3204-3215, 2013.

[7] Hamamci, N. Kucuk, K. Karaman, K. Engin and G. Unal, "Tumor-cut: Segmentation of brain tumors on contrast enhanced MR images for radiosurgery applications," IEEE Transactions on Medical Imaging, pp. 790-804, 2012.

[8] H. B. Menze, A. Jakab, S. Bauer, K. Farahani and K. Justin, "The Multimodal Brain Tumor Image Segmentation Benchmark," IEEE Transactions on Medical Imaging, pp. 1993-2024, 2014.