**Python code for Machine Learning model to detect brain tumor using Kaggle MRI images datasets**

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

# Input data files are available in the read-only "../input/" directory

# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))

#Importing important packages that will be used for building and running the model.

import keras

from keras.models import Sequential

from keras.layers import Conv2D,Flatten,Dense,MaxPooling2D,Dropout

from sklearn.metrics import accuracy\_score

#Importing important packages that will be used for building and running the model.

import keras

from keras.models import Sequential

from keras.layers import Conv2D,Flatten,Dense,MaxPooling2D,Dropout

from sklearn.metrics import accuracy\_score

X\_train = []

Y\_train = []

image\_size = 150

labels = ['glioma\_tumor','meningioma\_tumor','no\_tumor','pituitary\_tumor']

for i in labels:

folderPath = os.path.join('../input/brain-tumor-classification-mri/Training',i)

for j in os.listdir(folderPath):

img = cv2.imread(os.path.join(folderPath,j))

img = cv2.resize(img,(image\_size,image\_size))

X\_train.append(img)

Y\_train.append(i)

for i in labels:

folderPath = os.path.join('../input/brain-tumor-classification-mri/Testing',i)

for j in os.listdir(folderPath):

img = cv2.imread(os.path.join(folderPath,j))

img = cv2.resize(img,(image\_size,image\_size))

X\_train.append(img)

Y\_train.append(i)

X\_train = np.array(X\_train)

Y\_train = np.array(Y\_train)

X\_train,Y\_train = shuffle(X\_train,Y\_train,random\_state=101)

X\_train.shape

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X\_train,Y\_train,test\_size=0.1,random\_state=101)

y\_train\_new = []

for i in y\_train:

y\_train\_new.append(labels.index(i))

y\_train=y\_train\_new

y\_train = tf.keras.utils.to\_categorical(y\_train)

y\_test\_new = []

for i in y\_test:

y\_test\_new.append(labels.index(i))

y\_test=y\_test\_new

y\_test = tf.keras.utils.to\_categorical(y\_test)

model = Sequential()

model.add(Conv2D(32,(3,3),activation = 'relu',input\_shape=(150,150,3)))

model.add(Conv2D(64,(3,3),activation='relu'))

model.add(MaxPooling2D(2,2))

model.add(Dropout(0.3))

model.add(Conv2D(64,(3,3),activation='relu'))

model.add(Conv2D(64,(3,3),activation='relu'))

model.add(Dropout(0.3))

model.add(MaxPooling2D(2,2))

model.add(Dropout(0.3))

model.add(Conv2D(128,(3,3),activation='relu'))

model.add(Conv2D(128,(3,3),activation='relu'))

model.add(Conv2D(128,(3,3),activation='relu'))

model.add(MaxPooling2D(2,2))

model.add(Dropout(0.3))

model.add(Conv2D(128,(3,3),activation='relu'))

model.add(Conv2D(256,(3,3),activation='relu'))

model.add(MaxPooling2D(2,2))

model.add(Dropout(0.3))

model.add(Flatten())

model.add(Dense(512,activation = 'relu'))

model.add(Dense(512,activation = 'relu'))

model.add(Dropout(0.3))

model.add(Dense(4,activation='softmax'))

model.summary()

model.compile(loss='categorical\_crossentropy',optimizer='Adam',metrics=['accuracy'])

history = model.fit(X\_train,y\_train,epochs=20,validation\_split=0.1)

#model.save('braintumor.h5')

acc = history.history['accuracy']

val\_acc = history.history['val\_accuracy']

epochs = range(len(acc))

fig = plt.figure(figsize=(14,7))

plt.plot(epochs,acc,'r',label="Training Accuracy")

plt.plot(epochs,val\_acc,'b',label="Validation Accuracy")

plt.legend(loc='upper left')

plt.show()

loss = history.history['loss']

val\_loss = history.history['val\_loss']

epochs = range(len(loss))

fig = plt.figure(figsize=(14,7))

plt.plot(epochs,loss,'r',label="Training loss")

plt.plot(epochs,val\_loss,'b',label="Validation loss")

plt.legend(loc='upper left')

plt.show()

img = cv2.imread('../input/brain-tumor-classification-mri/Training/pituitary\_tumor/p (107).jpg')

img = cv2.resize(img,(150,150))

img\_array = np.array(img)

img\_array.shape

img\_array = img\_array.reshape(1,150,150,3)

img\_array.shape

from tensorflow.keras.preprocessing import image

img = image.load\_img('../input/brain-tumor-classification-mri/Training/pituitary\_tumor/p (107).jpg')

plt.imshow(img,interpolation='nearest')

plt.show()