```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator



#Define image size and batch
IMG_SIZE=224
BATCH_SIZE=32


#creating training data
train_datagen=ImageDataGenerator(
    rescale=1./225,
    validation_split=0.2
)


#creating training data with above parameters
#folder=parameters.flow_from_directorty(path,target_size,batch_size,class_mode,subset)
train_generator=train_datagen.flow_from_directory('/content/drive/MyDrive/Brain_Tumor_Detection/Train',
                                                  target_size=(IMG_SIZE,IMG_SIZE),
                                                  batch_size=BATCH_SIZE,
                                                  class_mode='binary',
                                                  subset='training'
)
```

```
Found 2400 images belonging to 2 classes.
```

```
valid_generator=train_datagen.flow_from_directory('/content/drive/MyDrive/Brain_Tumor_Detection/Train',
                                                  target_size=(IMG_SIZE,IMG_SIZE),
                                                  batch_size=BATCH_SIZE,
                                                  class_mode='binary',
                                                  subset='validation'
)
```

```
Found 600 images belonging to 2 classes.
```

```
#Define the  model
import keras
from keras import layers
model=keras.Sequential([
    layers.Conv2D(32,(3,3),activation='relu',input_shape=(IMG_SIZE,IMG_SIZE,3)),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(3,3),activation="relu"),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(128,(3,3),activation="relu"),
    layers.MaxPooling2D((2,2)),
    layers.Flatten(),
    layers.Dense(128,activation='relu'),
    layers.Dense(1,activation="sigmoid")
])


model.compile(optimizer="adam",loss="binary_crossentropy",metrics=(["accuracy"]))



model.fit(train_generator,validation_data=valid_generator,epochs=5)
```

```
Epoch 1/5
75/75 [==============================] - 590s 7s/step - loss: 0.5138 - accuracy: 0.7979 - val_loss: 0.4112 - val_accuracy: 0.8150
Epoch 2/5
75/75 [==============================] - 326s 4s/step - loss: 0.2251 - accuracy: 0.9108 - val_loss: 0.1802 - val_accuracy: 0.9333
Epoch 3/5
75/75 [==============================] - 321s 4s/step - loss: 0.1220 - accuracy: 0.9604 - val_loss: 0.1171 - val_accuracy: 0.9567
Epoch 4/5
75/75 [==============================] - 320s 4s/step - loss: 0.0845 - accuracy: 0.9700 - val_loss: 0.0555 - val_accuracy: 0.9867
Epoch 5/5
75/75 [==============================] - 323s 4s/step - loss: 0.0293 - accuracy: 0.9900 - val_loss: 0.0334 - val_accuracy: 0.9867
<keras.src.callbacks.History at 0x78babce1dde0>
```

```
model.save("Model.h5","label.txt")
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file vi
  saving_api.save_model(
```

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
#Load the saved model
model=load_model('/content/Model.h5')
#Load and preprocess the test image
test_image_path='/content/drive/MyDrive/Brain_Tumor_Detection/Test/pred/pred0.jpg'
img=image.load_img(test_image_path,target_size=(224,224))
img_array=image.img_to_array(img)
img_array=np.expand_dims(img_array,axis=0)
#Add batch dimension
img_array/=225. #Normalize the pixel value
#Make predictions
prediction=model.predict(img_array)
#print the prediction
print(prediction)
```

```
1/1 [==============================] - 0s 105ms/step
[[0.05741625]]
```

```
if prediction<0.5:
  print("Prediction: No Tumor (Probabilty:)",prediction[0][0])
else:
  print("Prediction: Tumor present (Probability:)",prediction[0][0])
```

```
Prediction: No Tumor (Probabilty:) 0.05741625
```