

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

```
IMG_SIZE=224
BATCH_SIZE=32
```

```
train_datagen =
ImageDataGenerator(rescale=1./255,validation_split=0.2)
```

```
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/brain/brain_tumor_dataset',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='training'
)
```

Found 914 images belonging to 1 classes.

```
val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/brain/brain_tumor_dataset',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='validation'
)
```

Found 228 images belonging to 1 classes.

```
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.summary()
```

Model: "sequential"

Layer (type)	Output Shape	

Param #		
conv2d_6 (Conv2D)	(None, 222, 222, 32)	
896		
max_pooling2d_6 (MaxPooling2D)	(None, 111, 111, 32)	
0		
conv2d_7 (Conv2D)	(None, 109, 109, 64)	
18,496		
max_pooling2d_7 (MaxPooling2D)	(None, 54, 54, 64)	
0		
conv2d_8 (Conv2D)	(None, 52, 52, 128)	
73,856		
max_pooling2d_8 (MaxPooling2D)	(None, 26, 26, 128)	
0		
flatten_1 (Flatten)	(None, 86528)	
0		
dense (Dense)	(None, 128)	
11,075,712		
dense_1 (Dense)	(None, 1)	
129		

Total params: 11,169,089 (42.61 MB)

Trainable params: 11,169,089 (42.61 MB)

Non-trainable params: 0 (0.00 B)

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

```
model.fit(train_generator,epochs=5,validation_data=val_generator,batch_size=BATCH_SIZE)
```

Epoch 1/5

```
29/29 _____ 418s 15s/step - accuracy: 1.0000 - loss: 2.7862e-33 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
```

Epoch 2/5

```
29/29 _____ 6s 225ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
```

Epoch 3/5

```
29/29 _____ 6s 197ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
```

Epoch 4/5

```
29/29 _____ 8s 259ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
```

Epoch 5/5

```
29/29 _____ 6s 198ms/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
```

```
<keras.src.callbacks.history.History at 0x7bdb500f07a0>
```

```
model.save('/content/drive/MyDrive/brain/jo.h5')
```

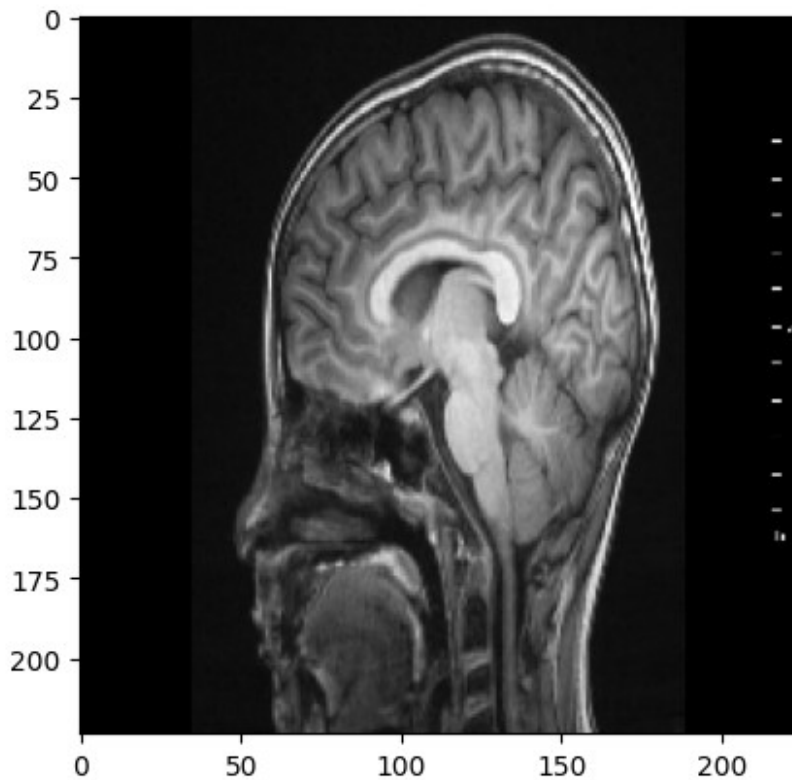
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np
model=load_model('/content/drive/MyDrive/brain/jo.h5')
print("Model Loaded")
```

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.

Model Loaded

```
test_image_path="/content/drive/MyDrive/test brain tumour.webp"
img=image.load_img(test_image_path,target_size=(224,224))
plt.imshow(img)
plt.axis()
plt.show()
```



```
img_array=image.img_to_array(img)
img_array=np.expand_dims(img_array,axis=0)
img_array /= 255
```

```
prediction=model.predict(img_array)
print(prediction)
```

```
1/1 ————— 0s 32ms/step
[[0.]]
```

```
if prediction >=0.5:
    print("You have brain tumour")
else:
    print("You do not have brain tumour")
```

```
You do not have brain tumour
```