

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

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

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

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

```
train_datagen=ImageDataGenerator(rescale=1.225,validation_split=0.2)
```

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

Found 203 images belonging to 2 classes.

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

Found 50 images belonging to 2 classes.

```
model=keras.Sequential([
    layers.Conv2D(32,kernel_size=(3,3),activation='relu',input_shape=(IMG_SIZE,IMG_SIZE,3)),
    layers.MaxPooling2D(pool_size=(2,2)),
    layers.Conv2D(64,kernel_size=(3,3),activation='relu'),
    layers.MaxPooling2D(pool_size=(2,2)),
    layers.Conv2D(128,kernel_size=(3,3),activation='relu'),
    layers.MaxPooling2D(pool_size=(2,2)),
    layers.Flatten(),
    layers.Dense(128,activation='relu'),
    layers.Dense(1,activation='sigmoid')
])
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
```

```
    super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
model.summary()
```

```
Model: "sequential"
```

Layer (type) Param #	Output Shape
conv2d (Conv2D) 896	(None, 222, 222, 32)
max_pooling2d (MaxPooling2D) 0	(None, 111, 111, 32)
conv2d_1 (Conv2D) 18,496	(None, 109, 109, 64)
max_pooling2d_1 (MaxPooling2D) 0	(None, 54, 54, 64)
conv2d_2 (Conv2D) 73,856	(None, 52, 52, 128)
max_pooling2d_2 (MaxPooling2D) 0	(None, 26, 26, 128)
flatten (Flatten) 0	(None, 86528)
dense (Dense) 11,075,712	(None, 128)

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=3,validation_data=val_generator,batch_size=BATCH_SIZE)
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/
data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset`
class should call `super().__init__(**kwargs)` in its constructor.
`**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will
be ignored.
self._warn_if_super_not_called()
```

Epoch 1/3

```
7/7 ----- 0s 5s/step - accuracy: 0.5301 - loss:
270.5665
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/
data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset`
class should call `super().__init__(**kwargs)` in its constructor.
`**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will
be ignored.
self._warn_if_super_not_called()
```

```
7/7 ----- 56s 7s/step - accuracy: 0.5334 - loss:
261.0916 - val_accuracy: 0.4800 - val_loss: 11.1933
```

Epoch 2/3

```
7/7 ----- 28s 4s/step - accuracy: 0.7090 - loss: 4.7055
- val_accuracy: 0.6200 - val_loss: 2.2123
```

Epoch 3/3

```
7/7 ----- 24s 3s/step - accuracy: 0.6766 - loss: 1.2096
- val_accuracy: 0.6200 - val_loss: 0.7159
```

```
<keras.src.callbacks.history.History at 0x7c501aef6c90>
```

```
model.save('/content/drive/MyDrive/brain_tumor_dataset/BrainTumor.h5')
```

```
WARNING:absl:You are saving your model as an HDF5 file via
`model.save()` or `keras.saving.save_model(model)`. This file format
```

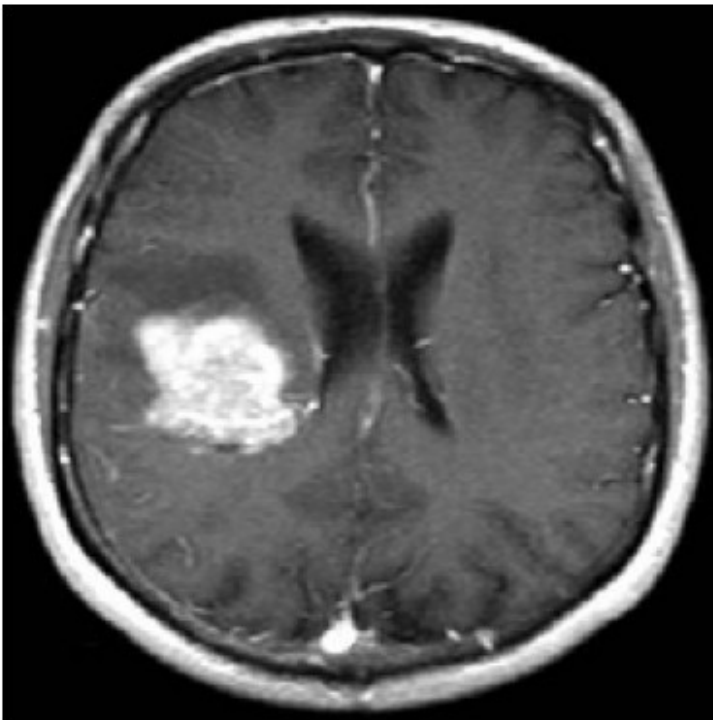
is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.

```
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_tumor_dataset/BrainTumor.h5')
print("Model Loaded Successfully")
```

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 Successfully

```
test_image_path='/content/drive/MyDrive/brain_tumor_dataset/yes/Y11.jpg'
img=image.load_img(test_image_path,target_size=(224,224))
plt.imshow(img)
plt.axis('off')
plt.show()
```



```
immg_array=image.img_to_array(img)
immg_array=np.expand_dims(immg_array,axis=0)
immg_array=immg_array/255.0
```

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

```
1/1 _____ 0s 167ms/step
[[0.50102556]]
```

```
if prediction>0.5:
    print("Tumor Detected")
else:
    print("No Tumor Detected")
```

Tumor Detected