```
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./255,validation_split=0.2)

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

```
Found 1486 images belonging to 1 classes.
```

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

```
Found 371 images belonging to 1 classes.
```

```
model = keras.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(254,254,
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_3"
```

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ conv2d_9 (Conv2D)               │ (None, 222, 222, 32)   │           896 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ max_pooling2d_9 (MaxPooling2D)  │ (None, 111, 111, 32)   │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_10 (Conv2D)              │ (None, 109, 109, 64)   │        18,496 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ max_pooling2d_10 (MaxPooling2D) │ (None, 54, 54, 64)     │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_11 (Conv2D)              │ (None, 52, 52, 128)    │        73,856 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ max_pooling2d_11 (MaxPooling2D) │ (None, 26, 26, 128)    │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ flatten_3 (Flatten)             │ (None, 86528)          │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_6 (Dense)                 │ (None, 128)            │    11,075,712 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_7 (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=train_generator,
batch_size=BATCH_SIZE)
```

```
Epoch 1/5
12/12 ━━━━━━━━━━━━━━━━━━━━ 78s 6s/step - accuracy: 0.7849 - loss:
0.1759 - val_accuracy: 1.0000 - val_loss: 6.8437e-33
Epoch 2/5
12/12 ━━━━━━━━━━━━━━━━━━━━ 50s 4s/step - accuracy: 1.0000 - loss:
0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 3/5
12/12 ━━━━━━━━━━━━━━━━━━━━ 51s 4s/step - accuracy: 1.0000 - loss:
0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 4/5
12/12 ━━━━━━━━━━━━━━━━━━━━ 49s 4s/step - accuracy: 1.0000 - loss:
0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 5/5
12/12 ━━━━━━━━━━━━━━━━━━━━ 60s 5s/step - accuracy: 1.0000 - loss:
0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00

<keras.src.callbacks.history.History at 0x79f4aecafa90>
```

```
model.save('/MyDrive/Brain_Tumour_Dataset.h5')
```

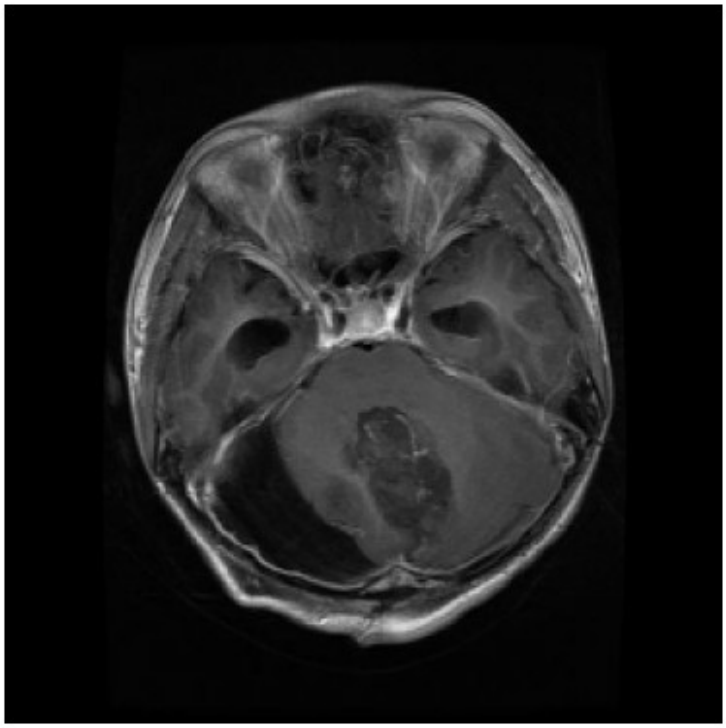
```
WARNING:absl:You are saving your model as an HDF5 file via
`model.save()` or `keras.saving.save_model(model)`. This file format
is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')` or
`keras.saving.save_model(model, 'my_model.keras')`.
```

```
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('/MyDrive/Brain_Tumour_Dataset.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/brain_tumour_dataset/
Brain_Tumor_Dataset/Positive/Te-glTr_0000.jpg"
img = image.load_img(test_image_path, target_size=(254,254))
plt.imshow(img)
plt.axis('off')
plt.show()
```



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

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

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 228ms/step
[[0.5048334]]
```

```
if prediction >=0.5:
  print(" you have brain Tumour Detected")
else:
  print(" you do not have brain Tumour Detected")
```

```
 you have brain Tumour Detected
```