

Brain Tumor

Mount Google Drive & Setup Paths

In [1]: `import kagglehub`

```
# Download latest version
path = kagglehub.dataset_download("ishans24/brain-tumor-dataset")

print("Path to dataset files:", path)
```

Using Colab cache for faster access to the 'brain-tumor-dataset' dataset.
Path to dataset files: /kaggle/input/brain-tumor-dataset

In [2]: `import os`

```
print(f"Contents of the downloaded dataset directory ({path}):")
for item in os.listdir(path):
    print(f"- {item}")
```

Contents of the downloaded dataset directory (/kaggle/input/brain-tumor-dataset):

- pituitary
- no_tumor
- meningioma
- glioma

Split Dataset (Train / Val / Test)

In [3]: `import os`

```
import shutil
import random
```

```
# Use the path from the kagglehub download directly as the input path
# The dataset structure has class folders directly under the main downloaded path
input_path = path
output_path = "/content/splitted_data"

train_ratio = 0.7
val_ratio = 0.15
test_ratio = 0.15

# Make sure output folders exist
os.makedirs(output_path, exist_ok=True)

classes = os.listdir(input_path)

for cls in classes:
    cls_path = os.path.join(input_path, cls)

    if not os.path.isdir(cls_path):
        continue

    images = [img for img in os.listdir(cls_path) if img.lower().endswith(('.jpg', '.png', '.jpeg'))]
    random.shuffle(images)

    train_end = int(len(images) * train_ratio)
    val_end = train_end + int(len(images) * val_ratio)

    splits = {
```

```

        "train": images[:train_end],
        "val": images[train_end:val_end],
        "test": images[val_end:]
    }

    for split_name, files in splits.items():
        split_dir = os.path.join(output_path, split_name, cls)
        os.makedirs(split_dir, exist_ok=True)

        for img in files:
            src = os.path.join(cls_path, img)
            dst = os.path.join(split_dir, img)
            shutil.copy(src, dst)

print("✅ Dataset splitting completed!")

```

✅ Dataset splitting completed!

Apply Data Augmentation

In [4]: `from tensorflow.keras.preprocessing.image import ImageDataGenerator`

```

train_dir = "/content/splitted_data/train"
val_dir = "/content/splitted_data/val"
test_dir = "/content/splitted_data/test"

# ⚠️ Apply augmentation ONLY to training data
train_datagen = ImageDataGenerator(
    rescale=1/255,
    rotation_range=20,
    zoom_range=0.2,
    shear_range=0.2,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True
)

# Validation and Test should NOT be augmented
test_val_datagen = ImageDataGenerator(rescale=1/255)

img_size = (256, 256)
batch_size = 32

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical'
)

val_generator = test_val_datagen.flow_from_directory(
    val_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical'
)

test_generator = test_val_datagen.flow_from_directory(
    test_dir,
    target_size=img_size,

```

```

    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)

```

Found 7390 images belonging to 4 classes.

Found 1582 images belonging to 4 classes.

Found 1588 images belonging to 4 classes.

Train & Test a CNN Model

```

In [ ]: import tensorflow as tf
        from tensorflow.keras import layers, models

model = models.Sequential([
    layers.Conv2D(32, (3,3), activation='relu', input_shape=(256, 256, 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.Dropout(0.3),

    layers.Dense(4, activation='softmax') # 4 classes
])

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

model.summary()

```

/usr/local/lib/python3.12/dist-packages/keras/src/layers/convolutional/base_conv.py:1
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: "sequential"

Layer (type)	Output Shape	Param
conv2d (Conv2D)	(None, 254, 254, 32)	89
max_pooling2d (MaxPooling2D)	(None, 127, 127, 32)	
conv2d_1 (Conv2D)	(None, 125, 125, 64)	18,49
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	
conv2d_2 (Conv2D)	(None, 60, 60, 128)	73,85
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 128)	
flatten (Flatten)	(None, 115200)	
dense (Dense)	(None, 128)	14,745,72
dropout (Dropout)	(None, 128)	
dense_1 (Dense)	(None, 4)	51

Total params: 14,839,492 (56.61 MB)

Trainable params: 14,839,492 (56.61 MB)

Non-trainable params: 0 (0.00 B)

Train the Model

```
In [ ]: history = model.fit(
    train_generator,
    epochs=15,
    validation_data=val_generator
)
```

```
/usr/local/lib/python3.12/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 t
will be ignored.
  self._warn_if_super_not_called()
```

Epoch 1/15
 231/231 
 729s 3s/step - accuracy: 0.4668 - loss: 1.2917 - val_accuracy: 0.6947 - val_loss: 0.7
 Epoch 2/15
 231/231 
 774s 3s/step - accuracy: 0.6587 - loss: 0.8161 - val_accuracy: 0.7263 - val_loss: 0.6
 Epoch 3/15
 231/231 
 732s 3s/step - accuracy: 0.6916 - loss: 0.7513 - val_accuracy: 0.7566 - val_loss: 0.6
 Epoch 4/15
 231/231 
 738s 3s/step - accuracy: 0.7172 - loss: 0.6823 - val_accuracy: 0.7731 - val_loss: 0.5
 Epoch 5/15
 231/231 
 730s 3s/step - accuracy: 0.7503 - loss: 0.6271 - val_accuracy: 0.7276 - val_loss: 0.6
 Epoch 6/15
 231/231 
 723s 3s/step - accuracy: 0.7372 - loss: 0.6432 - val_accuracy: 0.7560 - val_loss: 0.5
 Epoch 7/15
 231/231 
 724s 3s/step - accuracy: 0.7663 - loss: 0.5744 - val_accuracy: 0.8198 - val_loss: 0.4
 Epoch 8/15
 231/231 
 722s 3s/step - accuracy: 0.7703 - loss: 0.5469 - val_accuracy: 0.8249 - val_loss: 0.4
 Epoch 9/15
 231/231 
 725s 3s/step - accuracy: 0.7995 - loss: 0.5001 - val_accuracy: 0.8375 - val_loss: 0.3
 Epoch 10/15
 231/231 
 733s 3s/step - accuracy: 0.8149 - loss: 0.4609 - val_accuracy: 0.8344 - val_loss: 0.3
 Epoch 11/15
 231/231 
 743s 3s/step - accuracy: 0.8110 - loss: 0.4656 - val_accuracy: 0.8464 - val_loss: 0.3
 Epoch 12/15
 231/231 
 740s 3s/step - accuracy: 0.8393 - loss: 0.4264 - val_accuracy: 0.8502 - val_loss: 0.3
 Epoch 13/15
 231/231 
 738s 3s/step - accuracy: 0.8341 - loss: 0.4192 - val_accuracy: 0.8458 - val_loss: 0.4
 Epoch 14/15
 231/231 
 744s 3s/step - accuracy: 0.8473 - loss: 0.4010 - val_accuracy: 0.8034 - val_loss: 0.4
 Epoch 15/15
 231/231 
 805s 3s/step - accuracy: 0.8499 - loss: 0.3755 - val_accuracy: 0.8628 - val_loss: 0.3

Evaluate on Test Set

```
In [ ]: import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

# --- 1. Evaluate Model on Test Set ---
test_loss, test_acc = model.evaluate(test_generator, verbose=1)
print("💎 Test Accuracy:", test_acc)
print("💎 Test Loss:", test_loss)

# --- 2. Predictions ---
y_pred = model.predict(test_generator)
```

```

y_pred_classes = np.argmax(y_pred, axis=1)

# True Labels
y_true = test_generator.classes

# Class names
class_labels = list(test_generator.class_indices.keys())
print("Classes:", class_labels)

# --- 3. Classification Report ---
print("\n🔗 Classification Report:")
print(classification_report(y_true, y_pred_classes, target_names=class_labels))

# --- 4. Confusion Matrix ---
cm = confusion_matrix(y_true, y_pred_classes)

plt.figure(figsize=(7,5))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=class_labels,
            yticklabels=class_labels)
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()

```

50/50 ————— 43s 841ms/step - accuracy: 0.8290 - loss: 0.3871

◆ Test Accuracy: 0.8602015376091003

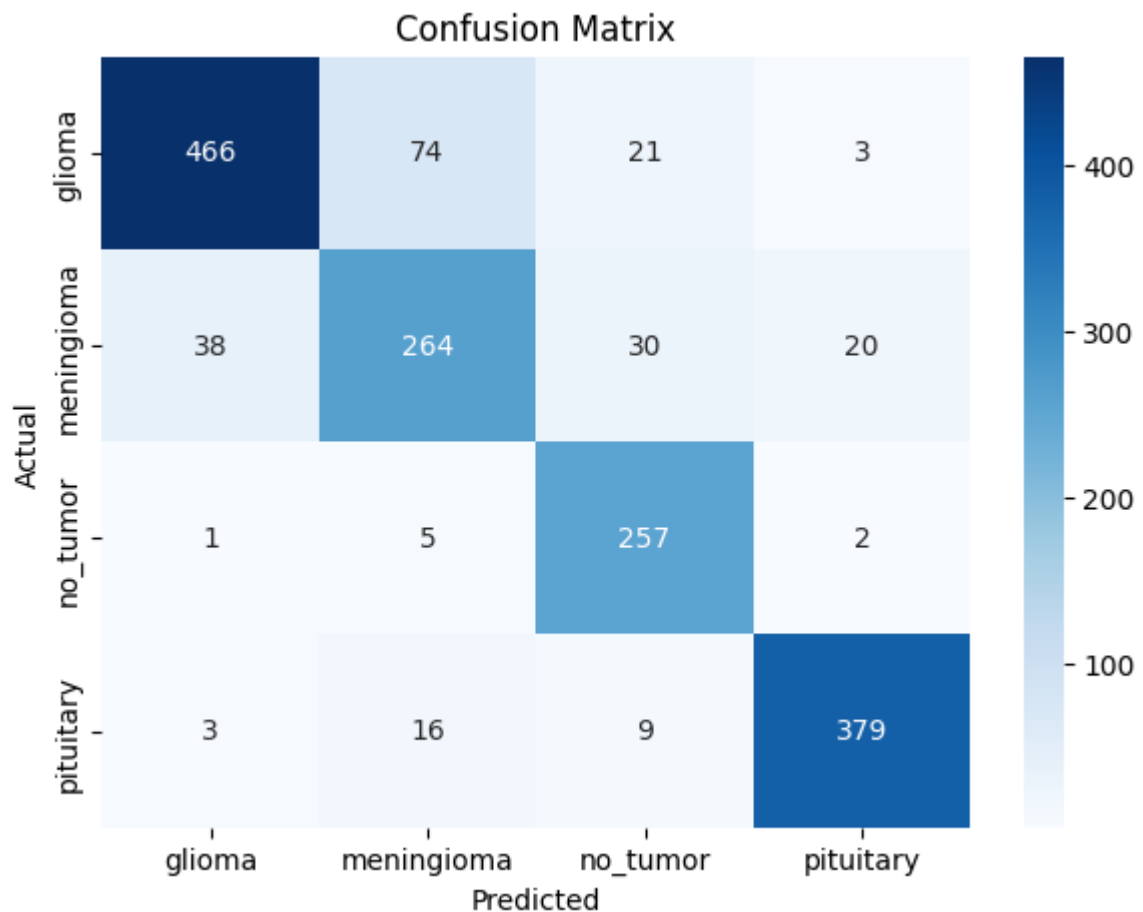
◆ Test Loss: 0.3655005395412445

50/50 ————— 44s 876ms/step

Classes: ['glioma', 'meningioma', 'no_tumor', 'pituitary']

🔗 Classification Report:

	precision	recall	f1-score	support
glioma	0.92	0.83	0.87	564
meningioma	0.74	0.75	0.74	352
no_tumor	0.81	0.97	0.88	265
pituitary	0.94	0.93	0.93	407
accuracy			0.86	1588
macro avg	0.85	0.87	0.86	1588
weighted avg	0.86	0.86	0.86	1588



```
In [7]: import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras.models import Model
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# =====
# DATA PATHS
# =====
train_dir = "/content/splitted_data/train"
test_dir = "/content/splitted_data/test"

# =====
# IMAGE DATA GENERATOR
# =====
train_gen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=15,
    zoom_range=0.15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True
)

test_gen = ImageDataGenerator(rescale=1./255)

train_data = train_gen.flow_from_directory(
```

```

        train_dir,
        target_size=(224, 224),
        batch_size=32,
        class_mode="categorical"
    )

test_data = test_gen.flow_from_directory(
    test_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode="categorical",
    shuffle=False
)

# =====
# RESNET50 MODEL
# =====
base_model = ResNet50(
    weights="imagenet",
    include_top=False,
    input_shape=(224, 224, 3)
)

base_model.trainable = False # Freeze weights for FAST training

x = GlobalAveragePooling2D()(base_model.output)
x = Dense(256, activation="relu")(x)
x = Dropout(0.3)(x)
output = Dense(4, activation="softmax")(x)

model = Model(inputs=base_model.input, outputs=output)

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

model.summary()

# =====
# TRAIN MODEL
# =====
history = model.fit(
    train_data,
    epochs=15, # Small number to stay FAST
    validation_data=test_data
)

# =====
# EVALUATE MODEL ON TEST SET
# =====
test_loss, test_acc = model.evaluate(test_data)
print("\n📊 Test Accuracy:", test_acc)
print("📉 Test Loss:", test_loss)

# =====
# CONFUSION MATRIX + CLASSIFICATION REPORT
# =====
pred_probs = model.predict(test_data)

```

```

pred_classes = np.argmax(pred_probs, axis=1)
true_classes = test_data.classes
class_labels = list(test_data.class_indices.keys())

cm = confusion_matrix(true_classes, pred_classes)

plt.figure(figsize=(7, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=class_labels, yticklabels=class_labels)
plt.title("Confusion Matrix - ResNet50")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

print("\n📄 Classification Report:\n")
print(classification_report(true_classes, pred_classes, target_names=class_labels))

```

Found 7390 images belonging to 4 classes.

Found 1588 images belonging to 4 classes.

Model: "functional_1"

Layer (type)	Output Shape	Param #	Connected to
input_layer_1 (InputLayer)	(None , 224, 224, 3)	0	-
conv1_pad (ZeroPadding2D)	(None , 230, 230, 3)	0	input_layer_1[0]
conv1_conv (Conv2D)	(None , 112, 112, 64)	9,472	conv1_pad[0][0]
conv1_bn (BatchNormalizatio...)	(None , 112, 112, 64)	256	conv1_conv[0][0]
conv1_relu (Activation)	(None , 112, 112, 64)	0	conv1_bn[0][0]
pool1_pad (ZeroPadding2D)	(None , 114, 114, 64)	0	conv1_relu[0][0]
pool1_pool (MaxPooling2D)	(None , 56, 56, 64)	0	pool1_pad[0][0]
conv2_block1_1_conv (Conv2D)	(None , 56, 56, 64)	4,160	pool1_pool[0][0]
conv2_block1_1_bn (BatchNormalizatio...)	(None , 56, 56, 64)	256	conv2_block1_1_c
conv2_block1_1_relu (Activation)	(None , 56, 56, 64)	0	conv2_block1_1_t
conv2_block1_2_conv (Conv2D)	(None , 56, 56, 64)	36,928	conv2_block1_1_r
conv2_block1_2_bn (BatchNormalizatio...)	(None , 56, 56, 64)	256	conv2_block1_2_c
conv2_block1_2_relu (Activation)	(None , 56, 56, 64)	0	conv2_block1_2_t
conv2_block1_0_conv (Conv2D)	(None , 56, 56, 256)	16,640	pool1_pool[0][0]
conv2_block1_3_conv (Conv2D)	(None , 56, 56, 256)	16,640	conv2_block1_2_r
conv2_block1_0_bn (BatchNormalizatio...)	(None , 56, 56, 256)	1,024	conv2_block1_0_c
conv2_block1_3_bn (BatchNormalizatio...)	(None , 56, 56, 256)	1,024	conv2_block1_3_c
conv2_block1_add (Add)	(None , 56, 56, 256)	0	conv2_block1_0_t conv2_block1_3_t
conv2_block1_out (Activation)	(None , 56, 56, 256)	0	conv2_block1_adc
conv2_block2_1_conv (Conv2D)	(None , 56, 56, 64)	16,448	conv2_block1_out

conv2_block2_1_bn (BatchNormalizatio...	(None, 56, 56, 64)	256	conv2_block2_1_c
conv2_block2_1_relu (Activation)	(None, 56, 56, 64)	0	conv2_block2_1_t
conv2_block2_2_conv (Conv2D)	(None, 56, 56, 64)	36,928	conv2_block2_1_r
conv2_block2_2_bn (BatchNormalizatio...	(None, 56, 56, 64)	256	conv2_block2_2_c
conv2_block2_2_relu (Activation)	(None, 56, 56, 64)	0	conv2_block2_2_t
conv2_block2_3_conv (Conv2D)	(None, 56, 56, 256)	16,640	conv2_block2_2_r
conv2_block2_3_bn (BatchNormalizatio...	(None, 56, 56, 256)	1,024	conv2_block2_3_c
conv2_block2_add (Add)	(None, 56, 56, 256)	0	conv2_block1_out conv2_block2_3_t
conv2_block2_out (Activation)	(None, 56, 56, 256)	0	conv2_block2_adc
conv2_block3_1_conv (Conv2D)	(None, 56, 56, 64)	16,448	conv2_block2_out
conv2_block3_1_bn (BatchNormalizatio...	(None, 56, 56, 64)	256	conv2_block3_1_c
conv2_block3_1_relu (Activation)	(None, 56, 56, 64)	0	conv2_block3_1_t
conv2_block3_2_conv (Conv2D)	(None, 56, 56, 64)	36,928	conv2_block3_1_r
conv2_block3_2_bn (BatchNormalizatio...	(None, 56, 56, 64)	256	conv2_block3_2_c
conv2_block3_2_relu (Activation)	(None, 56, 56, 64)	0	conv2_block3_2_t
conv2_block3_3_conv (Conv2D)	(None, 56, 56, 256)	16,640	conv2_block3_2_r
conv2_block3_3_bn (BatchNormalizatio...	(None, 56, 56, 256)	1,024	conv2_block3_3_c
conv2_block3_add (Add)	(None, 56, 56, 256)	0	conv2_block2_out conv2_block3_3_t
conv2_block3_out (Activation)	(None, 56, 56, 256)	0	conv2_block3_adc
conv3_block1_1_conv (Conv2D)	(None, 28, 28, 128)	32,896	conv2_block3_out
conv3_block1_1_bn	(None, 28, 28,	512	conv3_block1_1_c

(BatchNormalizatio...	128)		
conv3_block1_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block1_1_t
conv3_block1_2_conv (Conv2D)	(None, 28, 28, 128)	147,584	conv3_block1_1_r
conv3_block1_2_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block1_2_c
conv3_block1_2_relu (Activation)	(None, 28, 28, 128)	0	conv3_block1_2_t
conv3_block1_0_conv (Conv2D)	(None, 28, 28, 512)	131,584	conv2_block3_out
conv3_block1_3_conv (Conv2D)	(None, 28, 28, 512)	66,048	conv3_block1_2_r
conv3_block1_0_bn (BatchNormalizatio...	(None, 28, 28, 512)	2,048	conv3_block1_0_c
conv3_block1_3_bn (BatchNormalizatio...	(None, 28, 28, 512)	2,048	conv3_block1_3_c
conv3_block1_add (Add)	(None, 28, 28, 512)	0	conv3_block1_0_t conv3_block1_3_t
conv3_block1_out (Activation)	(None, 28, 28, 512)	0	conv3_block1_adc
conv3_block2_1_conv (Conv2D)	(None, 28, 28, 128)	65,664	conv3_block1_out
conv3_block2_1_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block2_1_c
conv3_block2_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block2_1_t
conv3_block2_2_conv (Conv2D)	(None, 28, 28, 128)	147,584	conv3_block2_1_r
conv3_block2_2_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block2_2_c
conv3_block2_2_relu (Activation)	(None, 28, 28, 128)	0	conv3_block2_2_t
conv3_block2_3_conv (Conv2D)	(None, 28, 28, 512)	66,048	conv3_block2_2_r
conv3_block2_3_bn (BatchNormalizatio...	(None, 28, 28, 512)	2,048	conv3_block2_3_c
conv3_block2_add (Add)	(None, 28, 28, 512)	0	conv3_block1_out conv3_block2_3_t
conv3_block2_out (Activation)	(None, 28, 28, 512)	0	conv3_block2_adc

conv3_block3_1_conv (Conv2D)	(None, 28, 28, 128)	65,664	conv3_block2_out
conv3_block3_1_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block3_1_c
conv3_block3_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block3_1_t
conv3_block3_2_conv (Conv2D)	(None, 28, 28, 128)	147,584	conv3_block3_1_r
conv3_block3_2_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block3_2_c
conv3_block3_2_relu (Activation)	(None, 28, 28, 128)	0	conv3_block3_2_t
conv3_block3_3_conv (Conv2D)	(None, 28, 28, 512)	66,048	conv3_block3_2_r
conv3_block3_3_bn (BatchNormalizatio...	(None, 28, 28, 512)	2,048	conv3_block3_3_c
conv3_block3_add (Add)	(None, 28, 28, 512)	0	conv3_block2_out conv3_block3_3_t
conv3_block3_out (Activation)	(None, 28, 28, 512)	0	conv3_block3_adc
conv3_block4_1_conv (Conv2D)	(None, 28, 28, 128)	65,664	conv3_block3_out
conv3_block4_1_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block4_1_c
conv3_block4_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block4_1_t
conv3_block4_2_conv (Conv2D)	(None, 28, 28, 128)	147,584	conv3_block4_1_r
conv3_block4_2_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block4_2_c
conv3_block4_2_relu (Activation)	(None, 28, 28, 128)	0	conv3_block4_2_t
conv3_block4_3_conv (Conv2D)	(None, 28, 28, 512)	66,048	conv3_block4_2_r
conv3_block4_3_bn (BatchNormalizatio...	(None, 28, 28, 512)	2,048	conv3_block4_3_c
conv3_block4_add (Add)	(None, 28, 28, 512)	0	conv3_block3_out conv3_block4_3_t
conv3_block4_out (Activation)	(None, 28, 28, 512)	0	conv3_block4_adc
conv4_block1_1_conv (Conv2D)	(None, 14, 14, 256)	131,328	conv3_block4_out

conv4_block1_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block1_1_c
conv4_block1_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block1_1_t
conv4_block1_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block1_1_r
conv4_block1_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block1_2_c
conv4_block1_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block1_2_t
conv4_block1_0_conv (Conv2D)	(None, 14, 14, 1024)	525,312	conv3_block4_out
conv4_block1_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block1_2_r
conv4_block1_0_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block1_0_c
conv4_block1_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block1_3_c
conv4_block1_add (Add)	(None, 14, 14, 1024)	0	conv4_block1_0_t conv4_block1_3_t
conv4_block1_out (Activation)	(None, 14, 14, 1024)	0	conv4_block1_adc
conv4_block2_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block1_out
conv4_block2_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block2_1_c
conv4_block2_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block2_1_t
conv4_block2_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block2_1_r
conv4_block2_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block2_2_c
conv4_block2_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block2_2_t
conv4_block2_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block2_2_r
conv4_block2_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block2_3_c
conv4_block2_add (Add)	(None, 14, 14, 1024)	0	conv4_block1_out conv4_block2_3_t
conv4_block2_out	(None, 14, 14, 1024)	0	conv4_block2_adc

(Activation)	1024)		
conv4_block3_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block2_out
conv4_block3_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block3_1_c
conv4_block3_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block3_1_t
conv4_block3_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block3_1_r
conv4_block3_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block3_2_c
conv4_block3_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block3_2_t
conv4_block3_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block3_2_r
conv4_block3_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block3_3_c
conv4_block3_add (Add)	(None, 14, 14, 1024)	0	conv4_block2_out conv4_block3_3_t
conv4_block3_out (Activation)	(None, 14, 14, 1024)	0	conv4_block3_adc
conv4_block4_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block3_out
conv4_block4_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block4_1_c
conv4_block4_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block4_1_t
conv4_block4_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block4_1_r
conv4_block4_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block4_2_c
conv4_block4_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block4_2_t
conv4_block4_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block4_2_r
conv4_block4_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block4_3_c
conv4_block4_add (Add)	(None, 14, 14, 1024)	0	conv4_block3_out conv4_block4_3_t
conv4_block4_out (Activation)	(None, 14, 14, 1024)	0	conv4_block4_adc

conv4_block5_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block4_out
conv4_block5_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block5_1_c
conv4_block5_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block5_1_t
conv4_block5_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block5_1_r
conv4_block5_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block5_2_c
conv4_block5_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block5_2_t
conv4_block5_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block5_2_r
conv4_block5_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block5_3_c
conv4_block5_add (Add)	(None, 14, 14, 1024)	0	conv4_block4_out conv4_block5_3_t
conv4_block5_out (Activation)	(None, 14, 14, 1024)	0	conv4_block5_adc
conv4_block6_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block5_out
conv4_block6_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block6_1_c
conv4_block6_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block6_1_t
conv4_block6_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block6_1_r
conv4_block6_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block6_2_c
conv4_block6_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block6_2_t
conv4_block6_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block6_2_r
conv4_block6_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block6_3_c
conv4_block6_add (Add)	(None, 14, 14, 1024)	0	conv4_block5_out conv4_block6_3_t
conv4_block6_out (Activation)	(None, 14, 14, 1024)	0	conv4_block6_adc
conv5_block1_1_conv (Conv2D)	(None, 7, 7, 512)	524,800	conv4_block6_out

conv5_block1_1_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block1_1_c
conv5_block1_1_relu (Activation)	(None, 7, 7, 512)	0	conv5_block1_1_t
conv5_block1_2_conv (Conv2D)	(None, 7, 7, 512)	2,359,808	conv5_block1_1_r
conv5_block1_2_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block1_2_c
conv5_block1_2_relu (Activation)	(None, 7, 7, 512)	0	conv5_block1_2_t
conv5_block1_0_conv (Conv2D)	(None, 7, 7, 2048)	2,099,200	conv4_block6_out
conv5_block1_3_conv (Conv2D)	(None, 7, 7, 2048)	1,050,624	conv5_block1_2_r
conv5_block1_0_bn (BatchNormalizatio...	(None, 7, 7, 2048)	8,192	conv5_block1_0_c
conv5_block1_3_bn (BatchNormalizatio...	(None, 7, 7, 2048)	8,192	conv5_block1_3_c
conv5_block1_add (Add)	(None, 7, 7, 2048)	0	conv5_block1_0_t conv5_block1_3_t
conv5_block1_out (Activation)	(None, 7, 7, 2048)	0	conv5_block1_adc
conv5_block2_1_conv (Conv2D)	(None, 7, 7, 512)	1,049,088	conv5_block1_out
conv5_block2_1_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block2_1_c
conv5_block2_1_relu (Activation)	(None, 7, 7, 512)	0	conv5_block2_1_t
conv5_block2_2_conv (Conv2D)	(None, 7, 7, 512)	2,359,808	conv5_block2_1_r
conv5_block2_2_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block2_2_c
conv5_block2_2_relu (Activation)	(None, 7, 7, 512)	0	conv5_block2_2_t
conv5_block2_3_conv (Conv2D)	(None, 7, 7, 2048)	1,050,624	conv5_block2_2_r
conv5_block2_3_bn (BatchNormalizatio...	(None, 7, 7, 2048)	8,192	conv5_block2_3_c
conv5_block2_add (Add)	(None, 7, 7, 2048)	0	conv5_block1_out conv5_block2_3_t
conv5_block2_out	(None, 7, 7,	0	conv5_block2_adc

(Activation)	2048)		
conv5_block3_1_conv (Conv2D)	(None, 7, 7, 512)	1,049,088	conv5_block2_out
conv5_block3_1_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block3_1_c
conv5_block3_1_relu (Activation)	(None, 7, 7, 512)	0	conv5_block3_1_t
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 512)	2,359,808	conv5_block3_1_r
conv5_block3_2_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block3_2_c
conv5_block3_2_relu (Activation)	(None, 7, 7, 512)	0	conv5_block3_2_t
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1,050,624	conv5_block3_2_r
conv5_block3_3_bn (BatchNormalizatio...	(None, 7, 7, 2048)	8,192	conv5_block3_3_c
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	conv5_block2_out conv5_block3_3_t
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	conv5_block3_adc
global_average_poo... (GlobalAveragePool...	(None, 2048)	0	conv5_block3_out
dense_2 (Dense)	(None, 256)	524,544	global_average_p
dropout_1 (Dropout)	(None, 256)	0	dense_2[0][0]
dense_3 (Dense)	(None, 4)	1,028	dropout_1[0][0]

Total params: 24,113,284 (91.98 MB)

Trainable params: 525,572 (2.00 MB)

Non-trainable params: 23,587,712 (89.98 MB)

```

/usr/local/lib/python3.12/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 t
will be ignored.
  self._warn_if_super_not_called()

```

Epoch 1/15
231/231  138s 554ms/step - accuracy: 0.4053 - loss: 1.2677 - val_accuracy: 0.5019 - val_loss: 1.0865

Epoch 2/15
231/231  119s 513ms/step - accuracy: 0.5034 - loss: 1.0945 - val_accuracy: 0.5989 - val_loss: 1.0407

Epoch 3/15
231/231  119s 513ms/step - accuracy: 0.5135 - loss: 1.0701 - val_accuracy: 0.5963 - val_loss: 1.0021

Epoch 4/15
231/231  119s 513ms/step - accuracy: 0.5369 - loss: 1.0484 - val_accuracy: 0.5982 - val_loss: 1.0090

Epoch 5/15
231/231  119s 514ms/step - accuracy: 0.5410 - loss: 1.0385 - val_accuracy: 0.5416 - val_loss: 0.9905

Epoch 6/15
231/231  119s 513ms/step - accuracy: 0.5428 - loss: 1.0169 - val_accuracy: 0.6285 - val_loss: 0.9195

Epoch 7/15
231/231  117s 506ms/step - accuracy: 0.5598 - loss: 0.9946 - val_accuracy: 0.6423 - val_loss: 0.9202

Epoch 8/15
231/231  143s 509ms/step - accuracy: 0.5721 - loss: 0.9830 - val_accuracy: 0.5882 - val_loss: 0.9447

Epoch 9/15
231/231  119s 514ms/step - accuracy: 0.5713 - loss: 0.9843 - val_accuracy: 0.6322 - val_loss: 0.9279

Epoch 10/15
231/231  119s 515ms/step - accuracy: 0.5736 - loss: 0.9639 - val_accuracy: 0.6455 - val_loss: 0.8775

Epoch 11/15
231/231  118s 508ms/step - accuracy: 0.5687 - loss: 0.9745 - val_accuracy: 0.6354 - val_loss: 0.9027

Epoch 12/15
231/231  118s 509ms/step - accuracy: 0.5790 - loss: 0.9440 - val_accuracy: 0.6530 - val_loss: 0.8290

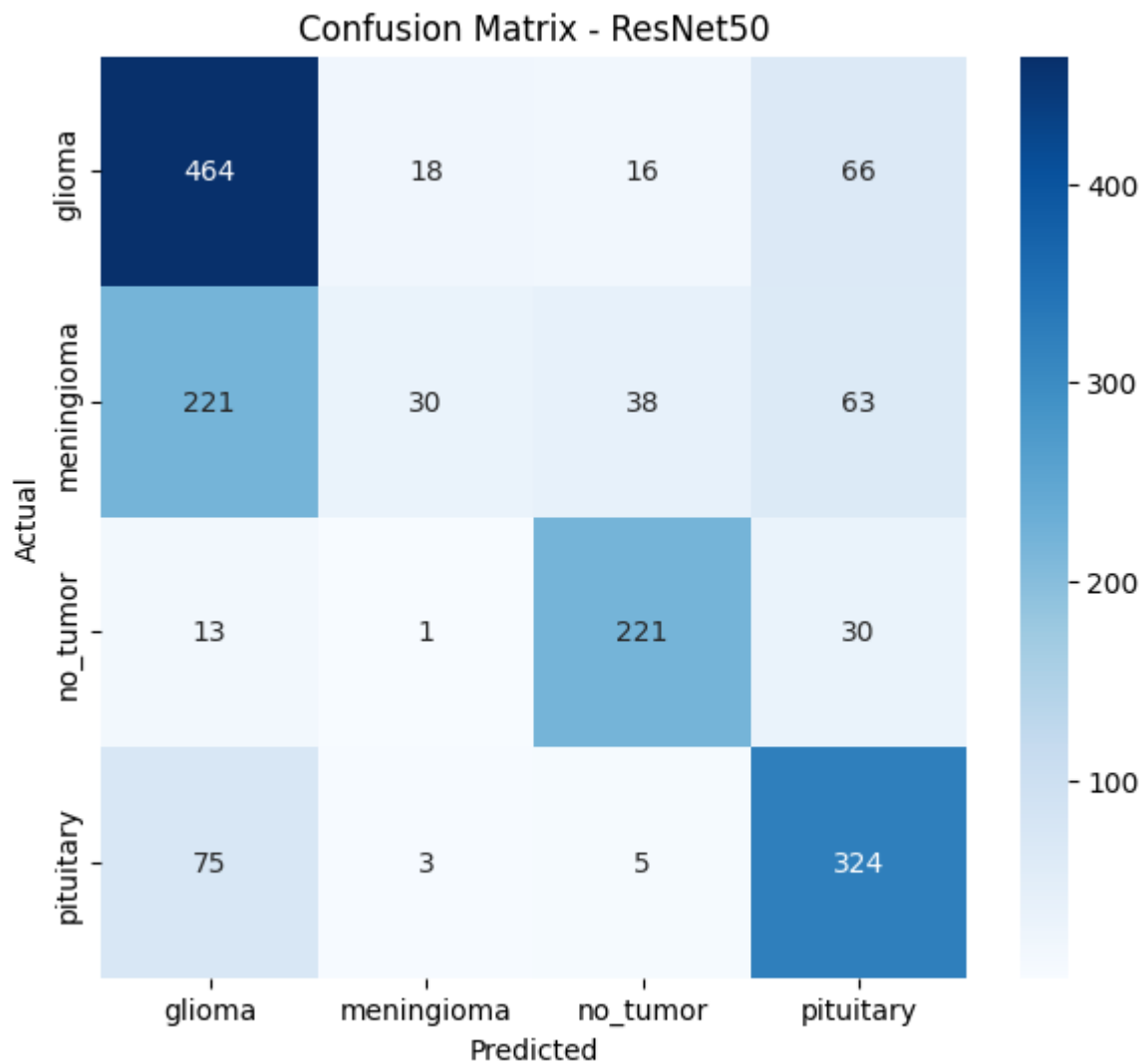
Epoch 13/15
231/231  119s 517ms/step - accuracy: 0.5965 - loss: 0.9356 - val_accuracy: 0.6669 - val_loss: 0.8843

Epoch 14/15
231/231  119s 514ms/step - accuracy: 0.5942 - loss: 0.9379 - val_accuracy: 0.6398 - val_loss: 0.8597

Epoch 15/15
231/231  118s 512ms/step - accuracy: 0.5950 - loss: 0.9303 - val_accuracy: 0.6543 - val_loss: 0.8468

50/50  9s 167ms/step - accuracy: 0.7100 - loss: 0.8175

 Test Accuracy: 0.6542820930480957
 Test Loss: 0.8468418717384338
50/50  16s 229ms/step



Classification Report:

	precision	recall	f1-score	support
glioma	0.60	0.82	0.69	564
meningioma	0.58	0.09	0.15	352
no_tumor	0.79	0.83	0.81	265
pituitary	0.67	0.80	0.73	407
accuracy			0.65	1588
macro avg	0.66	0.63	0.60	1588
weighted avg	0.64	0.65	0.60	1588