unet

June 13, 2024

[1]:
```
import os

import keras

print("Keras = {}".format(keras.__version__))
import tensorflow as tf

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'  # or any {'0', '1', '2'}
import matplotlib.pyplot as plt
import numpy as np
from keras.preprocessing.image import ImageDataGenerator
from keras.models import load_model
import seaborn as sns
import pandas as pd


# Print gpus
gpus = tf.config.experimental.list_physical_devices('GPU')
print("Num GPUs Available: ", len(gpus))

model_dir = './models/'
model_file = model_dir + 'unetpp.h5'
```

```
Keras = 2.15.0
Num GPUs Available:  1
```

[2]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[3]:
```
! unzip -o drive/MyDrive/dataset_19.zip

! pwd
```

```
Archive:  drive/MyDrive/dataset_19.zip
   creating: dataset_19/glioma/
  inflating: dataset_19/glioma/Te-gl_0010.jpg
  inflating: dataset_19/glioma/Te-gl_0016.jpg
```

```
inflating: dataset_19/glioma/Te-gl_0020.jpg
inflating: dataset_19/glioma/Te-gl_0026.jpg
inflating: dataset_19/glioma/Te-gl_0035.jpg
inflating: dataset_19/glioma/Te-gl_0041.jpg
inflating: dataset_19/glioma/Te-gl_0043.jpg
inflating: dataset_19/glioma/Te-gl_0044.jpg
inflating: dataset_19/glioma/Te-gl_0045.jpg
inflating: dataset_19/glioma/Te-gl_0057.jpg
inflating: dataset_19/glioma/Te-gl_0060.jpg
inflating: dataset_19/glioma/Te-gl_0075.jpg
inflating: dataset_19/glioma/Te-gl_0079.jpg
inflating: dataset_19/glioma/Te-gl_0085.jpg
inflating: dataset_19/glioma/Te-gl_0090.jpg
inflating: dataset_19/glioma/Te-gl_0107.jpg
inflating: dataset_19/glioma/Te-gl_0114.jpg
inflating: dataset_19/glioma/Te-gl_0125.jpg
inflating: dataset_19/glioma/Te-gl_0126.jpg
inflating: dataset_19/glioma/Te-gl_0127.jpg
inflating: dataset_19/glioma/Te-gl_0136.jpg
inflating: dataset_19/glioma/Te-gl_0144.jpg
inflating: dataset_19/glioma/Te-gl_0154.jpg
inflating: dataset_19/glioma/Te-gl_0155.jpg
inflating: dataset_19/glioma/Te-gl_0162.jpg
inflating: dataset_19/glioma/Te-gl_0166.jpg
inflating: dataset_19/glioma/Te-gl_0167.jpg
inflating: dataset_19/glioma/Te-gl_0168.jpg
inflating: dataset_19/glioma/Te-gl_0172.jpg
inflating: dataset_19/glioma/Te-gl_0183.jpg
inflating: dataset_19/glioma/Te-gl_0192.jpg
inflating: dataset_19/glioma/Te-gl_0195.jpg
inflating: dataset_19/glioma/Te-gl_0199.jpg
inflating: dataset_19/glioma/Tr-gl_0016.jpg
inflating: dataset_19/glioma/Tr-gl_0026.jpg
inflating: dataset_19/glioma/Tr-gl_0031.jpg
inflating: dataset_19/glioma/Tr-gl_0039.jpg
inflating: dataset_19/glioma/Tr-gl_0049.jpg
inflating: dataset_19/glioma/Tr-gl_0052.jpg
inflating: dataset_19/glioma/Tr-gl_0054.jpg
inflating: dataset_19/glioma/Tr-gl_0062.jpg
inflating: dataset_19/glioma/Tr-gl_0072.jpg
inflating: dataset_19/glioma/Tr-gl_0078.jpg
inflating: dataset_19/glioma/Tr-gl_0084.jpg
inflating: dataset_19/glioma/Tr-gl_0086.jpg
inflating: dataset_19/glioma/Tr-gl_0096.jpg
inflating: dataset_19/glioma/Tr-gl_0106.jpg
inflating: dataset_19/glioma/Tr-gl_0111.jpg
inflating: dataset_19/glioma/Tr-gl_0113.jpg
inflating: dataset_19/glioma/Tr-gl_0118.jpg
```

```
inflating: dataset_19/glioma/Tr-gl_0119.jpg
inflating: dataset_19/glioma/Tr-gl_0127(1).jpg
inflating: dataset_19/glioma/Tr-gl_0135.jpg
inflating: dataset_19/glioma/Tr-gl_0136.jpg
inflating: dataset_19/glioma/Tr-gl_0138.jpg
inflating: dataset_19/glioma/Tr-gl_0145.jpg
inflating: dataset_19/glioma/Tr-gl_0155.jpg
inflating: dataset_19/glioma/Tr-gl_0161.jpg
inflating: dataset_19/glioma/Tr-gl_0166.jpg
inflating: dataset_19/glioma/Tr-gl_0169.jpg
inflating: dataset_19/glioma/Tr-gl_0184.jpg
inflating: dataset_19/glioma/Tr-gl_0187.jpg
inflating: dataset_19/glioma/Tr-gl_0191.jpg
inflating: dataset_19/glioma/Tr-gl_0194.jpg
inflating: dataset_19/glioma/Tr-gl_0197.jpg
inflating: dataset_19/glioma/Tr-gl_0204.jpg
inflating: dataset_19/glioma/Tr-gl_0206.jpg
inflating: dataset_19/glioma/Tr-gl_0207.jpg
inflating: dataset_19/glioma/Tr-gl_0224.jpg
inflating: dataset_19/glioma/Tr-gl_0226.jpg
inflating: dataset_19/glioma/Tr-gl_0230.jpg
inflating: dataset_19/glioma/Tr-gl_0237.jpg
inflating: dataset_19/glioma/Tr-gl_0247.jpg
inflating: dataset_19/glioma/Tr-gl_0250.jpg
inflating: dataset_19/glioma/Tr-gl_0268.jpg
inflating: dataset_19/glioma/Tr-gl_0281.jpg
inflating: dataset_19/glioma/Tr-gl_0283.jpg
inflating: dataset_19/glioma/Tr-gl_0284.jpg
inflating: dataset_19/glioma/Tr-gl_0286.jpg
inflating: dataset_19/glioma/Tr-gl_0294.jpg
inflating: dataset_19/glioma/Tr-gl_0303.jpg
inflating: dataset_19/glioma/Tr-gl_0304.jpg
inflating: dataset_19/glioma/Tr-gl_0311.jpg
inflating: dataset_19/glioma/Tr-gl_0312.jpg
inflating: dataset_19/glioma/Tr-gl_0320.jpg
inflating: dataset_19/glioma/Tr-gl_0322.jpg
inflating: dataset_19/glioma/Tr-gl_0327.jpg
inflating: dataset_19/glioma/Tr-gl_0330.jpg
inflating: dataset_19/glioma/Tr-gl_0334.jpg
inflating: dataset_19/glioma/Tr-gl_0338.jpg
inflating: dataset_19/glioma/Tr-gl_0343.jpg
inflating: dataset_19/glioma/Tr-gl_0349.jpg
inflating: dataset_19/glioma/Tr-gl_0358.jpg
inflating: dataset_19/glioma/Tr-gl_0369.jpg
inflating: dataset_19/glioma/Tr-gl_0373.jpg
inflating: dataset_19/glioma/Tr-gl_0378.jpg
inflating: dataset_19/glioma/Tr-gl_0396.jpg
inflating: dataset_19/glioma/Tr-gl_0406.jpg
```

```
inflating: dataset_19/glioma/Tr-gl_0407.jpg
inflating: dataset_19/glioma/Tr-gl_0422.jpg
inflating: dataset_19/glioma/Tr-gl_0426.jpg
inflating: dataset_19/glioma/Tr-gl_0428.jpg
inflating: dataset_19/glioma/Tr-gl_0437.jpg
inflating: dataset_19/glioma/Tr-gl_0444.jpg
inflating: dataset_19/glioma/Tr-gl_0450.jpg
inflating: dataset_19/glioma/Tr-gl_0451.jpg
inflating: dataset_19/glioma/Tr-gl_0454.jpg
inflating: dataset_19/glioma/Tr-gl_0455.jpg
inflating: dataset_19/glioma/Tr-gl_0457.jpg
inflating: dataset_19/glioma/Tr-gl_0459.jpg
inflating: dataset_19/glioma/Tr-gl_0462.jpg
inflating: dataset_19/glioma/Tr-gl_0465.jpg
inflating: dataset_19/glioma/Tr-gl_0474.jpg
inflating: dataset_19/glioma/Tr-gl_0490.jpg
inflating: dataset_19/glioma/Tr-gl_0493.jpg
inflating: dataset_19/glioma/Tr-gl_0494(1).jpg
inflating: dataset_19/glioma/Tr-gl_0495.jpg
inflating: dataset_19/glioma/Tr-gl_0502.jpg
inflating: dataset_19/glioma/Tr-gl_0508.jpg
inflating: dataset_19/glioma/Tr-gl_0509.jpg
 creating: dataset_19/meningioma/
inflating: dataset_19/meningioma/Te-me_0014.jpg
inflating: dataset_19/meningioma/Te-me_0015.jpg
inflating: dataset_19/meningioma/Te-me_0016.jpg
inflating: dataset_19/meningioma/Te-me_0017(3).jpg
inflating: dataset_19/meningioma/Te-me_0017(4).jpg
inflating: dataset_19/meningioma/Te-me_0017.jpg
inflating: dataset_19/meningioma/Te-me_0026.jpg
inflating: dataset_19/meningioma/Te-me_0028.jpg
inflating: dataset_19/meningioma/Te-me_0032.jpg
inflating: dataset_19/meningioma/Te-me_0046.jpg
inflating: dataset_19/meningioma/Te-me_0053.jpg
inflating: dataset_19/meningioma/Te-me_0058.jpg
inflating: dataset_19/meningioma/Te-me_0068.jpg
inflating: dataset_19/meningioma/Te-me_0080.jpg
inflating: dataset_19/meningioma/Te-me_0082.jpg
inflating: dataset_19/meningioma/Te-me_0095.jpg
inflating: dataset_19/meningioma/Te-me_0097.jpg
inflating: dataset_19/meningioma/Te-me_0101.jpg
inflating: dataset_19/meningioma/Te-me_0104.jpg
inflating: dataset_19/meningioma/Te-me_0108.jpg
inflating: dataset_19/meningioma/Te-me_0119.jpg
inflating: dataset_19/meningioma/Te-me_0128.jpg
inflating: dataset_19/meningioma/Te-me_0140.jpg
inflating: dataset_19/meningioma/Te-me_0146(1).jpg
inflating: dataset_19/meningioma/Te-me_0147(1).jpg
```

```
inflating: dataset_19/meningioma/Te-me_0149.jpg
inflating: dataset_19/meningioma/Te-me_0151(1).jpg
inflating: dataset_19/meningioma/Te-me_0156.jpg
inflating: dataset_19/meningioma/Te-me_0160(1).jpg
inflating: dataset_19/meningioma/Te-me_0162(4).jpg
inflating: dataset_19/meningioma/Te-me_0165.jpg
inflating: dataset_19/meningioma/Te-me_0166.jpg
inflating: dataset_19/meningioma/Te-me_0169.jpg
inflating: dataset_19/meningioma/Te-me_0170(1).jpg
inflating: dataset_19/meningioma/Te-me_0170.jpg
inflating: dataset_19/meningioma/Te-me_0174(1).jpg
inflating: dataset_19/meningioma/Te-me_0175(1).jpg
inflating: dataset_19/meningioma/Te-me_0176.jpg
inflating: dataset_19/meningioma/Te-me_0182.jpg
inflating: dataset_19/meningioma/Te-me_0184(2).jpg
inflating: dataset_19/meningioma/Te-me_0184.jpg
inflating: dataset_19/meningioma/Te-me_0186.jpg
inflating: dataset_19/meningioma/Te-me_0192.jpg
inflating: dataset_19/meningioma/Te-me_0193.jpg
inflating: dataset_19/meningioma/Te-me_0198.jpg
inflating: dataset_19/meningioma/Te-me_0200(2).jpg
inflating: dataset_19/meningioma/Te-me_0202(1).jpg
inflating: dataset_19/meningioma/Tr-me_0013.jpg
inflating: dataset_19/meningioma/Tr-me_0027.jpg
inflating: dataset_19/meningioma/Tr-me_0029.jpg
inflating: dataset_19/meningioma/Tr-me_0030.jpg
inflating: dataset_19/meningioma/Tr-me_0039.jpg
inflating: dataset_19/meningioma/Tr-me_0043.jpg
inflating: dataset_19/meningioma/Tr-me_0067.jpg
inflating: dataset_19/meningioma/Tr-me_0068.jpg
inflating: dataset_19/meningioma/Tr-me_0069.jpg
inflating: dataset_19/meningioma/Tr-me_0071.jpg
inflating: dataset_19/meningioma/Tr-me_0073.jpg
inflating: dataset_19/meningioma/Tr-me_0076.jpg
inflating: dataset_19/meningioma/Tr-me_0098.jpg
inflating: dataset_19/meningioma/Tr-me_0099.jpg
inflating: dataset_19/meningioma/Tr-me_0101.jpg
inflating: dataset_19/meningioma/Tr-me_0107.jpg
inflating: dataset_19/meningioma/Tr-me_0108.jpg
inflating: dataset_19/meningioma/Tr-me_0113.jpg
inflating: dataset_19/meningioma/Tr-me_0119.jpg
inflating: dataset_19/meningioma/Tr-me_0121.jpg
inflating: dataset_19/meningioma/Tr-me_0125.jpg
inflating: dataset_19/meningioma/Tr-me_0134.jpg
inflating: dataset_19/meningioma/Tr-me_0140.jpg
inflating: dataset_19/meningioma/Tr-me_0147.jpg
inflating: dataset_19/meningioma/Tr-me_0149.jpg
inflating: dataset_19/meningioma/Tr-me_0159.jpg
```

```
inflating: dataset_19/meningioma/Tr-me_0167.jpg
inflating: dataset_19/meningioma/Tr-me_0182.jpg
inflating: dataset_19/meningioma/Tr-me_0183.jpg
inflating: dataset_19/meningioma/Tr-me_0187.jpg
inflating: dataset_19/meningioma/Tr-me_0199.jpg
inflating: dataset_19/meningioma/Tr-me_0211.jpg
inflating: dataset_19/meningioma/Tr-me_0223.jpg
inflating: dataset_19/meningioma/Tr-me_0227.jpg
inflating: dataset_19/meningioma/Tr-me_0237.jpg
inflating: dataset_19/meningioma/Tr-me_0247.jpg
inflating: dataset_19/meningioma/Tr-me_0250.jpg
inflating: dataset_19/meningioma/Tr-me_0251.jpg
inflating: dataset_19/meningioma/Tr-me_0256.jpg
inflating: dataset_19/meningioma/Tr-me_0261.jpg
inflating: dataset_19/meningioma/Tr-me_0272.jpg
inflating: dataset_19/meningioma/Tr-me_0282.jpg
inflating: dataset_19/meningioma/Tr-me_0284.jpg
inflating: dataset_19/meningioma/Tr-me_0301.jpg
inflating: dataset_19/meningioma/Tr-me_0303.jpg
inflating: dataset_19/meningioma/Tr-me_0317.jpg
inflating: dataset_19/meningioma/Tr-me_0318.jpg
inflating: dataset_19/meningioma/Tr-me_0323.jpg
inflating: dataset_19/meningioma/Tr-me_0338.jpg
inflating: dataset_19/meningioma/Tr-me_0347.jpg
inflating: dataset_19/meningioma/Tr-me_0361.jpg
inflating: dataset_19/meningioma/Tr-me_0368.jpg
inflating: dataset_19/meningioma/Tr-me_0371.jpg
inflating: dataset_19/meningioma/Tr-me_0378.jpg
inflating: dataset_19/meningioma/Tr-me_0393.jpg
inflating: dataset_19/meningioma/Tr-me_0400.jpg
inflating: dataset_19/meningioma/Tr-me_0402.jpg
inflating: dataset_19/meningioma/Tr-me_0420.jpg
inflating: dataset_19/meningioma/Tr-me_0422.jpg
inflating: dataset_19/meningioma/Tr-me_0432.jpg
inflating: dataset_19/meningioma/Tr-me_0433.jpg
inflating: dataset_19/meningioma/Tr-me_0434.jpg
inflating: dataset_19/meningioma/Tr-me_0441.jpg
inflating: dataset_19/meningioma/Tr-me_0458.jpg
inflating: dataset_19/meningioma/Tr-me_0461.jpg
inflating: dataset_19/meningioma/Tr-me_0466.jpg
inflating: dataset_19/meningioma/Tr-me_0467.jpg
inflating: dataset_19/meningioma/Tr-me_0479.jpg
inflating: dataset_19/meningioma/Tr-me_0480.jpg
inflating: dataset_19/meningioma/Tr-me_0485.jpg
inflating: dataset_19/meningioma/Tr-me_0492.jpg
inflating: dataset_19/meningioma/Tr-me_0494.jpg
inflating: dataset_19/meningioma/Tr-me_0510.jpg
 creating: dataset_19/notumor/
```

```
inflating: dataset_19/notumor/Te-no_0010.jpg
inflating: dataset_19/notumor/Te-no_0011.jpg
inflating: dataset_19/notumor/Te-no_0025.jpg
inflating: dataset_19/notumor/Te-no_0029.jpg
inflating: dataset_19/notumor/Te-no_0031.jpg
inflating: dataset_19/notumor/Te-no_0041.jpg
inflating: dataset_19/notumor/Te-no_0043.jpg
inflating: dataset_19/notumor/Te-no_0060.jpg
inflating: dataset_19/notumor/Te-no_0067.jpg
inflating: dataset_19/notumor/Te-no_0069.jpg
inflating: dataset_19/notumor/Te-no_0079.jpg
inflating: dataset_19/notumor/Te-no_0097.jpg
inflating: dataset_19/notumor/Te-no_0111.jpg
inflating: dataset_19/notumor/Te-no_0114.jpg
inflating: dataset_19/notumor/Te-no_0119.jpg
inflating: dataset_19/notumor/Te-no_0121.jpg
inflating: dataset_19/notumor/Te-no_0127.jpg
inflating: dataset_19/notumor/Te-no_0135.jpg
inflating: dataset_19/notumor/Te-no_0141.jpg
inflating: dataset_19/notumor/Te-no_0145.jpg
inflating: dataset_19/notumor/Te-no_0146.jpg
inflating: dataset_19/notumor/Te-no_0153.jpg
inflating: dataset_19/notumor/Te-no_0158.jpg
inflating: dataset_19/notumor/Te-no_0162.jpg
inflating: dataset_19/notumor/Te-no_0168.jpg
inflating: dataset_19/notumor/Te-no_0172.jpg
inflating: dataset_19/notumor/Te-no_0176.jpg
inflating: dataset_19/notumor/Te-no_0178.jpg
inflating: dataset_19/notumor/Te-no_0180.jpg
inflating: dataset_19/notumor/Te-no_0189.jpg
inflating: dataset_19/notumor/Te-no_0190.jpg
inflating: dataset_19/notumor/Te-no_0194.jpg
inflating: dataset_19/notumor/Te-no_0198.jpg
inflating: dataset_19/notumor/Te-no_0200.jpg
inflating: dataset_19/notumor/Te-no_0201.jpg
inflating: dataset_19/notumor/Te-no_0202.jpg
inflating: dataset_19/notumor/Te-no_0203.jpg
inflating: dataset_19/notumor/Tr-no_0032.jpg
inflating: dataset_19/notumor/Tr-no_0035.jpg
inflating: dataset_19/notumor/Tr-no_0036.jpg
inflating: dataset_19/notumor/Tr-no_0041.jpg
inflating: dataset_19/notumor/Tr-no_0044.jpg
inflating: dataset_19/notumor/Tr-no_0056.jpg
inflating: dataset_19/notumor/Tr-no_0063.jpg
inflating: dataset_19/notumor/Tr-no_0065.jpg
inflating: dataset_19/notumor/Tr-no_0076.jpg
inflating: dataset_19/notumor/Tr-no_0077.jpg
inflating: dataset_19/notumor/Tr-no_0079.jpg
```

```
inflating: dataset_19/notumor/Tr-no_0081.jpg
inflating: dataset_19/notumor/Tr-no_0090.jpg
inflating: dataset_19/notumor/Tr-no_0091.jpg
inflating: dataset_19/notumor/Tr-no_0103.jpg
inflating: dataset_19/notumor/Tr-no_0104.jpg
inflating: dataset_19/notumor/Tr-no_0118.jpg
inflating: dataset_19/notumor/Tr-no_0121.jpg
inflating: dataset_19/notumor/Tr-no_0124.jpg
inflating: dataset_19/notumor/Tr-no_0125.jpg
inflating: dataset_19/notumor/Tr-no_0130.jpg
inflating: dataset_19/notumor/Tr-no_0135.jpg
inflating: dataset_19/notumor/Tr-no_0137.jpg
inflating: dataset_19/notumor/Tr-no_0141.jpg
inflating: dataset_19/notumor/Tr-no_0142.jpg
inflating: dataset_19/notumor/Tr-no_0150.jpg
inflating: dataset_19/notumor/Tr-no_0157.jpg
inflating: dataset_19/notumor/Tr-no_0163.jpg
inflating: dataset_19/notumor/Tr-no_0165.jpg
inflating: dataset_19/notumor/Tr-no_0168.jpg
inflating: dataset_19/notumor/Tr-no_0170.jpg
inflating: dataset_19/notumor/Tr-no_0171.jpg
inflating: dataset_19/notumor/Tr-no_0174.jpg
inflating: dataset_19/notumor/Tr-no_0179.jpg
inflating: dataset_19/notumor/Tr-no_0183.jpg
inflating: dataset_19/notumor/Tr-no_0188.jpg
inflating: dataset_19/notumor/Tr-no_0189.jpg
inflating: dataset_19/notumor/Tr-no_0190.jpg
inflating: dataset_19/notumor/Tr-no_0194.jpg
inflating: dataset_19/notumor/Tr-no_0199.jpg
inflating: dataset_19/notumor/Tr-no_0200.jpg
inflating: dataset_19/notumor/Tr-no_0203.jpg
inflating: dataset_19/notumor/Tr-no_0211.jpg
inflating: dataset_19/notumor/Tr-no_0212.jpg
inflating: dataset_19/notumor/Tr-no_0214.jpg
inflating: dataset_19/notumor/Tr-no_0225(1).jpg
inflating: dataset_19/notumor/Tr-no_0233.jpg
inflating: dataset_19/notumor/Tr-no_0247.jpg
inflating: dataset_19/notumor/Tr-no_0256.jpg
inflating: dataset_19/notumor/Tr-no_0271.jpg
inflating: dataset_19/notumor/Tr-no_0272.jpg
inflating: dataset_19/notumor/Tr-no_0276.jpg
inflating: dataset_19/notumor/Tr-no_0289.jpg
inflating: dataset_19/notumor/Tr-no_0291.jpg
inflating: dataset_19/notumor/Tr-no_0295.jpg
inflating: dataset_19/notumor/Tr-no_0303.jpg
inflating: dataset_19/notumor/Tr-no_0306.jpg
inflating: dataset_19/notumor/Tr-no_0310.jpg
inflating: dataset_19/notumor/Tr-no_0328.jpg
```

```
inflating: dataset_19/notumor/Tr-no_0350.jpg
inflating: dataset_19/notumor/Tr-no_0351.jpg
inflating: dataset_19/notumor/Tr-no_0371.jpg
inflating: dataset_19/notumor/Tr-no_0373.jpg
inflating: dataset_19/notumor/Tr-no_0379.jpg
inflating: dataset_19/notumor/Tr-no_0386.jpg
inflating: dataset_19/notumor/Tr-no_0392.jpg
inflating: dataset_19/notumor/Tr-no_0393.jpg
inflating: dataset_19/notumor/Tr-no_0394.jpg
inflating: dataset_19/notumor/Tr-no_0398.jpg
inflating: dataset_19/notumor/Tr-no_0412.jpg
inflating: dataset_19/notumor/Tr-no_0421.jpg
inflating: dataset_19/notumor/Tr-no_0424.jpg
inflating: dataset_19/notumor/Tr-no_0425.jpg
inflating: dataset_19/notumor/Tr-no_0441.jpg
inflating: dataset_19/notumor/Tr-no_0444.jpg
inflating: dataset_19/notumor/Tr-no_0447.jpg
inflating: dataset_19/notumor/Tr-no_0457.jpg
inflating: dataset_19/notumor/Tr-no_0472.jpg
inflating: dataset_19/notumor/Tr-no_0478.jpg
inflating: dataset_19/notumor/Tr-no_0490.jpg
inflating: dataset_19/notumor/Tr-no_0492.jpg
inflating: dataset_19/notumor/Tr-no_0509.jpg
inflating: dataset_19/notumor/Tr-no_0510.jpg
 creating: dataset_19/pituitary/
inflating: dataset_19/pituitary/Te-pi_0015.jpg
inflating: dataset_19/pituitary/Te-pi_0017.jpg
inflating: dataset_19/pituitary/Te-pi_0023.jpg
inflating: dataset_19/pituitary/Te-pi_0036.jpg
inflating: dataset_19/pituitary/Te-pi_0047.jpg
inflating: dataset_19/pituitary/Te-pi_0050.jpg
inflating: dataset_19/pituitary/Te-pi_0052.jpg
inflating: dataset_19/pituitary/Te-pi_0062.jpg
inflating: dataset_19/pituitary/Te-pi_0064.jpg
inflating: dataset_19/pituitary/Te-pi_0088.jpg
inflating: dataset_19/pituitary/Te-pi_0091.jpg
inflating: dataset_19/pituitary/Te-pi_0093.jpg
inflating: dataset_19/pituitary/Te-pi_0103.jpg
inflating: dataset_19/pituitary/Te-pi_0108.jpg
inflating: dataset_19/pituitary/Te-pi_0116.jpg
inflating: dataset_19/pituitary/Te-pi_0123.jpg
inflating: dataset_19/pituitary/Te-pi_0132.jpg
inflating: dataset_19/pituitary/Te-pi_0144.jpg
inflating: dataset_19/pituitary/Te-pi_0146.jpg
inflating: dataset_19/pituitary/Te-pi_0148.jpg
inflating: dataset_19/pituitary/Te-pi_0153.jpg
inflating: dataset_19/pituitary/Te-pi_0160.jpg
inflating: dataset_19/pituitary/Te-pi_0162.jpg
```

```
inflating: dataset_19/pituitary/Te-pi_0171.jpg
inflating: dataset_19/pituitary/Te-pi_0174.jpg
inflating: dataset_19/pituitary/Te-pi_0182.jpg
inflating: dataset_19/pituitary/Te-pi_0202.jpg
inflating: dataset_19/pituitary/Te-pi_0204.jpg
inflating: dataset_19/pituitary/Te-pi_0206.jpg
inflating: dataset_19/pituitary/Tr-pi_0023.jpg
inflating: dataset_19/pituitary/Tr-pi_0035.jpg
inflating: dataset_19/pituitary/Tr-pi_0037.jpg
inflating: dataset_19/pituitary/Tr-pi_0039(3).jpg
inflating: dataset_19/pituitary/Tr-pi_0039.jpg
inflating: dataset_19/pituitary/Tr-pi_0040.jpg
inflating: dataset_19/pituitary/Tr-pi_0045.jpg
inflating: dataset_19/pituitary/Tr-pi_0051.jpg
inflating: dataset_19/pituitary/Tr-pi_0054.jpg
inflating: dataset_19/pituitary/Tr-pi_0067.jpg
inflating: dataset_19/pituitary/Tr-pi_0074.jpg
inflating: dataset_19/pituitary/Tr-pi_0086.jpg
inflating: dataset_19/pituitary/Tr-pi_0091.jpg
inflating: dataset_19/pituitary/Tr-pi_0098.jpg
inflating: dataset_19/pituitary/Tr-pi_0099.jpg
inflating: dataset_19/pituitary/Tr-pi_0105.jpg
inflating: dataset_19/pituitary/Tr-pi_0107.jpg
inflating: dataset_19/pituitary/Tr-pi_0110.jpg
inflating: dataset_19/pituitary/Tr-pi_0118.jpg
inflating: dataset_19/pituitary/Tr-pi_0119.jpg
inflating: dataset_19/pituitary/Tr-pi_0121.jpg
inflating: dataset_19/pituitary/Tr-pi_0122.jpg
inflating: dataset_19/pituitary/Tr-pi_0123.jpg
inflating: dataset_19/pituitary/Tr-pi_0126.jpg
inflating: dataset_19/pituitary/Tr-pi_0132.jpg
inflating: dataset_19/pituitary/Tr-pi_0145.jpg
inflating: dataset_19/pituitary/Tr-pi_0147.jpg
inflating: dataset_19/pituitary/Tr-pi_0149.jpg
inflating: dataset_19/pituitary/Tr-pi_0153.jpg
inflating: dataset_19/pituitary/Tr-pi_0156.jpg
inflating: dataset_19/pituitary/Tr-pi_0159(1).jpg
inflating: dataset_19/pituitary/Tr-pi_0159(2).jpg
inflating: dataset_19/pituitary/Tr-pi_0162.jpg
inflating: dataset_19/pituitary/Tr-pi_0165.jpg
inflating: dataset_19/pituitary/Tr-pi_0170.jpg
inflating: dataset_19/pituitary/Tr-pi_0171.jpg
inflating: dataset_19/pituitary/Tr-pi_0190.jpg
inflating: dataset_19/pituitary/Tr-pi_0194.jpg
inflating: dataset_19/pituitary/Tr-pi_0199.jpg
inflating: dataset_19/pituitary/Tr-pi_0204.jpg
inflating: dataset_19/pituitary/Tr-pi_0206.jpg
inflating: dataset_19/pituitary/Tr-pi_0213.jpg
```

```
inflating: dataset_19/pituitary/Tr-pi_0218.jpg
inflating: dataset_19/pituitary/Tr-pi_0224.jpg
inflating: dataset_19/pituitary/Tr-pi_0229.jpg
inflating: dataset_19/pituitary/Tr-pi_0238.jpg
inflating: dataset_19/pituitary/Tr-pi_0250.jpg
inflating: dataset_19/pituitary/Tr-pi_0260.jpg
inflating: dataset_19/pituitary/Tr-pi_0263.jpg
inflating: dataset_19/pituitary/Tr-pi_0274.jpg
inflating: dataset_19/pituitary/Tr-pi_0282.jpg
inflating: dataset_19/pituitary/Tr-pi_0298.jpg
inflating: dataset_19/pituitary/Tr-pi_0308.jpg
inflating: dataset_19/pituitary/Tr-pi_0311.jpg
inflating: dataset_19/pituitary/Tr-pi_0315.jpg
inflating: dataset_19/pituitary/Tr-pi_0326.jpg
inflating: dataset_19/pituitary/Tr-pi_0327.jpg
inflating: dataset_19/pituitary/Tr-pi_0330.jpg
inflating: dataset_19/pituitary/Tr-pi_0347.jpg
inflating: dataset_19/pituitary/Tr-pi_0356.jpg
inflating: dataset_19/pituitary/Tr-pi_0359.jpg
inflating: dataset_19/pituitary/Tr-pi_0362.jpg
inflating: dataset_19/pituitary/Tr-pi_0367.jpg
inflating: dataset_19/pituitary/Tr-pi_0374.jpg
inflating: dataset_19/pituitary/Tr-pi_0375.jpg
inflating: dataset_19/pituitary/Tr-pi_0388.jpg
inflating: dataset_19/pituitary/Tr-pi_0394.jpg
inflating: dataset_19/pituitary/Tr-pi_0396.jpg
inflating: dataset_19/pituitary/Tr-pi_0399.jpg
inflating: dataset_19/pituitary/Tr-pi_0400.jpg
inflating: dataset_19/pituitary/Tr-pi_0405.jpg
inflating: dataset_19/pituitary/Tr-pi_0420.jpg
inflating: dataset_19/pituitary/Tr-pi_0430.jpg
inflating: dataset_19/pituitary/Tr-pi_0450.jpg
inflating: dataset_19/pituitary/Tr-pi_0451.jpg
inflating: dataset_19/pituitary/Tr-pi_0452.jpg
inflating: dataset_19/pituitary/Tr-pi_0455.jpg
inflating: dataset_19/pituitary/Tr-pi_0462.jpg
inflating: dataset_19/pituitary/Tr-pi_0464.jpg
inflating: dataset_19/pituitary/Tr-pi_0469.jpg
inflating: dataset_19/pituitary/Tr-pi_0472.jpg
inflating: dataset_19/pituitary/Tr-pi_0479.jpg
inflating: dataset_19/pituitary/Tr-pi_0485.jpg
inflating: dataset_19/pituitary/Tr-pi_0486.jpg
inflating: dataset_19/pituitary/Tr-pi_0487.jpg
inflating: dataset_19/pituitary/Tr-pi_0489.jpg
inflating: dataset_19/pituitary/Tr-pi_0494.jpg
inflating: dataset_19/pituitary/Tr-pi_0496.jpg
inflating: dataset_19/pituitary/Tr-pi_0503.jpg
inflating: dataset_19/pituitary/Tr-pi_0507.jpg
```

```
  inflating: dataset_19/pituitary/Tr-pi_0509.jpg
/content
```

1 Load the data

[4]:
```
# Data Directories
dir = "dataset_19/"
```

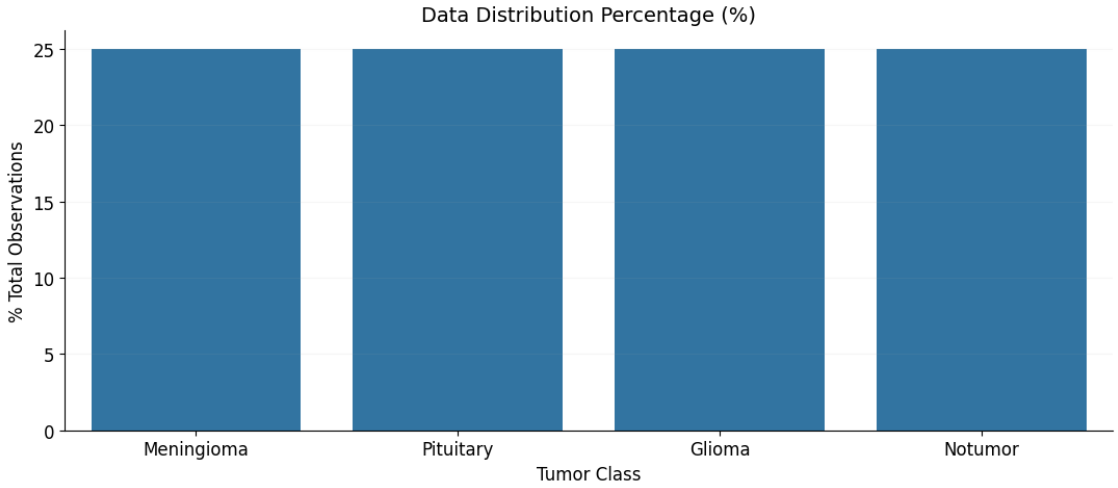
1.1 Data Distribution

[ ]:
```
data_distribution_count = pd.Series(
    {curr_index: len(os.listdir(os.path.join(dir, curr_index))) for curr_index␣
 ↪in os.listdir(dir)})

data_distribution_count
```

[ ]:
```
meningioma    120
pituitary     120
glioma        120
notumor       120
dtype: int64
```

[ ]:
```
fig, axis = plt.subplots(figsize=(13, 5))
axis.grid(True, alpha=0.1)
axis.set_title("Data Distribution Percentage (%)", fontsize=14)
sns.barplot(x=['\n'.join(curr_index.strip().split('_')).title() for curr_index␣
 ↪in data_distribution_count.index],
            y=100 * data_distribution_count / data_distribution_count.sum(),␣
 ↪ax=axis)
axis.set_xlabel("Tumor Class", fontsize=12)
axis.set_ylabel("% Total Observations", fontsize=12)
axis.tick_params(which='major', labelsize=12)
axis.text(2.5, 37, f'Total Observations: {data_distribution_count.sum()}',␣
 ↪fontdict=dict(size=12))
sns.despine()
```

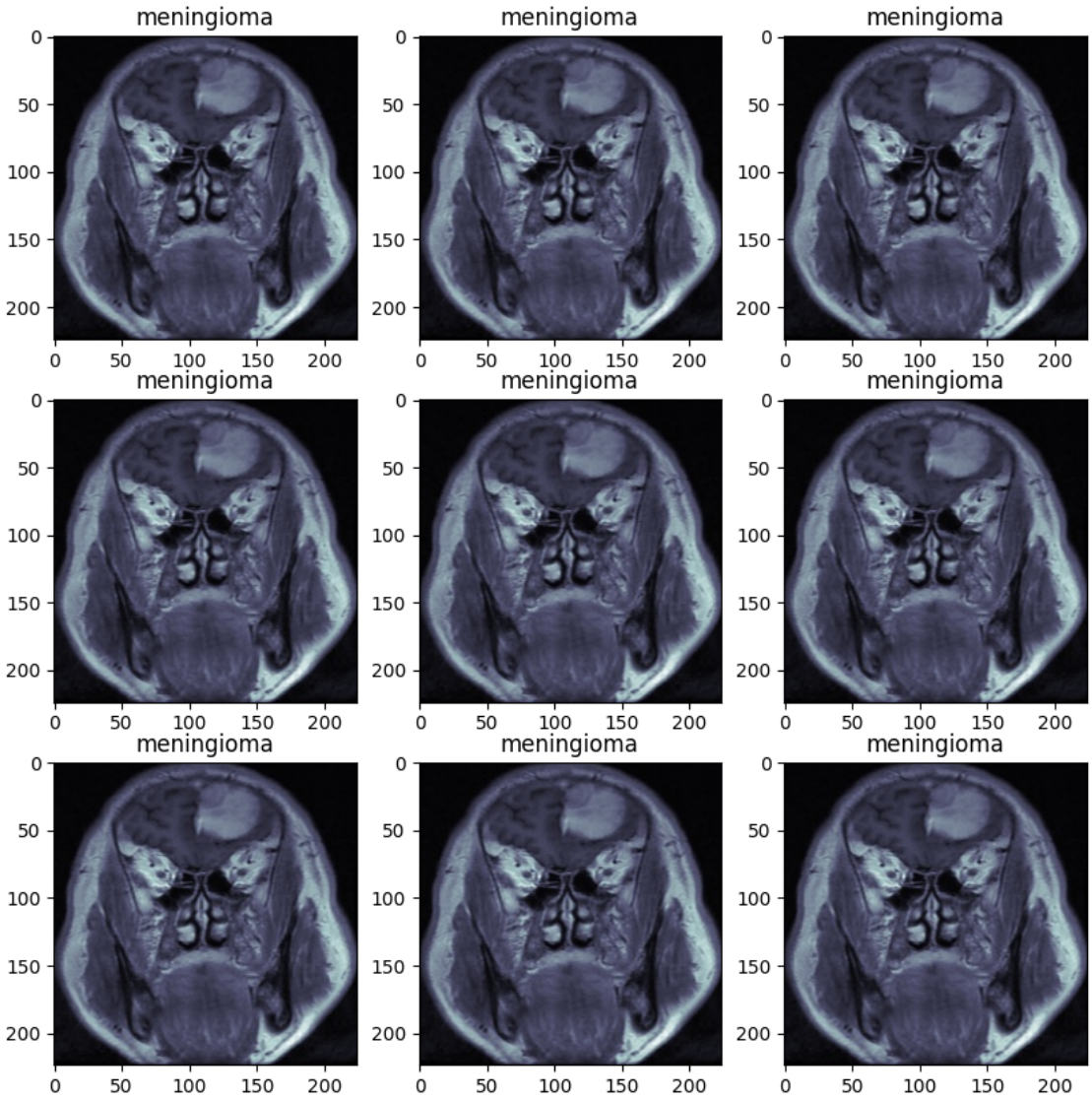
Total Observations: 480



2 Preprocess Data

[5]:
```
from tqdm import tqdm
import cv2
import imutils

def crop_img(img):

  # Find extreme points on the image and crop the rectangular out

  gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
  gray = cv2.GaussianBlur(gray, (3, 3), 0)

  # threshold the image, then perform a series of erosions +
  # dilations to remove any small regions of noise
  thresh = cv2.threshold(gray, 45, 255, cv2.THRESH_BINARY)[1]
  thresh = cv2.erode(thresh, None, iterations=2)
  thresh = cv2.dilate(thresh, None, iterations=2)

  # find contours in thresholded image, then grab the largest one
  cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.
↪CHAIN_APPROX_SIMPLE)
  cnts = imutils.grab_contours(cnts)
  c = max(cnts, key=cv2.contourArea)
```

```
  # find the extreme points
  extLeft = tuple(c[c[:, :, 0].argmin()][0])
  extRight = tuple(c[c[:, :, 0].argmax()][0])
  extTop = tuple(c[c[:, :, 1].argmin()][0])
  extBot = tuple(c[c[:, :, 1].argmax()][0])
  ADD_PIXELS = 0
  new_img = img[extTop[1]-ADD_PIXELS:extBot[1]+ADD_PIXELS,␣
 ↪extLeft[0]-ADD_PIXELS:extRight[0]+ADD_PIXELS].copy()

  return new_img

def preprocess_images(directory):
    for dir in os.listdir(directory):
        path = os.path.join(directory, dir)
        for img_name in os.listdir(path):
            img_path = os.path.join(path, img_name)
            img = cv2.imread(img_path)
            cropped_img = crop_img(img)
            processed_img = cv2.cvtColor(cropped_img, cv2.COLOR_RGB2GRAY)
            processed_img = cv2.bilateralFilter(processed_img, 2, 50, 50)
            processed_img = cv2.applyColorMap(processed_img, cv2.COLORMAP_BONE)
            processed_img = cv2.resize(processed_img, (224, 224))
            cv2.imwrite(img_path, processed_img)


# Preprocess the images before generating data
preprocess_images(dir)

# Display 9 image using matplotlib
plt.figure(figsize=(10, 10))
for i in range(9):
    plt.subplot(3, 3, i + 1)
    for curr_index in os.listdir(dir):
        path = os.path.join(dir, curr_index)
        for img_name in os.listdir(path):
            img_path = os.path.join(path, img_name)
            image = plt.imread(img_path)
            plt.imshow(image)
            plt.title(curr_index)
            break
        break
```



2.1 Splitting the data

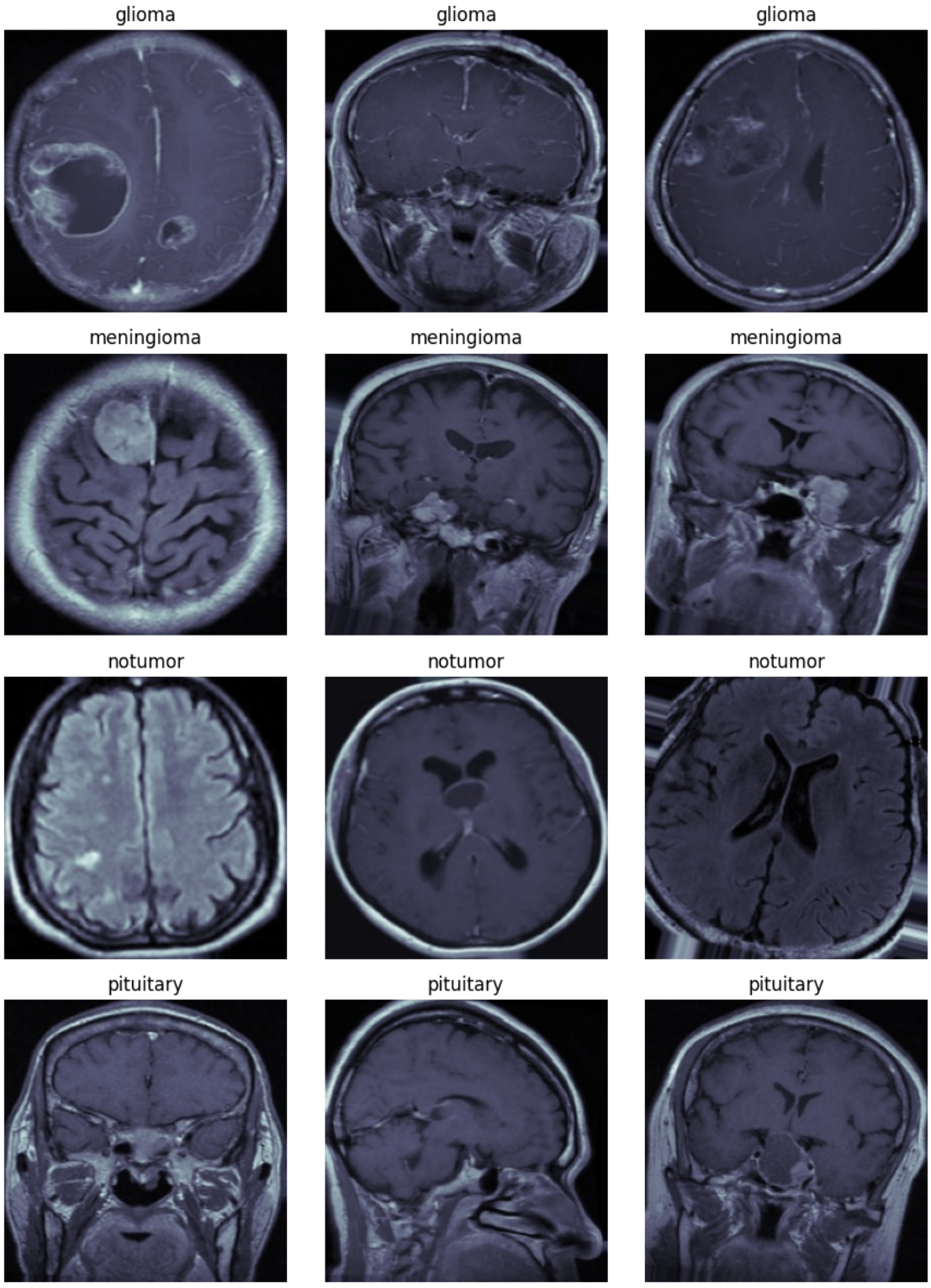
[6]:
```
classes = os.listdir(dir)

batch_size = 10

train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    horizontal_flip=True,
    rotation_range=20,
    validation_split=0.2)
```

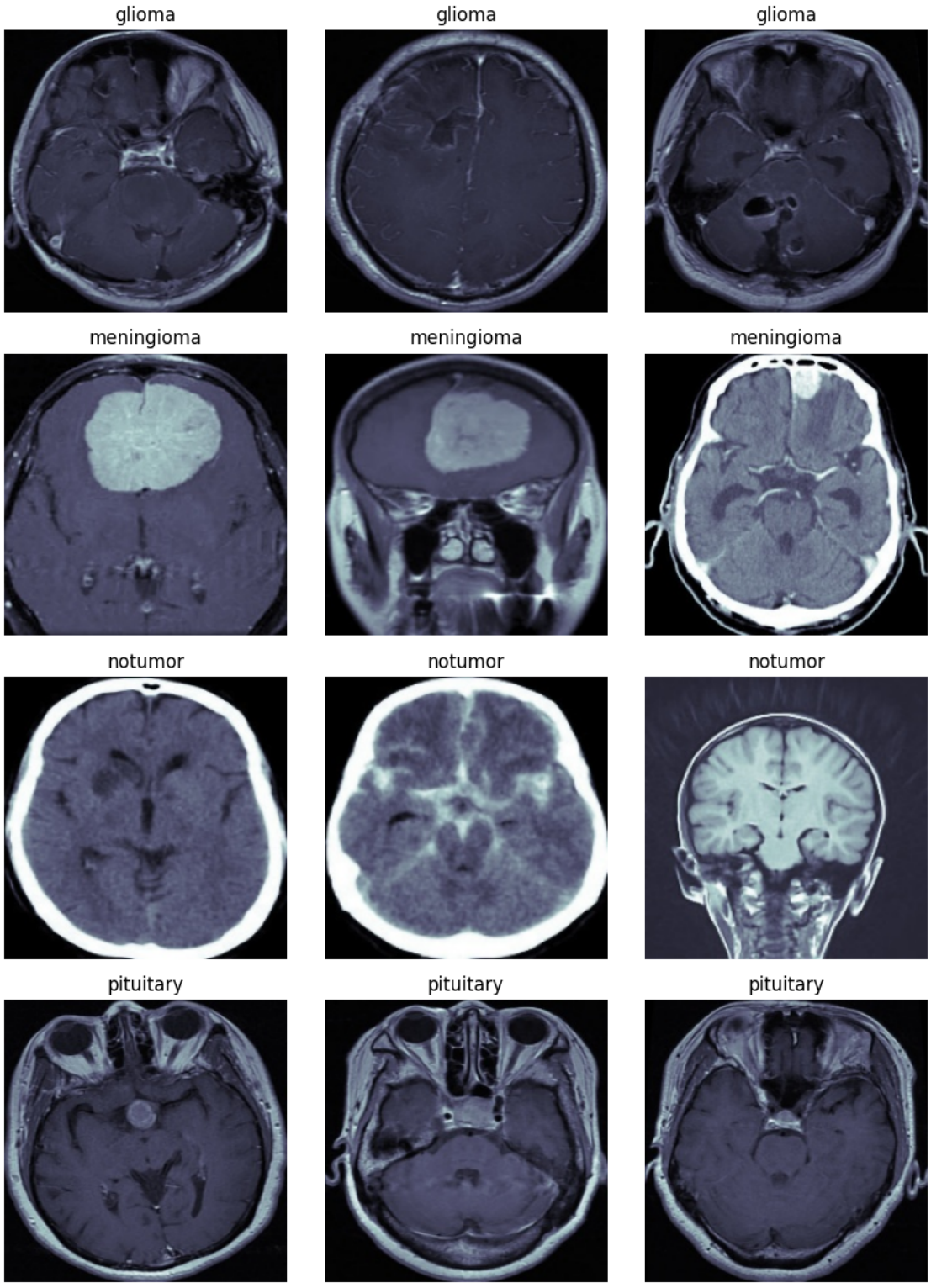
```
validation_datagen = ImageDataGenerator(rescale=1. / 255,
                                        validation_split=0.2)

train_generator = train_datagen.flow_from_directory(
    dir,
    target_size=(224, 224),
    batch_size=batch_size,
    seed=42,
    subset='training'
)

test_generator = validation_datagen.flow_from_directory(
    dir,
    target_size=(224, 224),
    batch_size=batch_size,
    seed=42,
    shuffle = False,
    subset='validation')


print(test_generator.class_indices)
```

```
Found 384 images belonging to 4 classes.
Found 96 images belonging to 4 classes.
{'glioma': 0, 'meningioma': 1, 'notumor': 2, 'pituitary': 3}
```

[10]:
```
import matplotlib.pyplot as plt
import numpy as np

def plot_images_from_generator(generator, num_images_per_class=3):
    # Get the class labels
    class_indices = generator.class_indices
    class_names = list(class_indices.keys())
    num_classes = len(class_names)

    # Initialize a dictionary to keep track of the images plotted for each class
    images_per_class = {class_name: 0 for class_name in class_names}

    # Prepare a figure for plotting
    fig, axes = plt.subplots(num_classes, num_images_per_class,␣
 ↪figsize=(num_images_per_class*3, num_classes*3))

    # Ensure the axes array is always 2D
    if num_classes == 1:
        axes = np.expand_dims(axes, axis=0)

    if num_images_per_class == 1:
```

```
        axes = np.expand_dims(axes, axis=1)

    # Loop through batches of images from the generator
    for batch_x, batch_y in generator:
        for i in range(len(batch_x)):
            # Get the image and its corresponding label
            img = batch_x[i]
            label_index = np.argmax(batch_y[i])
            label_name = class_names[label_index]

            # Check if we have already plotted enough images for this class
            if images_per_class[label_name] < num_images_per_class:
                row = list(class_indices.keys()).index(label_name)
                col = images_per_class[label_name]
                axes[row, col].imshow(img)
                axes[row, col].axis('off')
                axes[row, col].set_title(label_name)

                # Update the count of images plotted for this class
                images_per_class[label_name] += 1

            # Check if we have enough images for all classes
            if all(count >= num_images_per_class for count in images_per_class.
 ↪values()):
                break
        else:
            continue
        break

    plt.tight_layout()
    plt.show()

# Visualize images from the training generator
plot_images_from_generator(train_generator, num_images_per_class=3)

# Visualize images from the validation generator
plot_images_from_generator(test_generator, num_images_per_class=3)
```


glioma
glioma
glioma
meningioma
meningioma
meningioma
notumor
notumor
notumor
pituitary
pituitary
pituitary


glioma
glioma
glioma
meningioma
meningioma
meningioma
notumor
notumor
notumor
pituitary
pituitary
pituitary

3 Model Training

[ ]:
```
!pip install segmentation-models
```

```
Collecting segmentation-models
  Downloading segmentation_models-1.0.1-py3-none-any.whl (33 kB)
Collecting keras-applications<=1.0.8,>=1.0.7 (from segmentation-models)
  Downloading Keras_Applications-1.0.8-py3-none-any.whl (50 kB)
                              50.7/50.7 kB
6.9 MB/s eta 0:00:00
Collecting image-classifiers==1.0.0 (from segmentation-models)
  Downloading image_classifiers-1.0.0-py3-none-any.whl (19 kB)
Collecting efficientnet==1.0.0 (from segmentation-models)
  Downloading efficientnet-1.0.0-py3-none-any.whl (17 kB)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist-
packages (from efficientnet==1.0.0->segmentation-models) (0.19.3)
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.10/dist-
packages (from keras-applications<=1.0.8,>=1.0.7->segmentation-models) (1.25.2)
Requirement already satisfied: h5py in /usr/local/lib/python3.10/dist-packages
(from keras-applications<=1.0.8,>=1.0.7->segmentation-models) (3.9.0)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-
packages (from scikit-image->efficientnet==1.0.0->segmentation-models) (1.11.4)
Requirement already satisfied: networkx>=2.2 in /usr/local/lib/python3.10/dist-
packages (from scikit-image->efficientnet==1.0.0->segmentation-models) (3.3)
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,!=8.3.0,>=6.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-
image->efficientnet==1.0.0->segmentation-models) (9.4.0)
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-
packages (from scikit-image->efficientnet==1.0.0->segmentation-models) (2.31.6)
Requirement already satisfied: tifffile>=2019.7.26 in
/usr/local/lib/python3.10/dist-packages (from scikit-
image->efficientnet==1.0.0->segmentation-models) (2024.5.22)
Requirement already satisfied: PyWavelets>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from scikit-
image->efficientnet==1.0.0->segmentation-models) (1.6.0)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-
image->efficientnet==1.0.0->segmentation-models) (24.0)
Installing collected packages: keras-applications, image-classifiers,
efficientnet, segmentation-models
Successfully installed efficientnet-1.0.0 image-classifiers-1.0.0 keras-
applications-1.0.8 segmentation-models-1.0.1
```

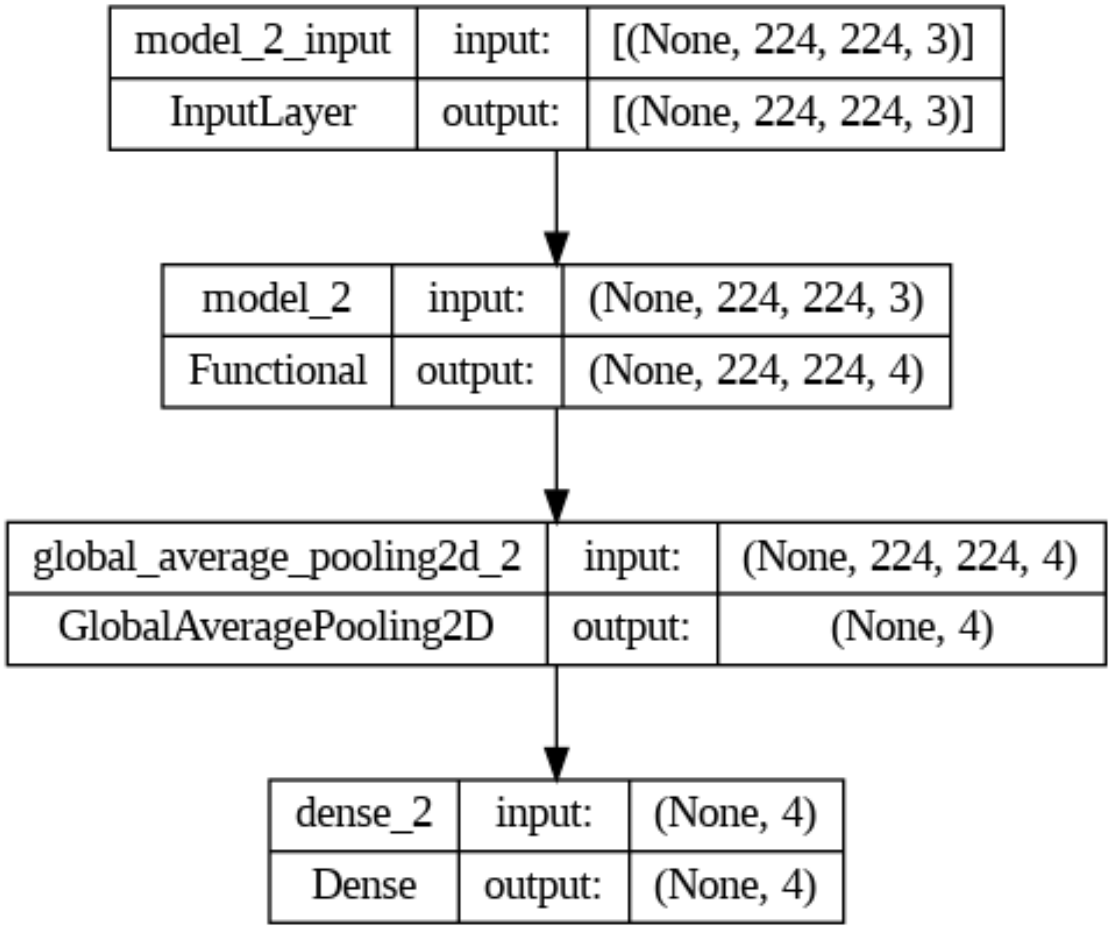
[ ]:
```
import os
os.environ["SM_FRAMEWORK"] = "tf.keras"

from tensorflow import keras
from segmentation_models import Unet
```

```
from tensorflow.keras.regularizers import l2

# Model parameters
input_shape = (224, 224, 3)
num_classes = 4
initial_learning_rate = 1e-4

# Define the U-Net++ model
model = Unet('efficientnetb1', input_shape=input_shape, classes=num_classes,␣
 ↪activation=None)
tf.keras.utils.plot_model(model, to_file='unetpp_model.png', show_shapes=True,␣
 ↪show_layer_names=True)

# Add a GlobalAveragePooling2D layer to collapse spatial dimensions
model = keras.Sequential([
    model,
    keras.layers.GlobalAveragePooling2D()  # Pool across spatial dimensions
])

# Add a Dense layer with softmax activation for final classification
model.add(keras.layers.Dense(num_classes, activation='softmax'))

# Compile the model
optimizer = keras.optimizers.Adam(learning_rate=initial_learning_rate)
model.compile(optimizer=optimizer,
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Summary of the model
model.summary()
tf.keras.utils.plot_model(model, to_file='unetpp_model2.png', show_shapes=True,␣
 ↪show_layer_names=True)
```

```
dot: graph is too large for cairo-renderer bitmaps. Scaling by 0.794525 to fit

Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 model_2 (Functional)        (None, 224, 224, 4)       12641604

 global_average_pooling2d_2  (None, 4)                 0
  (GlobalAveragePooling2D)

 dense_2 (Dense)             (None, 4)                 20

=================================================================
```

```
Total params: 12641624 (48.22 MB)
Trainable params: 12577592 (47.98 MB)
Non-trainable params: 64032 (250.12 KB)
_________________________________________________________________
```

[ ]:



[ ]:
```
from keras.callbacks import *

# Callbacks
reduce_lr = ReduceLROnPlateau(monitor='val_loss', patience=10, factor=0.3,␣
 ↪min_lr=1e-6, verbose=1)
checkpoint = ModelCheckpoint(model_file, monitor='val_loss', verbose=1,␣
 ↪save_best_only=True)
early_stopping = EarlyStopping(monitor='val_loss', min_delta=0, patience=20,␣
 ↪verbose=1, mode='auto')

# Train the model
history = model.fit(
    train_generator,
```

```
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    epochs=100,
    validation_data=test_generator,
    validation_steps=test_generator.samples // test_generator.batch_size,
    callbacks=[checkpoint, early_stopping, reduce_lr]
)
```

```
Epoch 1/100
38/38 [==============================] - ETA: 0s - loss: 1.3388 - accuracy:
0.3182
Epoch 1: val_loss improved from inf to 1.22903, saving model to
./models/unetpp.h5
38/38 [==============================] - 55s 193ms/step - loss: 1.3388 -
accuracy: 0.3182 - val_loss: 1.2290 - val_accuracy: 0.3222 - lr: 1.0000e-04
Epoch 2/100
38/38 [==============================] - ETA: 0s - loss: 1.0598 - accuracy:
0.6257
Epoch 2: val_loss did not improve from 1.22903
38/38 [==============================] - 5s 117ms/step - loss: 1.0598 -
accuracy: 0.6257 - val_loss: 1.5023 - val_accuracy: 0.4222 - lr: 1.0000e-04
Epoch 3/100
38/38 [==============================] - ETA: 0s - loss: 0.8673 - accuracy:
0.7166
Epoch 3: val_loss improved from 1.22903 to 0.85093, saving model to
./models/unetpp.h5
38/38 [==============================] - 6s 145ms/step - loss: 0.8673 -
accuracy: 0.7166 - val_loss: 0.8509 - val_accuracy: 0.6778 - lr: 1.0000e-04
Epoch 4/100
38/38 [==============================] - ETA: 0s - loss: 0.7575 - accuracy:
0.7995
Epoch 4: val_loss improved from 0.85093 to 0.74367, saving model to
./models/unetpp.h5
38/38 [==============================] - 6s 147ms/step - loss: 0.7575 -
accuracy: 0.7995 - val_loss: 0.7437 - val_accuracy: 0.6889 - lr: 1.0000e-04
Epoch 5/100
38/38 [==============================] - ETA: 0s - loss: 0.7159 - accuracy:
0.8182
Epoch 5: val_loss did not improve from 0.74367
38/38 [==============================] - 4s 112ms/step - loss: 0.7159 -
accuracy: 0.8182 - val_loss: 0.7468 - val_accuracy: 0.7444 - lr: 1.0000e-04
Epoch 6/100
38/38 [==============================] - ETA: 0s - loss: 0.5693 - accuracy:
0.8717
Epoch 6: val_loss improved from 0.74367 to 0.64111, saving model to
./models/unetpp.h5
38/38 [==============================] - 6s 144ms/step - loss: 0.5693 -
accuracy: 0.8717 - val_loss: 0.6411 - val_accuracy: 0.7444 - lr: 1.0000e-04
Epoch 7/100
```

```
38/38 [==============================] - ETA: 0s - loss: 0.5355 - accuracy:
0.8743
Epoch 7: val_loss improved from 0.64111 to 0.60537, saving model to
./models/unetpp.h5
38/38 [==============================] - 6s 146ms/step - loss: 0.5355 -
accuracy: 0.8743 - val_loss: 0.6054 - val_accuracy: 0.7778 - lr: 1.0000e-04
Epoch 8/100
38/38 [==============================] - ETA: 0s - loss: 0.4354 - accuracy:
0.9225
Epoch 8: val_loss improved from 0.60537 to 0.53137, saving model to
./models/unetpp.h5
38/38 [==============================] - 6s 145ms/step - loss: 0.4354 -
accuracy: 0.9225 - val_loss: 0.5314 - val_accuracy: 0.7667 - lr: 1.0000e-04
Epoch 9/100
38/38 [==============================] - ETA: 0s - loss: 0.4749 - accuracy:
0.8797
Epoch 9: val_loss improved from 0.53137 to 0.50809, saving model to
./models/unetpp.h5
38/38 [==============================] - 6s 145ms/step - loss: 0.4749 -
accuracy: 0.8797 - val_loss: 0.5081 - val_accuracy: 0.8000 - lr: 1.0000e-04
Epoch 10/100
38/38 [==============================] - ETA: 0s - loss: 0.4342 - accuracy:
0.9091
Epoch 10: val_loss improved from 0.50809 to 0.48974, saving model to
./models/unetpp.h5
38/38 [==============================] - 6s 143ms/step - loss: 0.4342 -
accuracy: 0.9091 - val_loss: 0.4897 - val_accuracy: 0.8111 - lr: 1.0000e-04
Epoch 11/100
38/38 [==============================] - ETA: 0s - loss: 0.3889 - accuracy:
0.9064
Epoch 11: val_loss improved from 0.48974 to 0.39200, saving model to
./models/unetpp.h5
38/38 [==============================] - 6s 143ms/step - loss: 0.3889 -
accuracy: 0.9064 - val_loss: 0.3920 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 12/100
38/38 [==============================] - ETA: 0s - loss: 0.4113 - accuracy:
0.8877
Epoch 12: val_loss did not improve from 0.39200
38/38 [==============================] - 4s 113ms/step - loss: 0.4113 -
accuracy: 0.8877 - val_loss: 0.4505 - val_accuracy: 0.8778 - lr: 1.0000e-04
Epoch 13/100
38/38 [==============================] - ETA: 0s - loss: 0.3902 - accuracy:
0.9091
Epoch 13: val_loss did not improve from 0.39200
38/38 [==============================] - 4s 118ms/step - loss: 0.3902 -
accuracy: 0.9091 - val_loss: 0.4038 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 14/100
38/38 [==============================] - ETA: 0s - loss: 0.3354 - accuracy:
```

```
0.9011
Epoch 14: val_loss improved from 0.39200 to 0.35025, saving model to
./models/unetpp.h5
38/38 [==============================] - 6s 145ms/step - loss: 0.3354 -
accuracy: 0.9011 - val_loss: 0.3502 - val_accuracy: 0.8778 - lr: 1.0000e-04
Epoch 15/100
38/38 [==============================] - ETA: 0s - loss: 0.3424 - accuracy:
0.9278
Epoch 15: val_loss did not improve from 0.35025
38/38 [==============================] - 4s 115ms/step - loss: 0.3424 -
accuracy: 0.9278 - val_loss: 0.3584 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 16/100
38/38 [==============================] - ETA: 0s - loss: 0.2569 - accuracy:
0.9465
Epoch 16: val_loss did not improve from 0.35025
38/38 [==============================] - 4s 113ms/step - loss: 0.2569 -
accuracy: 0.9465 - val_loss: 0.3984 - val_accuracy: 0.8889 - lr: 1.0000e-04
Epoch 17/100
38/38 [==============================] - ETA: 0s - loss: 0.2859 - accuracy:
0.9251
Epoch 17: val_loss did not improve from 0.35025
38/38 [==============================] - 4s 114ms/step - loss: 0.2859 -
accuracy: 0.9251 - val_loss: 0.3738 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 18/100
38/38 [==============================] - ETA: 0s - loss: 0.2323 - accuracy:
0.9572
Epoch 18: val_loss improved from 0.35025 to 0.29634, saving model to
./models/unetpp.h5
38/38 [==============================] - 5s 143ms/step - loss: 0.2323 -
accuracy: 0.9572 - val_loss: 0.2963 - val_accuracy: 0.9000 - lr: 1.0000e-04
Epoch 19/100
38/38 [==============================] - ETA: 0s - loss: 0.2620 - accuracy:
0.9545
Epoch 19: val_loss improved from 0.29634 to 0.27448, saving model to
./models/unetpp.h5
38/38 [==============================] - 6s 146ms/step - loss: 0.2620 -
accuracy: 0.9545 - val_loss: 0.2745 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 20/100
38/38 [==============================] - ETA: 0s - loss: 0.2566 - accuracy:
0.9385
Epoch 20: val_loss did not improve from 0.27448
38/38 [==============================] - 5s 117ms/step - loss: 0.2566 -
accuracy: 0.9385 - val_loss: 0.3840 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 21/100
38/38 [==============================] - ETA: 0s - loss: 0.2519 - accuracy:
0.9465
Epoch 21: val_loss improved from 0.27448 to 0.24982, saving model to
./models/unetpp.h5
```

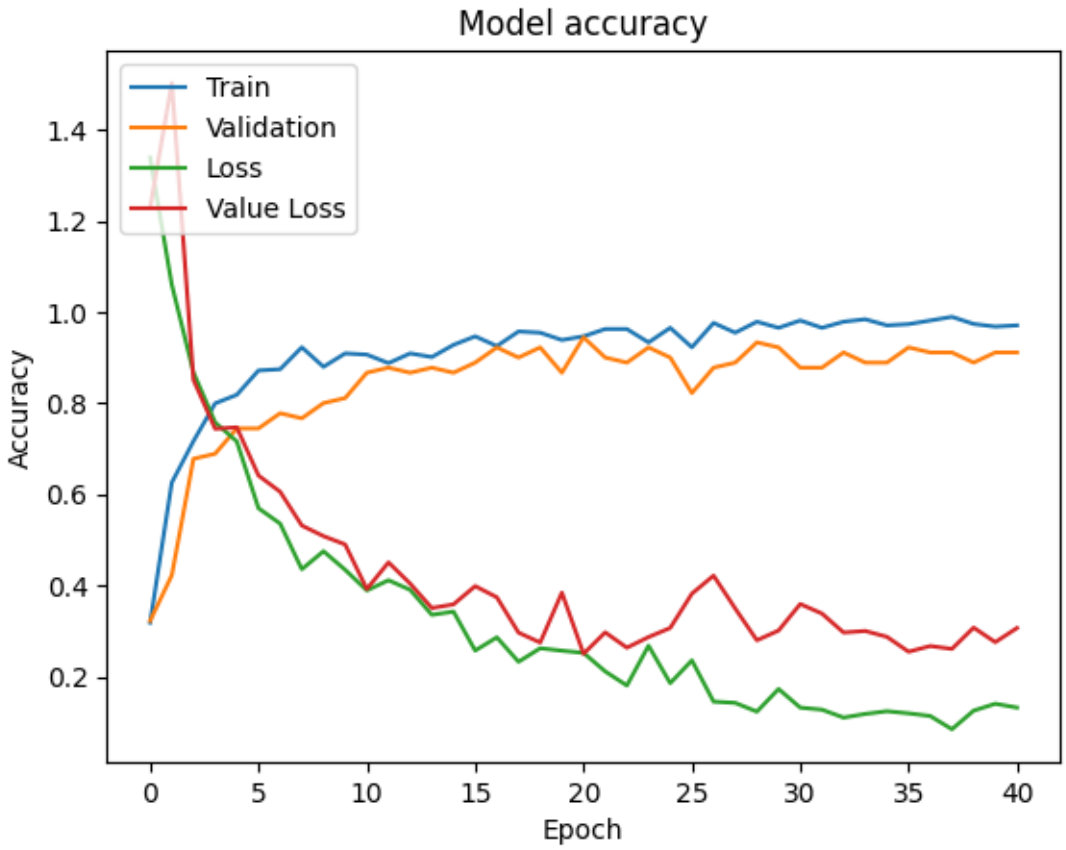
```
38/38 [==============================] - 6s 161ms/step - loss: 0.2519 -
accuracy: 0.9465 - val_loss: 0.2498 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 22/100
38/38 [==============================] - ETA: 0s - loss: 0.2112 - accuracy:
0.9626
Epoch 22: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 112ms/step - loss: 0.2112 -
accuracy: 0.9626 - val_loss: 0.2969 - val_accuracy: 0.9000 - lr: 1.0000e-04
Epoch 23/100
38/38 [==============================] - ETA: 0s - loss: 0.1802 - accuracy:
0.9626
Epoch 23: val_loss did not improve from 0.24982
38/38 [==============================] - 5s 119ms/step - loss: 0.1802 -
accuracy: 0.9626 - val_loss: 0.2632 - val_accuracy: 0.8889 - lr: 1.0000e-04
Epoch 24/100
38/38 [==============================] - ETA: 0s - loss: 0.2676 - accuracy:
0.9332
Epoch 24: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 116ms/step - loss: 0.2676 -
accuracy: 0.9332 - val_loss: 0.2866 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 25/100
38/38 [==============================] - ETA: 0s - loss: 0.1854 - accuracy:
0.9652
Epoch 25: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 114ms/step - loss: 0.1854 -
accuracy: 0.9652 - val_loss: 0.3066 - val_accuracy: 0.9000 - lr: 1.0000e-04
Epoch 26/100
38/38 [==============================] - ETA: 0s - loss: 0.2354 - accuracy:
0.9225
Epoch 26: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 116ms/step - loss: 0.2354 -
accuracy: 0.9225 - val_loss: 0.3818 - val_accuracy: 0.8222 - lr: 1.0000e-04
Epoch 27/100
38/38 [==============================] - ETA: 0s - loss: 0.1449 - accuracy:
0.9759
Epoch 27: val_loss did not improve from 0.24982
38/38 [==============================] - 5s 116ms/step - loss: 0.1449 -
accuracy: 0.9759 - val_loss: 0.4214 - val_accuracy: 0.8778 - lr: 1.0000e-04
Epoch 28/100
38/38 [==============================] - ETA: 0s - loss: 0.1421 - accuracy:
0.9545
Epoch 28: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 113ms/step - loss: 0.1421 -
accuracy: 0.9545 - val_loss: 0.3492 - val_accuracy: 0.8889 - lr: 1.0000e-04
Epoch 29/100
38/38 [==============================] - ETA: 0s - loss: 0.1229 - accuracy:
0.9786
Epoch 29: val_loss did not improve from 0.24982
```

```
38/38 [==============================] - 5s 117ms/step - loss: 0.1229 -
accuracy: 0.9786 - val_loss: 0.2795 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 30/100
38/38 [==============================] - ETA: 0s - loss: 0.1726 - accuracy:
0.9652
Epoch 30: val_loss did not improve from 0.24982
38/38 [==============================] - 5s 117ms/step - loss: 0.1726 -
accuracy: 0.9652 - val_loss: 0.3009 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 31/100
38/38 [==============================] - ETA: 0s - loss: 0.1317 - accuracy:
0.9813
Epoch 31: val_loss did not improve from 0.24982

Epoch 31: ReduceLROnPlateau reducing learning rate to 2.9999999242136255e-05.
38/38 [==============================] - 4s 114ms/step - loss: 0.1317 -
accuracy: 0.9813 - val_loss: 0.3591 - val_accuracy: 0.8778 - lr: 1.0000e-04
Epoch 32/100
38/38 [==============================] - ETA: 0s - loss: 0.1273 - accuracy:
0.9652
Epoch 32: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 116ms/step - loss: 0.1273 -
accuracy: 0.9652 - val_loss: 0.3382 - val_accuracy: 0.8778 - lr: 3.0000e-05
Epoch 33/100
38/38 [==============================] - ETA: 0s - loss: 0.1099 - accuracy:
0.9786
Epoch 33: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 116ms/step - loss: 0.1099 -
accuracy: 0.9786 - val_loss: 0.2966 - val_accuracy: 0.9111 - lr: 3.0000e-05
Epoch 34/100
38/38 [==============================] - ETA: 0s - loss: 0.1179 - accuracy:
0.9840
Epoch 34: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 114ms/step - loss: 0.1179 -
accuracy: 0.9840 - val_loss: 0.2999 - val_accuracy: 0.8889 - lr: 3.0000e-05
Epoch 35/100
38/38 [==============================] - ETA: 0s - loss: 0.1238 - accuracy:
0.9706
Epoch 35: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 114ms/step - loss: 0.1238 -
accuracy: 0.9706 - val_loss: 0.2871 - val_accuracy: 0.8889 - lr: 3.0000e-05
Epoch 36/100
38/38 [==============================] - ETA: 0s - loss: 0.1192 - accuracy:
0.9733
Epoch 36: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 114ms/step - loss: 0.1192 -
accuracy: 0.9733 - val_loss: 0.2544 - val_accuracy: 0.9222 - lr: 3.0000e-05
Epoch 37/100
38/38 [==============================] - ETA: 0s - loss: 0.1133 - accuracy:
```

```
0.9813
Epoch 37: val_loss did not improve from 0.24982
38/38 [==============================] - 5s 116ms/step - loss: 0.1133 -
accuracy: 0.9813 - val_loss: 0.2669 - val_accuracy: 0.9111 - lr: 3.0000e-05
Epoch 38/100
38/38 [==============================] - ETA: 0s - loss: 0.0843 - accuracy:
0.9893
Epoch 38: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 114ms/step - loss: 0.0843 -
accuracy: 0.9893 - val_loss: 0.2606 - val_accuracy: 0.9111 - lr: 3.0000e-05
Epoch 39/100
38/38 [==============================] - ETA: 0s - loss: 0.1250 - accuracy:
0.9737
Epoch 39: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 116ms/step - loss: 0.1250 -
accuracy: 0.9737 - val_loss: 0.3073 - val_accuracy: 0.8889 - lr: 3.0000e-05
Epoch 40/100
38/38 [==============================] - ETA: 0s - loss: 0.1402 - accuracy:
0.9679
Epoch 40: val_loss did not improve from 0.24982
38/38 [==============================] - 4s 112ms/step - loss: 0.1402 -
accuracy: 0.9679 - val_loss: 0.2754 - val_accuracy: 0.9111 - lr: 3.0000e-05
Epoch 41/100
38/38 [==============================] - ETA: 0s - loss: 0.1318 - accuracy:
0.9706
Epoch 41: val_loss did not improve from 0.24982

Epoch 41: ReduceLROnPlateau reducing learning rate to 8.999999772640877e-06.
38/38 [==============================] - 4s 115ms/step - loss: 0.1318 -
accuracy: 0.9706 - val_loss: 0.3064 - val_accuracy: 0.9111 - lr: 3.0000e-05
Epoch 41: early stopping
```

4 Testing the Model

[ ]:
```
# Learning curve
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])

# Loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])

plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation','Loss', 'Value Loss'], loc='upper left')
plt.show()
```



[ ]:
```
# Validate the model with test data

model = keras.models.load_model(model_file)
model.evaluate(test_generator)
```

```
# Predict and Display image using matplotlib
# plt.figure(figsize=(20, 20))
# for i in range(9):
#     plt.subplot(3, 3, i + 1)
#     for X_batch, Y_batch in test_generator:
#         image = X_batch[0]
#         # Print Class
#         plt.title("Predicted: " + classes[predictions[i]] + "\nActual: " + classes[np.argmax(Y_batch[i])])

#         plt.imshow(image)
#         break
```

```
10/10 [==============================] - 2s 23ms/step - loss: 0.2657 - accuracy: 0.9375
```

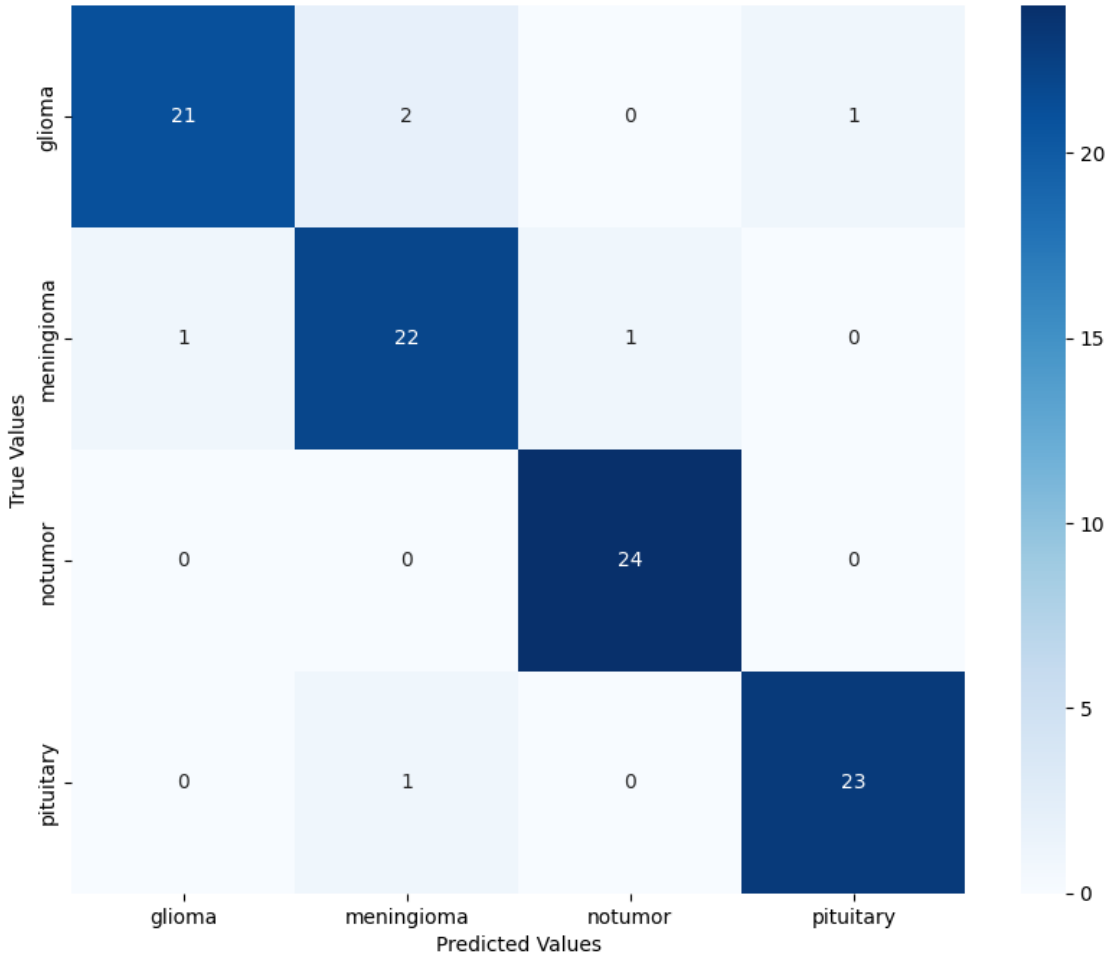
[ ]:
```
[0.26569023728370667, 0.9375]
```

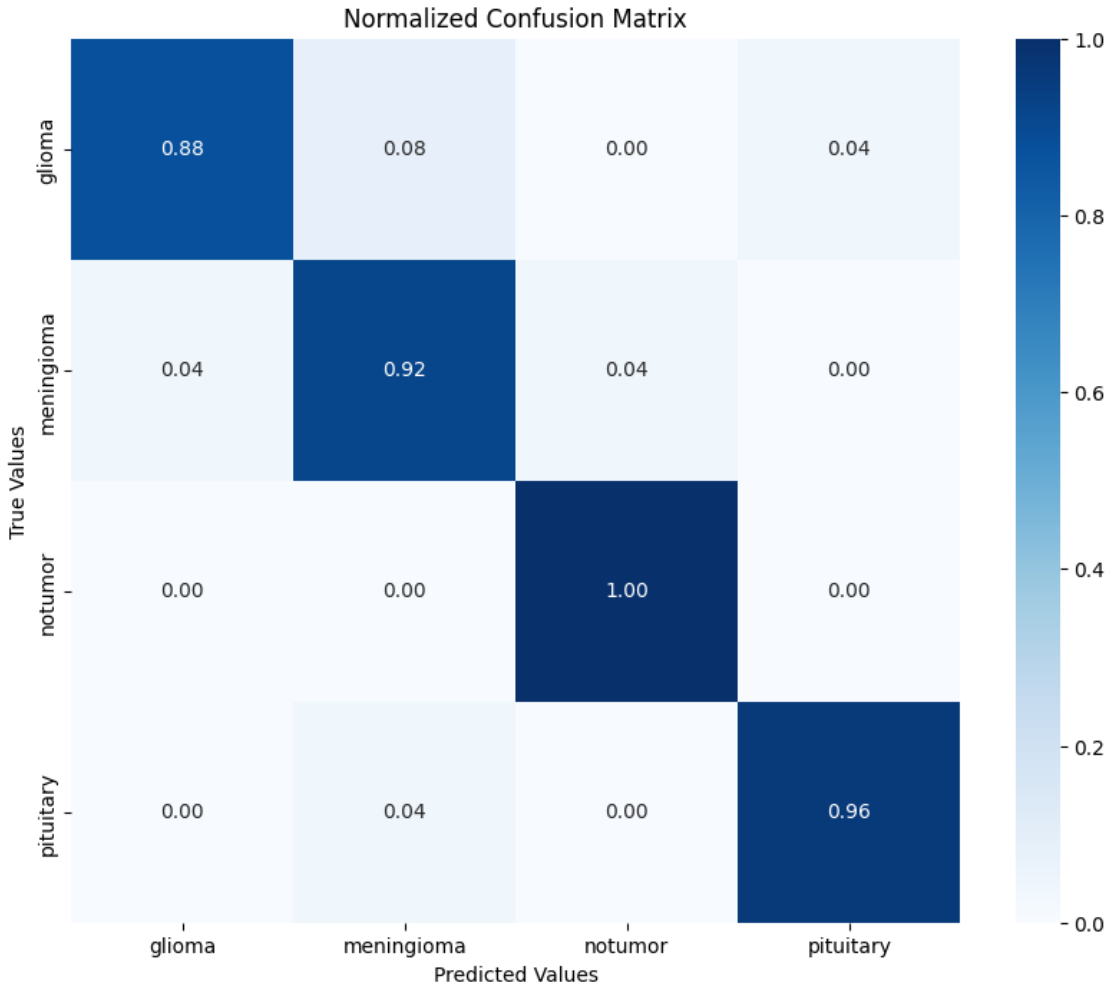
5 Model Visualisation (Evaluation)

[ ]:
```
import seaborn as sns
from sklearn.metrics import confusion_matrix, accuracy_score

classes = ['glioma', 'meningioma', 'notumor', 'pituitary']

def calculate_metrics(y_true, y_pred):
    # Confusion matrix
    cm = confusion_matrix(y_true, y_pred)

    # Plot the Confusion Matrix
    plt.figure(figsize=(10, 8))
    sns.heatmap(cm, annot=True, fmt='g', cmap='Blues', xticklabels=classes, yticklabels=classes)
    plt.xlabel('Predicted Values')
    plt.ylabel('True Values')
    plt.show()

    # Normalize the confusion matrix
    cm_normalized = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    # Plot the normalized confusion matrix
    plt.figure(figsize=(10, 8))
    sns.heatmap(cm_normalized, annot=True, fmt='.2f', cmap='Blues', xticklabels=classes, yticklabels=classes)
    plt.xlabel('Predicted Values')
    plt.ylabel('True Values')
```

```
    plt.title('Normalized Confusion Matrix')
    plt.show()

    # Calculate metrics for each class and average them
    dsc = np.mean([2.0 * cm[i, i] / (np.sum(cm[i, :]) + np.sum(cm[:, i])) for i␣
 ↪in range(cm.shape[0])])
    sensitivity = np.mean([cm[i, i] / np.sum(cm[i, :]) for i in range(cm.
 ↪shape[0])])
    specificity = np.mean([np.sum(np.delete(np.delete(cm, j, 0), j, 1)) / np.
 ↪sum(np.delete(cm, j, 0)) for j in range(cm.shape[0])])

    # Accuracy
    accuracy = accuracy_score(y_true, y_pred)

    return dsc, sensitivity, specificity, accuracy

# Predict the output
predictions_prob = model.predict(test_generator)
predictions = np.argmax(predictions_prob, axis=1)

dsc, sensitivity, specificity, accuracy = calculate_metrics(test_generator.
 ↪classes, predictions)
print(f"DSC: {dsc}, Sensitivity: {sensitivity}, Specificity: {specificity},␣
 ↪Accuracy: {accuracy}")
```

```
10/10 [==============================] - 2s 24ms/step
```

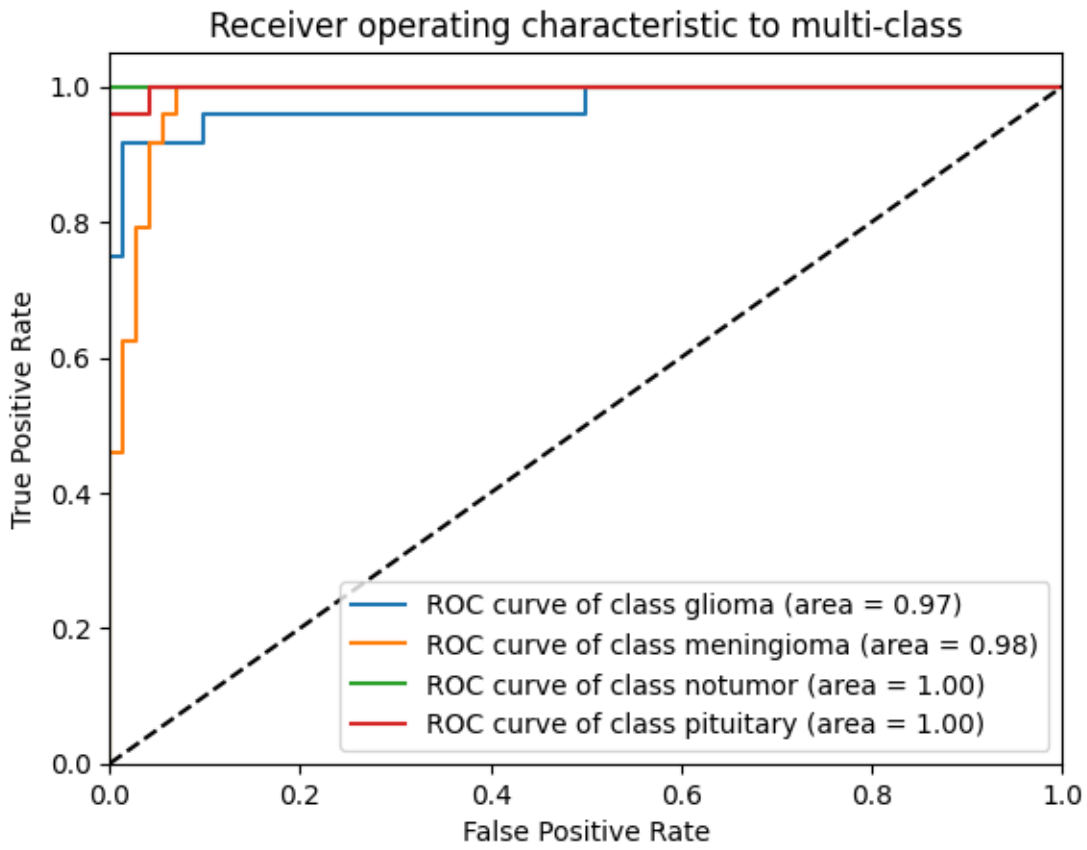

glioma
meningioma
notumor
pituitary
True Values
21
2
0
1
1
22
1
0
0
0
24
0
0
1
0
23
glioma
meningioma
notumor
pituitary
Predicted Values
20
15
10
5
0



DSC: 0.9372319580005916, Sensitivity: 0.9375, Specificity: 0.9791666666666667, Accuracy: 0.9375

[ ]:
```
from sklearn.metrics import roc_curve, auc, classification_report
from sklearn.preprocessing import LabelBinarizer
import matplotlib.pyplot as plt
import numpy as np

# Binarize the output
lb = LabelBinarizer()
y_test = lb.fit_transform(test_generator.classes)
y_pred = lb.transform(predictions)

# Compute ROC curve and ROC area for each class
fpr = dict()
tpr = dict()
```

```
roc_auc = dict()
for i in range(len(classes)):
    fpr[i], tpr[i], _ = roc_curve(y_test[:, i], predictions_prob[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Plot all ROC curves
plt.figure()
for i, class_name in enumerate(classes):
    plt.plot(fpr[i], tpr[i],
             label='ROC curve of class {0} (area = {1:0.2f})'
             ''.format(class_name, roc_auc[i]))

plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic to multi-class')
plt.legend(loc="lower right")
plt.show()

# Print classification report
print(classification_report(y_test, y_pred, target_names=classes))
```



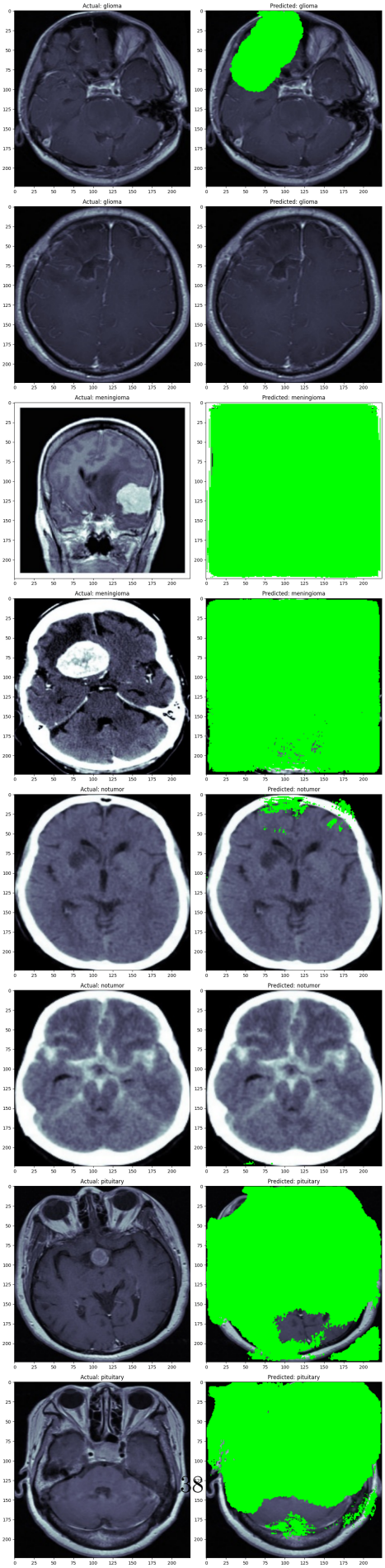
	precision	recall	f1-score	support
glioma	0.95	0.88	0.91	24
meningioma	0.88	0.92	0.90	24
notumor	0.96	1.00	0.98	24
pituitary	0.96	0.96	0.96	24
micro avg	0.94	0.94	0.94	96
macro avg	0.94	0.94	0.94	96
weighted avg	0.94	0.94	0.94	96
samples avg	0.94	0.94	0.94	96

[ ]:
```
import os
import numpy as np
import cv2
import matplotlib.pyplot as plt

# Define the class names
classes = ['glioma', 'meningioma', 'notumor', 'pituitary']
```

```
# Get predictions from the model
predictions_model_2 = model.layers[0].predict(test_generator)
predictions_prob = model.predict(test_generator)
predictions = np.argmax(predictions_prob, axis=1)

# Thresholding
threshold = 0.9  # Adjust threshold as needed for binary mask (if needed)
binary_mask = (predictions_model_2 > threshold).astype(np.uint8)

# Track displayed classes and their counts
displayed_classes_counts = {class_name: 0 for class_name in classes}

# Number of images to display per class
num_images_per_class = 2

# Prepare to display the images
plt.figure(figsize=(12, 6 * len(classes) * num_images_per_class))
image_index = 0

for i in range(len(test_generator.filenames)):
    # Get the actual class index and name
    actual_class_index = test_generator.classes[i]
    actual_class_name = classes[actual_class_index]

    # Check if we have already displayed the required number of images for this␣
 ↪class
    if displayed_classes_counts[actual_class_name] >= num_images_per_class:
        continue

    # Load original image
    img_path = os.path.join(dir, test_generator.filenames[i])
    original_image = cv2.imread(img_path)
    original_image_rgb = cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB)

    # Reshape binary mask to match original image shape
    reshaped_binary_mask = binary_mask[i][:, :, 0]

    # Overlay mask on original image
    overlay = original_image_rgb.copy()
    overlay[reshaped_binary_mask == 1] = (0, 255, 0)  # Overlay mask in green␣
 ↪color

    # Get predicted class name
    predicted_class_name = classes[predictions[i]]

    # Plot original image with overlayed mask
```

```
    plt.subplot(len(classes) * num_images_per_class, 2, 2 * image_index + 1)
    plt.imshow(original_image_rgb)
    plt.title(f'Actual: {actual_class_name}')

    plt.subplot(len(classes) * num_images_per_class, 2, 2 * image_index + 2)
    plt.imshow(overlay)
    plt.title(f'Predicted: {predicted_class_name}')

    # Mark this class as displayed
    displayed_classes_counts[actual_class_name] += 1
    image_index += 1

    # Break the loop if we've displayed the required number of images for all␣
 ↪classes
    if all(count >= num_images_per_class for count in displayed_classes_counts.
 ↪values()):
        break

plt.tight_layout()
plt.show()
```

```
10/10 [==============================] - 0s 22ms/step
10/10 [==============================] - 0s 24ms/step
```


Actual: glioma
Predicted: glioma
Actual: glioma
Predicted: glioma
Actual: meningioma
Predicted: meningioma
Actual: meningioma
Predicted: meningioma
Actual: notumor
Predicted: notumor
Actual: notumor
Predicted: notumor
Actual: pituitary
Predicted: pituitary
Actual: pituitary
Predicted: pituitary

6 Optimisation

[ ]:
```
!pip install optuna tensorflow-addons
```

```
Collecting optuna
  Downloading optuna-3.6.1-py3-none-any.whl (380 kB)
                             380.1/380.1

kB 8.2 MB/s eta 0:00:00
Collecting tensorflow-addons
  Downloading tensorflow_addons-0.23.0-cp310-cp310-manylinux_2_17_x86_64.manylin
ux2014_x86_64.whl (611 kB)
                             611.8/611.8

kB 43.2 MB/s eta 0:00:00
Collecting alembic>=1.5.0 (from optuna)
  Downloading alembic-1.13.1-py3-none-any.whl (233 kB)
                             233.4/233.4

kB 31.8 MB/s eta 0:00:00
Collecting colorlog (from optuna)
  Downloading colorlog-6.8.2-py3-none-any.whl (11 kB)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from optuna) (1.25.2)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from optuna) (24.0)
Requirement already satisfied: sqlalchemy>=1.3.0 in
/usr/local/lib/python3.10/dist-packages (from optuna) (2.0.30)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from optuna) (4.66.4)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages
(from optuna) (6.0.1)
Collecting typeguard<3.0.0,>=2.7 (from tensorflow-addons)
  Downloading typeguard-2.13.3-py3-none-any.whl (17 kB)
Collecting Mako (from alembic>=1.5.0->optuna)
  Downloading Mako-1.3.5-py3-none-any.whl (78 kB)
                             78.6/78.6 kB
13.1 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions>=4 in
/usr/local/lib/python3.10/dist-packages (from alembic>=1.5.0->optuna) (4.12.1)
Requirement already satisfied: greenlet!=0.4.17 in
/usr/local/lib/python3.10/dist-packages (from sqlalchemy>=1.3.0->optuna) (3.0.3)
Requirement already satisfied: MarkupSafe>=0.9.2 in
/usr/local/lib/python3.10/dist-packages (from Mako->alembic>=1.5.0->optuna)
(2.1.5)
Installing collected packages: typeguard, Mako, colorlog, tensorflow-addons,
```

```
alembic, optuna
Successfully installed Mako-1.3.5 alembic-1.13.1 colorlog-6.8.2 optuna-3.6.1
tensorflow-addons-0.23.0 typeguard-2.13.3
```

[ ]:
```python
import os
os.environ["SM_FRAMEWORK"] = "tf.keras"
import optuna
import tensorflow as tf
import tensorflow_addons as tfa
from tensorflow import keras
from segmentation_models import Unet
from tensorflow.keras.callbacks import ReduceLROnPlateau, ModelCheckpoint,␣
 ↪EarlyStopping

# Model parameters
input_shape = (224, 224, 3)
num_classes = 4
initial_learning_rate = 1e-4

def create_model(model):
    # Define the U-Net++ model
    model = Unet(model, input_shape=input_shape, classes=num_classes,␣
 ↪activation=None)

    # Add a GlobalAveragePooling2D layer to collapse spatial dimensions
    model = keras.Sequential([
        model,
        keras.layers.GlobalAveragePooling2D()  # Pool across spatial dimensions
    ])

    # Add a Dense layer with softmax activation for final classification
    model.add(keras.layers.Dense(num_classes, activation='softmax'))

    # Select optimizer
    optimizer = keras.optimizers.Adam(learning_rate=initial_learning_rate)

    # Compile the model
    model.compile(optimizer=optimizer,
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
    return model

def objective(trial):
    # Suggest Models (Other models other than efficientnet)
    base_model = trial.suggest_categorical('base_model', ['vgg16', 'resnet18',␣
 ↪'inceptionv3', 'mobilenetv2', 'densenet121'])
```

```
    # Create model
    model = create_model(base_model)

    # Callbacks
    reduce_lr = ReduceLROnPlateau(monitor='val_loss', patience=10, factor=0.3,␣
 ↪min_lr=1e-6, verbose=1)
    checkpoint = ModelCheckpoint(model_file, monitor='val_loss', verbose=1,␣
 ↪save_best_only=True)
    early_stopping = EarlyStopping(monitor='val_loss', min_delta=0,␣
 ↪patience=20, verbose=1, mode='auto')

    # Train the model
    history = model.fit(
        train_generator,
        steps_per_epoch=train_generator.samples // train_generator.batch_size,
        epochs=35,
        validation_data=test_generator,
        validation_steps=test_generator.samples // test_generator.batch_size,
        callbacks=[early_stopping, reduce_lr],
        verbose=1
    )

    # Return the best validation loss
    return min(history.history['val_loss'])

# Optuna study
study = optuna.create_study(direction='minimize')
study.optimize(objective, n_trials=5)

# Print the best trial
best_trial = study.best_trial
print(f"Best trial: {best_trial.value}")
print("Best parameters: ")
for key, value in best_trial.params.items():
    print(f"    {key}: {value}")
```

```
[I 2024-06-08 07:30:57,681] A new study created in memory with name: no-
name-143a6787-1abf-47aa-97ab-ea451ce02136

Downloading data from https://github.com/fchollet/deep-learning-
models/releases/download/v0.1/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [==============================] - 0s 0us/step
Epoch 1/35
38/38 [==============================] - 26s 256ms/step - loss: 1.2479 -
accuracy: 0.5027 - val_loss: 1.1361 - val_accuracy: 0.6111 - lr: 1.0000e-04
Epoch 2/35
38/38 [==============================] - 5s 122ms/step - loss: 1.0934 -
accuracy: 0.6203 - val_loss: 2.1893 - val_accuracy: 0.2333 - lr: 1.0000e-04
```

```
Epoch 3/35
38/38 [==============================] - 5s 122ms/step - loss: 0.9902 -
accuracy: 0.6551 - val_loss: 2.4874 - val_accuracy: 0.5000 - lr: 1.0000e-04
Epoch 4/35
38/38 [==============================] - 5s 122ms/step - loss: 0.9325 -
accuracy: 0.6551 - val_loss: 1.0617 - val_accuracy: 0.5444 - lr: 1.0000e-04
Epoch 5/35
38/38 [==============================] - 5s 123ms/step - loss: 0.8052 -
accuracy: 0.7112 - val_loss: 0.9246 - val_accuracy: 0.5889 - lr: 1.0000e-04
Epoch 6/35
38/38 [==============================] - 5s 127ms/step - loss: 0.8802 -
accuracy: 0.7059 - val_loss: 0.8703 - val_accuracy: 0.6333 - lr: 1.0000e-04
Epoch 7/35
38/38 [==============================] - 5s 122ms/step - loss: 0.7921 -
accuracy: 0.7166 - val_loss: 1.4683 - val_accuracy: 0.5222 - lr: 1.0000e-04
Epoch 8/35
38/38 [==============================] - 5s 123ms/step - loss: 0.7911 -
accuracy: 0.7166 - val_loss: 1.0678 - val_accuracy: 0.5667 - lr: 1.0000e-04
Epoch 9/35
38/38 [==============================] - 5s 124ms/step - loss: 0.7004 -
accuracy: 0.7579 - val_loss: 4.3393 - val_accuracy: 0.2667 - lr: 1.0000e-04
Epoch 10/35
38/38 [==============================] - 5s 122ms/step - loss: 0.6489 -
accuracy: 0.7781 - val_loss: 1.8751 - val_accuracy: 0.5000 - lr: 1.0000e-04
Epoch 11/35
38/38 [==============================] - 5s 124ms/step - loss: 0.5990 -
accuracy: 0.8075 - val_loss: 1.1523 - val_accuracy: 0.5333 - lr: 1.0000e-04
Epoch 12/35
38/38 [==============================] - 5s 123ms/step - loss: 0.6217 -
accuracy: 0.7868 - val_loss: 0.7008 - val_accuracy: 0.7444 - lr: 1.0000e-04
Epoch 13/35
38/38 [==============================] - 5s 121ms/step - loss: 0.5864 -
accuracy: 0.8048 - val_loss: 4.9328 - val_accuracy: 0.2333 - lr: 1.0000e-04
Epoch 14/35
38/38 [==============================] - 5s 121ms/step - loss: 0.5738 -
accuracy: 0.8316 - val_loss: 1.6377 - val_accuracy: 0.5111 - lr: 1.0000e-04
Epoch 15/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4975 -
accuracy: 0.8449 - val_loss: 1.8760 - val_accuracy: 0.3889 - lr: 1.0000e-04
Epoch 16/35
38/38 [==============================] - 5s 122ms/step - loss: 0.5835 -
accuracy: 0.7914 - val_loss: 0.8464 - val_accuracy: 0.6778 - lr: 1.0000e-04
Epoch 17/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4882 -
accuracy: 0.8422 - val_loss: 2.0833 - val_accuracy: 0.4889 - lr: 1.0000e-04
Epoch 18/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4904 -
accuracy: 0.8476 - val_loss: 1.0082 - val_accuracy: 0.6333 - lr: 1.0000e-04
```

```
Epoch 19/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4493 -
accuracy: 0.8770 - val_loss: 1.6683 - val_accuracy: 0.5111 - lr: 1.0000e-04
Epoch 20/35
38/38 [==============================] - 5s 121ms/step - loss: 0.3905 -
accuracy: 0.8850 - val_loss: 1.2608 - val_accuracy: 0.5556 - lr: 1.0000e-04
Epoch 21/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4665 -
accuracy: 0.8369 - val_loss: 1.4244 - val_accuracy: 0.5889 - lr: 1.0000e-04
Epoch 22/35
38/38 [==============================] - ETA: 0s - loss: 0.3893 - accuracy:
0.8711
Epoch 22: ReduceLROnPlateau reducing learning rate to 2.9999999242136255e-05.
38/38 [==============================] - 5s 123ms/step - loss: 0.3893 -
accuracy: 0.8711 - val_loss: 1.3803 - val_accuracy: 0.5556 - lr: 1.0000e-04
Epoch 23/35
38/38 [==============================] - 5s 121ms/step - loss: 0.2936 -
accuracy: 0.9332 - val_loss: 0.9165 - val_accuracy: 0.6778 - lr: 3.0000e-05
Epoch 24/35
38/38 [==============================] - 5s 121ms/step - loss: 0.2832 -
accuracy: 0.9251 - val_loss: 0.5906 - val_accuracy: 0.7889 - lr: 3.0000e-05
Epoch 25/35
38/38 [==============================] - 5s 121ms/step - loss: 0.3174 -
accuracy: 0.9198 - val_loss: 1.4536 - val_accuracy: 0.5667 - lr: 3.0000e-05
Epoch 26/35
38/38 [==============================] - 5s 121ms/step - loss: 0.2810 -
accuracy: 0.9332 - val_loss: 0.8005 - val_accuracy: 0.7556 - lr: 3.0000e-05
Epoch 27/35
38/38 [==============================] - 5s 121ms/step - loss: 0.2956 -
accuracy: 0.9358 - val_loss: 0.6852 - val_accuracy: 0.7556 - lr: 3.0000e-05
Epoch 28/35
38/38 [==============================] - 5s 125ms/step - loss: 0.2742 -
accuracy: 0.9332 - val_loss: 0.6176 - val_accuracy: 0.7778 - lr: 3.0000e-05
Epoch 29/35
38/38 [==============================] - 5s 122ms/step - loss: 0.2695 -
accuracy: 0.9305 - val_loss: 0.9909 - val_accuracy: 0.6556 - lr: 3.0000e-05
Epoch 30/35
38/38 [==============================] - 5s 123ms/step - loss: 0.2644 -
accuracy: 0.9289 - val_loss: 0.7274 - val_accuracy: 0.7000 - lr: 3.0000e-05
Epoch 31/35
38/38 [==============================] - 5s 121ms/step - loss: 0.2689 -
accuracy: 0.9251 - val_loss: 0.8863 - val_accuracy: 0.6556 - lr: 3.0000e-05
Epoch 32/35
38/38 [==============================] - 5s 124ms/step - loss: 0.2754 -
accuracy: 0.9251 - val_loss: 0.5812 - val_accuracy: 0.7444 - lr: 3.0000e-05
Epoch 33/35
38/38 [==============================] - 5s 121ms/step - loss: 0.2049 -
accuracy: 0.9519 - val_loss: 0.6373 - val_accuracy: 0.7556 - lr: 3.0000e-05
```

```
Epoch 34/35
38/38 [==============================] - 5s 123ms/step - loss: 0.1686 -
accuracy: 0.9842 - val_loss: 0.8351 - val_accuracy: 0.7222 - lr: 3.0000e-05
Epoch 35/35
38/38 [==============================] - 5s 121ms/step - loss: 0.1973 -
accuracy: 0.9492 - val_loss: 0.4694 - val_accuracy: 0.8222 - lr: 3.0000e-05

[I 2024-06-08 07:34:06,960] Trial 0 finished with value: 0.469392865896225 and
parameters: {'base_model': 'vgg16'}. Best is trial 0 with value:
0.469392865896225.

Epoch 1/35
38/38 [==============================] - 38s 202ms/step - loss: 1.2545 -
accuracy: 0.4037 - val_loss: 1.3064 - val_accuracy: 0.3444 - lr: 1.0000e-04
Epoch 2/35
38/38 [==============================] - 4s 114ms/step - loss: 0.9793 -
accuracy: 0.6604 - val_loss: 1.3408 - val_accuracy: 0.4222 - lr: 1.0000e-04
Epoch 3/35
38/38 [==============================] - 4s 113ms/step - loss: 0.8129 -
accuracy: 0.8102 - val_loss: 1.5538 - val_accuracy: 0.4111 - lr: 1.0000e-04
Epoch 4/35
38/38 [==============================] - 4s 115ms/step - loss: 0.6518 -
accuracy: 0.8658 - val_loss: 1.1854 - val_accuracy: 0.5778 - lr: 1.0000e-04
Epoch 5/35
38/38 [==============================] - 4s 115ms/step - loss: 0.5665 -
accuracy: 0.8957 - val_loss: 1.1837 - val_accuracy: 0.5667 - lr: 1.0000e-04
Epoch 6/35
38/38 [==============================] - 4s 113ms/step - loss: 0.4694 -
accuracy: 0.9198 - val_loss: 1.0173 - val_accuracy: 0.6444 - lr: 1.0000e-04
Epoch 7/35
38/38 [==============================] - 4s 114ms/step - loss: 0.4298 -
accuracy: 0.9144 - val_loss: 0.8966 - val_accuracy: 0.6667 - lr: 1.0000e-04
Epoch 8/35
38/38 [==============================] - 4s 115ms/step - loss: 0.4539 -
accuracy: 0.8904 - val_loss: 1.0604 - val_accuracy: 0.6667 - lr: 1.0000e-04
Epoch 9/35
38/38 [==============================] - 4s 112ms/step - loss: 0.4025 -
accuracy: 0.9037 - val_loss: 0.9204 - val_accuracy: 0.6889 - lr: 1.0000e-04
Epoch 10/35
38/38 [==============================] - 4s 114ms/step - loss: 0.4044 -
accuracy: 0.9198 - val_loss: 0.7372 - val_accuracy: 0.7444 - lr: 1.0000e-04
Epoch 11/35
38/38 [==============================] - 4s 114ms/step - loss: 0.3815 -
accuracy: 0.9198 - val_loss: 0.7379 - val_accuracy: 0.7111 - lr: 1.0000e-04
Epoch 12/35
38/38 [==============================] - 4s 112ms/step - loss: 0.3428 -
accuracy: 0.9358 - val_loss: 0.7277 - val_accuracy: 0.6333 - lr: 1.0000e-04
Epoch 13/35
38/38 [==============================] - 4s 114ms/step - loss: 0.3608 -
```

```
accuracy: 0.9064 - val_loss: 0.6543 - val_accuracy: 0.7556 - lr: 1.0000e-04
Epoch 14/35
38/38 [==============================] - 4s 114ms/step - loss: 0.2679 -
accuracy: 0.9358 - val_loss: 0.7929 - val_accuracy: 0.6444 - lr: 1.0000e-04
Epoch 15/35
38/38 [==============================] - 4s 112ms/step - loss: 0.3150 -
accuracy: 0.9251 - val_loss: 0.9718 - val_accuracy: 0.7000 - lr: 1.0000e-04
Epoch 16/35
38/38 [==============================] - 4s 114ms/step - loss: 0.2238 -
accuracy: 0.9679 - val_loss: 1.1633 - val_accuracy: 0.6222 - lr: 1.0000e-04
Epoch 17/35
38/38 [==============================] - 4s 113ms/step - loss: 0.2804 -
accuracy: 0.9332 - val_loss: 0.8878 - val_accuracy: 0.6444 - lr: 1.0000e-04
Epoch 18/35
38/38 [==============================] - 4s 109ms/step - loss: 0.2666 -
accuracy: 0.9492 - val_loss: 1.1751 - val_accuracy: 0.5889 - lr: 1.0000e-04
Epoch 19/35
38/38 [==============================] - 4s 112ms/step - loss: 0.2417 -
accuracy: 0.9492 - val_loss: 0.9593 - val_accuracy: 0.7111 - lr: 1.0000e-04
Epoch 20/35
38/38 [==============================] - 4s 115ms/step - loss: 0.2773 -
accuracy: 0.9225 - val_loss: 0.7719 - val_accuracy: 0.7556 - lr: 1.0000e-04
Epoch 21/35
38/38 [==============================] - 4s 112ms/step - loss: 0.1797 -
accuracy: 0.9652 - val_loss: 0.8172 - val_accuracy: 0.6667 - lr: 1.0000e-04
Epoch 22/35
38/38 [==============================] - 4s 116ms/step - loss: 0.1790 -
accuracy: 0.9679 - val_loss: 0.8162 - val_accuracy: 0.6444 - lr: 1.0000e-04
Epoch 23/35
38/38 [==============================] - ETA: 0s - loss: 0.1512 - accuracy:
0.9759
Epoch 23: ReduceLROnPlateau reducing learning rate to 2.9999999242136255e-05.
38/38 [==============================] - 4s 112ms/step - loss: 0.1512 -
accuracy: 0.9759 - val_loss: 1.0564 - val_accuracy: 0.6556 - lr: 1.0000e-04
Epoch 24/35
38/38 [==============================] - 4s 113ms/step - loss: 0.1532 -
accuracy: 0.9733 - val_loss: 0.8402 - val_accuracy: 0.7333 - lr: 3.0000e-05
Epoch 25/35
38/38 [==============================] - 4s 115ms/step - loss: 0.1532 -
accuracy: 0.9733 - val_loss: 0.7125 - val_accuracy: 0.7667 - lr: 3.0000e-05
Epoch 26/35
38/38 [==============================] - 4s 113ms/step - loss: 0.1683 -
accuracy: 0.9733 - val_loss: 0.7961 - val_accuracy: 0.7000 - lr: 3.0000e-05
Epoch 27/35
38/38 [==============================] - 4s 114ms/step - loss: 0.1365 -
accuracy: 0.9733 - val_loss: 0.6971 - val_accuracy: 0.7556 - lr: 3.0000e-05
Epoch 28/35
38/38 [==============================] - 4s 113ms/step - loss: 0.1598 -
```

```
accuracy: 0.9706 - val_loss: 0.6604 - val_accuracy: 0.7333 - lr: 3.0000e-05
Epoch 29/35
38/38 [==============================] - 4s 112ms/step - loss: 0.1077 -
accuracy: 0.9893 - val_loss: 0.6181 - val_accuracy: 0.8000 - lr: 3.0000e-05
Epoch 30/35
38/38 [==============================] - 4s 112ms/step - loss: 0.1163 -
accuracy: 0.9786 - val_loss: 0.5885 - val_accuracy: 0.8111 - lr: 3.0000e-05
Epoch 31/35
38/38 [==============================] - 4s 115ms/step - loss: 0.1992 -
accuracy: 0.9545 - val_loss: 0.4978 - val_accuracy: 0.8333 - lr: 3.0000e-05
Epoch 32/35
38/38 [==============================] - 4s 112ms/step - loss: 0.1338 -
accuracy: 0.9733 - val_loss: 0.4801 - val_accuracy: 0.8222 - lr: 3.0000e-05
Epoch 33/35
38/38 [==============================] - 4s 112ms/step - loss: 0.1836 -
accuracy: 0.9519 - val_loss: 0.4891 - val_accuracy: 0.8444 - lr: 3.0000e-05
Epoch 34/35
38/38 [==============================] - 4s 115ms/step - loss: 0.1639 -
accuracy: 0.9706 - val_loss: 0.5248 - val_accuracy: 0.8556 - lr: 3.0000e-05
Epoch 35/35
38/38 [==============================] - 4s 112ms/step - loss: 0.1088 -
accuracy: 0.9840 - val_loss: 0.4158 - val_accuracy: 0.8222 - lr: 3.0000e-05

[I 2024-06-08 07:37:16,971] Trial 1 finished with value: 0.41584599018096924 and
parameters: {'base_model': 'mobilenetv2'}. Best is trial 1 with value:
0.41584599018096924.

Downloading data from https://github.com/keras-team/keras-applications/releases/
download/densenet/densenet121_weights_tf_dim_ordering_tf_kernels_notop.h5
29084464/29084464 [==============================] - 0s 0us/step
Epoch 1/35
38/38 [==============================] - 85s 353ms/step - loss: 1.1730 -
accuracy: 0.4385 - val_loss: 1.4873 - val_accuracy: 0.2667 - lr: 1.0000e-04
Epoch 2/35
38/38 [==============================] - 5s 118ms/step - loss: 0.8247 -
accuracy: 0.8048 - val_loss: 1.0241 - val_accuracy: 0.6444 - lr: 1.0000e-04
Epoch 3/35
38/38 [==============================] - 5s 118ms/step - loss: 0.6565 -
accuracy: 0.8770 - val_loss: 1.0412 - val_accuracy: 0.6222 - lr: 1.0000e-04
Epoch 4/35
38/38 [==============================] - 5s 123ms/step - loss: 0.5418 -
accuracy: 0.8984 - val_loss: 0.8066 - val_accuracy: 0.6889 - lr: 1.0000e-04
Epoch 5/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4723 -
accuracy: 0.8930 - val_loss: 0.8194 - val_accuracy: 0.7000 - lr: 1.0000e-04
Epoch 6/35
38/38 [==============================] - 5s 118ms/step - loss: 0.4503 -
accuracy: 0.8984 - val_loss: 0.7051 - val_accuracy: 0.7111 - lr: 1.0000e-04
Epoch 7/35
```

```
38/38 [==============================] - 5s 120ms/step - loss: 0.4099 -
accuracy: 0.9064 - val_loss: 0.5722 - val_accuracy: 0.8222 - lr: 1.0000e-04
Epoch 8/35
38/38 [==============================] - 5s 121ms/step - loss: 0.3902 -
accuracy: 0.9144 - val_loss: 0.6681 - val_accuracy: 0.7778 - lr: 1.0000e-04
Epoch 9/35
38/38 [==============================] - 5s 120ms/step - loss: 0.3552 -
accuracy: 0.9278 - val_loss: 0.9685 - val_accuracy: 0.7000 - lr: 1.0000e-04
Epoch 10/35
38/38 [==============================] - 5s 120ms/step - loss: 0.3182 -
accuracy: 0.9305 - val_loss: 1.0614 - val_accuracy: 0.6556 - lr: 1.0000e-04
Epoch 11/35
38/38 [==============================] - 5s 118ms/step - loss: 0.2989 -
accuracy: 0.9439 - val_loss: 0.6674 - val_accuracy: 0.8111 - lr: 1.0000e-04
Epoch 12/35
38/38 [==============================] - 5s 122ms/step - loss: 0.3686 -
accuracy: 0.9064 - val_loss: 0.7870 - val_accuracy: 0.7667 - lr: 1.0000e-04
Epoch 13/35
38/38 [==============================] - 5s 118ms/step - loss: 0.2687 -
accuracy: 0.9572 - val_loss: 0.7009 - val_accuracy: 0.7333 - lr: 1.0000e-04
Epoch 14/35
38/38 [==============================] - 5s 120ms/step - loss: 0.2634 -
accuracy: 0.9421 - val_loss: 0.5048 - val_accuracy: 0.8444 - lr: 1.0000e-04
Epoch 15/35
38/38 [==============================] - 5s 120ms/step - loss: 0.2416 -
accuracy: 0.9519 - val_loss: 0.5281 - val_accuracy: 0.8333 - lr: 1.0000e-04
Epoch 16/35
38/38 [==============================] - 5s 117ms/step - loss: 0.2019 -
accuracy: 0.9679 - val_loss: 0.3697 - val_accuracy: 0.8333 - lr: 1.0000e-04
Epoch 17/35
38/38 [==============================] - 5s 121ms/step - loss: 0.1909 -
accuracy: 0.9626 - val_loss: 0.4074 - val_accuracy: 0.8333 - lr: 1.0000e-04
Epoch 18/35
38/38 [==============================] - 5s 119ms/step - loss: 0.2971 -
accuracy: 0.9278 - val_loss: 0.6125 - val_accuracy: 0.7889 - lr: 1.0000e-04
Epoch 19/35
38/38 [==============================] - 5s 118ms/step - loss: 0.1943 -
accuracy: 0.9572 - val_loss: 0.4634 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 20/35
38/38 [==============================] - 5s 120ms/step - loss: 0.2005 -
accuracy: 0.9492 - val_loss: 0.6485 - val_accuracy: 0.7556 - lr: 1.0000e-04
Epoch 21/35
38/38 [==============================] - 5s 118ms/step - loss: 0.2077 -
accuracy: 0.9385 - val_loss: 0.4219 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 22/35
38/38 [==============================] - 5s 117ms/step - loss: 0.1929 -
accuracy: 0.9519 - val_loss: 0.5491 - val_accuracy: 0.8444 - lr: 1.0000e-04
Epoch 23/35
```

```
38/38 [==============================] - 5s 122ms/step - loss: 0.1676 -
accuracy: 0.9706 - val_loss: 0.5475 - val_accuracy: 0.8333 - lr: 1.0000e-04
Epoch 24/35
38/38 [==============================] - 5s 116ms/step - loss: 0.1625 -
accuracy: 0.9679 - val_loss: 0.4974 - val_accuracy: 0.8444 - lr: 1.0000e-04
Epoch 25/35
38/38 [==============================] - 5s 117ms/step - loss: 0.1912 -
accuracy: 0.9465 - val_loss: 0.7879 - val_accuracy: 0.7778 - lr: 1.0000e-04
Epoch 26/35
38/38 [==============================] - ETA: 0s - loss: 0.2266 - accuracy:
0.9385
Epoch 26: ReduceLROnPlateau reducing learning rate to 2.9999999242136255e-05.
38/38 [==============================] - 5s 119ms/step - loss: 0.2266 -
accuracy: 0.9385 - val_loss: 0.4998 - val_accuracy: 0.8222 - lr: 1.0000e-04
Epoch 27/35
38/38 [==============================] - 4s 115ms/step - loss: 0.1415 -
accuracy: 0.9813 - val_loss: 0.3684 - val_accuracy: 0.8556 - lr: 3.0000e-05
Epoch 28/35
38/38 [==============================] - 5s 118ms/step - loss: 0.1545 -
accuracy: 0.9545 - val_loss: 0.4192 - val_accuracy: 0.8667 - lr: 3.0000e-05
Epoch 29/35
38/38 [==============================] - 5s 116ms/step - loss: 0.0932 -
accuracy: 0.9920 - val_loss: 0.3869 - val_accuracy: 0.8556 - lr: 3.0000e-05
Epoch 30/35
38/38 [==============================] - 5s 117ms/step - loss: 0.1145 -
accuracy: 0.9786 - val_loss: 0.4019 - val_accuracy: 0.8556 - lr: 3.0000e-05
Epoch 31/35
38/38 [==============================] - 5s 119ms/step - loss: 0.1027 -
accuracy: 0.9840 - val_loss: 0.4002 - val_accuracy: 0.8556 - lr: 3.0000e-05
Epoch 32/35
38/38 [==============================] - 5s 121ms/step - loss: 0.0795 -
accuracy: 0.9893 - val_loss: 0.3828 - val_accuracy: 0.8556 - lr: 3.0000e-05
Epoch 33/35
38/38 [==============================] - 5s 117ms/step - loss: 0.0866 -
accuracy: 0.9947 - val_loss: 0.4234 - val_accuracy: 0.8778 - lr: 3.0000e-05
Epoch 34/35
38/38 [==============================] - 5s 119ms/step - loss: 0.0713 -
accuracy: 0.9920 - val_loss: 0.4155 - val_accuracy: 0.8667 - lr: 3.0000e-05
Epoch 35/35
38/38 [==============================] - 5s 119ms/step - loss: 0.0949 -
accuracy: 0.9866 - val_loss: 0.3806 - val_accuracy: 0.8667 - lr: 3.0000e-05

[I 2024-06-08 07:41:26,039] Trial 2 finished with value: 0.3683975338935852 and
parameters: {'base_model': 'densenet121'}. Best is trial 2 with value:
0.3683975338935852.

Epoch 1/35
38/38 [==============================] - 33s 140ms/step - loss: 1.1957 -
accuracy: 0.4519 - val_loss: 1.5635 - val_accuracy: 0.4222 - lr: 1.0000e-04
```

```
Epoch 2/35
38/38 [==============================] - 4s 113ms/step - loss: 0.8267 -
accuracy: 0.6979 - val_loss: 1.5192 - val_accuracy: 0.5333 - lr: 1.0000e-04
Epoch 3/35
38/38 [==============================] - 4s 117ms/step - loss: 0.7468 -
accuracy: 0.7273 - val_loss: 1.4432 - val_accuracy: 0.5111 - lr: 1.0000e-04
Epoch 4/35
38/38 [==============================] - 4s 115ms/step - loss: 0.6681 -
accuracy: 0.7701 - val_loss: 3.0160 - val_accuracy: 0.4000 - lr: 1.0000e-04
Epoch 5/35
38/38 [==============================] - 4s 112ms/step - loss: 0.6152 -
accuracy: 0.7674 - val_loss: 3.0779 - val_accuracy: 0.3667 - lr: 1.0000e-04
Epoch 6/35
38/38 [==============================] - 4s 114ms/step - loss: 0.5390 -
accuracy: 0.7727 - val_loss: 1.4322 - val_accuracy: 0.5667 - lr: 1.0000e-04
Epoch 7/35
38/38 [==============================] - 4s 113ms/step - loss: 0.5392 -
accuracy: 0.8235 - val_loss: 1.3116 - val_accuracy: 0.5000 - lr: 1.0000e-04
Epoch 8/35
38/38 [==============================] - 4s 113ms/step - loss: 0.5382 -
accuracy: 0.8155 - val_loss: 0.9010 - val_accuracy: 0.6222 - lr: 1.0000e-04
Epoch 9/35
38/38 [==============================] - 4s 116ms/step - loss: 0.5396 -
accuracy: 0.8316 - val_loss: 0.8812 - val_accuracy: 0.6444 - lr: 1.0000e-04
Epoch 10/35
38/38 [==============================] - 4s 112ms/step - loss: 0.5130 -
accuracy: 0.8850 - val_loss: 0.8966 - val_accuracy: 0.5778 - lr: 1.0000e-04
Epoch 11/35
38/38 [==============================] - 4s 113ms/step - loss: 0.4526 -
accuracy: 0.9198 - val_loss: 0.7296 - val_accuracy: 0.6667 - lr: 1.0000e-04
Epoch 12/35
38/38 [==============================] - 4s 114ms/step - loss: 0.4087 -
accuracy: 0.9251 - val_loss: 0.8254 - val_accuracy: 0.6444 - lr: 1.0000e-04
Epoch 13/35
38/38 [==============================] - 4s 114ms/step - loss: 0.4322 -
accuracy: 0.8984 - val_loss: 1.1473 - val_accuracy: 0.6444 - lr: 1.0000e-04
Epoch 14/35
38/38 [==============================] - 4s 113ms/step - loss: 0.3553 -
accuracy: 0.9332 - val_loss: 1.6578 - val_accuracy: 0.5444 - lr: 1.0000e-04
Epoch 15/35
38/38 [==============================] - 4s 112ms/step - loss: 0.4244 -
accuracy: 0.8957 - val_loss: 1.7018 - val_accuracy: 0.6000 - lr: 1.0000e-04
Epoch 16/35
38/38 [==============================] - 4s 112ms/step - loss: 0.4186 -
accuracy: 0.8984 - val_loss: 1.1005 - val_accuracy: 0.6889 - lr: 1.0000e-04
Epoch 17/35
38/38 [==============================] - 4s 115ms/step - loss: 0.3422 -
accuracy: 0.9358 - val_loss: 0.7901 - val_accuracy: 0.6889 - lr: 1.0000e-04
```

```
Epoch 18/35
38/38 [==============================] - 4s 113ms/step - loss: 0.3899 -
accuracy: 0.9144 - val_loss: 0.8187 - val_accuracy: 0.7222 - lr: 1.0000e-04
Epoch 19/35
38/38 [==============================] - 4s 114ms/step - loss: 0.3233 -
accuracy: 0.9492 - val_loss: 0.9699 - val_accuracy: 0.7111 - lr: 1.0000e-04
Epoch 20/35
38/38 [==============================] - 4s 113ms/step - loss: 0.2812 -
accuracy: 0.9492 - val_loss: 0.7039 - val_accuracy: 0.7667 - lr: 1.0000e-04
Epoch 21/35
38/38 [==============================] - 4s 111ms/step - loss: 0.2747 -
accuracy: 0.9599 - val_loss: 0.8337 - val_accuracy: 0.6667 - lr: 1.0000e-04
Epoch 22/35
38/38 [==============================] - 4s 115ms/step - loss: 0.2946 -
accuracy: 0.9332 - val_loss: 0.7322 - val_accuracy: 0.7889 - lr: 1.0000e-04
Epoch 23/35
38/38 [==============================] - 4s 114ms/step - loss: 0.2811 -
accuracy: 0.9385 - val_loss: 0.9786 - val_accuracy: 0.7556 - lr: 1.0000e-04
Epoch 24/35
38/38 [==============================] - 4s 110ms/step - loss: 0.2957 -
accuracy: 0.9332 - val_loss: 0.7824 - val_accuracy: 0.7889 - lr: 1.0000e-04
Epoch 25/35
38/38 [==============================] - 4s 115ms/step - loss: 0.2666 -
accuracy: 0.9278 - val_loss: 1.1794 - val_accuracy: 0.7000 - lr: 1.0000e-04
Epoch 26/35
38/38 [==============================] - 4s 113ms/step - loss: 0.2284 -
accuracy: 0.9439 - val_loss: 1.2853 - val_accuracy: 0.7222 - lr: 1.0000e-04
Epoch 27/35
38/38 [==============================] - 4s 110ms/step - loss: 0.1914 -
accuracy: 0.9706 - val_loss: 0.8329 - val_accuracy: 0.8000 - lr: 1.0000e-04
Epoch 28/35
38/38 [==============================] - 4s 113ms/step - loss: 0.2541 -
accuracy: 0.9385 - val_loss: 1.4328 - val_accuracy: 0.7222 - lr: 1.0000e-04
Epoch 29/35
38/38 [==============================] - 4s 116ms/step - loss: 0.2371 -
accuracy: 0.9412 - val_loss: 1.2576 - val_accuracy: 0.7333 - lr: 1.0000e-04
Epoch 30/35
38/38 [==============================] - ETA: 0s - loss: 0.2174 - accuracy:
0.9599
Epoch 30: ReduceLROnPlateau reducing learning rate to 2.9999999242136255e-05.
38/38 [==============================] - 4s 114ms/step - loss: 0.2174 -
accuracy: 0.9599 - val_loss: 0.9748 - val_accuracy: 0.7111 - lr: 1.0000e-04
Epoch 31/35
38/38 [==============================] - 4s 114ms/step - loss: 0.2281 -
accuracy: 0.9385 - val_loss: 0.9225 - val_accuracy: 0.7333 - lr: 3.0000e-05
Epoch 32/35
38/38 [==============================] - 4s 114ms/step - loss: 0.1681 -
accuracy: 0.9786 - val_loss: 0.8718 - val_accuracy: 0.7667 - lr: 3.0000e-05
```

```
Epoch 33/35
38/38 [==============================] - 4s 110ms/step - loss: 0.1569 -
accuracy: 0.9786 - val_loss: 0.8626 - val_accuracy: 0.7778 - lr: 3.0000e-05
Epoch 34/35
38/38 [==============================] - 4s 115ms/step - loss: 0.1791 -
accuracy: 0.9706 - val_loss: 0.8848 - val_accuracy: 0.7556 - lr: 3.0000e-05
Epoch 35/35
38/38 [==============================] - 4s 115ms/step - loss: 0.1407 -
accuracy: 0.9893 - val_loss: 0.9530 - val_accuracy: 0.7556 - lr: 3.0000e-05

[I 2024-06-08 07:44:29,789] Trial 3 finished with value: 0.7039403319358826 and
parameters: {'base_model': 'mobilenetv2'}. Best is trial 2 with value:
0.3683975338935852.

Epoch 1/35
38/38 [==============================] - 14s 134ms/step - loss: 1.3049 -
accuracy: 0.3904 - val_loss: 2.2623 - val_accuracy: 0.2667 - lr: 1.0000e-04
Epoch 2/35
38/38 [==============================] - 5s 122ms/step - loss: 1.1220 -
accuracy: 0.5348 - val_loss: 1.1507 - val_accuracy: 0.4444 - lr: 1.0000e-04
Epoch 3/35
38/38 [==============================] - 5s 124ms/step - loss: 1.0215 -
accuracy: 0.5763 - val_loss: 10.0723 - val_accuracy: 0.2667 - lr: 1.0000e-04
Epoch 4/35
38/38 [==============================] - 5s 122ms/step - loss: 1.0006 -
accuracy: 0.5561 - val_loss: 7.5805 - val_accuracy: 0.2667 - lr: 1.0000e-04
Epoch 5/35
38/38 [==============================] - 5s 123ms/step - loss: 0.9584 -
accuracy: 0.6070 - val_loss: 1.5792 - val_accuracy: 0.3556 - lr: 1.0000e-04
Epoch 6/35
38/38 [==============================] - 5s 123ms/step - loss: 0.8687 -
accuracy: 0.6283 - val_loss: 3.0405 - val_accuracy: 0.4333 - lr: 1.0000e-04
Epoch 7/35
38/38 [==============================] - 5s 122ms/step - loss: 0.8583 -
accuracy: 0.6444 - val_loss: 1.1220 - val_accuracy: 0.5444 - lr: 1.0000e-04
Epoch 8/35
38/38 [==============================] - 5s 123ms/step - loss: 0.8557 -
accuracy: 0.6310 - val_loss: 0.8945 - val_accuracy: 0.6556 - lr: 1.0000e-04
Epoch 9/35
38/38 [==============================] - 5s 122ms/step - loss: 0.8404 -
accuracy: 0.6444 - val_loss: 1.8309 - val_accuracy: 0.3444 - lr: 1.0000e-04
Epoch 10/35
38/38 [==============================] - 5s 122ms/step - loss: 0.7703 -
accuracy: 0.6551 - val_loss: 1.2670 - val_accuracy: 0.5333 - lr: 1.0000e-04
Epoch 11/35
38/38 [==============================] - 5s 121ms/step - loss: 0.7678 -
accuracy: 0.6337 - val_loss: 1.5618 - val_accuracy: 0.4889 - lr: 1.0000e-04
Epoch 12/35
38/38 [==============================] - 5s 121ms/step - loss: 0.8115 -
```

```
accuracy: 0.6417 - val_loss: 1.2568 - val_accuracy: 0.5000 - lr: 1.0000e-04
Epoch 13/35
38/38 [==============================] - 5s 125ms/step - loss: 0.7349 -
accuracy: 0.6658 - val_loss: 1.0434 - val_accuracy: 0.5667 - lr: 1.0000e-04
Epoch 14/35
38/38 [==============================] - 5s 120ms/step - loss: 0.7412 -
accuracy: 0.6604 - val_loss: 1.6460 - val_accuracy: 0.4000 - lr: 1.0000e-04
Epoch 15/35
38/38 [==============================] - 5s 122ms/step - loss: 0.6299 -
accuracy: 0.7193 - val_loss: 0.7757 - val_accuracy: 0.6556 - lr: 1.0000e-04
Epoch 16/35
38/38 [==============================] - 5s 122ms/step - loss: 0.6985 -
accuracy: 0.6524 - val_loss: 0.7236 - val_accuracy: 0.6889 - lr: 1.0000e-04
Epoch 17/35
38/38 [==============================] - 5s 124ms/step - loss: 0.6618 -
accuracy: 0.7032 - val_loss: 1.7551 - val_accuracy: 0.4444 - lr: 1.0000e-04
Epoch 18/35
38/38 [==============================] - 5s 122ms/step - loss: 0.6671 -
accuracy: 0.6816 - val_loss: 1.0085 - val_accuracy: 0.5889 - lr: 1.0000e-04
Epoch 19/35
38/38 [==============================] - 5s 120ms/step - loss: 0.6137 -
accuracy: 0.7246 - val_loss: 0.8230 - val_accuracy: 0.6778 - lr: 1.0000e-04
Epoch 20/35
38/38 [==============================] - 5s 121ms/step - loss: 0.5857 -
accuracy: 0.7299 - val_loss: 0.8931 - val_accuracy: 0.6000 - lr: 1.0000e-04
Epoch 21/35
38/38 [==============================] - 5s 122ms/step - loss: 0.6615 -
accuracy: 0.6898 - val_loss: 0.9484 - val_accuracy: 0.6333 - lr: 1.0000e-04
Epoch 22/35
38/38 [==============================] - 5s 123ms/step - loss: 0.5502 -
accuracy: 0.7193 - val_loss: 0.9260 - val_accuracy: 0.5778 - lr: 1.0000e-04
Epoch 23/35
38/38 [==============================] - 5s 121ms/step - loss: 0.6351 -
accuracy: 0.7246 - val_loss: 0.7004 - val_accuracy: 0.6778 - lr: 1.0000e-04
Epoch 24/35
38/38 [==============================] - 5s 121ms/step - loss: 0.6037 -
accuracy: 0.7139 - val_loss: 0.9531 - val_accuracy: 0.6556 - lr: 1.0000e-04
Epoch 25/35
38/38 [==============================] - 5s 121ms/step - loss: 0.5587 -
accuracy: 0.7540 - val_loss: 1.4842 - val_accuracy: 0.5111 - lr: 1.0000e-04
Epoch 26/35
38/38 [==============================] - 5s 121ms/step - loss: 0.5518 -
accuracy: 0.7433 - val_loss: 0.9693 - val_accuracy: 0.6444 - lr: 1.0000e-04
Epoch 27/35
38/38 [==============================] - 5s 121ms/step - loss: 0.5460 -
accuracy: 0.7701 - val_loss: 0.7202 - val_accuracy: 0.6778 - lr: 1.0000e-04
Epoch 28/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4633 -
```

```
accuracy: 0.7834 - val_loss: 1.0733 - val_accuracy: 0.6333 - lr: 1.0000e-04
Epoch 29/35
38/38 [==============================] - 5s 122ms/step - loss: 0.4821 -
accuracy: 0.7781 - val_loss: 0.8798 - val_accuracy: 0.7444 - lr: 1.0000e-04
Epoch 30/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4823 -
accuracy: 0.7995 - val_loss: 0.7515 - val_accuracy: 0.7667 - lr: 1.0000e-04
Epoch 31/35
38/38 [==============================] - 5s 121ms/step - loss: 0.5264 -
accuracy: 0.7727 - val_loss: 0.7730 - val_accuracy: 0.7444 - lr: 1.0000e-04
Epoch 32/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4960 -
accuracy: 0.7647 - val_loss: 1.0305 - val_accuracy: 0.6556 - lr: 1.0000e-04
Epoch 33/35
38/38 [==============================] - ETA: 0s - loss: 0.4634 - accuracy:
0.8342
Epoch 33: ReduceLROnPlateau reducing learning rate to 2.9999999242136255e-05.
38/38 [==============================] - 5s 121ms/step - loss: 0.4634 -
accuracy: 0.8342 - val_loss: 0.7953 - val_accuracy: 0.6667 - lr: 1.0000e-04
Epoch 34/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4334 -
accuracy: 0.8396 - val_loss: 0.7137 - val_accuracy: 0.7000 - lr: 3.0000e-05
Epoch 35/35
38/38 [==============================] - 5s 121ms/step - loss: 0.4282 -
accuracy: 0.8422 - val_loss: 0.7402 - val_accuracy: 0.8222 - lr: 3.0000e-05

[I 2024-06-08 07:47:26,167] Trial 4 finished with value: 0.700380265712738 and
parameters: {'base_model': 'vgg16'}. Best is trial 2 with value:
0.3683975338935852.

Best trial: 0.3683975338935852
Best parameters:
    base_model: densenet121
```

7 K-Folds Validation

[ ]:
```python
import os
import numpy as np
import pandas as pd
from sklearn.model_selection import KFold
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from segmentation_models import Unet
from tensorflow import keras
from tensorflow.keras.callbacks import ReduceLROnPlateau, ModelCheckpoint,␣
 ↪EarlyStopping

# Set parameters
dir = 'dataset_19'  # Update this to your dataset directory
```

```
batch_size = 10
input_shape = (224, 224, 3)
num_classes = 4
initial_learning_rate = 1e-4
k = 5

# Prepare data generators
datagen = ImageDataGenerator(
    rescale=1. / 255,
    horizontal_flip=True,
    rotation_range=20
)

# List all images and labels
all_images = []
all_labels = []

for class_index, class_name in enumerate(os.listdir(dir)):
    class_dir = os.path.join(dir, class_name)
    for image_name in os.listdir(class_dir):
        all_images.append(os.path.join(class_dir, image_name))
        all_labels.append(str(class_index))  # Convert class index to string

# Convert lists to numpy arrays
all_images = np.array(all_images)
all_labels = np.array(all_labels)

# Define K-Fold cross-validation
kf = KFold(n_splits=k, shuffle=True, random_state=42)

# Initialize lists to store metrics
fold_accuracies = []
fold_losses = []

# Iterate over each fold
for fold, (train_index, val_index) in enumerate(kf.split(all_images)):
    print(f'Fold {fold + 1}/{k}')

    train_images, val_images = all_images[train_index], all_images[val_index]
    train_labels, val_labels = all_labels[train_index], all_labels[val_index]

    # Create DataFrames for training and validation data
    train_df = pd.DataFrame({'filename': train_images, 'class': train_labels})
    val_df = pd.DataFrame({'filename': val_images, 'class': val_labels})

    # Create training and validation data generators
    train_generator = datagen.flow_from_dataframe(
```

```
        dataframe=train_df,
        x_col='filename',
        y_col='class',
        target_size=(224, 224),
        batch_size=batch_size,
        class_mode='categorical',
        shuffle=True
    )

    validation_generator = datagen.flow_from_dataframe(
        dataframe=val_df,
        x_col='filename',
        y_col='class',
        target_size=(224, 224),
        batch_size=batch_size,
        class_mode='categorical',
        shuffle=False
    )

    # Define the U-Net++ model
    model = Unet('efficientnetb1', input_shape=input_shape,␣
↪classes=num_classes, activation=None)

    # Add a GlobalAveragePooling2D layer and Dense layer with softmax activation
    model = keras.Sequential([
        model,
        keras.layers.GlobalAveragePooling2D(),  # Pool across spatial dimensions
        keras.layers.Dense(num_classes, activation='softmax')  # Final␣
↪classification layer
    ])

    # Compile the model
    optimizer = keras.optimizers.Adam(learning_rate=initial_learning_rate)
    model.compile(optimizer=optimizer, loss='categorical_crossentropy',␣
↪metrics=['accuracy'])

    # Callbacks
    reduce_lr = ReduceLROnPlateau(monitor='val_loss', patience=10, factor=0.3,␣
↪min_lr=1e-6, verbose=1)
    # checkpoint = ModelCheckpoint(f'model_fold_{fold + 1}.h5',␣
↪monitor='val_loss', verbose=1, save_best_only=True)
    early_stopping = EarlyStopping(monitor='val_loss', patience=15, verbose=1,␣
↪mode='auto')

    # Train the model
    history = model.fit(
```

```
        train_generator,
        steps_per_epoch=train_generator.samples // train_generator.batch_size,
        epochs=35,
        validation_data=validation_generator,
        validation_steps=validation_generator.samples // validation_generator.
 ↪batch_size,
        callbacks=[early_stopping, reduce_lr]
    )

    # Store the best validation accuracy and loss for the current fold
    best_val_accuracy = max(history.history['val_accuracy'])
    best_val_loss = min(history.history['val_loss'])
    fold_accuracies.append(best_val_accuracy)
    fold_losses.append(best_val_loss)

# Calculate mean and standard deviation of accuracies and losses
mean_accuracy = np.mean(fold_accuracies)
std_accuracy = np.std(fold_accuracies)
mean_loss = np.mean(fold_losses)
std_loss = np.std(fold_losses)

print(f'Validation Accuracy: {mean_accuracy:.4f} ± {std_accuracy:.4f}')
print(f'Validation Loss: {mean_loss:.4f} ± {std_loss:.4f}')
```

```
Fold 1/5
Found 384 validated image filenames belonging to 4 classes.
Found 96 validated image filenames belonging to 4 classes.
Epoch 1/35
38/38 [==============================] - 57s 190ms/step - loss: 1.2399 -
accuracy: 0.3984 - val_loss: 1.8823 - val_accuracy: 0.2111 - lr: 1.0000e-04
Epoch 2/35
38/38 [==============================] - 5s 136ms/step - loss: 0.9369 -
accuracy: 0.6471 - val_loss: 1.6458 - val_accuracy: 0.3778 - lr: 1.0000e-04
Epoch 3/35
38/38 [==============================] - 5s 135ms/step - loss: 0.7928 -
accuracy: 0.7166 - val_loss: 2.7122 - val_accuracy: 0.2111 - lr: 1.0000e-04
Epoch 4/35
38/38 [==============================] - 5s 139ms/step - loss: 0.6554 -
accuracy: 0.7727 - val_loss: 2.0891 - val_accuracy: 0.3778 - lr: 1.0000e-04
Epoch 5/35
38/38 [==============================] - 5s 134ms/step - loss: 0.5753 -
accuracy: 0.8182 - val_loss: 1.1983 - val_accuracy: 0.5667 - lr: 1.0000e-04
Epoch 6/35
38/38 [==============================] - 5s 139ms/step - loss: 0.5400 -
accuracy: 0.8209 - val_loss: 1.3590 - val_accuracy: 0.5000 - lr: 1.0000e-04
Epoch 7/35
38/38 [==============================] - 5s 135ms/step - loss: 0.4551 -
accuracy: 0.8930 - val_loss: 1.1837 - val_accuracy: 0.5444 - lr: 1.0000e-04
```

```
Epoch 8/35
38/38 [==============================] - 5s 135ms/step - loss: 0.4439 -
accuracy: 0.8930 - val_loss: 1.1023 - val_accuracy: 0.5778 - lr: 1.0000e-04
Epoch 9/35
38/38 [==============================] - 5s 138ms/step - loss: 0.3773 -
accuracy: 0.9278 - val_loss: 0.5592 - val_accuracy: 0.7556 - lr: 1.0000e-04
Epoch 10/35
38/38 [==============================] - 5s 140ms/step - loss: 0.3848 -
accuracy: 0.9198 - val_loss: 0.7113 - val_accuracy: 0.7444 - lr: 1.0000e-04
Epoch 11/35
38/38 [==============================] - 5s 138ms/step - loss: 0.3590 -
accuracy: 0.9091 - val_loss: 0.6510 - val_accuracy: 0.7667 - lr: 1.0000e-04
Epoch 12/35
38/38 [==============================] - 5s 140ms/step - loss: 0.3546 -
accuracy: 0.8930 - val_loss: 0.5668 - val_accuracy: 0.8111 - lr: 1.0000e-04
Epoch 13/35
38/38 [==============================] - 5s 138ms/step - loss: 0.3032 -
accuracy: 0.9225 - val_loss: 0.5003 - val_accuracy: 0.8333 - lr: 1.0000e-04
Epoch 14/35
38/38 [==============================] - 5s 136ms/step - loss: 0.3266 -
accuracy: 0.9171 - val_loss: 0.4488 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 15/35
38/38 [==============================] - 5s 137ms/step - loss: 0.3337 -
accuracy: 0.9144 - val_loss: 0.5258 - val_accuracy: 0.8889 - lr: 1.0000e-04
Epoch 16/35
38/38 [==============================] - 5s 134ms/step - loss: 0.2583 -
accuracy: 0.9412 - val_loss: 0.4446 - val_accuracy: 0.8778 - lr: 1.0000e-04
Epoch 17/35
38/38 [==============================] - 5s 138ms/step - loss: 0.2600 -
accuracy: 0.9412 - val_loss: 0.3909 - val_accuracy: 0.8778 - lr: 1.0000e-04
Epoch 18/35
38/38 [==============================] - 5s 138ms/step - loss: 0.2657 -
accuracy: 0.9278 - val_loss: 0.4241 - val_accuracy: 0.8556 - lr: 1.0000e-04
Epoch 19/35
38/38 [==============================] - 5s 136ms/step - loss: 0.2117 -
accuracy: 0.9679 - val_loss: 0.4133 - val_accuracy: 0.8778 - lr: 1.0000e-04
Epoch 20/35
38/38 [==============================] - 5s 138ms/step - loss: 0.2542 -
accuracy: 0.9412 - val_loss: 0.4431 - val_accuracy: 0.8556 - lr: 1.0000e-04
Epoch 21/35
38/38 [==============================] - 5s 136ms/step - loss: 0.2442 -
accuracy: 0.9439 - val_loss: 0.3850 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 22/35
38/38 [==============================] - 5s 135ms/step - loss: 0.1966 -
accuracy: 0.9572 - val_loss: 0.4043 - val_accuracy: 0.8889 - lr: 1.0000e-04
Epoch 23/35
38/38 [==============================] - 5s 135ms/step - loss: 0.1758 -
accuracy: 0.9626 - val_loss: 0.4189 - val_accuracy: 0.8778 - lr: 1.0000e-04
```

```
Epoch 24/35
38/38 [==============================] - 5s 131ms/step - loss: 0.2093 -
accuracy: 0.9332 - val_loss: 0.3785 - val_accuracy: 0.8889 - lr: 1.0000e-04
Epoch 25/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1869 -
accuracy: 0.9652 - val_loss: 0.5038 - val_accuracy: 0.8222 - lr: 1.0000e-04
Epoch 26/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1755 -
accuracy: 0.9759 - val_loss: 0.3520 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 27/35
38/38 [==============================] - 5s 140ms/step - loss: 0.2165 -
accuracy: 0.9519 - val_loss: 0.3945 - val_accuracy: 0.8889 - lr: 1.0000e-04
Epoch 28/35
38/38 [==============================] - 5s 135ms/step - loss: 0.1674 -
accuracy: 0.9572 - val_loss: 0.3791 - val_accuracy: 0.8778 - lr: 1.0000e-04
Epoch 29/35
38/38 [==============================] - 5s 139ms/step - loss: 0.1617 -
accuracy: 0.9632 - val_loss: 0.4668 - val_accuracy: 0.8556 - lr: 1.0000e-04
Epoch 30/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1703 -
accuracy: 0.9519 - val_loss: 0.4105 - val_accuracy: 0.8556 - lr: 1.0000e-04
Epoch 31/35
38/38 [==============================] - 5s 134ms/step - loss: 0.1206 -
accuracy: 0.9733 - val_loss: 0.4047 - val_accuracy: 0.8778 - lr: 1.0000e-04
Epoch 32/35
38/38 [==============================] - 5s 139ms/step - loss: 0.1894 -
accuracy: 0.9599 - val_loss: 0.4956 - val_accuracy: 0.8333 - lr: 1.0000e-04
Epoch 33/35
38/38 [==============================] - 5s 140ms/step - loss: 0.1501 -
accuracy: 0.9652 - val_loss: 0.4337 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 34/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1333 -
accuracy: 0.9733 - val_loss: 0.4001 - val_accuracy: 0.8778 - lr: 1.0000e-04
Epoch 35/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1218 -
accuracy: 0.9786 - val_loss: 0.2722 - val_accuracy: 0.9000 - lr: 1.0000e-04
Fold 2/5
Found 384 validated image filenames belonging to 4 classes.
Found 96 validated image filenames belonging to 4 classes.
Epoch 1/35
38/38 [==============================] - 58s 192ms/step - loss: 1.2892 -
accuracy: 0.3128 - val_loss: 1.3893 - val_accuracy: 0.1778 - lr: 1.0000e-04
Epoch 2/35
38/38 [==============================] - 5s 138ms/step - loss: 0.8810 -
accuracy: 0.7684 - val_loss: 1.0304 - val_accuracy: 0.6222 - lr: 1.0000e-04
Epoch 3/35
38/38 [==============================] - 5s 137ms/step - loss: 0.6350 -
accuracy: 0.8476 - val_loss: 1.1861 - val_accuracy: 0.4778 - lr: 1.0000e-04
```

```
Epoch 4/35
38/38 [==============================] - 5s 136ms/step - loss: 0.5703 -
accuracy: 0.8369 - val_loss: 0.8186 - val_accuracy: 0.7000 - lr: 1.0000e-04
Epoch 5/35
38/38 [==============================] - 5s 138ms/step - loss: 0.4640 -
accuracy: 0.8824 - val_loss: 0.7069 - val_accuracy: 0.6889 - lr: 1.0000e-04
Epoch 6/35
38/38 [==============================] - 5s 136ms/step - loss: 0.4226 -
accuracy: 0.8984 - val_loss: 1.0807 - val_accuracy: 0.6333 - lr: 1.0000e-04
Epoch 7/35
38/38 [==============================] - 5s 139ms/step - loss: 0.4159 -
accuracy: 0.8797 - val_loss: 0.5211 - val_accuracy: 0.8000 - lr: 1.0000e-04
Epoch 8/35
38/38 [==============================] - 5s 139ms/step - loss: 0.2990 -
accuracy: 0.9358 - val_loss: 0.3450 - val_accuracy: 0.8333 - lr: 1.0000e-04
Epoch 9/35
38/38 [==============================] - 5s 135ms/step - loss: 0.2885 -
accuracy: 0.9171 - val_loss: 0.2715 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 10/35
38/38 [==============================] - 5s 139ms/step - loss: 0.3251 -
accuracy: 0.9064 - val_loss: 0.2755 - val_accuracy: 0.9111 - lr: 1.0000e-04
Epoch 11/35
38/38 [==============================] - 5s 139ms/step - loss: 0.2877 -
accuracy: 0.9519 - val_loss: 0.3364 - val_accuracy: 0.8556 - lr: 1.0000e-04
Epoch 12/35
38/38 [==============================] - 5s 140ms/step - loss: 0.2332 -
accuracy: 0.9492 - val_loss: 0.2494 - val_accuracy: 0.9000 - lr: 1.0000e-04
Epoch 13/35
38/38 [==============================] - 5s 136ms/step - loss: 0.2922 -
accuracy: 0.9251 - val_loss: 0.4312 - val_accuracy: 0.8333 - lr: 1.0000e-04
Epoch 14/35
38/38 [==============================] - 5s 138ms/step - loss: 0.2159 -
accuracy: 0.9599 - val_loss: 0.2044 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 15/35
38/38 [==============================] - 5s 137ms/step - loss: 0.2291 -
accuracy: 0.9332 - val_loss: 0.2290 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 16/35
38/38 [==============================] - 5s 135ms/step - loss: 0.2077 -
accuracy: 0.9465 - val_loss: 0.3648 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 17/35
38/38 [==============================] - 5s 138ms/step - loss: 0.1758 -
accuracy: 0.9652 - val_loss: 0.1405 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 18/35
38/38 [==============================] - 5s 135ms/step - loss: 0.1763 -
accuracy: 0.9545 - val_loss: 0.3239 - val_accuracy: 0.8889 - lr: 1.0000e-04
Epoch 19/35
38/38 [==============================] - 5s 138ms/step - loss: 0.2455 -
accuracy: 0.9251 - val_loss: 0.5013 - val_accuracy: 0.8556 - lr: 1.0000e-04
```

```
Epoch 20/35
38/38 [==============================] - 5s 135ms/step - loss: 0.2114 -
accuracy: 0.9439 - val_loss: 0.2086 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 21/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1898 -
accuracy: 0.9545 - val_loss: 0.1746 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 22/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1446 -
accuracy: 0.9786 - val_loss: 0.2200 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 23/35
38/38 [==============================] - 5s 134ms/step - loss: 0.1895 -
accuracy: 0.9492 - val_loss: 0.1859 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 24/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1595 -
accuracy: 0.9599 - val_loss: 0.1230 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 25/35
38/38 [==============================] - 5s 132ms/step - loss: 0.1253 -
accuracy: 0.9733 - val_loss: 0.2291 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 26/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1259 -
accuracy: 0.9706 - val_loss: 0.1250 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 27/35
38/38 [==============================] - 5s 140ms/step - loss: 0.1143 -
accuracy: 0.9813 - val_loss: 0.2505 - val_accuracy: 0.9111 - lr: 1.0000e-04
Epoch 28/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1351 -
accuracy: 0.9652 - val_loss: 0.2583 - val_accuracy: 0.9111 - lr: 1.0000e-04
Epoch 29/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1293 -
accuracy: 0.9679 - val_loss: 0.2453 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 30/35
38/38 [==============================] - 5s 134ms/step - loss: 0.1371 -
accuracy: 0.9759 - val_loss: 0.2666 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 31/35
38/38 [==============================] - 5s 138ms/step - loss: 0.0910 -
accuracy: 0.9866 - val_loss: 0.2009 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 32/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1388 -
accuracy: 0.9652 - val_loss: 0.2769 - val_accuracy: 0.9000 - lr: 1.0000e-04
Epoch 33/35
38/38 [==============================] - 5s 139ms/step - loss: 0.1176 -
accuracy: 0.9759 - val_loss: 0.1460 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 34/35
38/38 [==============================] - ETA: 0s - loss: 0.1194 - accuracy:
0.9733
Epoch 34: ReduceLROnPlateau reducing learning rate to 2.9999999242136255e-05.
38/38 [==============================] - 5s 138ms/step - loss: 0.1194 -
accuracy: 0.9733 - val_loss: 0.2064 - val_accuracy: 0.9111 - lr: 1.0000e-04
```

```
Epoch 35/35
38/38 [==============================] - 5s 136ms/step - loss: 0.0850 -
accuracy: 0.9813 - val_loss: 0.1733 - val_accuracy: 0.9556 - lr: 3.0000e-05
Fold 3/5
Found 384 validated image filenames belonging to 4 classes.
Found 96 validated image filenames belonging to 4 classes.
Epoch 1/35
38/38 [==============================] - 57s 195ms/step - loss: 1.2982 -
accuracy: 0.3663 - val_loss: 1.3627 - val_accuracy: 0.3889 - lr: 1.0000e-04
Epoch 2/35
38/38 [==============================] - 5s 140ms/step - loss: 1.0139 -
accuracy: 0.5882 - val_loss: 1.5761 - val_accuracy: 0.3222 - lr: 1.0000e-04
Epoch 3/35
38/38 [==============================] - 6s 149ms/step - loss: 0.7978 -
accuracy: 0.7861 - val_loss: 1.4460 - val_accuracy: 0.3556 - lr: 1.0000e-04
Epoch 4/35
38/38 [==============================] - 5s 141ms/step - loss: 0.6344 -
accuracy: 0.8610 - val_loss: 1.9368 - val_accuracy: 0.4111 - lr: 1.0000e-04
Epoch 5/35
38/38 [==============================] - 6s 143ms/step - loss: 0.6087 -
accuracy: 0.8369 - val_loss: 1.7242 - val_accuracy: 0.4667 - lr: 1.0000e-04
Epoch 6/35
38/38 [==============================] - 5s 139ms/step - loss: 0.5309 -
accuracy: 0.8824 - val_loss: 1.5913 - val_accuracy: 0.5000 - lr: 1.0000e-04
Epoch 7/35
38/38 [==============================] - 5s 137ms/step - loss: 0.4240 -
accuracy: 0.9064 - val_loss: 1.0169 - val_accuracy: 0.6000 - lr: 1.0000e-04
Epoch 8/35
38/38 [==============================] - 5s 136ms/step - loss: 0.3966 -
accuracy: 0.9332 - val_loss: 0.6869 - val_accuracy: 0.7667 - lr: 1.0000e-04
Epoch 9/35
38/38 [==============================] - 5s 140ms/step - loss: 0.3895 -
accuracy: 0.9144 - val_loss: 0.6433 - val_accuracy: 0.8000 - lr: 1.0000e-04
Epoch 10/35
38/38 [==============================] - 5s 134ms/step - loss: 0.3121 -
accuracy: 0.9492 - val_loss: 0.3834 - val_accuracy: 0.8667 - lr: 1.0000e-04
Epoch 11/35
38/38 [==============================] - 5s 137ms/step - loss: 0.3847 -
accuracy: 0.8957 - val_loss: 0.3918 - val_accuracy: 0.9000 - lr: 1.0000e-04
Epoch 12/35
38/38 [==============================] - 5s 134ms/step - loss: 0.3125 -
accuracy: 0.9358 - val_loss: 0.2694 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 13/35
38/38 [==============================] - 5s 137ms/step - loss: 0.3319 -
accuracy: 0.9144 - val_loss: 0.3977 - val_accuracy: 0.8444 - lr: 1.0000e-04
Epoch 14/35
38/38 [==============================] - 5s 136ms/step - loss: 0.3497 -
accuracy: 0.9053 - val_loss: 0.2863 - val_accuracy: 0.9333 - lr: 1.0000e-04
```

```
Epoch 15/35
38/38 [==============================] - 5s 137ms/step - loss: 0.2879 -
accuracy: 0.9332 - val_loss: 0.2340 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 16/35
38/38 [==============================] - 5s 137ms/step - loss: 0.2616 -
accuracy: 0.9492 - val_loss: 0.1837 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 17/35
38/38 [==============================] - 5s 138ms/step - loss: 0.2441 -
accuracy: 0.9412 - val_loss: 0.2427 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 18/35
38/38 [==============================] - 5s 139ms/step - loss: 0.2083 -
accuracy: 0.9465 - val_loss: 0.2219 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 19/35
38/38 [==============================] - 5s 139ms/step - loss: 0.2425 -
accuracy: 0.9439 - val_loss: 0.2542 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 20/35
38/38 [==============================] - 5s 140ms/step - loss: 0.2563 -
accuracy: 0.9225 - val_loss: 0.2547 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 21/35
38/38 [==============================] - 5s 136ms/step - loss: 0.2043 -
accuracy: 0.9572 - val_loss: 0.1383 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 22/35
38/38 [==============================] - 5s 133ms/step - loss: 0.1661 -
accuracy: 0.9626 - val_loss: 0.1533 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 23/35
38/38 [==============================] - 5s 138ms/step - loss: 0.1647 -
accuracy: 0.9706 - val_loss: 0.1951 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 24/35
38/38 [==============================] - 5s 137ms/step - loss: 0.2452 -
accuracy: 0.9385 - val_loss: 0.2085 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 25/35
38/38 [==============================] - 5s 140ms/step - loss: 0.1333 -
accuracy: 0.9813 - val_loss: 0.2215 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 26/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1577 -
accuracy: 0.9572 - val_loss: 0.2615 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 27/35
38/38 [==============================] - 5s 139ms/step - loss: 0.1476 -
accuracy: 0.9652 - val_loss: 0.1430 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 28/35
38/38 [==============================] - 5s 135ms/step - loss: 0.1609 -
accuracy: 0.9626 - val_loss: 0.1335 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 29/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1676 -
accuracy: 0.9572 - val_loss: 0.1816 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 30/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1671 -
accuracy: 0.9626 - val_loss: 0.1794 - val_accuracy: 0.9333 - lr: 1.0000e-04
```

```
Epoch 31/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1677 -
accuracy: 0.9599 - val_loss: 0.2026 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 32/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1237 -
accuracy: 0.9679 - val_loss: 0.1665 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 33/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1184 -
accuracy: 0.9759 - val_loss: 0.1745 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 34/35
38/38 [==============================] - 5s 138ms/step - loss: 0.1274 -
accuracy: 0.9733 - val_loss: 0.1455 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 35/35
38/38 [==============================] - 5s 139ms/step - loss: 0.1069 -
accuracy: 0.9759 - val_loss: 0.1422 - val_accuracy: 0.9556 - lr: 1.0000e-04
Fold 4/5
Found 384 validated image filenames belonging to 4 classes.
Found 96 validated image filenames belonging to 4 classes.
Epoch 1/35
38/38 [==============================] - 58s 191ms/step - loss: 1.2535 -
accuracy: 0.5374 - val_loss: 2.7770 - val_accuracy: 0.2556 - lr: 1.0000e-04
Epoch 2/35
38/38 [==============================] - 5s 134ms/step - loss: 0.9078 -
accuracy: 0.7326 - val_loss: 1.5406 - val_accuracy: 0.2889 - lr: 1.0000e-04
Epoch 3/35
38/38 [==============================] - 5s 135ms/step - loss: 0.7100 -
accuracy: 0.8075 - val_loss: 1.5076 - val_accuracy: 0.4333 - lr: 1.0000e-04
Epoch 4/35
38/38 [==============================] - 5s 138ms/step - loss: 0.6325 -
accuracy: 0.8422 - val_loss: 1.0268 - val_accuracy: 0.6222 - lr: 1.0000e-04
Epoch 5/35
38/38 [==============================] - 5s 140ms/step - loss: 0.4816 -
accuracy: 0.8816 - val_loss: 1.2346 - val_accuracy: 0.5778 - lr: 1.0000e-04
Epoch 6/35
38/38 [==============================] - 5s 134ms/step - loss: 0.4791 -
accuracy: 0.8717 - val_loss: 0.7495 - val_accuracy: 0.7333 - lr: 1.0000e-04
Epoch 7/35
38/38 [==============================] - 5s 139ms/step - loss: 0.3507 -
accuracy: 0.9251 - val_loss: 0.5378 - val_accuracy: 0.8111 - lr: 1.0000e-04
Epoch 8/35
38/38 [==============================] - 5s 136ms/step - loss: 0.3733 -
accuracy: 0.9011 - val_loss: 0.5638 - val_accuracy: 0.8333 - lr: 1.0000e-04
Epoch 9/35
38/38 [==============================] - 5s 135ms/step - loss: 0.3708 -
accuracy: 0.8984 - val_loss: 0.3457 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 10/35
38/38 [==============================] - 5s 136ms/step - loss: 0.3428 -
accuracy: 0.9225 - val_loss: 0.3329 - val_accuracy: 0.8667 - lr: 1.0000e-04
```

```
Epoch 11/35
38/38 [==============================] - 5s 136ms/step - loss: 0.3321 -
accuracy: 0.9064 - val_loss: 0.2922 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 12/35
38/38 [==============================] - 5s 138ms/step - loss: 0.3595 -
accuracy: 0.8984 - val_loss: 0.3011 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 13/35
38/38 [==============================] - 5s 136ms/step - loss: 0.2547 -
accuracy: 0.9358 - val_loss: 0.2698 - val_accuracy: 0.9111 - lr: 1.0000e-04
Epoch 14/35
38/38 [==============================] - 5s 137ms/step - loss: 0.2545 -
accuracy: 0.9332 - val_loss: 0.2431 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 15/35
38/38 [==============================] - 5s 137ms/step - loss: 0.2632 -
accuracy: 0.9412 - val_loss: 0.3159 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 16/35
38/38 [==============================] - 5s 135ms/step - loss: 0.2704 -
accuracy: 0.9385 - val_loss: 0.2589 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 17/35
38/38 [==============================] - 5s 138ms/step - loss: 0.2381 -
accuracy: 0.9439 - val_loss: 0.2320 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 18/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1896 -
accuracy: 0.9599 - val_loss: 0.2004 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 19/35
38/38 [==============================] - 5s 137ms/step - loss: 0.2151 -
accuracy: 0.9332 - val_loss: 0.2037 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 20/35
38/38 [==============================] - 5s 139ms/step - loss: 0.2450 -
accuracy: 0.9421 - val_loss: 0.2088 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 21/35
38/38 [==============================] - 5s 139ms/step - loss: 0.1882 -
accuracy: 0.9652 - val_loss: 0.1021 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 22/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1427 -
accuracy: 0.9679 - val_loss: 0.1396 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 23/35
38/38 [==============================] - 5s 135ms/step - loss: 0.1782 -
accuracy: 0.9572 - val_loss: 0.1710 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 24/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1892 -
accuracy: 0.9519 - val_loss: 0.1637 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 25/35
38/38 [==============================] - 5s 133ms/step - loss: 0.1332 -
accuracy: 0.9733 - val_loss: 0.1435 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 26/35
38/38 [==============================] - 5s 135ms/step - loss: 0.1179 -
accuracy: 0.9840 - val_loss: 0.1350 - val_accuracy: 0.9556 - lr: 1.0000e-04
```

```
Epoch 27/35
38/38 [==============================] - 5s 139ms/step - loss: 0.1797 -
accuracy: 0.9492 - val_loss: 0.1750 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 28/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1800 -
accuracy: 0.9519 - val_loss: 0.2239 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 29/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1406 -
accuracy: 0.9759 - val_loss: 0.1644 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 30/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1192 -
accuracy: 0.9679 - val_loss: 0.2364 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 31/35
38/38 [==============================] - ETA: 0s - loss: 0.1179 - accuracy:
0.9733
Epoch 31: ReduceLROnPlateau reducing learning rate to 2.9999999242136255e-05.
38/38 [==============================] - 5s 137ms/step - loss: 0.1179 -
accuracy: 0.9733 - val_loss: 0.2886 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 32/35
38/38 [==============================] - 5s 133ms/step - loss: 0.1160 -
accuracy: 0.9706 - val_loss: 0.2078 - val_accuracy: 0.9333 - lr: 3.0000e-05
Epoch 33/35
38/38 [==============================] - 5s 139ms/step - loss: 0.1023 -
accuracy: 0.9866 - val_loss: 0.1989 - val_accuracy: 0.9556 - lr: 3.0000e-05
Epoch 34/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1004 -
accuracy: 0.9813 - val_loss: 0.2255 - val_accuracy: 0.9444 - lr: 3.0000e-05
Epoch 35/35
38/38 [==============================] - 5s 135ms/step - loss: 0.0970 -
accuracy: 0.9759 - val_loss: 0.1606 - val_accuracy: 0.9667 - lr: 3.0000e-05
Fold 5/5
Found 384 validated image filenames belonging to 4 classes.
Found 96 validated image filenames belonging to 4 classes.
Epoch 1/35
38/38 [==============================] - 56s 187ms/step - loss: 1.1876 -
accuracy: 0.4840 - val_loss: 1.4566 - val_accuracy: 0.2556 - lr: 1.0000e-04
Epoch 2/35
38/38 [==============================] - 5s 135ms/step - loss: 0.8170 -
accuracy: 0.6925 - val_loss: 3.3628 - val_accuracy: 0.3111 - lr: 1.0000e-04
Epoch 3/35
38/38 [==============================] - 5s 136ms/step - loss: 0.6898 -
accuracy: 0.7166 - val_loss: 1.7661 - val_accuracy: 0.4667 - lr: 1.0000e-04
Epoch 4/35
38/38 [==============================] - 5s 137ms/step - loss: 0.6767 -
accuracy: 0.7005 - val_loss: 1.4600 - val_accuracy: 0.4889 - lr: 1.0000e-04
Epoch 5/35
38/38 [==============================] - 5s 136ms/step - loss: 0.5777 -
accuracy: 0.7727 - val_loss: 1.2615 - val_accuracy: 0.5111 - lr: 1.0000e-04
```

```
Epoch 6/35
38/38 [==============================] - 5s 135ms/step - loss: 0.5373 -
accuracy: 0.7995 - val_loss: 1.4967 - val_accuracy: 0.5667 - lr: 1.0000e-04
Epoch 7/35
38/38 [==============================] - 5s 139ms/step - loss: 0.4362 -
accuracy: 0.8984 - val_loss: 0.8511 - val_accuracy: 0.6444 - lr: 1.0000e-04
Epoch 8/35
38/38 [==============================] - 5s 133ms/step - loss: 0.4949 -
accuracy: 0.8449 - val_loss: 0.5425 - val_accuracy: 0.7889 - lr: 1.0000e-04
Epoch 9/35
38/38 [==============================] - 5s 135ms/step - loss: 0.3850 -
accuracy: 0.9118 - val_loss: 0.3731 - val_accuracy: 0.8889 - lr: 1.0000e-04
Epoch 10/35
38/38 [==============================] - 5s 138ms/step - loss: 0.3814 -
accuracy: 0.8984 - val_loss: 0.2692 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 11/35
38/38 [==============================] - 5s 137ms/step - loss: 0.3470 -
accuracy: 0.9225 - val_loss: 0.2815 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 12/35
38/38 [==============================] - 5s 139ms/step - loss: 0.2815 -
accuracy: 0.9385 - val_loss: 0.3079 - val_accuracy: 0.9111 - lr: 1.0000e-04
Epoch 13/35
38/38 [==============================] - 5s 135ms/step - loss: 0.2874 -
accuracy: 0.9251 - val_loss: 0.2806 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 14/35
38/38 [==============================] - 5s 137ms/step - loss: 0.3182 -
accuracy: 0.9037 - val_loss: 0.5138 - val_accuracy: 0.7889 - lr: 1.0000e-04
Epoch 15/35
38/38 [==============================] - 5s 135ms/step - loss: 0.2703 -
accuracy: 0.9332 - val_loss: 0.2201 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 16/35
38/38 [==============================] - 5s 138ms/step - loss: 0.2169 -
accuracy: 0.9572 - val_loss: 0.2412 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 17/35
38/38 [==============================] - 5s 133ms/step - loss: 0.2423 -
accuracy: 0.9412 - val_loss: 0.2288 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 18/35
38/38 [==============================] - 5s 138ms/step - loss: 0.3102 -
accuracy: 0.9091 - val_loss: 0.1685 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 19/35
38/38 [==============================] - 5s 136ms/step - loss: 0.2798 -
accuracy: 0.9305 - val_loss: 0.1849 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 20/35
38/38 [==============================] - 5s 139ms/step - loss: 0.2472 -
accuracy: 0.9465 - val_loss: 0.2249 - val_accuracy: 0.9444 - lr: 1.0000e-04
Epoch 21/35
38/38 [==============================] - 5s 137ms/step - loss: 0.2458 -
accuracy: 0.9465 - val_loss: 0.1702 - val_accuracy: 0.9556 - lr: 1.0000e-04
```

```
Epoch 22/35
38/38 [==============================] - 5s 140ms/step - loss: 0.2486 -
accuracy: 0.9278 - val_loss: 0.2283 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 23/35
38/38 [==============================] - 5s 140ms/step - loss: 0.1953 -
accuracy: 0.9439 - val_loss: 0.1958 - val_accuracy: 0.9556 - lr: 1.0000e-04
Epoch 24/35
38/38 [==============================] - 5s 134ms/step - loss: 0.2218 -
accuracy: 0.9465 - val_loss: 0.2485 - val_accuracy: 0.9111 - lr: 1.0000e-04
Epoch 25/35
38/38 [==============================] - 5s 137ms/step - loss: 0.1297 -
accuracy: 0.9786 - val_loss: 0.2162 - val_accuracy: 0.9222 - lr: 1.0000e-04
Epoch 26/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1937 -
accuracy: 0.9492 - val_loss: 0.2602 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 27/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1166 -
accuracy: 0.9840 - val_loss: 0.1687 - val_accuracy: 0.9667 - lr: 1.0000e-04
Epoch 28/35
38/38 [==============================] - ETA: 0s - loss: 0.2057 - accuracy:
0.9545
Epoch 28: ReduceLROnPlateau reducing learning rate to 2.9999999242136255e-05.
38/38 [==============================] - 5s 137ms/step - loss: 0.2057 -
accuracy: 0.9545 - val_loss: 0.2308 - val_accuracy: 0.9333 - lr: 1.0000e-04
Epoch 29/35
38/38 [==============================] - 5s 134ms/step - loss: 0.2008 -
accuracy: 0.9412 - val_loss: 0.2219 - val_accuracy: 0.9444 - lr: 3.0000e-05
Epoch 30/35
38/38 [==============================] - 5s 138ms/step - loss: 0.1503 -
accuracy: 0.9545 - val_loss: 0.1811 - val_accuracy: 0.9444 - lr: 3.0000e-05
Epoch 31/35
38/38 [==============================] - 5s 138ms/step - loss: 0.1207 -
accuracy: 0.9706 - val_loss: 0.2094 - val_accuracy: 0.9333 - lr: 3.0000e-05
Epoch 32/35
38/38 [==============================] - 5s 136ms/step - loss: 0.1551 -
accuracy: 0.9706 - val_loss: 0.1776 - val_accuracy: 0.9556 - lr: 3.0000e-05
Epoch 33/35
38/38 [==============================] - 5s 133ms/step - loss: 0.1156 -
accuracy: 0.9786 - val_loss: 0.1790 - val_accuracy: 0.9667 - lr: 3.0000e-05
Epoch 33: early stopping
Validation Accuracy: 0.9533 ± 0.0267
Validation Loss: 0.1599 ± 0.0601
```

[ ]:
```python
import matplotlib.pyplot as plt

# Plot validation accuracies
plt.figure(figsize=(12, 6))
```

```
# Accuracy
plt.subplot(1, 2, 1)
plt.plot(range(1, k + 1), fold_accuracies, marker='o', linestyle='--',␣
 ↪color='b')
plt.title('Validation Accuracy per Fold')
plt.xlabel('Fold')
plt.ylabel('Accuracy')
plt.ylim([0, 1])
plt.yticks(np.arange(0, 1.1, 0.05))
plt.grid(True)

# Loss
plt.subplot(1, 2, 2)
plt.plot(range(1, k + 1), fold_losses, marker='o', linestyle='--', color='r')
plt.title('Validation Loss per Fold')
plt.xlabel('Fold')
plt.ylabel('Loss')
plt.grid(True)

# Show plots
plt.tight_layout()
plt.show()
```