eimer-mri-detection-efficientnetb0

March 28, 2024

[1]:
```
!pip install tensorflow==2.9.1
```

```
Collecting tensorflow==2.9.1
  Obtaining dependency information for tensorflow==2.9.1 from https://files.pyth
onhosted.org/packages/0b/3f/57bb9bbd2dfecba3659aedffb54bb837e83ccba81f217cb744f2
6d453048/tensorflow-2.9.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64
.whl.metadata
  Downloading tensorflow-2.9.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x
86_64.whl.metadata (3.0 kB)
Requirement already satisfied: absl-py>=1.0.0 in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.9.1) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.9.1) (1.6.3)
Collecting flatbuffers<2,>=1.12 (from tensorflow==2.9.1)
  Obtaining dependency information for flatbuffers<2,>=1.12 from https://files.p
ythonhosted.org/packages/eb/26/712e578c5f14e26ae3314c39a1bdc4eb2ec2f4ddc89b708cf
8e0a0d20423/flatbuffers-1.12-py2.py3-none-any.whl.metadata
  Downloading flatbuffers-1.12-py2.py3-none-any.whl.metadata (872 bytes)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.9.1) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.9.1) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.9.1) (1.51.1)
Requirement already satisfied: h5py>=2.9.0 in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.9.1) (3.9.0)
Collecting keras<2.10.0,>=2.9.0rc0 (from tensorflow==2.9.1)
  Obtaining dependency information for keras<2.10.0,>=2.9.0rc0 from https://file
s.pythonhosted.org/packages/ff/ff/f25909606aed26981a8bd6d263f89d64a20ca5e5316e6a
afb4c75d9ec8ae/keras-2.9.0-py2.py3-none-any.whl.metadata
  Downloading keras-2.9.0-py2.py3-none-any.whl.metadata (1.3 kB)
Collecting keras-preprocessing>=1.1.1 (from tensorflow==2.9.1)
  Obtaining dependency information for keras-preprocessing>=1.1.1 from https://f
iles.pythonhosted.org/packages/79/4c/7c3275a01e12ef9368a892926ab932b33bb13d55794
881e3573482b378a7/Keras_Preprocessing-1.1.2-py2.py3-none-any.whl.metadata
  Downloading Keras_Preprocessing-1.1.2-py2.py3-none-any.whl.metadata (1.9 kB)
Requirement already satisfied: libclang>=13.0.0 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.9.1) (16.0.6)
```

```
Requirement already satisfied: numpy>=1.20 in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.9.1) (1.24.3)
Requirement already satisfied: opt-einsum>=2.3.2 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.9.1) (3.3.0)
Requirement already satisfied: packaging in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.9.1) (21.3)
Collecting protobuf<3.20,>=3.9.2 (from tensorflow==2.9.1)
  Obtaining dependency information for protobuf<3.20,>=3.9.2 from https://files.
pythonhosted.org/packages/26/ef/bd6ba3b4ff9a35944bdd325e2c9ee56f71e855757f7d4393
8232499f0278/protobuf-3.19.6-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86
_64.whl.metadata
  Downloading protobuf-3.19.6-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl.metadata (787 bytes)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.9.1) (68.1.2)
Requirement already satisfied: six>=1.12.0 in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.9.1) (1.16.0)
Collecting tensorboard<2.10,>=2.9 (from tensorflow==2.9.1)
  Obtaining dependency information for tensorboard<2.10,>=2.9 from https://files
.pythonhosted.org/packages/ee/0d/23812e6ce63b3d87c39bc9fee83e28c499052fa83fdddd
7daea21a6d620/tensorboard-2.9.1-py3-none-any.whl.metadata
  Downloading tensorboard-2.9.1-py3-none-any.whl.metadata (1.9 kB)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.9.1) (0.34.0)
Collecting tensorflow-estimator<2.10.0,>=2.9.0rc0 (from tensorflow==2.9.1)
  Obtaining dependency information for tensorflow-estimator<2.10.0,>=2.9.0rc0
from https://files.pythonhosted.org/packages/61/e1/a72ec68403d91ba433018db58859f
d4706642aa9d0fb44ff778934fc4c2c/tensorflow_estimator-2.9.0-py2.py3-none-
any.whl.metadata
  Downloading tensorflow_estimator-2.9.0-py2.py3-none-any.whl.metadata (1.3 kB)
Requirement already satisfied: termcolor>=1.1.0 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.9.1) (2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.9.1) (4.5.0)
Requirement already satisfied: wrapt>=1.11.0 in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.9.1) (1.15.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/opt/conda/lib/python3.10/site-packages (from
astunparse>=1.6.0->tensorflow==2.9.1) (0.41.2)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<2.10,>=2.9->tensorflow==2.9.1) (2.22.0)
Collecting google-auth-oauthlib<0.5,>=0.4.1 (from
tensorboard<2.10,>=2.9->tensorflow==2.9.1)
  Obtaining dependency information for google-auth-oauthlib<0.5,>=0.4.1 from htt
ps://files.pythonhosted.org/packages/b1/0e/0636cc1448a7abc444fb1b3a63655e294e0d2
d49092dc3de05241be6d43c/google_auth_oauthlib-0.4.6-py2.py3-none-any.whl.metadata
  Downloading google_auth_oauthlib-0.4.6-py2.py3-none-any.whl.metadata (2.7 kB)
```

```
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<2.10,>=2.9->tensorflow==2.9.1) (3.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<2.10,>=2.9->tensorflow==2.9.1) (2.31.0)
Collecting tensorboard-data-server<0.7.0,>=0.6.0 (from
tensorboard<2.10,>=2.9->tensorflow==2.9.1)
  Obtaining dependency information for tensorboard-data-server<0.7.0,>=0.6.0
from https://files.pythonhosted.org/packages/60/f9/802efd84988bffd9f644c03b6e66f
de8e76c3aa33db4279ddd11c5d61f4b/tensorboard_data_server-0.6.1-py3-none-
manylinux2010_x86_64.whl.metadata
  Downloading tensorboard_data_server-0.6.1-py3-none-
manylinux2010_x86_64.whl.metadata (1.1 kB)
Collecting tensorboard-plugin-wit>=1.6.0 (from
tensorboard<2.10,>=2.9->tensorflow==2.9.1)
  Obtaining dependency information for tensorboard-plugin-wit>=1.6.0 from https:
//files.pythonhosted.org/packages/e0/68/e8ecfac5dd594b676c23a7f07ea34c197d7d69b3
313afdf8ac1b0a9905a2/tensorboard_plugin_wit-1.8.1-py3-none-any.whl.metadata
  Downloading tensorboard_plugin_wit-1.8.1-py3-none-any.whl.metadata (873 bytes)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<2.10,>=2.9->tensorflow==2.9.1) (3.0.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/opt/conda/lib/python3.10/site-packages (from packaging->tensorflow==2.9.1)
(3.0.9)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/lib/python3.10/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow==2.9.1) (4.2.4)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/lib/python3.10/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow==2.9.1) (0.2.7)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/lib/python3.10/site-
packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow==2.9.1)
(4.9)
Requirement already satisfied: urllib3<2.0 in /opt/conda/lib/python3.10/site-
packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow==2.9.1)
(1.26.15)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/lib/python3.10/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow==2.9.1) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/conda/lib/python3.10/site-packages (from
requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.1) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-
packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.1)
(3.4)
Requirement already satisfied: certifi>=2017.4.17 in
```

```
/opt/conda/lib/python3.10/site-packages (from
requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow==2.9.1) (2023.11.17)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/opt/conda/lib/python3.10/site-packages (from
werkzeug>=1.0.1->tensorboard<2.10,>=2.9->tensorflow==2.9.1) (2.1.3)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/opt/conda/lib/python3.10/site-packages (from pyasn1-modules>=0.2.1->google-
auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow==2.9.1) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/lib/python3.10/site-packages (from requests-oauthlib>=0.7.0->google-
auth-oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow==2.9.1) (3.2.2)
Downloading
tensorflow-2.9.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(511.7 MB)
                         511.7/511.7 MB
2.4 MB/s eta 0:00:00:00:0100:01
Downloading flatbuffers-1.12-py2.py3-none-any.whl (15 kB)
Downloading keras-2.9.0-py2.py3-none-any.whl (1.6 MB)
                         1.6/1.6 MB
67.0 MB/s eta 0:00:00
Downloading Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
                         42.6/42.6 kB
4.0 MB/s eta 0:00:00
Downloading
protobuf-3.19.6-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.1
MB)
                         1.1/1.1 MB
59.0 MB/s eta 0:00:00
Downloading tensorboard-2.9.1-py3-none-any.whl (5.8 MB)
                         5.8/5.8 MB
91.0 MB/s eta 0:00:00:00:01
Downloading tensorflow_estimator-2.9.0-py2.py3-none-any.whl (438 kB)
                         438.7/438.7 kB
37.6 MB/s eta 0:00:00
Downloading google_auth_oauthlib-0.4.6-py2.py3-none-any.whl (18 kB)
Downloading tensorboard_data_server-0.6.1-py3-none-manylinux2010_x86_64.whl (4.9
MB)
                         4.9/4.9 MB
84.0 MB/s eta 0:00:00:00:01
Downloading tensorboard_plugin_wit-1.8.1-py3-none-any.whl (781 kB)
                         781.3/781.3 kB
54.9 MB/s eta 0:00:00
Installing collected packages: tensorboard-plugin-wit, keras, flatbuffers,
tensorflow-estimator, tensorboard-data-server, protobuf, keras-preprocessing,
google-auth-oauthlib, tensorboard, tensorflow
  Attempting uninstall: keras
    Found existing installation: keras 2.13.1
    Uninstalling keras-2.13.1:
```

```
    Successfully uninstalled keras-2.13.1
Attempting uninstall: flatbuffers
  Found existing installation: flatbuffers 23.5.26
  Uninstalling flatbuffers-23.5.26:
    Successfully uninstalled flatbuffers-23.5.26
Attempting uninstall: tensorflow-estimator
  Found existing installation: tensorflow-estimator 2.13.0
  Uninstalling tensorflow-estimator-2.13.0:
    Successfully uninstalled tensorflow-estimator-2.13.0
Attempting uninstall: tensorboard-data-server
  Found existing installation: tensorboard-data-server 0.7.1
  Uninstalling tensorboard-data-server-0.7.1:
    Successfully uninstalled tensorboard-data-server-0.7.1
Attempting uninstall: protobuf
  Found existing installation: protobuf 3.20.3
  Uninstalling protobuf-3.20.3:
    Successfully uninstalled protobuf-3.20.3
Attempting uninstall: google-auth-oauthlib
  Found existing installation: google-auth-oauthlib 1.0.0
  Uninstalling google-auth-oauthlib-1.0.0:
    Successfully uninstalled google-auth-oauthlib-1.0.0
Attempting uninstall: tensorboard
  Found existing installation: tensorboard 2.13.0
  Uninstalling tensorboard-2.13.0:
    Successfully uninstalled tensorboard-2.13.0
Attempting uninstall: tensorflow
  Found existing installation: tensorflow 2.13.0
  Uninstalling tensorflow-2.13.0:
    Successfully uninstalled tensorflow-2.13.0
```

```
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
cudf 23.8.0 requires cupy-cuda11x>=12.0.0, which is not installed.
cuml 23.8.0 requires cupy-cuda11x>=12.0.0, which is not installed.
dask-cudf 23.8.0 requires cupy-cuda11x>=12.0.0, which is not installed.
apache-beam 2.46.0 requires dill<0.3.2,>=0.3.1.1, but you have dill 0.3.7 which
is incompatible.
apache-beam 2.46.0 requires pyarrow<10.0.0,>=3.0.0, but you have pyarrow 11.0.0
which is incompatible.
cudf 23.8.0 requires pandas<1.6.0dev0,>=1.3, but you have pandas 2.0.3 which is
incompatible.
cudf 23.8.0 requires protobuf<5,>=4.21, but you have protobuf 3.19.6 which is
incompatible.
cuml 23.8.0 requires dask==2023.7.1, but you have dask 2023.12.0 which is
incompatible.
cuml 23.8.0 requires distributed==2023.7.1, but you have distributed 2023.12.0
which is incompatible.
dask-cudf 23.8.0 requires dask==2023.7.1, but you have dask 2023.12.0 which is
incompatible.
dask-cudf 23.8.0 requires distributed==2023.7.1, but you have distributed
2023.12.0 which is incompatible.
dask-cudf 23.8.0 requires pandas<1.6.0dev0,>=1.3, but you have pandas 2.0.3
which is incompatible.
gcsfs 2023.6.0 requires fsspec==2023.6.0, but you have fsspec 2023.12.2 which is
incompatible.
google-cloud-aiplatform 0.6.0a1 requires google-api-
core[grpc]<2.0.0dev,>=1.22.2, but you have google-api-core 2.11.1 which is
incompatible.
google-cloud-automl 1.0.1 requires google-api-core[grpc]<2.0.0dev,>=1.14.0, but
you have google-api-core 2.11.1 which is incompatible.
google-cloud-pubsub 2.18.3 requires grpcio<2.0dev,>=1.51.3, but you have grpcio
1.51.1 which is incompatible.
kfp 2.0.1 requires google-cloud-storage<3,>=2.2.1, but you have google-cloud-
storage 1.44.0 which is incompatible.
onnx 1.15.0 requires protobuf>=3.20.2, but you have protobuf 3.19.6 which is
```

```
keras-2.9.0 keras-preprocessing-1.1.2 protobuf-3.19.6 tensorboard-2.9.1
tensorboard-data-server-0.6.1 tensorboard-plugin-wit-1.8.1 tensorflow-2.9.1
tensorflow-estimator-2.9.0
```

[2]:

```
# import system libs
import os
import time
import shutil
import pathlib
import itertools
from PIL import Image

# import data handling tools
import cv2
import numpy as np
import pandas as pd
import seaborn as sns
sns.set_style('darkgrid')
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
# import Deep learning Libraries
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam, Adamax
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,␣
  ↪Activation, Dropout, BatchNormalization
from tensorflow.keras import regularizers

# Ignore Warnings
import warnings
warnings.filterwarnings("ignore")

print ('modules loaded')
```

```
/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146: UserWarning: A
NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy
(detected version 1.24.3
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

```
modules loaded
```

[3]:

```
train_dir = '/kaggle/input/alzheimers-dataset-4-class-of-images/Alzheimer_s␣
  ↪Dataset/train'
filepaths = []
```

```
labels = []

folds = os.listdir(train_dir)
for fold in folds:
    foldpath = os.path.join(train_dir, fold)
    filelist = os.listdir(foldpath)
    for file in filelist:
        fpath = os.path.join(foldpath, file)
        filepaths.append(fpath)
        labels.append(fold)

# Concatenate data paths with labels into one dataframe
Fseries = pd.Series(filepaths, name= 'filepaths')
Lseries = pd.Series(labels, name='labels')
train_df = pd.concat([Fseries, Lseries], axis= 1)
```

[4]:

```
train_df
```

[4]:

```
                                              filepaths             labels
0     /kaggle/input/alzheimers-dataset-4-class-of-im…  ModerateDemented
1     /kaggle/input/alzheimers-dataset-4-class-of-im…  ModerateDemented
2     /kaggle/input/alzheimers-dataset-4-class-of-im…  ModerateDemented
3     /kaggle/input/alzheimers-dataset-4-class-of-im…  ModerateDemented
4     /kaggle/input/alzheimers-dataset-4-class-of-im…  ModerateDemented
…                                                  …                …
5116  /kaggle/input/alzheimers-dataset-4-class-of-im…      MildDemented
5117  /kaggle/input/alzheimers-dataset-4-class-of-im…      MildDemented
5118  /kaggle/input/alzheimers-dataset-4-class-of-im…      MildDemented
5119  /kaggle/input/alzheimers-dataset-4-class-of-im…      MildDemented
5120  /kaggle/input/alzheimers-dataset-4-class-of-im…      MildDemented

[5121 rows x 2 columns]
```

[5]:

```
# Generate  test data paths with labels
test_dir = '/kaggle/input/alzheimers-dataset-4-class-of-images/Alzheimer_s␣
 ↪Dataset/test'
filepaths = []
labels = []

folds = os.listdir(test_dir)
for fold in folds:
    foldpath = os.path.join(test_dir, fold)
    filelist = os.listdir(foldpath)
    for file in filelist:
        fpath = os.path.join(foldpath, file)
        filepaths.append(fpath)
        labels.append(fold)
```

```
# Concatenate data paths with labels into one dataframe
Fseries = pd.Series(filepaths, name= 'filepaths')
Lseries = pd.Series(labels, name='labels')
test_df = pd.concat([Fseries, Lseries], axis= 1)
```

[6]:

```
test_df
```

[6]:

```
                                              filepaths            labels
0     /kaggle/input/alzheimers-dataset-4-class-of-im…  ModerateDemented
1     /kaggle/input/alzheimers-dataset-4-class-of-im…  ModerateDemented
2     /kaggle/input/alzheimers-dataset-4-class-of-im…  ModerateDemented
3     /kaggle/input/alzheimers-dataset-4-class-of-im…  ModerateDemented
4     /kaggle/input/alzheimers-dataset-4-class-of-im…  ModerateDemented
…                                                 …                …
1274  /kaggle/input/alzheimers-dataset-4-class-of-im…      MildDemented
1275  /kaggle/input/alzheimers-dataset-4-class-of-im…      MildDemented
1276  /kaggle/input/alzheimers-dataset-4-class-of-im…      MildDemented
1277  /kaggle/input/alzheimers-dataset-4-class-of-im…      MildDemented
1278  /kaggle/input/alzheimers-dataset-4-class-of-im…      MildDemented

[1279 rows x 2 columns]
```

[7]:

```
# valid and test dataframe
valid_df, test_df = train_test_split(test_df,  train_size= 0.6, shuffle= True,␣
 ↪random_state= 123)
```

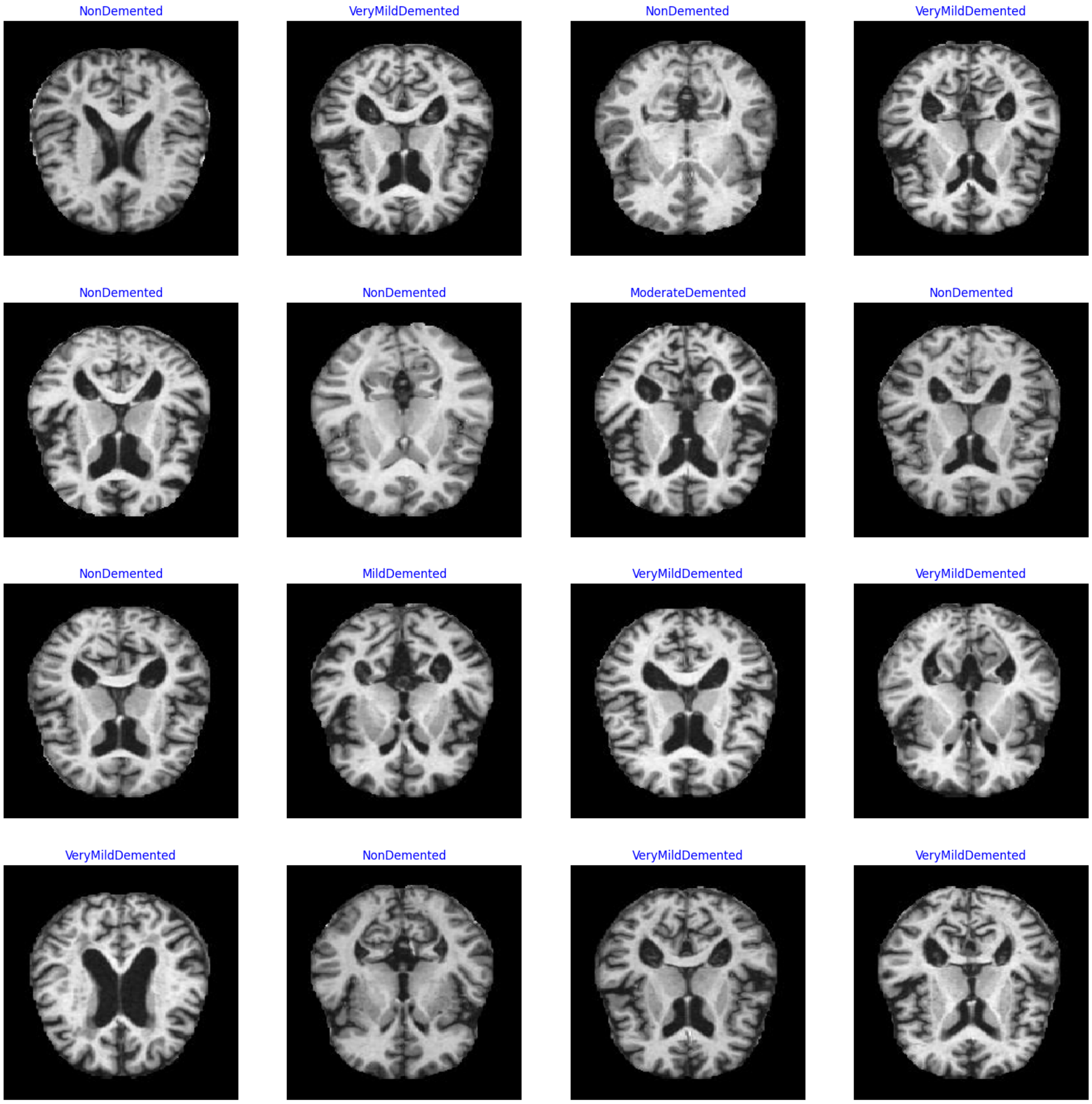
[8]:

```
# crobed image size
batch_size = 16
img_size = (224, 224)
channels = 3
img_shape = (img_size[0], img_size[1], channels)

tr_gen = ImageDataGenerator()
ts_gen = ImageDataGenerator()
train_gen = tr_gen.flow_from_dataframe( train_df, x_col= 'filepaths', y_col=␣
 ↪'labels', target_size= img_size, class_mode= 'categorical',
                                    color_mode= 'rgb', shuffle= True,␣
 ↪batch_size= batch_size)

valid_gen = ts_gen.flow_from_dataframe( valid_df, x_col= 'filepaths', y_col=␣
 ↪'labels', target_size= img_size, class_mode= 'categorical',
                                    color_mode= 'rgb', shuffle= True,␣
 ↪batch_size= batch_size)

test_gen = ts_gen.flow_from_dataframe( test_df, x_col= 'filepaths', y_col=␣
 ↪'labels', target_size= img_size, class_mode= 'categorical',
                                    color_mode= 'rgb', shuffle= False,␣
 ↪batch_size= batch_size)
```

```
Found 5121 validated image filenames belonging to 4 classes.
Found 767 validated image filenames belonging to 4 classes.
Found 512 validated image filenames belonging to 4 classes.
```

[9]:

```
g_dict = train_gen.class_indices      # defines dictionary {'class': index}
classes = list(g_dict.keys())       # defines list of dictionary's kays␣
 ↪(classes), classes names : string
images, labels = next(train_gen)      # get a batch size samples from the␣
 ↪generator

plt.figure(figsize= (20, 20))

for i in range(16):
    plt.subplot(4, 4, i + 1)
    image = images[i] / 255       # scales data to range (0 - 255)
    plt.imshow(image)
    index = np.argmax(labels[i])  # get image index
    class_name = classes[index]   # get class of image
    plt.title(class_name, color= 'blue', fontsize= 12)
    plt.axis('off')
plt.show()
```



[10]:

```python
# Create Model Structure
img_size = (224, 224)
channels = 3
img_shape = (img_size[0], img_size[1], channels)
class_count = len(list(train_gen.class_indices.keys())) # to define number of classes in dense layer

# create pre-trained model (you can built on pretrained model such as :  efficientnet, VGG , Resnet )
# we will use efficientnetb3 from EfficientNet family.
base_model = tf.keras.applications.efficientnet.EfficientNetB0(include_top= False, weights= "imagenet", input_shape= img_shape, pooling= 'max')
```

```
# base_model.trainable = False
model = Sequential([
    base_model,
    BatchNormalization(axis= -1, momentum= 0.99, epsilon= 0.001),
    Dense(256, kernel_regularizer= regularizers.l2(l= 0.016),␣
 ↪activity_regularizer= regularizers.l1(0.006),
                bias_regularizer= regularizers.l1(0.006), activation= 'relu'),
    Dropout(rate= 0.45, seed= 123),
    Dense(class_count, activation= 'softmax')
])

model.compile(Adamax(learning_rate= 0.001), loss= 'categorical_crossentropy',␣
 ↪metrics= ['accuracy'])

model.summary()
```

```
Downloading data from https://storage.googleapis.com/keras-
applications/efficientnetb0_notop.h5
16705208/16705208 [==============================] - 0s 0us/step
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 efficientnetb0 (Functional)  (None, 1280)             4049571

 batch_normalization (BatchN  (None, 1280)             5120
 ormalization)

 dense (Dense)               (None, 256)               327936

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 4)                 1028

=================================================================
Total params: 4,383,655
Trainable params: 4,339,072
Non-trainable params: 44,583
_________________________________________________________________
```

[11]:

```
batch_size = 20   # set batch size for training
epochs =  50  # number of all epochs in training

history = model.fit(x= train_gen, epochs= epochs, verbose= 1, validation_data=␣
 ↪valid_gen,
                    validation_steps= None, shuffle= False)
```

```
Epoch 1/50
321/321 [==============================] - 56s 141ms/step - loss: 6.7986 -
accuracy: 0.5419 - val_loss: 5.5256 - val_accuracy: 0.3533
Epoch 2/50
321/321 [==============================] - 40s 125ms/step - loss: 4.1742 -
accuracy: 0.6919 - val_loss: 3.5147 - val_accuracy: 0.6206
Epoch 3/50
321/321 [==============================] - 40s 124ms/step - loss: 2.5934 -
accuracy: 0.8350 - val_loss: 2.4967 - val_accuracy: 0.6506
Epoch 4/50
321/321 [==============================] - 40s 125ms/step - loss: 1.5684 -
accuracy: 0.9166 - val_loss: 1.8653 - val_accuracy: 0.6623
Epoch 5/50
321/321 [==============================] - 40s 125ms/step - loss: 0.9645 -
accuracy: 0.9443 - val_loss: 1.6672 - val_accuracy: 0.6454
Epoch 6/50
321/321 [==============================] - 40s 125ms/step - loss: 0.6344 -
accuracy: 0.9617 - val_loss: 1.2727 - val_accuracy: 0.7027
Epoch 7/50
321/321 [==============================] - 40s 125ms/step - loss: 0.4448 -
accuracy: 0.9717 - val_loss: 1.2796 - val_accuracy: 0.6558
Epoch 8/50
321/321 [==============================] - 40s 125ms/step - loss: 0.3507 -
accuracy: 0.9717 - val_loss: 1.1285 - val_accuracy: 0.6910
Epoch 9/50
321/321 [==============================] - 40s 125ms/step - loss: 0.2885 -
accuracy: 0.9738 - val_loss: 0.7688 - val_accuracy: 0.7784
Epoch 10/50
321/321 [==============================] - 40s 124ms/step - loss: 0.2415 -
accuracy: 0.9820 - val_loss: 1.0604 - val_accuracy: 0.6949
Epoch 11/50
321/321 [==============================] - 40s 125ms/step - loss: 0.2369 -
accuracy: 0.9762 - val_loss: 0.9829 - val_accuracy: 0.7236
Epoch 12/50
321/321 [==============================] - 40s 124ms/step - loss: 0.1930 -
accuracy: 0.9877 - val_loss: 0.8137 - val_accuracy: 0.7497
Epoch 13/50
321/321 [==============================] - 40s 125ms/step - loss: 0.1848 -
accuracy: 0.9857 - val_loss: 0.9165 - val_accuracy: 0.7301
Epoch 14/50
321/321 [==============================] - 40s 125ms/step - loss: 0.1771 -
accuracy: 0.9848 - val_loss: 1.0629 - val_accuracy: 0.7184
Epoch 15/50
321/321 [==============================] - 40s 125ms/step - loss: 0.1669 -
accuracy: 0.9867 - val_loss: 0.9389 - val_accuracy: 0.7392
Epoch 16/50
321/321 [==============================] - 40s 125ms/step - loss: 0.1587 -
accuracy: 0.9875 - val_loss: 0.8062 - val_accuracy: 0.7392
```

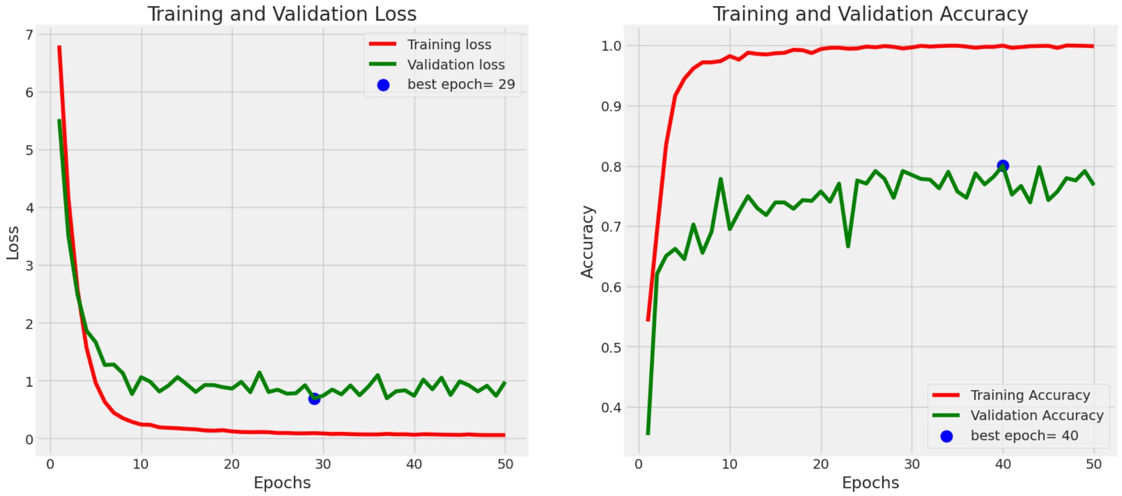
```
Epoch 17/50
321/321 [==============================] - 40s 125ms/step - loss: 0.1397 -
accuracy: 0.9926 - val_loss: 0.9258 - val_accuracy: 0.7288
Epoch 18/50
321/321 [==============================] - 40s 125ms/step - loss: 0.1357 -
accuracy: 0.9918 - val_loss: 0.9238 - val_accuracy: 0.7432
Epoch 19/50
321/321 [==============================] - 40s 124ms/step - loss: 0.1434 -
accuracy: 0.9871 - val_loss: 0.8869 - val_accuracy: 0.7419
Epoch 20/50
321/321 [==============================] - 40s 124ms/step - loss: 0.1227 -
accuracy: 0.9938 - val_loss: 0.8649 - val_accuracy: 0.7575
Epoch 21/50
321/321 [==============================] - 40s 125ms/step - loss: 0.1133 -
accuracy: 0.9957 - val_loss: 0.9805 - val_accuracy: 0.7405
Epoch 22/50
321/321 [==============================] - 40s 124ms/step - loss: 0.1089 -
accuracy: 0.9959 - val_loss: 0.8012 - val_accuracy: 0.7705
Epoch 23/50
321/321 [==============================] - 40s 125ms/step - loss: 0.1116 -
accuracy: 0.9943 - val_loss: 1.1427 - val_accuracy: 0.6662
Epoch 24/50
321/321 [==============================] - 40s 125ms/step - loss: 0.1092 -
accuracy: 0.9947 - val_loss: 0.8052 - val_accuracy: 0.7757
Epoch 25/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0959 -
accuracy: 0.9979 - val_loss: 0.8444 - val_accuracy: 0.7705
Epoch 26/50
321/321 [==============================] - 40s 125ms/step - loss: 0.0960 -
accuracy: 0.9967 - val_loss: 0.7761 - val_accuracy: 0.7914
Epoch 27/50
321/321 [==============================] - 40s 126ms/step - loss: 0.0887 -
accuracy: 0.9986 - val_loss: 0.7832 - val_accuracy: 0.7784
Epoch 28/50
321/321 [==============================] - 40s 125ms/step - loss: 0.0893 -
accuracy: 0.9975 - val_loss: 0.9219 - val_accuracy: 0.7471
Epoch 29/50
321/321 [==============================] - 40s 126ms/step - loss: 0.0927 -
accuracy: 0.9947 - val_loss: 0.6947 - val_accuracy: 0.7914
Epoch 30/50
321/321 [==============================] - 40s 125ms/step - loss: 0.0870 -
accuracy: 0.9965 - val_loss: 0.7410 - val_accuracy: 0.7849
Epoch 31/50
321/321 [==============================] - 40s 125ms/step - loss: 0.0786 -
accuracy: 0.9990 - val_loss: 0.8467 - val_accuracy: 0.7784
Epoch 32/50
321/321 [==============================] - 40s 125ms/step - loss: 0.0818 -
accuracy: 0.9979 - val_loss: 0.7632 - val_accuracy: 0.7771
```

```
Epoch 33/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0762 -
accuracy: 0.9986 - val_loss: 0.9185 - val_accuracy: 0.7627
Epoch 34/50
321/321 [==============================] - 40s 126ms/step - loss: 0.0722 -
accuracy: 0.9992 - val_loss: 0.7478 - val_accuracy: 0.7901
Epoch 35/50
321/321 [==============================] - 40s 125ms/step - loss: 0.0707 -
accuracy: 0.9992 - val_loss: 0.9037 - val_accuracy: 0.7575
Epoch 36/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0711 -
accuracy: 0.9979 - val_loss: 1.0971 - val_accuracy: 0.7471
Epoch 37/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0799 -
accuracy: 0.9959 - val_loss: 0.6961 - val_accuracy: 0.7875
Epoch 38/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0727 -
accuracy: 0.9975 - val_loss: 0.8164 - val_accuracy: 0.7692
Epoch 39/50
321/321 [==============================] - 40s 125ms/step - loss: 0.0733 -
accuracy: 0.9975 - val_loss: 0.8341 - val_accuracy: 0.7823
Epoch 40/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0648 -
accuracy: 0.9994 - val_loss: 0.7398 - val_accuracy: 0.8005
Epoch 41/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0744 -
accuracy: 0.9957 - val_loss: 1.0191 - val_accuracy: 0.7523
Epoch 42/50
321/321 [==============================] - 40s 126ms/step - loss: 0.0720 -
accuracy: 0.9971 - val_loss: 0.8514 - val_accuracy: 0.7666
Epoch 43/50
321/321 [==============================] - 40s 126ms/step - loss: 0.0670 -
accuracy: 0.9984 - val_loss: 1.0520 - val_accuracy: 0.7392
Epoch 44/50
321/321 [==============================] - 40s 125ms/step - loss: 0.0651 -
accuracy: 0.9988 - val_loss: 0.7540 - val_accuracy: 0.7979
Epoch 45/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0620 -
accuracy: 0.9990 - val_loss: 0.9897 - val_accuracy: 0.7432
Epoch 46/50
321/321 [==============================] - 40s 126ms/step - loss: 0.0711 -
accuracy: 0.9957 - val_loss: 0.9208 - val_accuracy: 0.7575
Epoch 47/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0616 -
accuracy: 0.9996 - val_loss: 0.8143 - val_accuracy: 0.7797
Epoch 48/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0591 -
accuracy: 0.9994 - val_loss: 0.9131 - val_accuracy: 0.7757
```

```
Epoch 49/50
321/321 [==============================] - 40s 124ms/step - loss: 0.0597 -
accuracy: 0.9990 - val_loss: 0.7396 - val_accuracy: 0.7914
Epoch 50/50
321/321 [==============================] - 40s 125ms/step - loss: 0.0591 -
accuracy: 0.9984 - val_loss: 0.9808 - val_accuracy: 0.7679
```

[12]:

```python
# Define needed variables
tr_acc = history.history['accuracy']
tr_loss = history.history['loss']
val_acc = history.history['val_accuracy']
val_loss = history.history['val_loss']
index_loss = np.argmin(val_loss)
val_lowest = val_loss[index_loss]
index_acc = np.argmax(val_acc)
acc_highest = val_acc[index_acc]
Epochs = [i+1 for i in range(len(tr_acc))]
loss_label = f'best epoch= {str(index_loss + 1)}'
acc_label = f'best epoch= {str(index_acc + 1)}'
# Plot training history
plt.figure(figsize= (20, 8))
plt.style.use('fivethirtyeight')
plt.subplot(1, 2, 1)
plt.plot(Epochs, tr_loss, 'r', label= 'Training loss')
plt.plot(Epochs, val_loss, 'g', label= 'Validation loss')
plt.scatter(index_loss + 1, val_lowest, s= 150, c= 'blue', label= loss_label)
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.subplot(1, 2, 2)
plt.plot(Epochs, tr_acc, 'r', label= 'Training Accuracy')
plt.plot(Epochs, val_acc, 'g', label= 'Validation Accuracy')
plt.scatter(index_acc + 1 , acc_highest, s= 150, c= 'blue', label= acc_label)
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.tight_layout
plt.show()
```



[13]:

```python
ts_length = len(test_df)
test_batch_size = max(sorted([ts_length // n for n in range(1, ts_length + 1)␣
 ↪if ts_length%n == 0 and ts_length/n <= 80]))
test_steps = ts_length // test_batch_size

train_score = model.evaluate(train_gen, steps= test_steps, verbose= 1)
valid_score = model.evaluate(valid_gen, steps= test_steps, verbose= 1)
test_score = model.evaluate(test_gen, steps= test_steps, verbose= 1)

print("Train Loss: ", train_score[0])
print("Train Accuracy: ", train_score[1])
print('-' * 20)
print("Validation Loss: ", valid_score[0])
print("Validation Accuracy: ", valid_score[1])
print('-' * 20)
print("Test Loss: ", test_score[0])
print("Test Accuracy: ", test_score[1])
```

```
8/8 [==============================] - 0s 30ms/step - loss: 0.0798 - accuracy:
0.9844
8/8 [==============================] - 0s 30ms/step - loss: 1.0943 - accuracy:
0.7266
8/8 [==============================] - 1s 102ms/step - loss: 0.8280 - accuracy:
0.7734
Train Loss:  0.07983075082302094
Train Accuracy:  0.984375
--------------------
Validation Loss:  1.0943078994750977
Validation Accuracy:  0.7265625
--------------------
```

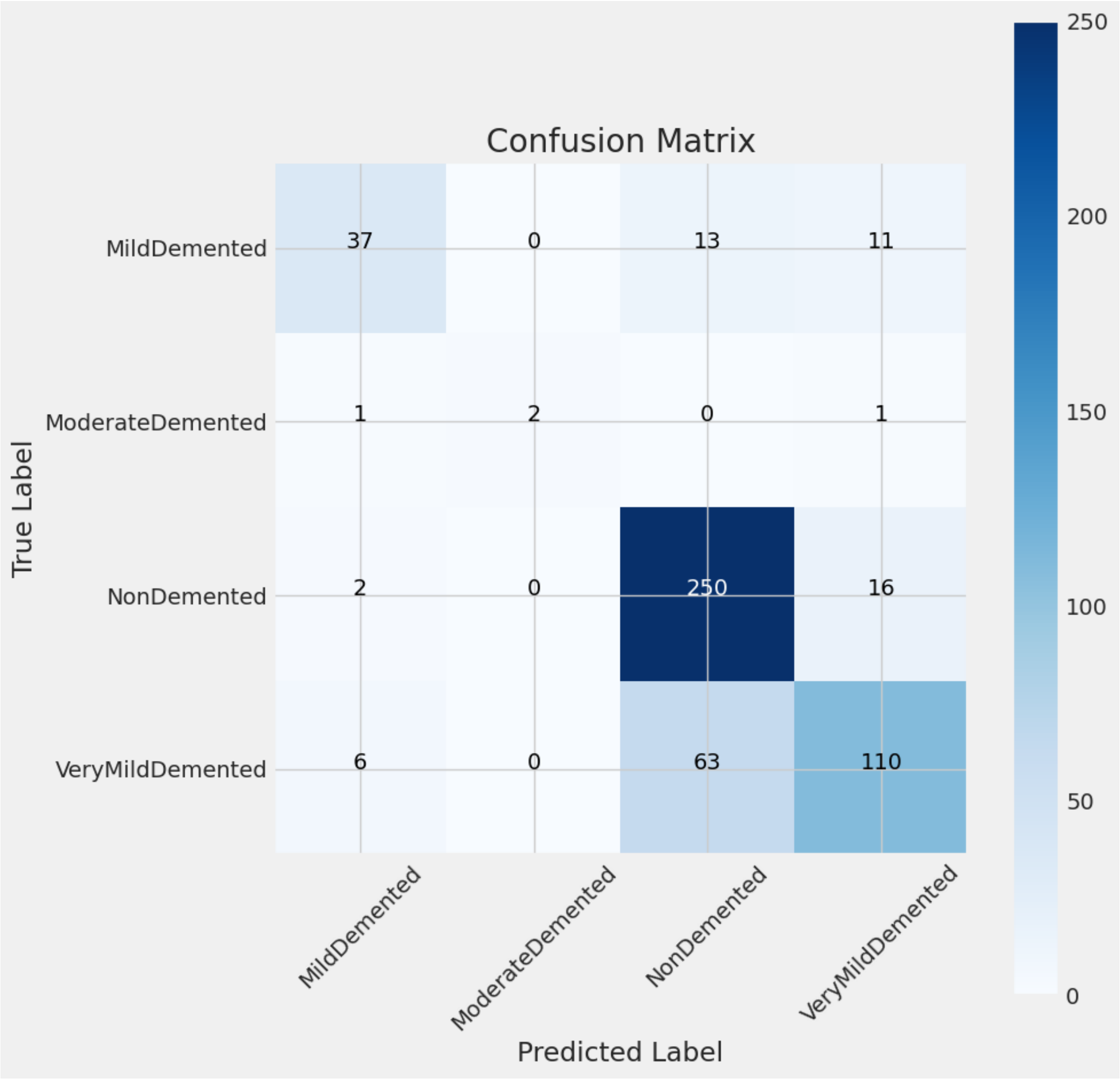
```
Test Loss:  0.8280189633369446
Test Accuracy:  0.7734375
```

[14]:

```
preds = model.predict_generator(test_gen)
y_pred = np.argmax(preds, axis=1)
```

[15]:

```
g_dict = test_gen.class_indices
classes = list(g_dict.keys())

# Confusion matrix
cm = confusion_matrix(test_gen.classes, y_pred)

plt.figure(figsize= (10, 10))
plt.imshow(cm, interpolation= 'nearest', cmap= plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()

tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation= 45)
plt.yticks(tick_marks, classes)
thresh = cm.max() / 2.
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
    plt.text(j, i, cm[i, j], horizontalalignment= 'center', color= 'white' if␣
 ↪cm[i, j] > thresh else 'black')

plt.tight_layout()
plt.ylabel('True Label')
plt.xlabel('Predicted Label')

plt.show()
```



[16]:

```
# Classification report
print(classification_report(test_gen.classes, y_pred, target_names= classes))
```

```
                  precision    recall  f1-score   support

    MildDemented       0.80      0.61      0.69        61
ModerateDemented       1.00      0.50      0.67         4
     NonDemented       0.77      0.93      0.84       268
VeryMildDemented       0.80      0.61      0.69       179

        accuracy                           0.78       512
       macro avg       0.84      0.66      0.72       512
    weighted avg       0.78      0.78      0.77       512
```

[17]:

```
#Save the model
model.save('model.h5')
```

[18]:

```
loaded_model = tf.keras.models.load_model('/kaggle/working/model.h5',␣
 ↪compile=False)
loaded_model.compile(Adamax(learning_rate= 0.001), loss=␣
 ↪'categorical_crossentropy', metrics= ['accuracy'])
```

[19]:

```
image_path = '/kaggle/input/alzheimers-dataset-4-class-of-images/Alzheimer_s␣
 ↪Dataset/test/VeryMildDemented/26 (46).jpg'
image = Image.open(image_path)

# Preprocess the image
img = image.resize((224, 224))
img_array = tf.keras.preprocessing.image.img_to_array(img)
img_array = tf.expand_dims(img_array, 0)

# Make predictions
predictions = loaded_model.predict(img_array)
class_labels = classes
score = tf.nn.softmax(predictions[0])
print(f"{class_labels[tf.argmax(score)]}")
```

```
1/1 [==============================] - 1s 1s/step
VeryMildDemented
```