99-8-acc-alzheimer-detection-and-classification

May 19, 2022

1 Alzheimer MRI Preprocessed(detection and classification)

2 LIBRARIES

[1]:
```
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np
import os
import tensorflow as tf
from tensorflow import keras
from keras import layers
import matplotlib.image as img
%matplotlib inline
```

3 DATA LOAD

[3]:
```
!pip install split-folders
import splitfolders
splitfolders.ratio('Dataset', output="output", seed=1345, ratio=(.8, 0.1,0.1))
```

```
WARNING: Ignoring invalid distribution -5py (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -orchvision
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -orch (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution - (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -arkupsafe
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ackaging
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -5py (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -orchvision
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -orch (c:\programdata\anaconda3\lib\site-
```

```
packages)
WARNING: Ignoring invalid distribution - (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -arkupsafe
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ackaging
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -5py (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -orchvision
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -orch (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution - (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -arkupsafe
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ackaging
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -5py (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -orchvision
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -orch (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution - (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -arkupsafe
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ackaging
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -5py (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -orchvision
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -orch (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution - (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -arkupsafe
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ackaging
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -5py (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -orchvision
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -orch (c:\programdata\anaconda3\lib\site-
```

```
packages)
WARNING: Ignoring invalid distribution - (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -arkupsafe
(c:\programdata\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ackaging
(c:\programdata\anaconda3\lib\site-packages)
```

```
Requirement already satisfied: split-folders in
c:\programdata\anaconda3\lib\site-packages (0.5.1)
```

```
Copying files: 6400 files [00:32, 199.11 files/s]
```

[4]:

```
IMG_HEIGHT = 128
IMG_WIDTH = 128
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
"./output/train",
seed=123,
image_size=(IMG_HEIGHT, IMG_WIDTH),
batch_size=64
)

test_ds = tf.keras.preprocessing.image_dataset_from_directory(
"./output/test",
seed=123,
image_size=(IMG_HEIGHT, IMG_WIDTH),
batch_size=64
)

val_ds = tf.keras.preprocessing.image_dataset_from_directory(
"./output/val",
seed=123,
image_size=(IMG_HEIGHT, IMG_WIDTH),
batch_size=64
)
```

```
Found 5119 files belonging to 4 classes.
Found 642 files belonging to 4 classes.
Found 639 files belonging to 4 classes.
```

[5]:

```
class_names = train_ds.class_names
print(class_names)
train_ds
```

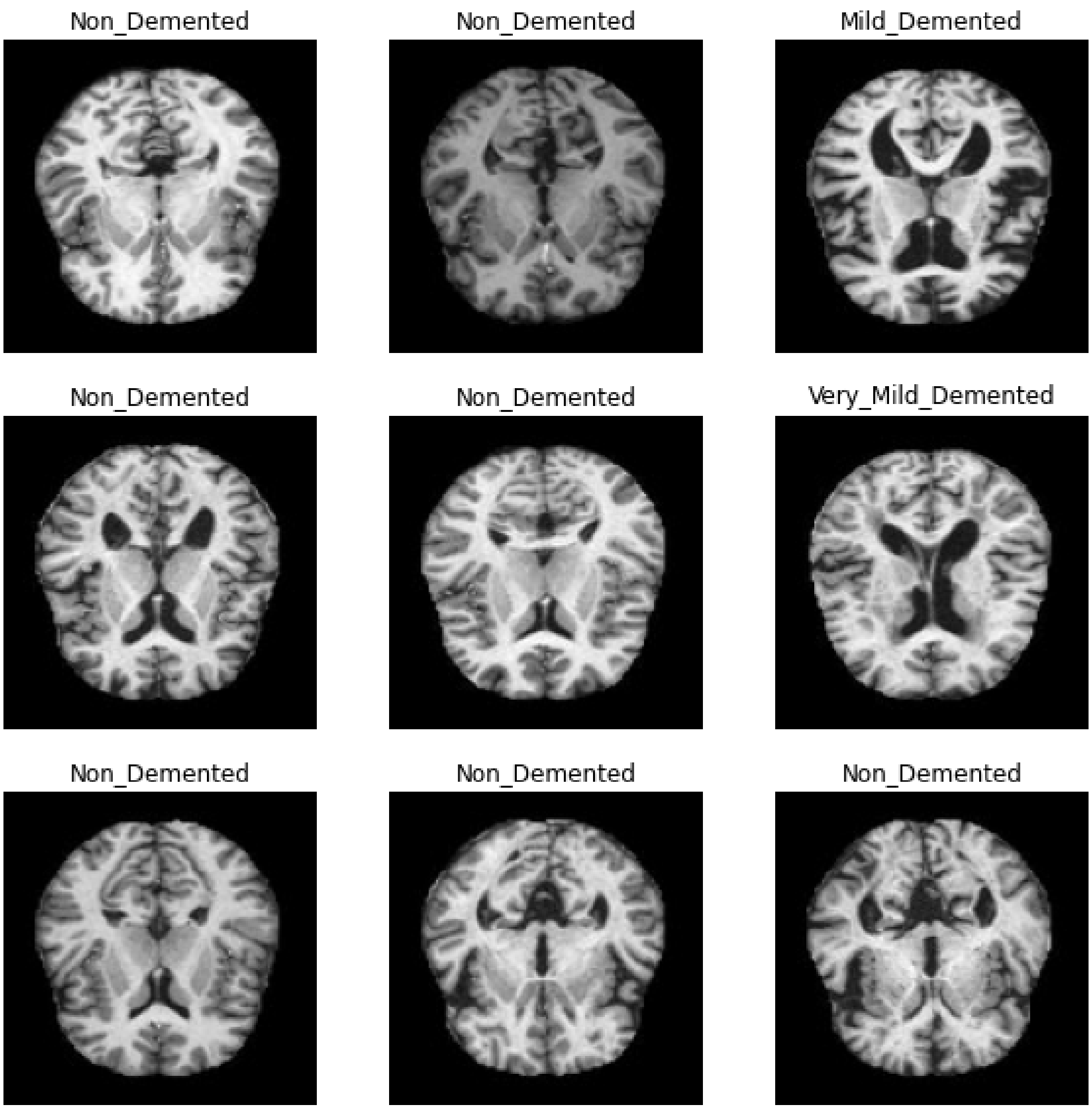
```
['Mild_Demented', 'Moderate_Demented', 'Non_Demented', 'Very_Mild_Demented']
```

[5]:

```
<BatchDataset element_spec=(TensorSpec(shape=(None, 128, 128, 3),
dtype=tf.float32, name=None), TensorSpec(shape=(None,), dtype=tf.int32,
name=None))>
```

4 EXAMPLE IMAGE

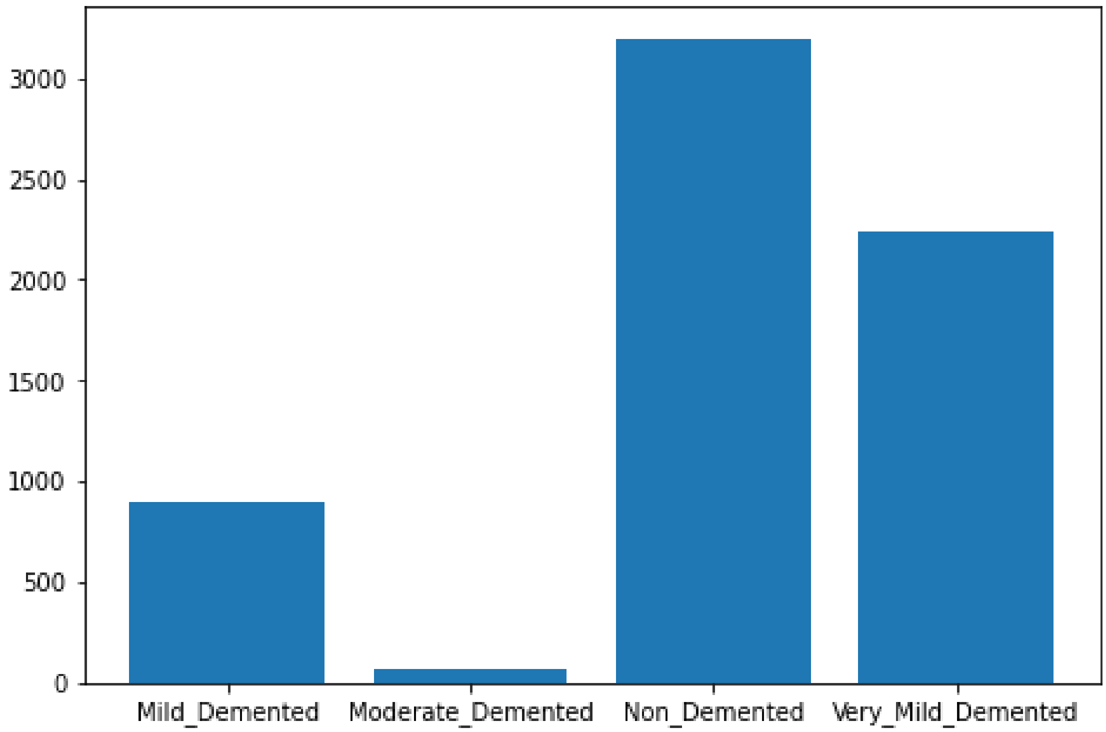
[6]:
```
plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
```



[7]:
```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
```

```
size = [896,64,3200,2240]
ax.bar(class_names,size)
plt.show
```

[7]: <function matplotlib.pyplot.show(close=None, block=None)>



5 MODEL

[8]:
```
model = keras.models.Sequential()
model.add(keras.layers.experimental.preprocessing.Rescaling(1./255,␣
↪input_shape=(IMG_HEIGHT,IMG_WIDTH, 3)))
model.add(keras.layers.
↪Conv2D(filters=16,kernel_size=(3,3),padding='same',activation='relu',kernel_initializer="he
model.add(keras.layers.MaxPooling2D(pool_size=(2,2)))


model.add(keras.layers.
↪Conv2D(filters=32,kernel_size=(3,3),padding='same',activation='relu',kernel_initializer="he
model.add(keras.layers.MaxPooling2D(pool_size=(2,2)))

model.add(keras.layers.Dropout(0.20))
```

```
model.add(keras.layers.
 ↪Conv2D(filters=64,kernel_size=(3,3),padding='same',activation='relu',kernel_initializer="he
model.add(keras.layers.MaxPooling2D(pool_size=(2,2)))

model.add(keras.layers.Dropout(0.25))
model.add(keras.layers.Flatten())
model.add(keras.layers.
 ↪Dense(128,activation="relu",kernel_initializer="he_normal"))
model.add(keras.layers.Dense(64,"relu"))
model.add(keras.layers.Dense(4,"softmax"))
```

[9]:
```
model.compile(loss="sparse_categorical_crossentropy",
              optimizer = "Adam",metrics=["accuracy"])
```

[10]:
```
model.summary()
```

```
Model: "sequential"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling (Rescaling)       (None, 128, 128, 3)       0

 conv2d (Conv2D)             (None, 128, 128, 16)      448

 max_pooling2d (MaxPooling2D  (None, 64, 64, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 64, 64, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 32, 32, 32)       0
 2D)

 dropout (Dropout)           (None, 32, 32, 32)        0

 conv2d_2 (Conv2D)           (None, 32, 32, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 16, 16, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 16, 16, 64)        0

 flatten (Flatten)           (None, 16384)             0

 dense (Dense)               (None, 128)               2097280

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 4)                 260
```

```
=================================================================
Total params: 2,129,380
Trainable params: 2,129,380
Non-trainable params: 0
_________________________________________________________________
```

[11]:
```
hist = model.fit(train_ds,validation_data=val_ds,epochs=100, batch_size=64,␣
 ↪verbose=1)
```

```
Epoch 1/100
80/80 [==============================] - 34s 400ms/step - loss: 1.1069 -
accuracy: 0.4989 - val_loss: 0.9679 - val_accuracy: 0.5837
Epoch 2/100
80/80 [==============================] - 30s 377ms/step - loss: 0.9084 -
accuracy: 0.5679 - val_loss: 0.8599 - val_accuracy: 0.6103
Epoch 3/100
80/80 [==============================] - 31s 382ms/step - loss: 0.8467 -
accuracy: 0.6095 - val_loss: 0.7424 - val_accuracy: 0.6886
Epoch 4/100
80/80 [==============================] - 31s 385ms/step - loss: 0.7464 -
accuracy: 0.6730 - val_loss: 0.7297 - val_accuracy: 0.6823
Epoch 5/100
80/80 [==============================] - 31s 385ms/step - loss: 0.6275 -
accuracy: 0.7259 - val_loss: 0.5482 - val_accuracy: 0.7966
Epoch 6/100
80/80 [==============================] - 32s 397ms/step - loss: 0.5262 -
accuracy: 0.7738 - val_loss: 0.4868 - val_accuracy: 0.8075
Epoch 7/100
80/80 [==============================] - 34s 421ms/step - loss: 0.3883 -
accuracy: 0.8412 - val_loss: 0.3425 - val_accuracy: 0.8623
Epoch 8/100
80/80 [==============================] - 32s 394ms/step - loss: 0.3274 -
accuracy: 0.8711 - val_loss: 0.2788 - val_accuracy: 0.9030
Epoch 9/100
80/80 [==============================] - 32s 397ms/step - loss: 0.2486 -
accuracy: 0.9012 - val_loss: 0.1804 - val_accuracy: 0.9437
Epoch 10/100
80/80 [==============================] - 32s 397ms/step - loss: 0.2127 -
accuracy: 0.9219 - val_loss: 0.1765 - val_accuracy: 0.9374
Epoch 11/100
80/80 [==============================] - 32s 394ms/step - loss: 0.1790 -
accuracy: 0.9349 - val_loss: 0.0916 - val_accuracy: 0.9718
Epoch 12/100
80/80 [==============================] - 32s 397ms/step - loss: 0.1386 -
accuracy: 0.9500 - val_loss: 0.1035 - val_accuracy: 0.9765
Epoch 13/100
80/80 [==============================] - 32s 401ms/step - loss: 0.1044 -
```

```
accuracy: 0.9633 - val_loss: 0.0685 - val_accuracy: 0.9812
Epoch 14/100
80/80 [==============================] - 32s 400ms/step - loss: 0.1258 -
accuracy: 0.9537 - val_loss: 0.0872 - val_accuracy: 0.9765
Epoch 15/100
80/80 [==============================] - 32s 401ms/step - loss: 0.0897 -
accuracy: 0.9666 - val_loss: 0.0707 - val_accuracy: 0.9765
Epoch 16/100
80/80 [==============================] - 33s 416ms/step - loss: 0.0816 -
accuracy: 0.9709 - val_loss: 0.1023 - val_accuracy: 0.9593
Epoch 17/100
80/80 [==============================] - 32s 400ms/step - loss: 0.0754 -
accuracy: 0.9721 - val_loss: 0.0561 - val_accuracy: 0.9781
Epoch 18/100
80/80 [==============================] - 32s 397ms/step - loss: 0.0576 -
accuracy: 0.9801 - val_loss: 0.0326 - val_accuracy: 0.9875
Epoch 19/100
80/80 [==============================] - 32s 397ms/step - loss: 0.0714 -
accuracy: 0.9732 - val_loss: 0.0814 - val_accuracy: 0.9703
Epoch 20/100
80/80 [==============================] - 32s 397ms/step - loss: 0.0567 -
accuracy: 0.9822 - val_loss: 0.0470 - val_accuracy: 0.9812
Epoch 21/100
80/80 [==============================] - 32s 396ms/step - loss: 0.0706 -
accuracy: 0.9754 - val_loss: 0.0577 - val_accuracy: 0.9844
Epoch 22/100
80/80 [==============================] - 32s 395ms/step - loss: 0.0485 -
accuracy: 0.9834 - val_loss: 0.0350 - val_accuracy: 0.9922
Epoch 23/100
80/80 [==============================] - 32s 395ms/step - loss: 0.0591 -
accuracy: 0.9805 - val_loss: 0.0390 - val_accuracy: 0.9875
Epoch 24/100
80/80 [==============================] - 32s 398ms/step - loss: 0.0480 -
accuracy: 0.9830 - val_loss: 0.0460 - val_accuracy: 0.9828
Epoch 25/100
80/80 [==============================] - 32s 397ms/step - loss: 0.0431 -
accuracy: 0.9857 - val_loss: 0.0282 - val_accuracy: 0.9906
Epoch 26/100
80/80 [==============================] - 32s 397ms/step - loss: 0.0379 -
accuracy: 0.9871 - val_loss: 0.0218 - val_accuracy: 0.9953
Epoch 27/100
80/80 [==============================] - 32s 401ms/step - loss: 0.0334 -
accuracy: 0.9883 - val_loss: 0.0393 - val_accuracy: 0.9875
Epoch 28/100
80/80 [==============================] - 32s 398ms/step - loss: 0.0373 -
accuracy: 0.9867 - val_loss: 0.0366 - val_accuracy: 0.9844
Epoch 29/100
80/80 [==============================] - 32s 400ms/step - loss: 0.0366 -
```

```
accuracy: 0.9889 - val_loss: 0.0728 - val_accuracy: 0.9718
Epoch 30/100
80/80 [==============================] - 32s 396ms/step - loss: 0.0334 -
accuracy: 0.9891 - val_loss: 0.0273 - val_accuracy: 0.9906
Epoch 31/100
80/80 [==============================] - 32s 395ms/step - loss: 0.0431 -
accuracy: 0.9848 - val_loss: 0.0164 - val_accuracy: 0.9969
Epoch 32/100
80/80 [==============================] - 32s 396ms/step - loss: 0.0245 -
accuracy: 0.9924 - val_loss: 0.0357 - val_accuracy: 0.9906
Epoch 33/100
80/80 [==============================] - 32s 398ms/step - loss: 0.0317 -
accuracy: 0.9891 - val_loss: 0.0171 - val_accuracy: 0.9922
Epoch 34/100
80/80 [==============================] - 32s 399ms/step - loss: 0.0269 -
accuracy: 0.9908 - val_loss: 0.0087 - val_accuracy: 0.9984
Epoch 35/100
80/80 [==============================] - 32s 397ms/step - loss: 0.0288 -
accuracy: 0.9893 - val_loss: 0.0129 - val_accuracy: 0.9969
Epoch 36/100
80/80 [==============================] - 32s 401ms/step - loss: 0.0313 -
accuracy: 0.9898 - val_loss: 0.0133 - val_accuracy: 0.9953
Epoch 37/100
80/80 [==============================] - 32s 398ms/step - loss: 0.0367 -
accuracy: 0.9865 - val_loss: 0.0101 - val_accuracy: 0.9984
Epoch 38/100
80/80 [==============================] - 34s 418ms/step - loss: 0.0337 -
accuracy: 0.9896 - val_loss: 0.0145 - val_accuracy: 0.9969
Epoch 39/100
80/80 [==============================] - 38s 478ms/step - loss: 0.0301 -
accuracy: 0.9891 - val_loss: 0.0243 - val_accuracy: 0.9890
Epoch 40/100
80/80 [==============================] - 51s 640ms/step - loss: 0.0295 -
accuracy: 0.9893 - val_loss: 0.0165 - val_accuracy: 0.9937
Epoch 41/100
80/80 [==============================] - 52s 642ms/step - loss: 0.0192 -
accuracy: 0.9939 - val_loss: 0.0104 - val_accuracy: 0.9984
Epoch 42/100
80/80 [==============================] - 51s 634ms/step - loss: 0.0150 -
accuracy: 0.9941 - val_loss: 0.0053 - val_accuracy: 1.0000
Epoch 43/100
80/80 [==============================] - 51s 638ms/step - loss: 0.0229 -
accuracy: 0.9922 - val_loss: 0.0517 - val_accuracy: 0.9828
Epoch 44/100
80/80 [==============================] - 51s 637ms/step - loss: 0.0265 -
accuracy: 0.9906 - val_loss: 0.0113 - val_accuracy: 0.9984
Epoch 45/100
80/80 [==============================] - 52s 649ms/step - loss: 0.0233 -
```

```
accuracy: 0.9908 - val_loss: 0.0156 - val_accuracy: 0.9953
Epoch 46/100
80/80 [==============================] - 52s 648ms/step - loss: 0.0181 -
accuracy: 0.9937 - val_loss: 0.0179 - val_accuracy: 0.9953
Epoch 47/100
80/80 [==============================] - 51s 639ms/step - loss: 0.0260 -
accuracy: 0.9914 - val_loss: 0.0152 - val_accuracy: 0.9937
Epoch 48/100
80/80 [==============================] - 51s 637ms/step - loss: 0.0133 -
accuracy: 0.9947 - val_loss: 0.0073 - val_accuracy: 0.9984
Epoch 49/100
80/80 [==============================] - 52s 642ms/step - loss: 0.0184 -
accuracy: 0.9941 - val_loss: 0.0093 - val_accuracy: 0.9984
Epoch 50/100
80/80 [==============================] - 52s 642ms/step - loss: 0.0195 -
accuracy: 0.9936 - val_loss: 0.0155 - val_accuracy: 0.9953
Epoch 51/100
80/80 [==============================] - 51s 640ms/step - loss: 0.0204 -
accuracy: 0.9924 - val_loss: 0.0133 - val_accuracy: 0.9969
Epoch 52/100
80/80 [==============================] - 52s 641ms/step - loss: 0.0230 -
accuracy: 0.9922 - val_loss: 0.0172 - val_accuracy: 0.9969
Epoch 53/100
80/80 [==============================] - 51s 640ms/step - loss: 0.0326 -
accuracy: 0.9889 - val_loss: 0.0675 - val_accuracy: 0.9797
Epoch 54/100
80/80 [==============================] - 51s 635ms/step - loss: 0.0356 -
accuracy: 0.9891 - val_loss: 0.0477 - val_accuracy: 0.9828
Epoch 55/100
80/80 [==============================] - 52s 642ms/step - loss: 0.0264 -
accuracy: 0.9920 - val_loss: 0.0362 - val_accuracy: 0.9890
Epoch 56/100
80/80 [==============================] - 52s 643ms/step - loss: 0.0109 -
accuracy: 0.9969 - val_loss: 0.0051 - val_accuracy: 1.0000
Epoch 57/100
80/80 [==============================] - 51s 639ms/step - loss: 0.0180 -
accuracy: 0.9932 - val_loss: 0.0462 - val_accuracy: 0.9812
Epoch 58/100
80/80 [==============================] - 51s 638ms/step - loss: 0.0186 -
accuracy: 0.9945 - val_loss: 0.0206 - val_accuracy: 0.9937
Epoch 59/100
80/80 [==============================] - 51s 637ms/step - loss: 0.0155 -
accuracy: 0.9937 - val_loss: 0.0116 - val_accuracy: 0.9953
Epoch 60/100
80/80 [==============================] - 51s 639ms/step - loss: 0.0198 -
accuracy: 0.9943 - val_loss: 0.0045 - val_accuracy: 1.0000
Epoch 61/100
80/80 [==============================] - 51s 637ms/step - loss: 0.0096 -
```

```
accuracy: 0.9965 - val_loss: 0.0189 - val_accuracy: 0.9937
Epoch 62/100
80/80 [==============================] - 51s 637ms/step - loss: 0.0137 -
accuracy: 0.9949 - val_loss: 0.0154 - val_accuracy: 0.9937
Epoch 63/100
80/80 [==============================] - 51s 639ms/step - loss: 0.0094 -
accuracy: 0.9955 - val_loss: 0.0379 - val_accuracy: 0.9922
Epoch 64/100
80/80 [==============================] - 52s 641ms/step - loss: 0.0407 -
accuracy: 0.9867 - val_loss: 0.0397 - val_accuracy: 0.9875
Epoch 65/100
80/80 [==============================] - 52s 641ms/step - loss: 0.0120 -
accuracy: 0.9963 - val_loss: 0.0048 - val_accuracy: 0.9984
Epoch 66/100
80/80 [==============================] - 51s 639ms/step - loss: 0.0053 -
accuracy: 0.9979 - val_loss: 0.0036 - val_accuracy: 1.0000
Epoch 67/100
80/80 [==============================] - 51s 637ms/step - loss: 0.0097 -
accuracy: 0.9969 - val_loss: 0.0128 - val_accuracy: 0.9937
Epoch 68/100
80/80 [==============================] - 51s 637ms/step - loss: 0.0253 -
accuracy: 0.9893 - val_loss: 0.0170 - val_accuracy: 0.9937
Epoch 69/100
80/80 [==============================] - 51s 638ms/step - loss: 0.0245 -
accuracy: 0.9914 - val_loss: 0.0130 - val_accuracy: 0.9969
Epoch 70/100
80/80 [==============================] - 51s 635ms/step - loss: 0.0248 -
accuracy: 0.9910 - val_loss: 0.0168 - val_accuracy: 0.9906
Epoch 71/100
80/80 [==============================] - 51s 638ms/step - loss: 0.0216 -
accuracy: 0.9926 - val_loss: 0.0183 - val_accuracy: 0.9922
Epoch 72/100
80/80 [==============================] - 75s 936ms/step - loss: 0.0139 -
accuracy: 0.9953 - val_loss: 0.0421 - val_accuracy: 0.9890
Epoch 73/100
80/80 [==============================] - 89s 1s/step - loss: 0.0106 - accuracy:
0.9967 - val_loss: 0.0222 - val_accuracy: 0.9906
Epoch 74/100
80/80 [==============================] - 57s 708ms/step - loss: 0.0090 -
accuracy: 0.9975 - val_loss: 0.0209 - val_accuracy: 0.9937
Epoch 75/100
80/80 [==============================] - 52s 653ms/step - loss: 0.0102 -
accuracy: 0.9961 - val_loss: 0.0089 - val_accuracy: 0.9969
Epoch 76/100
80/80 [==============================] - 53s 656ms/step - loss: 0.0180 -
accuracy: 0.9949 - val_loss: 0.0121 - val_accuracy: 0.9969
Epoch 77/100
80/80 [==============================] - 53s 654ms/step - loss: 0.0100 -
```

```
accuracy: 0.9973 - val_loss: 0.0103 - val_accuracy: 0.9969
Epoch 78/100
80/80 [==============================] - 53s 658ms/step - loss: 0.0092 -
accuracy: 0.9969 - val_loss: 0.0427 - val_accuracy: 0.9890
Epoch 79/100
80/80 [==============================] - 52s 650ms/step - loss: 0.0187 -
accuracy: 0.9939 - val_loss: 0.0215 - val_accuracy: 0.9922
Epoch 80/100
80/80 [==============================] - 52s 643ms/step - loss: 0.0133 -
accuracy: 0.9941 - val_loss: 0.0210 - val_accuracy: 0.9953
Epoch 81/100
80/80 [==============================] - 52s 645ms/step - loss: 0.0122 -
accuracy: 0.9967 - val_loss: 0.0226 - val_accuracy: 0.9953
Epoch 82/100
80/80 [==============================] - 52s 648ms/step - loss: 0.0115 -
accuracy: 0.9953 - val_loss: 0.0310 - val_accuracy: 0.9906
Epoch 83/100
80/80 [==============================] - 44s 547ms/step - loss: 0.0177 -
accuracy: 0.9924 - val_loss: 0.0110 - val_accuracy: 0.9969
Epoch 84/100
80/80 [==============================] - 36s 453ms/step - loss: 0.0147 -
accuracy: 0.9943 - val_loss: 0.0097 - val_accuracy: 0.9953
Epoch 85/100
80/80 [==============================] - 36s 450ms/step - loss: 0.0146 -
accuracy: 0.9949 - val_loss: 0.0083 - val_accuracy: 0.9953
Epoch 86/100
80/80 [==============================] - 36s 452ms/step - loss: 0.0141 -
accuracy: 0.9943 - val_loss: 0.0078 - val_accuracy: 0.9984
Epoch 87/100
80/80 [==============================] - 36s 449ms/step - loss: 0.0216 -
accuracy: 0.9939 - val_loss: 0.0062 - val_accuracy: 0.9984
Epoch 88/100
80/80 [==============================] - 37s 455ms/step - loss: 0.0056 -
accuracy: 0.9979 - val_loss: 0.0016 - val_accuracy: 1.0000
Epoch 89/100
80/80 [==============================] - 36s 452ms/step - loss: 0.0138 -
accuracy: 0.9953 - val_loss: 0.0195 - val_accuracy: 0.9906
Epoch 90/100
80/80 [==============================] - 36s 454ms/step - loss: 0.0114 -
accuracy: 0.9965 - val_loss: 0.0017 - val_accuracy: 1.0000
Epoch 91/100
80/80 [==============================] - 36s 450ms/step - loss: 0.0200 -
accuracy: 0.9920 - val_loss: 0.0114 - val_accuracy: 0.9969
Epoch 92/100
80/80 [==============================] - 37s 456ms/step - loss: 0.0179 -
accuracy: 0.9939 - val_loss: 0.0089 - val_accuracy: 0.9953
Epoch 93/100
80/80 [==============================] - 36s 451ms/step - loss: 0.0096 -
```

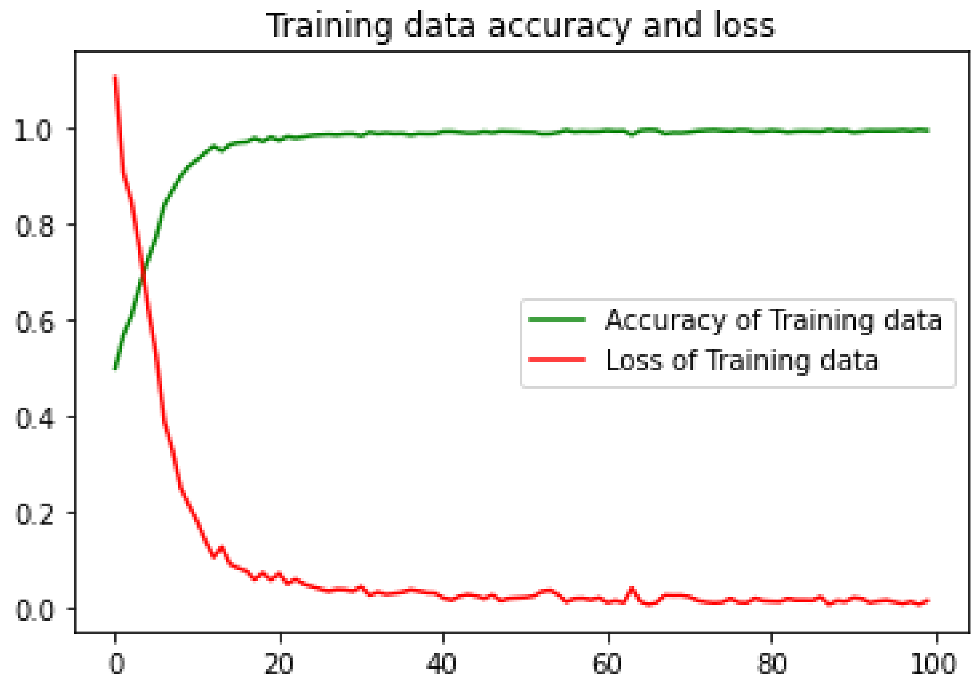
```
accuracy: 0.9961 - val_loss: 0.0172 - val_accuracy: 0.9953
Epoch 94/100
80/80 [==============================] - 36s 450ms/step - loss: 0.0126 -
accuracy: 0.9957 - val_loss: 0.0043 - val_accuracy: 0.9984
Epoch 95/100
80/80 [==============================] - 36s 450ms/step - loss: 0.0145 -
accuracy: 0.9961 - val_loss: 0.0383 - val_accuracy: 0.9890
Epoch 96/100
80/80 [==============================] - 36s 454ms/step - loss: 0.0110 -
accuracy: 0.9961 - val_loss: 0.0071 - val_accuracy: 0.9969
Epoch 97/100
80/80 [==============================] - 36s 450ms/step - loss: 0.0071 -
accuracy: 0.9973 - val_loss: 0.0122 - val_accuracy: 0.9984
Epoch 98/100
80/80 [==============================] - 36s 452ms/step - loss: 0.0117 -
accuracy: 0.9959 - val_loss: 0.0084 - val_accuracy: 0.9984
Epoch 99/100
80/80 [==============================] - 37s 460ms/step - loss: 0.0057 -
accuracy: 0.9979 - val_loss: 0.0084 - val_accuracy: 0.9969
Epoch 100/100
80/80 [==============================] - 36s 446ms/step - loss: 0.0134 -
accuracy: 0.9965 - val_loss: 0.0086 - val_accuracy: 0.9984
```

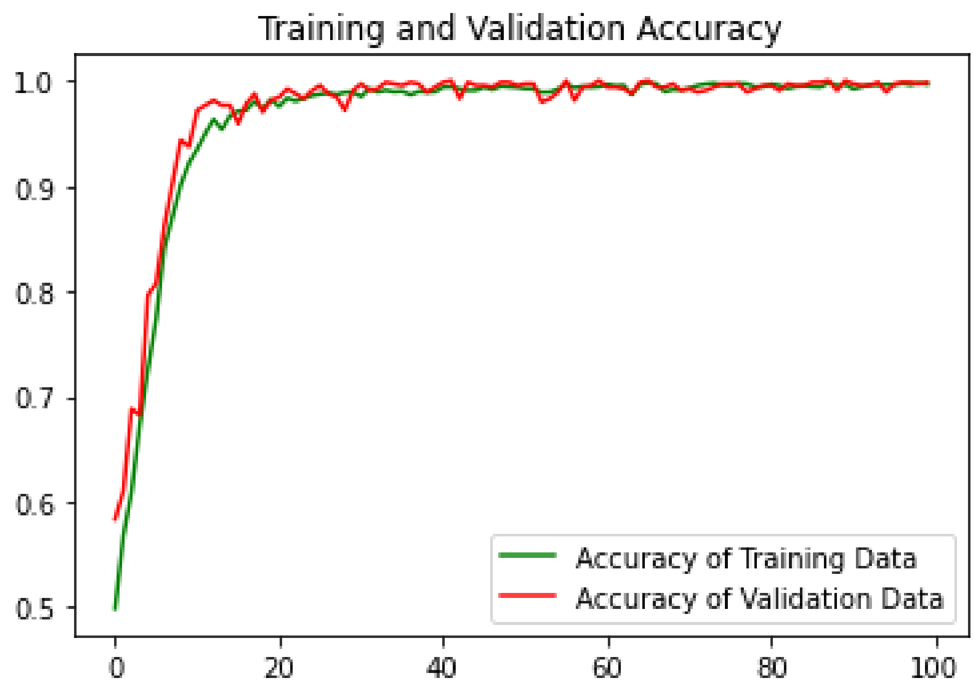
6 Plot the result

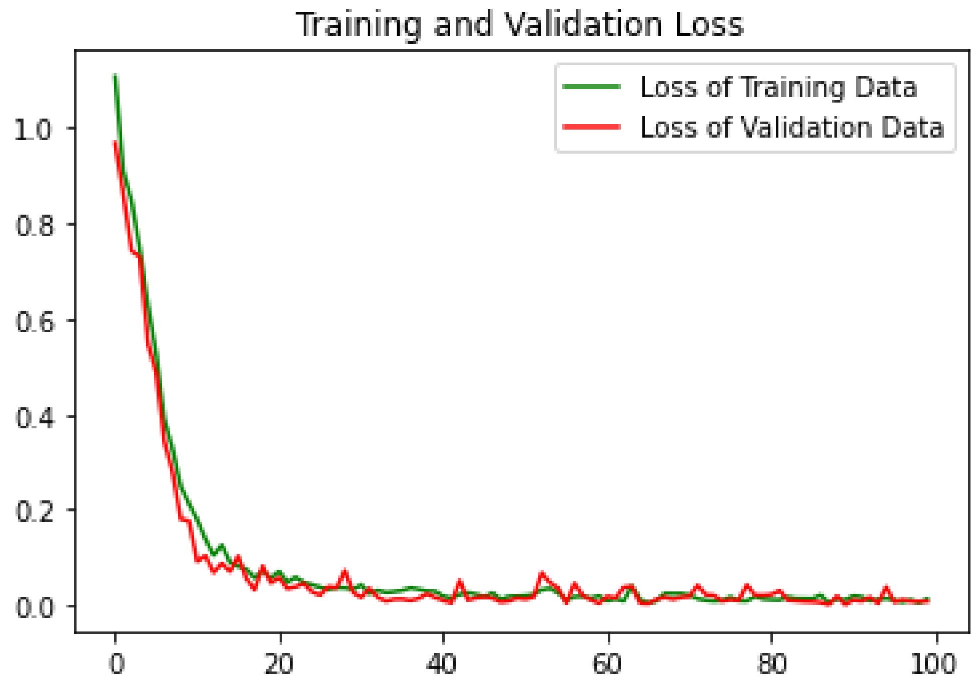
[12]:
```
get_ac = hist.history['accuracy']
get_los = hist.history['loss']
val_acc = hist.history['val_accuracy']
val_loss = hist.history['val_loss']
```

[13]:
```
epochs = range(len(get_ac))
plt.plot(epochs, get_ac, 'g', label='Accuracy of Training data')
plt.plot(epochs, get_los, 'r', label='Loss of Training data')
plt.title('Training data accuracy and loss')
plt.legend(loc=0)
plt.figure()

plt.plot(epochs, get_ac, 'g', label='Accuracy of Training Data')
plt.plot(epochs, val_acc, 'r', label='Accuracy of Validation Data')
plt.title('Training and Validation Accuracy')
plt.legend(loc=0)
plt.figure()

plt.plot(epochs, get_los, 'g', label='Loss of Training Data')
plt.plot(epochs, val_loss, 'r', label='Loss of Validation Data')
plt.title('Training and Validation Loss')
plt.legend(loc=0)
```

```
plt.figure()
plt.show()
```








```
<Figure size 432x288 with 0 Axes>
```

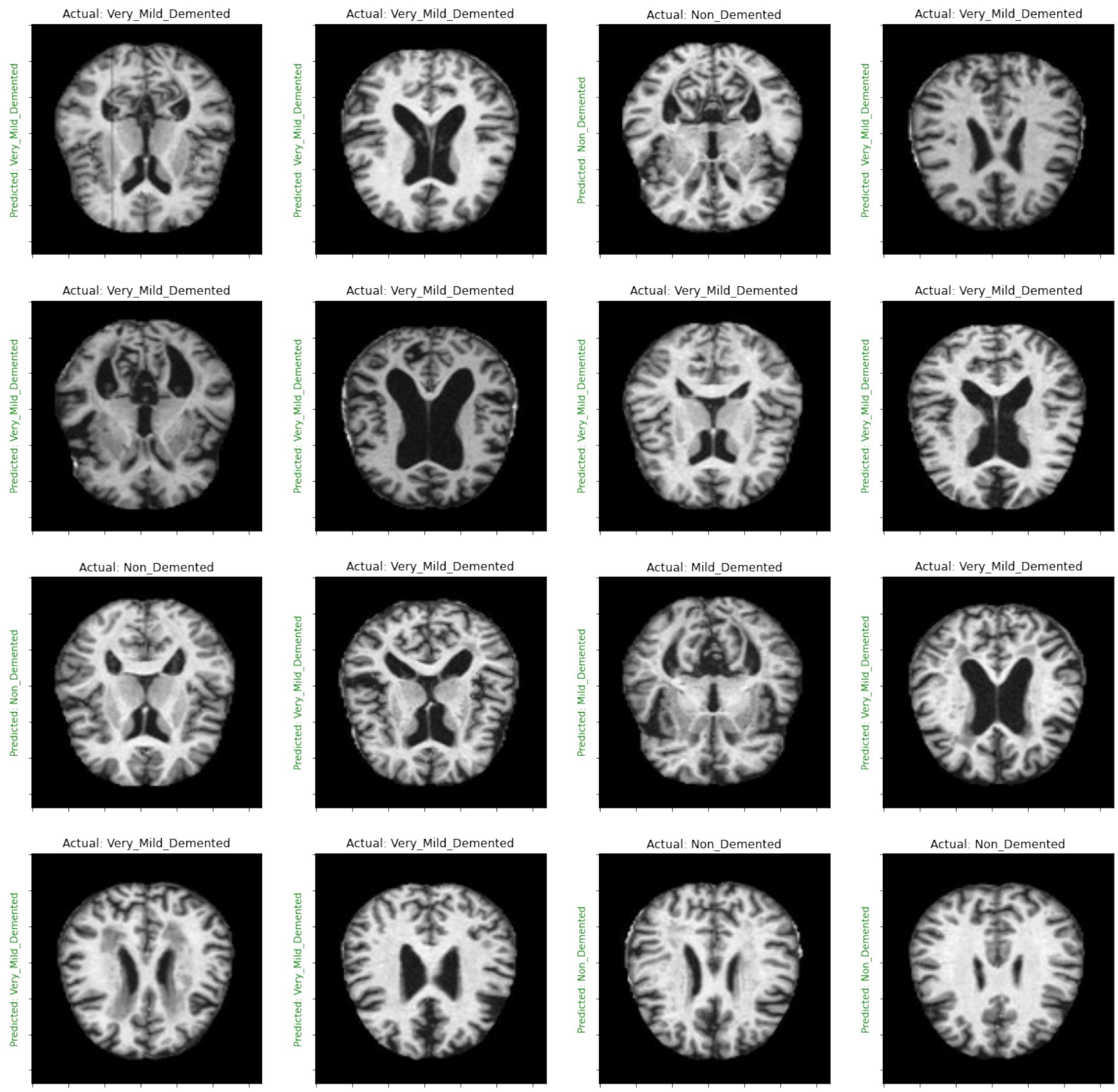
7 Predictions

[14]:
```
loss, accuracy = model.evaluate(test_ds)
```

```
11/11 [==============================] - 2s 93ms/step - loss: 0.0193 - accuracy:
0.9938
```

[15]:
```
plt.figure(figsize=(20, 20))
for images, labels in test_ds.take(1):
    for i in range(16):
        ax = plt.subplot(4, 4, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        predictions = model.predict(tf.expand_dims(images[i], 0))
        score = tf.nn.softmax(predictions[0])
        if(class_names[labels[i]]==class_names[np.argmax(score)]):
            plt.title("Actual: "+class_names[labels[i]])
            plt.ylabel("Predicted: "+class_names[np.
 ↪argmax(score)],fontdict={'color':'green'})

        else:
            plt.title("Actual: "+class_names[labels[i]])
            plt.ylabel("Predicted: "+class_names[np.
 ↪argmax(score)],fontdict={'color':'red'})
```

```
    plt.gca().axes.yaxis.set_ticklabels([])
    plt.gca().axes.xaxis.set_ticklabels([])
```



[ ]:

```
```