classification-using-resnet-50-v2

March 28, 2024

[35]:

```
trainpath=r"D:\Winter Semester 3\AIML\Project\Alzheiemer_image\Alzheimer_s␣
  ↪Dataset\train"
testpath=r"D:\Winter Semester 3\AIML\Project\Alzheiemer_image\Alzheimer_s␣
  ↪Dataset\test"
```

[36]:

```
import tensorflow
from tensorflow.keras.layers import Dense,Flatten,Input,Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator as IDG,␣
  ↪load_img
from tensorflow.keras.applications import Xception
from tensorflow.keras import layers,models
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import␣
  ↪SeparableConv2D,BatchNormalization,GlobalAveragePooling2D
from tensorflow.keras.models import load_model
from PIL import Image
from tensorflow.keras.applications.xception import preprocess_input
import numpy as np
```

[37]:

```
import tensorflow
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
img_size=180
img_size_dim=[180,180]
dimension=(img_size,img_size)
zoom=[0.99,1.01]
bright=[0.8,1.2]
fill_mode='constant'
data_format='channels_last'
dir="D:/Winter Semester 3/AIML/Project\Alzheiemer_image/Alzheimer_s Dataset/
  ↪train"
data_gen = ImageDataGenerator(rescale=1./
  ↪255,brightness_range=bright,zoom_range=zoom,data_format=data_format,fill_mode=fill_mode,hor
```

```
#data_gen=IDG(rescale=1./
↪255,brightness_range=bright,zoom_range=zoom,data_format=data_format,fill_mode=fill_mode,hor
train_data_gen=data_gen.flow_from_directory(directory="D:/Winter Semester 3/
↪AIML/Project/Alzheiemer_image/Alzheimer_s Dataset/
↪train",target_size=dimension,batch_size=6000,shuffle=False)
```

```
Found 5121 images belonging to 4 classes.
```

[38]:

```
# Fetch a batch of training data and labels
train_data, train_labels = train_data_gen.__next__()
# Print the shape of the training data and labels
print(train_data.shape, train_labels.shape)
```

```
(5121, 180, 180, 3) (5121, 4)
```

[39]:

```
#handling imbalanced classes
from imblearn.over_sampling import SMOTE
sm=SMOTE(random_state=47)
train_data,train_labels=sm.fit_resample(train_data.
↪reshape(-1,img_size*img_size*3),train_labels)
train_data=train_data.reshape(-1,img_size,img_size,3)
print(train_data.shape,train_labels.shape)
```

```
(10240, 180, 180, 3) (10240, 4)
```

[40]:

```
from sklearn.model_selection import train_test_split
train_data,test_data,train_labels,test_labels =␣
↪train_test_split(train_data,train_labels,test_size=0.2,random_state=47)
train_data,val_data,train_labels,val_labels =␣
↪train_test_split(train_data,train_labels,test_size=0.2,random_state=47)
```

[41]:

```
import tensorflow as tf
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define input shape and number of classes
input_shape = (224, 224, 3)
num_classes = 4  # Alzheimer's and non-Alzheimer's

# Load ResNet-50 model with pre-trained weights
base_model = ResNet50(weights='imagenet', include_top=False,␣
↪input_shape=input_shape)

# Freeze some layers and fine-tune others
for layer in base_model.layers[:-10]:
```

```python
    layer.trainable = False

# Add custom classification layers on top of ResNet-50
x = Flatten()(base_model.output)
x = Dense(256, activation='relu')(x)  # Increase the number of neurons
x = Dense(128, activation='relu')(x)
predictions = Dense(num_classes, activation='softmax')(x)

# Create the final model
model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model with a lower learning rate
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001),
              loss='sparse_categorical_crossentropy',  # for integer labels
              metrics=['accuracy'])
```

[57]:

```python
model.summary()
```

```
Model: "sequential_3"


 Layer (type)                      Output Shape                  Param #

 conv2d_4 (Conv2D)                 (None, 178, 178, 32)              896

 max_pooling2d_4 (MaxPooling2D)    (None, 89, 89, 32)                  0

 conv2d_5 (Conv2D)                 (None, 87, 87, 64)             18,496

 max_pooling2d_5 (MaxPooling2D)    (None, 43, 43, 64)                  0

 flatten_4 (Flatten)               (None, 118336)                      0

 dense_19 (Dense)                  (None, 64)                  7,573,568

 dense_20 (Dense)                  (None, 4)                         260


 Total params: 7,593,220 (28.97 MB)

 Trainable params: 7,593,220 (28.97 MB)

 Non-trainable params: 0 (0.00 B)
```

[46]:

```
mt=[tensorflow.keras.metrics.CategoricalAccuracy(name='Accuracy'),tensorflow.
 ↪keras.metrics.AUC(name='AUC')]
```

[1]:

```
import tensorflow as tf
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define input shape and number of classes
input_shape = (224, 224, 3)
num_classes = 4  # Alzheimer's and non-Alzheimer's

# Load ResNet-50 model with pre-trained weights
base_model = ResNet50(weights='imagenet', include_top=False,␣
 ↪input_shape=input_shape)

# Freeze some layers and fine-tune others
for layer in base_model.layers[:-10]:
    layer.trainable = False

# Add custom classification layers on top of ResNet-50
x = Flatten()(base_model.output)
x = Dense(256, activation='relu')(x)  # Increase the number of neurons
x = Dense(128, activation='relu')(x)
predictions = Dense(num_classes, activation='softmax')(x)

# Create the final model
model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model with a lower learning rate
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001),
              loss='sparse_categorical_crossentropy',  # for integer labels
              metrics=['accuracy'])

# Define data generators with data augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)
train_generator = train_datagen.flow_from_directory(
    "D:/Winter Semester 3/AIML/Project/Alzheiemer_image/Alzheimer_s Dataset/
  ↪train",
    target_size=(224, 224),
    batch_size=32,
    class_mode='binary',  # binary classification
    subset='training'
)
validation_generator = train_datagen.flow_from_directory(
    r"D:/Winter Semester 3/AIML/Project/Alzheiemer_image/Alzheimer_s Dataset/
  ↪train",
    target_size=(224, 224),
    batch_size=32,
    class_mode='binary',  # binary classification
    subset='validation'
)

# Train the model with increased epochs
history=model.fit(train_generator, epochs=10,␣
  ↪validation_data=validation_generator)
```

```
Found 4098 images belonging to 4 classes.
Found 1023 images belonging to 4 classes.
Epoch 1/10
```

```
c:\Users\harsa\AppData\Local\Programs\Python\Python310\lib\site-
packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:122:
UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in
its constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will be
ignored.
  self._warn_if_super_not_called()
```

```
129/129              286s 2s/step -
accuracy: 0.4609 - loss: 1.2016 - val_accuracy: 0.4907 - val_loss: 1.0399
Epoch 2/10
129/129              285s 2s/step -
accuracy: 0.5348 - loss: 0.9941 - val_accuracy: 0.4213 - val_loss: 1.1297
Epoch 3/10
129/129              299s 2s/step -
accuracy: 0.5073 - loss: 0.9986 - val_accuracy: 0.4985 - val_loss: 1.1067
Epoch 4/10
129/129              286s 2s/step -
accuracy: 0.4944 - loss: 1.0155 - val_accuracy: 0.4936 - val_loss: 1.2130
Epoch 5/10
129/129              293s 2s/step -
accuracy: 0.5255 - loss: 0.9826 - val_accuracy: 0.3910 - val_loss: 1.1155
Epoch 6/10
129/129              296s 2s/step -
```

```
accuracy: 0.5347 - loss: 0.9494 - val_accuracy: 0.4936 - val_loss: 1.3215
Epoch 7/10
129/129             292s 2s/step -
accuracy: 0.5414 - loss: 0.9495 - val_accuracy: 0.5249 - val_loss: 1.0896
Epoch 8/10
129/129             284s 2s/step -
accuracy: 0.5404 - loss: 0.9491 - val_accuracy: 0.4976 - val_loss: 1.2133
Epoch 9/10
129/129             285s 2s/step -
accuracy: 0.5413 - loss: 0.9352 - val_accuracy: 0.4946 - val_loss: 1.2045
Epoch 10/10
129/129             286s 2s/step -
accuracy: 0.5481 - loss: 0.9118 - val_accuracy: 0.3822 - val_loss: 1.1716
```

[58]:

```python
from keras.models import Model

def freeze_layers(model):
    for i in model.layers:
        i.trainable = False
        if isinstance(i, Model):
            freeze_layers(i)
    return model
```

[59]:

```python
model_freezed = freeze_layers(model)
model_freezed.save('alzheimer_res3.h5')
```

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

[60]:

```python
model_freezed.summary()
```

```
Model: "functional_24"


 Layer (type)          Output Shape            Param #   Connected to

 input_layer_14        (None, 224, 224,              0   -
 (InputLayer)          3)

 conv1_pad             (None, 230, 230,              0   input_layer_14[0…
 (ZeroPadding2D)       3)

 conv1_conv (Conv2D)   (None, 112, 112,          9,472   conv1_pad[0][0]
                       64)
```

```
conv1_bn              (None, 112, 112,          256  conv1_conv[0][0]
(BatchNormalizatio…   64)

conv1_relu            (None, 112, 112,            0  conv1_bn[0][0]
(Activation)          64)

pool1_pad             (None, 114, 114,            0  conv1_relu[0][0]
(ZeroPadding2D)       64)

pool1_pool            (None, 56, 56,              0  pool1_pad[0][0]
(MaxPooling2D)        64)

conv2_block1_1_conv   (None, 56, 56,          4,160  pool1_pool[0][0]
(Conv2D)              64)

conv2_block1_1_bn     (None, 56, 56,            256  conv2_block1_1_c…
(BatchNormalizatio…   64)

conv2_block1_1_relu   (None, 56, 56,              0  conv2_block1_1_b…
(Activation)          64)

conv2_block1_2_conv   (None, 56, 56,         36,928  conv2_block1_1_r…
(Conv2D)              64)

conv2_block1_2_bn     (None, 56, 56,            256  conv2_block1_2_c…
(BatchNormalizatio…   64)

conv2_block1_2_relu   (None, 56, 56,              0  conv2_block1_2_b…
(Activation)          64)

conv2_block1_0_conv   (None, 56, 56,         16,640  pool1_pool[0][0]
(Conv2D)              256)

conv2_block1_3_conv   (None, 56, 56,         16,640  conv2_block1_2_r…
(Conv2D)              256)

conv2_block1_0_bn     (None, 56, 56,          1,024  conv2_block1_0_c…
(BatchNormalizatio…   256)

conv2_block1_3_bn     (None, 56, 56,          1,024  conv2_block1_3_c…
(BatchNormalizatio…   256)

conv2_block1_add      (None, 56, 56,              0  conv2_block1_0_b…
(Add)                 256)                           conv2_block1_3_b…

conv2_block1_out      (None, 56, 56,              0  conv2_block1_add…
(Activation)          256)
```

```
conv2_block2_1_conv   (None, 56, 56,         16,448   conv2_block1_out…
(Conv2D)              64)

conv2_block2_1_bn     (None, 56, 56,            256   conv2_block2_1_c…
(BatchNormalizatio…   64)

conv2_block2_1_relu   (None, 56, 56,              0   conv2_block2_1_b…
(Activation)          64)

conv2_block2_2_conv   (None, 56, 56,         36,928   conv2_block2_1_r…
(Conv2D)              64)

conv2_block2_2_bn     (None, 56, 56,            256   conv2_block2_2_c…
(BatchNormalizatio…   64)

conv2_block2_2_relu   (None, 56, 56,              0   conv2_block2_2_b…
(Activation)          64)

conv2_block2_3_conv   (None, 56, 56,         16,640   conv2_block2_2_r…
(Conv2D)              256)

conv2_block2_3_bn     (None, 56, 56,          1,024   conv2_block2_3_c…
(BatchNormalizatio…   256)

conv2_block2_add      (None, 56, 56,              0   conv2_block1_out…
(Add)                 256)                            conv2_block2_3_b…

conv2_block2_out      (None, 56, 56,              0   conv2_block2_add…
(Activation)          256)

conv2_block3_1_conv   (None, 56, 56,         16,448   conv2_block2_out…
(Conv2D)              64)

conv2_block3_1_bn     (None, 56, 56,            256   conv2_block3_1_c…
(BatchNormalizatio…   64)

conv2_block3_1_relu   (None, 56, 56,              0   conv2_block3_1_b…
(Activation)          64)

conv2_block3_2_conv   (None, 56, 56,         36,928   conv2_block3_1_r…
(Conv2D)              64)

conv2_block3_2_bn     (None, 56, 56,            256   conv2_block3_2_c…
(BatchNormalizatio…   64)

conv2_block3_2_relu   (None, 56, 56,              0   conv2_block3_2_b…
(Activation)          64)
```

```
conv2_block3_3_conv   (None, 56, 56,          16,640  conv2_block3_2_r…
(Conv2D)              256)

conv2_block3_3_bn     (None, 56, 56,           1,024  conv2_block3_3_c…
(BatchNormalizatio…   256)

conv2_block3_add      (None, 56, 56,               0  conv2_block2_out…
(Add)                 256)                            conv2_block3_3_b…

conv2_block3_out      (None, 56, 56,               0  conv2_block3_add…
(Activation)          256)

conv3_block1_1_conv   (None, 28, 28,          32,896  conv2_block3_out…
(Conv2D)              128)

conv3_block1_1_bn     (None, 28, 28,             512  conv3_block1_1_c…
(BatchNormalizatio…   128)

conv3_block1_1_relu   (None, 28, 28,               0  conv3_block1_1_b…
(Activation)          128)

conv3_block1_2_conv   (None, 28, 28,         147,584  conv3_block1_1_r…
(Conv2D)              128)

conv3_block1_2_bn     (None, 28, 28,             512  conv3_block1_2_c…
(BatchNormalizatio…   128)

conv3_block1_2_relu   (None, 28, 28,               0  conv3_block1_2_b…
(Activation)          128)

conv3_block1_0_conv   (None, 28, 28,         131,584  conv2_block3_out…
(Conv2D)              512)

conv3_block1_3_conv   (None, 28, 28,          66,048  conv3_block1_2_r…
(Conv2D)              512)

conv3_block1_0_bn     (None, 28, 28,           2,048  conv3_block1_0_c…
(BatchNormalizatio…   512)

conv3_block1_3_bn     (None, 28, 28,           2,048  conv3_block1_3_c…
(BatchNormalizatio…   512)

conv3_block1_add      (None, 28, 28,               0  conv3_block1_0_b…
(Add)                 512)                            conv3_block1_3_b…

conv3_block1_out      (None, 28, 28,               0  conv3_block1_add…
(Activation)          512)
```

```
conv3_block2_1_conv   (None, 28, 28,         65,664   conv3_block1_out…
(Conv2D)              128)

conv3_block2_1_bn     (None, 28, 28,            512   conv3_block2_1_c…
(BatchNormalizatio…   128)

conv3_block2_1_relu   (None, 28, 28,              0   conv3_block2_1_b…
(Activation)          128)

conv3_block2_2_conv   (None, 28, 28,        147,584   conv3_block2_1_r…
(Conv2D)              128)

conv3_block2_2_bn     (None, 28, 28,            512   conv3_block2_2_c…
(BatchNormalizatio…   128)

conv3_block2_2_relu   (None, 28, 28,              0   conv3_block2_2_b…
(Activation)          128)

conv3_block2_3_conv   (None, 28, 28,         66,048   conv3_block2_2_r…
(Conv2D)              512)

conv3_block2_3_bn     (None, 28, 28,          2,048   conv3_block2_3_c…
(BatchNormalizatio…   512)

conv3_block2_add      (None, 28, 28,              0   conv3_block1_out…
(Add)                 512)                            conv3_block2_3_b…

conv3_block2_out      (None, 28, 28,              0   conv3_block2_add…
(Activation)          512)

conv3_block3_1_conv   (None, 28, 28,         65,664   conv3_block2_out…
(Conv2D)              128)

conv3_block3_1_bn     (None, 28, 28,            512   conv3_block3_1_c…
(BatchNormalizatio…   128)

conv3_block3_1_relu   (None, 28, 28,              0   conv3_block3_1_b…
(Activation)          128)

conv3_block3_2_conv   (None, 28, 28,        147,584   conv3_block3_1_r…
(Conv2D)              128)

conv3_block3_2_bn     (None, 28, 28,            512   conv3_block3_2_c…
(BatchNormalizatio…   128)

conv3_block3_2_relu   (None, 28, 28,              0   conv3_block3_2_b…
(Activation)          128)
```

```
conv3_block3_3_conv   (None, 28, 28,          66,048  conv3_block3_2_r…
(Conv2D)              512)

conv3_block3_3_bn     (None, 28, 28,           2,048  conv3_block3_3_c…
(BatchNormalizatio…   512)

conv3_block3_add      (None, 28, 28,               0  conv3_block2_out…
(Add)                 512)                            conv3_block3_3_b…

conv3_block3_out      (None, 28, 28,               0  conv3_block3_add…
(Activation)          512)

conv3_block4_1_conv   (None, 28, 28,          65,664  conv3_block3_out…
(Conv2D)              128)

conv3_block4_1_bn     (None, 28, 28,             512  conv3_block4_1_c…
(BatchNormalizatio…   128)

conv3_block4_1_relu   (None, 28, 28,               0  conv3_block4_1_b…
(Activation)          128)

conv3_block4_2_conv   (None, 28, 28,         147,584  conv3_block4_1_r…
(Conv2D)              128)

conv3_block4_2_bn     (None, 28, 28,             512  conv3_block4_2_c…
(BatchNormalizatio…   128)

conv3_block4_2_relu   (None, 28, 28,               0  conv3_block4_2_b…
(Activation)          128)

conv3_block4_3_conv   (None, 28, 28,          66,048  conv3_block4_2_r…
(Conv2D)              512)

conv3_block4_3_bn     (None, 28, 28,           2,048  conv3_block4_3_c…
(BatchNormalizatio…   512)

conv3_block4_add      (None, 28, 28,               0  conv3_block3_out…
(Add)                 512)                            conv3_block4_3_b…

conv3_block4_out      (None, 28, 28,               0  conv3_block4_add…
(Activation)          512)

conv4_block1_1_conv   (None, 14, 14,         131,328  conv3_block4_out…
(Conv2D)              256)

conv4_block1_1_bn     (None, 14, 14,           1,024  conv4_block1_1_c…
(BatchNormalizatio…   256)
```

```
conv4_block1_1_relu   (None, 14, 14,              0   conv4_block1_1_b…
(Activation)          256)

conv4_block1_2_conv   (None, 14, 14,        590,080   conv4_block1_1_r…
(Conv2D)              256)

conv4_block1_2_bn     (None, 14, 14,          1,024   conv4_block1_2_c…
(BatchNormalizatio…   256)

conv4_block1_2_relu   (None, 14, 14,              0   conv4_block1_2_b…
(Activation)          256)

conv4_block1_0_conv   (None, 14, 14,        525,312   conv3_block4_out…
(Conv2D)              1024)

conv4_block1_3_conv   (None, 14, 14,        263,168   conv4_block1_2_r…
(Conv2D)              1024)

conv4_block1_0_bn     (None, 14, 14,          4,096   conv4_block1_0_c…
(BatchNormalizatio…   1024)

conv4_block1_3_bn     (None, 14, 14,          4,096   conv4_block1_3_c…
(BatchNormalizatio…   1024)

conv4_block1_add      (None, 14, 14,              0   conv4_block1_0_b…
(Add)                 1024)                           conv4_block1_3_b…

conv4_block1_out      (None, 14, 14,              0   conv4_block1_add…
(Activation)          1024)

conv4_block2_1_conv   (None, 14, 14,        262,400   conv4_block1_out…
(Conv2D)              256)

conv4_block2_1_bn     (None, 14, 14,          1,024   conv4_block2_1_c…
(BatchNormalizatio…   256)

conv4_block2_1_relu   (None, 14, 14,              0   conv4_block2_1_b…
(Activation)          256)

conv4_block2_2_conv   (None, 14, 14,        590,080   conv4_block2_1_r…
(Conv2D)              256)

conv4_block2_2_bn     (None, 14, 14,          1,024   conv4_block2_2_c…
(BatchNormalizatio…   256)

conv4_block2_2_relu   (None, 14, 14,              0   conv4_block2_2_b…
(Activation)          256)
```

```
conv4_block2_3_conv   (None, 14, 14,          263,168  conv4_block2_2_r…
(Conv2D)              1024)

conv4_block2_3_bn     (None, 14, 14,            4,096  conv4_block2_3_c…
(BatchNormalizatio…   1024)

conv4_block2_add      (None, 14, 14,                0  conv4_block1_out…
(Add)                 1024)                            conv4_block2_3_b…

conv4_block2_out      (None, 14, 14,                0  conv4_block2_add…
(Activation)          1024)

conv4_block3_1_conv   (None, 14, 14,          262,400  conv4_block2_out…
(Conv2D)              256)

conv4_block3_1_bn     (None, 14, 14,            1,024  conv4_block3_1_c…
(BatchNormalizatio…   256)

conv4_block3_1_relu   (None, 14, 14,                0  conv4_block3_1_b…
(Activation)          256)

conv4_block3_2_conv   (None, 14, 14,          590,080  conv4_block3_1_r…
(Conv2D)              256)

conv4_block3_2_bn     (None, 14, 14,            1,024  conv4_block3_2_c…
(BatchNormalizatio…   256)

conv4_block3_2_relu   (None, 14, 14,                0  conv4_block3_2_b…
(Activation)          256)

conv4_block3_3_conv   (None, 14, 14,          263,168  conv4_block3_2_r…
(Conv2D)              1024)

conv4_block3_3_bn     (None, 14, 14,            4,096  conv4_block3_3_c…
(BatchNormalizatio…   1024)

conv4_block3_add      (None, 14, 14,                0  conv4_block2_out…
(Add)                 1024)                            conv4_block3_3_b…

conv4_block3_out      (None, 14, 14,                0  conv4_block3_add…
(Activation)          1024)

conv4_block4_1_conv   (None, 14, 14,          262,400  conv4_block3_out…
(Conv2D)              256)

conv4_block4_1_bn     (None, 14, 14,            1,024  conv4_block4_1_c…
(BatchNormalizatio…   256)
```

```
conv4_block4_1_relu   (None, 14, 14,              0  conv4_block4_1_b…
(Activation)          256)

conv4_block4_2_conv   (None, 14, 14,        590,080  conv4_block4_1_r…
(Conv2D)              256)

conv4_block4_2_bn     (None, 14, 14,          1,024  conv4_block4_2_c…
(BatchNormalizatio…   256)

conv4_block4_2_relu   (None, 14, 14,              0  conv4_block4_2_b…
(Activation)          256)

conv4_block4_3_conv   (None, 14, 14,        263,168  conv4_block4_2_r…
(Conv2D)              1024)

conv4_block4_3_bn     (None, 14, 14,          4,096  conv4_block4_3_c…
(BatchNormalizatio…   1024)

conv4_block4_add      (None, 14, 14,              0  conv4_block3_out…
(Add)                 1024)                          conv4_block4_3_b…

conv4_block4_out      (None, 14, 14,              0  conv4_block4_add…
(Activation)          1024)

conv4_block5_1_conv   (None, 14, 14,        262,400  conv4_block4_out…
(Conv2D)              256)

conv4_block5_1_bn     (None, 14, 14,          1,024  conv4_block5_1_c…
(BatchNormalizatio…   256)

conv4_block5_1_relu   (None, 14, 14,              0  conv4_block5_1_b…
(Activation)          256)

conv4_block5_2_conv   (None, 14, 14,        590,080  conv4_block5_1_r…
(Conv2D)              256)

conv4_block5_2_bn     (None, 14, 14,          1,024  conv4_block5_2_c…
(BatchNormalizatio…   256)

conv4_block5_2_relu   (None, 14, 14,              0  conv4_block5_2_b…
(Activation)          256)

conv4_block5_3_conv   (None, 14, 14,        263,168  conv4_block5_2_r…
(Conv2D)              1024)

conv4_block5_3_bn     (None, 14, 14,          4,096  conv4_block5_3_c…
(BatchNormalizatio…   1024)
```

```
conv4_block5_add      (None, 14, 14,               0  conv4_block4_out…
(Add)                 1024)                           conv4_block5_3_b…

conv4_block5_out      (None, 14, 14,               0  conv4_block5_add…
(Activation)          1024)

conv4_block6_1_conv   (None, 14, 14,         262,400  conv4_block5_out…
(Conv2D)              256)

conv4_block6_1_bn     (None, 14, 14,           1,024  conv4_block6_1_c…
(BatchNormalizatio…   256)

conv4_block6_1_relu   (None, 14, 14,               0  conv4_block6_1_b…
(Activation)          256)

conv4_block6_2_conv   (None, 14, 14,         590,080  conv4_block6_1_r…
(Conv2D)              256)

conv4_block6_2_bn     (None, 14, 14,           1,024  conv4_block6_2_c…
(BatchNormalizatio…   256)

conv4_block6_2_relu   (None, 14, 14,               0  conv4_block6_2_b…
(Activation)          256)

conv4_block6_3_conv   (None, 14, 14,         263,168  conv4_block6_2_r…
(Conv2D)              1024)

conv4_block6_3_bn     (None, 14, 14,           4,096  conv4_block6_3_c…
(BatchNormalizatio…   1024)

conv4_block6_add      (None, 14, 14,               0  conv4_block5_out…
(Add)                 1024)                           conv4_block6_3_b…

conv4_block6_out      (None, 14, 14,               0  conv4_block6_add…
(Activation)          1024)

conv5_block1_1_conv   (None, 7, 7, 512)      524,800  conv4_block6_out…
(Conv2D)

conv5_block1_1_bn     (None, 7, 7, 512)        2,048  conv5_block1_1_c…
(BatchNormalizatio…

conv5_block1_1_relu   (None, 7, 7, 512)            0  conv5_block1_1_b…
(Activation)

conv5_block1_2_conv   (None, 7, 7, 512)    2,359,808  conv5_block1_1_r…
(Conv2D)
```

```
conv5_block1_2_bn     (None, 7, 7, 512)       2,048   conv5_block1_2_c…
(BatchNormalizatio…

conv5_block1_2_relu   (None, 7, 7, 512)           0   conv5_block1_2_b…
(Activation)

conv5_block1_0_conv   (None, 7, 7,        2,099,200   conv4_block6_out…
(Conv2D)              2048)

conv5_block1_3_conv   (None, 7, 7,        1,050,624   conv5_block1_2_r…
(Conv2D)              2048)

conv5_block1_0_bn     (None, 7, 7,            8,192   conv5_block1_0_c…
(BatchNormalizatio…   2048)

conv5_block1_3_bn     (None, 7, 7,            8,192   conv5_block1_3_c…
(BatchNormalizatio…   2048)

conv5_block1_add      (None, 7, 7,                0   conv5_block1_0_b…
(Add)                 2048)                           conv5_block1_3_b…

conv5_block1_out      (None, 7, 7,                0   conv5_block1_add…
(Activation)          2048)

conv5_block2_1_conv   (None, 7, 7, 512)   1,049,088   conv5_block1_out…
(Conv2D)

conv5_block2_1_bn     (None, 7, 7, 512)       2,048   conv5_block2_1_c…
(BatchNormalizatio…

conv5_block2_1_relu   (None, 7, 7, 512)           0   conv5_block2_1_b…
(Activation)

conv5_block2_2_conv   (None, 7, 7, 512)   2,359,808   conv5_block2_1_r…
(Conv2D)

conv5_block2_2_bn     (None, 7, 7, 512)       2,048   conv5_block2_2_c…
(BatchNormalizatio…

conv5_block2_2_relu   (None, 7, 7, 512)           0   conv5_block2_2_b…
(Activation)

conv5_block2_3_conv   (None, 7, 7,        1,050,624   conv5_block2_2_r…
(Conv2D)              2048)

conv5_block2_3_bn     (None, 7, 7,            8,192   conv5_block2_3_c…
(BatchNormalizatio…   2048)
```

```
conv5_block2_add      (None, 7, 7,                    0   conv5_block1_out…
(Add)                 2048)                               conv5_block2_3_b…

conv5_block2_out      (None, 7, 7,                    0   conv5_block2_add…
(Activation)          2048)

conv5_block3_1_conv   (None, 7, 7, 512)   1,049,088   conv5_block2_out…
(Conv2D)

conv5_block3_1_bn     (None, 7, 7, 512)       2,048   conv5_block3_1_c…
(BatchNormalizatio…

conv5_block3_1_relu   (None, 7, 7, 512)           0   conv5_block3_1_b…
(Activation)

conv5_block3_2_conv   (None, 7, 7, 512)   2,359,808   conv5_block3_1_r…
(Conv2D)

conv5_block3_2_bn     (None, 7, 7, 512)       2,048   conv5_block3_2_c…
(BatchNormalizatio…

conv5_block3_2_relu   (None, 7, 7, 512)           0   conv5_block3_2_b…
(Activation)

conv5_block3_3_conv   (None, 7, 7,        1,050,624   conv5_block3_2_r…
(Conv2D)              2048)

conv5_block3_3_bn     (None, 7, 7,            8,192   conv5_block3_3_c…
(BatchNormalizatio…   2048)

conv5_block3_add      (None, 7, 7,                0   conv5_block2_out…
(Add)                 2048)                           conv5_block3_3_b…

conv5_block3_out      (None, 7, 7,                0   conv5_block3_add…
(Activation)          2048)

flatten_14            (None, 100352)              0   conv5_block3_out…
(Flatten)

dense_38 (Dense)      (None, 256)        25,690,368   flatten_14[0][0]

dense_39 (Dense)      (None, 128)            32,896   dense_38[0][0]

dense_40 (Dense)      (None, 4)                 516   dense_39[0][0]
```

```
Total params: 109,690,382 (418.44 MB)

Trainable params: 0 (0.00 B)

Non-trainable params: 49,311,492 (188.11 MB)

Optimizer params: 60,378,890 (230.33 MB)
```

[61]:

```
test_model=load_model(r"C:\Users\harsa\Downloads\alzheimer_res3.h5")
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be
built. `model.compile_metrics` will be empty until you train or evaluate the
model.
```

[62]:

```
img=image.load_img(r"D:\Winter Semester␣
 ↪3\AIML\Project\Alzheimer_image\Alzheimer_s Dataset\test\NonDemented\26 (62).
 ↪jpg")
img=img.resize((180,180))
```

[63]:

```
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
img_data=preprocess_input(x)
```

[64]:

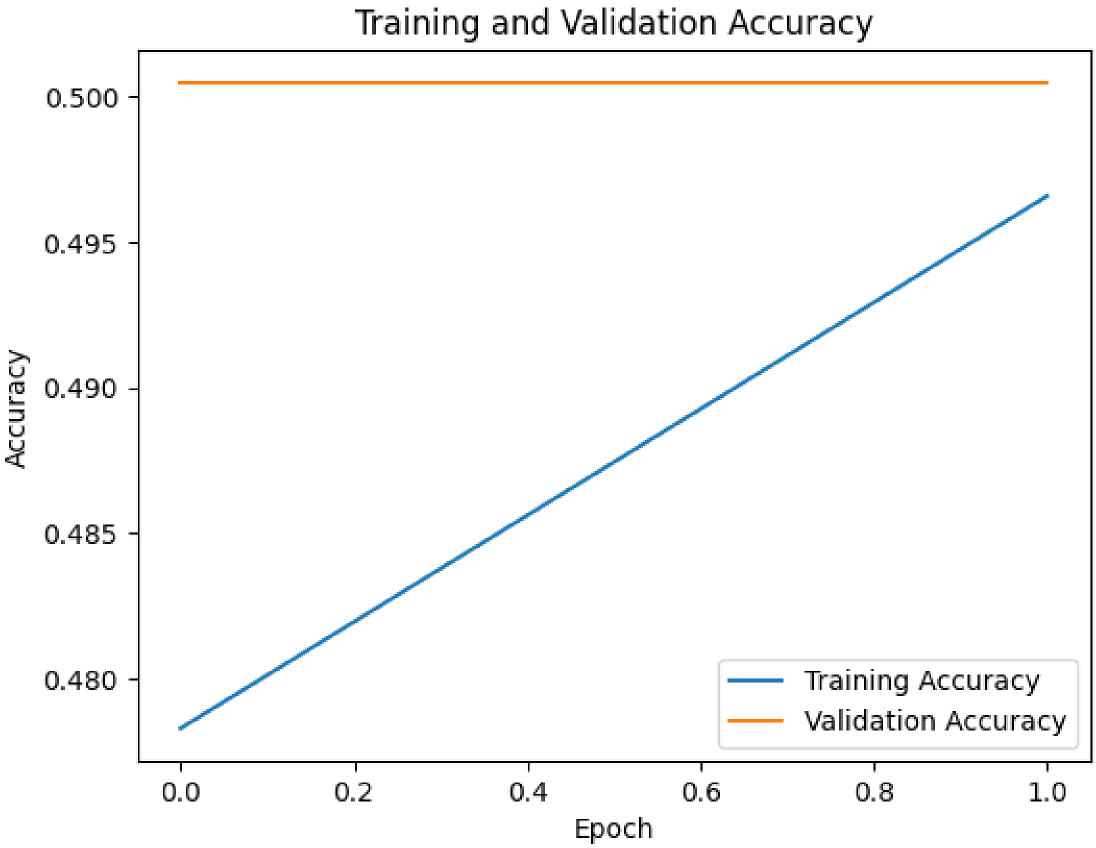
```
from tensorflow.image import resize

# Resize input data to match the model's input shape (224x224)
x_resized = resize(x, (224, 224))

# Preprocess input data if needed (e.g., normalize, scale)
x_preprocessed = preprocess_input(x_resized)

# Make predictions using the preprocessed input data
predictions = test_model.predict(x_preprocessed)
print(np.argmax(predictions, axis=1)[0])  # Example: Get the predicted class
```

```
1/1               2s 2s/step
2
```

[67]:

```
import numpy as np
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
test_generator = train_datagen.flow_from_directory(
    "D:/Winter Semester 3/AIML/Project/Alzheimer_image/Alzheimer_s Dataset/
 ↪test",
    target_size=(224, 224),
    batch_size=32,
    class_mode='binary',  # binary classification
    subset='training'
)
validation_generator = train_datagen.flow_from_directory(
    "D:/Winter Semester 3/AIML/Project/Alzheiemer_image/Alzheimer_s Dataset/
 ↪test",
    target_size=(224, 224),
    batch_size=32,
    class_mode='binary',  # binary classification
    subset='validation'
)
# Train the model
#history = model.fit(train_generator, epochs=2,␣
 ↪validation_data=validation_generator)

# Plot training and validation accuracy
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.title('Training and Validation Accuracy')
plt.show()

# Evaluate the model on test data
loss, accuracy = model.evaluate(test_generator)
print(f"Test Loss: {loss}")
print(f"Test Accuracy: {accuracy}")

# Generate predictions for test data
predictions = model.predict(test_generator)
predicted_classes = np.argmax(predictions, axis=1)

# Get true classes from the generator
true_classes = test_generator.classes
class_labels = list(test_generator.class_indices.keys())

# Generate classification report
report = classification_report(true_classes, predicted_classes,␣
 ↪target_names=class_labels)
print("Classification Report:\n", report)
```

```
Found 1025 images belonging to 4 classes.
Found 254 images belonging to 4 classes.
```



```
c:\Users\harsa\AppData\Local\Programs\Python\Python310\lib\site-
packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:122:
UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in
its constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will be
ignored.
  self._warn_if_super_not_called()

33/33                 47s 1s/step -
accuracy: 0.4869 - loss: 1.1885
Test Loss: 1.1624151468276978
Test Accuracy: 0.4995121955871582
33/33                 51s 1s/step
Classification Report:
                   precision    recall  f1-score   support

    MildDemented       0.00      0.00      0.00       144
ModerateDemented       0.00      0.00      0.00        10
     NonDemented       0.50      1.00      0.67       512
VeryMildDemented       0.00      0.00      0.00       359
```

```
    accuracy                           0.50      1025
   macro avg       0.12      0.25      0.17      1025
weighted avg       0.25      0.50      0.33      1025
```

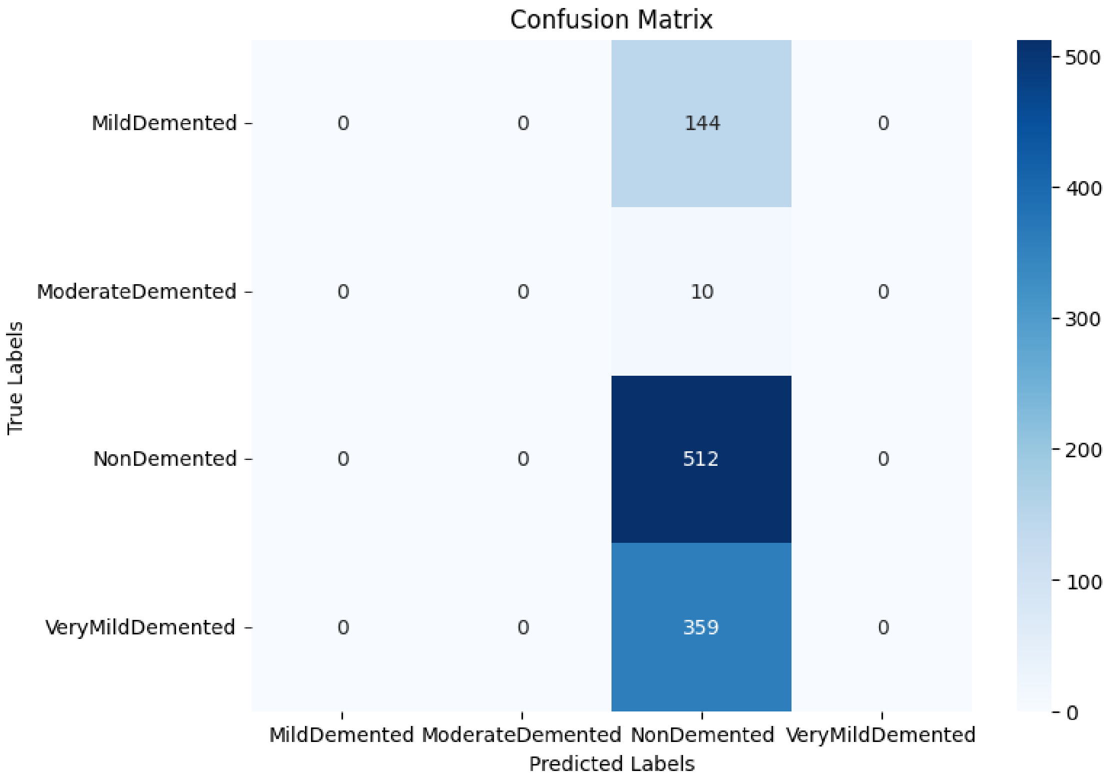
```
c:\Users\harsa\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\metrics\_classification.py:1471: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\Users\harsa\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\metrics\_classification.py:1471: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\Users\harsa\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\metrics\_classification.py:1471: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

[68]:

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

[69]:

```
# Generate confusion matrix
conf_matrix = confusion_matrix(true_classes, predicted_classes)

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',␣
  ↪xticklabels=class_labels, yticklabels=class_labels)
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('Confusion Matrix')
plt.show()
```



[70]:

```
# Load test data
test_datagen = ImageDataGenerator(rescale=1./255)
test_generator = test_datagen.flow_from_directory(
    "D:/Winter Semester 3/AIML/Project/Alzheiemer_image/Alzheimer_s Dataset/
  ↪test",
    target_size=(224, 224),
    batch_size=32,
    class_mode='binary'  # binary classification
)

# Evaluate the model on test data
loss, accuracy = model.evaluate(test_generator)

print(f"Test Loss: {loss}")
print(f"Test Accuracy: {accuracy}")
```

```
Found 1279 images belonging to 4 classes.
```

```
c:\Users\harsa\AppData\Local\Programs\Python\Python310\lib\site-
packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:122:
UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in
its constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will be
```

```
ignored.
  self._warn_if_super_not_called()
```

```
40/40                 54s 1s/step -
accuracy: 0.5035 - loss: 1.0979
Test Loss: 1.106929063796997
Test Accuracy: 0.5003909468650818
```

[71]:

```python
# Generate predictions for test data
predictions = model.predict(test_generator)
predicted_classes = np.argmax(predictions, axis=1)

# Get true classes from the generator
true_classes = test_generator.classes
class_labels = list(test_generator.class_indices.keys())

# Generate classification report
report = classification_report(true_classes, predicted_classes,␣
 ↪target_names=class_labels)
print("Classification Report:\n", report)
```

```
40/40                 55s 1s/step
Classification Report:
                   precision    recall  f1-score   support

    MildDemented       0.00      0.00      0.00       179
ModerateDemented       0.00      0.00      0.00        12
     NonDemented       0.50      1.00      0.67       640
VeryMildDemented       0.00      0.00      0.00       448

        accuracy                           0.50      1279
       macro avg       0.13      0.25      0.17      1279
    weighted avg       0.25      0.50      0.33      1279
```

```
c:\Users\harsa\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\metrics\_classification.py:1471: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\Users\harsa\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\metrics\_classification.py:1471: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\Users\harsa\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\metrics\_classification.py:1471: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```