

```
In [ ]: import os
import numpy
import pandas
import random
import zipfile
import datetime
import tensorflow
import matplotlib.pyplot

from PIL import Image

import tensorflow as tf
from keras.layers import Dense, Conv2D, Flatten, MaxPool2D, Dropout
from keras.models import Sequential
from tensorflow.keras.preprocessing import image
from keras.callbacks import ModelCheckpoint
from keras.models import load_model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import (
    Input,
    Conv2D,
    MaxPooling2D,
    BatchNormalization,
    Flatten,
    Dense,
    Dropout,
)
```

Since the contents of `testing_data` and `validation_data` does not contain any structured labels, we have decided not to use them.

Meanwhile three folds inside the `training_data` will be used in the following way :

- `fold_0` : Training Data
- `fold_1` : Testing Data
- `fold_2` : Validation Data

```
In [ ]: # Respective directories instantiated as string variables for future use
training_data_dir='./C-NMC_Leukemia/training_data/fold_0'
testing_data_dir='./C-NMC_Leukemia/training_data/fold_1'
validation_data_dir='./C-NMC_Leukemia/training_data/fold_2'
```

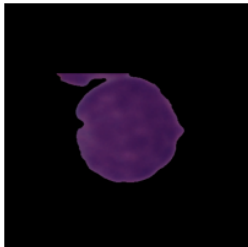
```
In [ ]: # available labels
labels=['all','hem']
```

```
In [ ]: # Visualize images in the dataset
def random_img_finder(url,typ):
    category=random.choice(typ)
    return category,random.choice(list(os.walk(f'{url}/{category}'))[0][2])

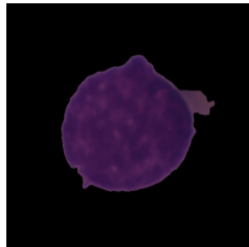
def img_plotter(url,nrows,ncols):
    fig,ax=matplotlib.pyplot.subplots(nrows=nrows,ncols=ncols,figsize=(20,12))
    for i in range(ncols):
        for j in range(nrows):
            cat,img=random_img_finder(url,labels)
            ax[i][j].imshow(numpy.asarray(Image.open(f'{url}/{cat}/{img}')))
            ax[i][j].set_title(f'Category : {cat}')
            ax[i][j].axis('off')
```

```
In [ ]: img_plotter(training_data_dir,4,4)
```

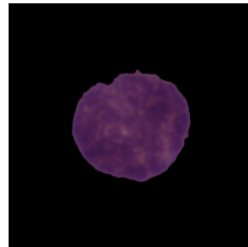
Category : all



Category : all



Category : hem



Category : hem



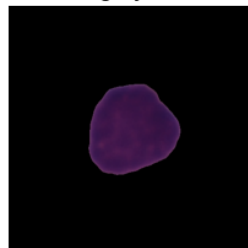
Category : hem



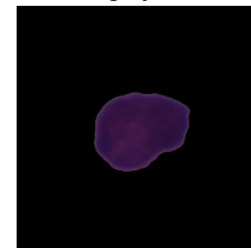
Category : all



Category : hem



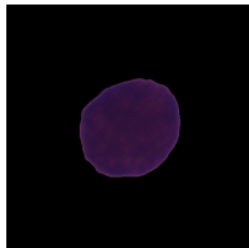
Category : hem



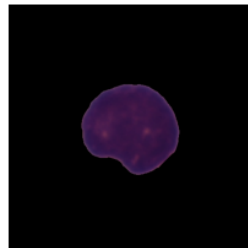
Category : hem



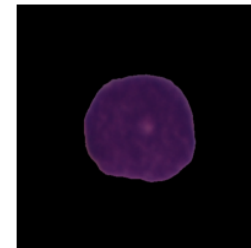
Category : all



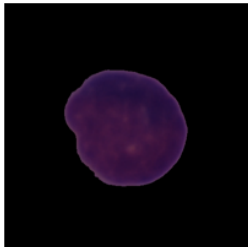
Category : hem



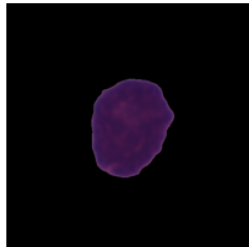
Category : all



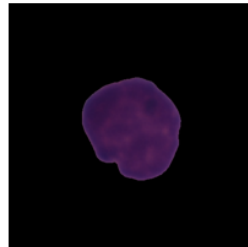
Category : all



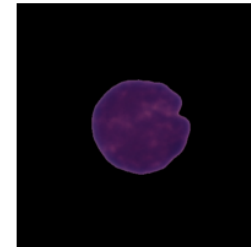
Category : hem



Category : hem



Category : hem



```

In [ ]: # Describe the quantitative aspect of the dataset
dataset_contents={
    # Train Data all
    'training_data_all':[i for i in os.walk('./C-NMC_Leukemia/training_data/fold_0/all')][0][2],
    # Train Data hem
    'training_data_hem':[i for i in os.walk('./C-NMC_Leukemia/training_data/fold_0/hem')][0][2],
    # Testing Data all
    'testing_data_all':[i for i in os.walk('./C-NMC_Leukemia/training_data/fold_1/all')][0][2],
    # Testing Data hem
    'testing_data_hem':[i for i in os.walk('./C-NMC_Leukemia/training_data/fold_1/hem')][0][2],
    # Validation Data all
    'validation_data_all':[i for i in os.walk('./C-NMC_Leukemia/training_data/fold_2/all')][0][2],
    # Validation Data hem
    'validation_data_hem':[i for i in os.walk('./C-NMC_Leukemia/training_data/fold_2/hem')][0][2],
}
## Major Elements of the database
print('Images Available in Training Data : ',len(dataset_contents['training_data_all'])+len(dataset_contents['training_data_hem']))
print('Division of Training set into all and hem would be like : ',len(dataset_contents['training_data_all']),len(dataset_contents['training_data_hem']))
print()
print()
print('Images Available in Testing Data : ',len(dataset_contents['testing_data_all'])+len(dataset_contents['testing_data_hem']))
print('Division of Testing set into all and hem would be like : ',len(dataset_contents['testing_data_all']),len(dataset_contents['testing_data_hem']))
print()
print()
print('Images Available in Validation Data : ',len(dataset_contents['validation_data_all'])+len(dataset_contents['validation_data_hem']))
print('Division of Validation set into all and hem would be like : ',len(dataset_contents['validation_data_all']),len(dataset_contents['validation_data_hem']))

```

Images Available in Training Data : 3527

Division of Training set into all and hem would be like : 2397 and 1130

Images Available in Testing Data : 3581

Division of Testing set into all and hem would be like : 2418 and 1163

Images Available in Validation Data : 3553

Division of Validation set into all and hem would be like : 2457 and 1096

```
In [ ]: # parameters
batch_size=64
img_height=180
img_width=180
train_ds=tensorflow.keras.utils.image_dataset_from_directory(
    training_data_dir,
    label_mode='binary',
    image_size=(img_height, img_width),
    batch_size=batch_size)
```

Found 3527 files belonging to 2 classes.

```
In [ ]:
```

```
In [ ]: test_ds=tensorflow.keras.utils.image_dataset_from_directory(
    testing_data_dir,
    label_mode='binary',
    image_size=(img_height, img_width),
    batch_size=batch_size)
```

Found 3581 files belonging to 2 classes.

```
In [ ]: validation_ds=tensorflow.keras.utils.image_dataset_from_directory(
    validation_data_dir,
    image_size=(img_height, img_width),
    batch_size=batch_size)
```

Found 3553 files belonging to 2 classes.

```
In [ ]: # Classes available
train_ds.class_names
```

```
Out[ ]: ['all', 'hem']
```

```
In [ ]: for image_batch, labels_batch in train_ds:
    print(image_batch.shape)
    print(labels_batch.shape)
    break
```

```
(64, 180, 180, 3)
(64, 1)
```

```
In [ ]: #Callback
from tensorflow.keras.callbacks import EarlyStopping
es = EarlyStopping(monitor='val_accuracy', patience=10, mode='max', verbose=1)
```

Standardize the data by normalizing it to be in the range [0,1]

```
In [ ]: AUTOTUNE = tensorflow.data.AUTOTUNE

train_ds = train_ds.cache().prefetch(buffer_size=AUTOTUNE)
test_ds = test_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

CNN

```
In [ ]: from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l2

# Define the input shape based on your 180x180 images
input_shape = (180, 180, 3)

# Define the model
model = Sequential([
    # Data normalization layer
    tf.keras.layers.experimental.preprocessing.Rescaling(1./255, input_shape=input_shape),

    # Convolutional layers with batch normalization
    Conv2D(64, (3, 3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 2)),

    Conv2D(128, (3, 3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 2)),

    Conv2D(256, (3, 3), activation='relu', padding='same'),
    BatchNormalization(),
```

```

MaxPooling2D(pool_size=(2, 2)),

Conv2D(512, (3, 3), activation='relu', padding='same'),
BatchNormalization(),
MaxPooling2D(pool_size=(2, 2)),

# Flatten the output and add dense layers
Flatten(),
Dense(512, activation='relu', kernel_regularizer=l2(0.01)),
Dropout(0.5), # Dropout for regularization
Dense(128, activation='relu', kernel_regularizer=l2(0.01)),
Dropout(0.5),

# Output layer for binary classification
Dense(1, activation='sigmoid')
])

# Compile the model
model.compile(
    optimizer=Adam(learning_rate=0.0001), # Adjust learning rate as needed
    loss='binary_crossentropy',
    metrics=['accuracy']
)

# Print model summary
model.summary()

```

WARNING:absl:At this time, the v2.11+ optimizer `tf.keras.optimizers.Adam` runs slowly on M1/M2 Macs, please use the legacy Keras optimizer instead, located at `tf.keras.optimizers.legacy.Adam`.

WARNING:absl:There is a known slowdown when using v2.11+ Keras optimizers on M1/M2 Macs. Falling back to the legacy Keras optimizer, i.e., `tf.keras.optimizers.legacy.Adam`.

Model: "sequential_1"

| Layer (type) | Output Shape | Param # |
|-------------------------|---------------------|---------|
| ===== | | |
| rescaling_3 (Rescaling) | (None, 180, 180, 3) | 0 |
| ===== | | |
| Layer (type) | Output Shape | Param # |
| ===== | | |
| rescaling_3 (Rescaling) | (None, 180, 180, 3) | 0 |

| | | |
|--|----------------------|----------|
| conv2d_100 (Conv2D) | (None, 180, 180, 64) | 1792 |
| batch_normalization_98 (BatchNormalization) | (None, 180, 180, 64) | 256 |
| max_pooling2d_9 (MaxPooling2D) | (None, 90, 90, 64) | 0 |
| conv2d_101 (Conv2D) | (None, 90, 90, 128) | 73856 |
| batch_normalization_99 (BatchNormalization) | (None, 90, 90, 128) | 512 |
| max_pooling2d_10 (MaxPooling2D) | (None, 45, 45, 128) | 0 |
| conv2d_102 (Conv2D) | (None, 45, 45, 256) | 295168 |
| batch_normalization_100 (BatchNormalization) | (None, 45, 45, 256) | 1024 |
| max_pooling2d_11 (MaxPooling2D) | (None, 22, 22, 256) | 0 |
| conv2d_103 (Conv2D) | (None, 22, 22, 512) | 1180160 |
| batch_normalization_101 (BatchNormalization) | (None, 22, 22, 512) | 2048 |
| max_pooling2d_12 (MaxPooling2D) | (None, 11, 11, 512) | 0 |
| flatten_1 (Flatten) | (None, 61952) | 0 |
| dense_5 (Dense) | (None, 512) | 31719936 |
| dropout_4 (Dropout) | (None, 512) | 0 |
| dense_6 (Dense) | (None, 128) | 65664 |

| | | |
|---------------------|-------------|-----|
| dropout_5 (Dropout) | (None, 128) | 0 |
| dense_7 (Dense) | (None, 1) | 129 |

```
=====
Total params: 33340545 (127.18 MB)
Trainable params: 33338625 (127.18 MB)
Non-trainable params: 1920 (7.50 KB)
=====
```

```
In [ ]: mc = ModelCheckpoint(filepath="cnn.hdf5", monitor= 'val_accuracy', verbose= 1, save_best_only= True, mode
call_back = [ mc,es ];
```

```
In [ ]: history=model.fit(
    train_ds,
    validation_data=test_ds,
    epochs=100,
    callbacks=call_back
)
```

Epoch 1/100

```
2023-10-30 16:13:00.429190: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:11
4] Plugin optimizer for device_type GPU is enabled.
```

```
56/56 [=====] - ETA: 0s - loss: 24.4869 - accuracy: 0.7287
```

```
2023-10-30 16:13:33.843854: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:11
4] Plugin optimizer for device_type GPU is enabled.
```

```
Epoch 1: val_accuracy improved from -inf to 0.32477, saving model to cnn.hdf5
```

```
/Users/rajitkuthiala/miniforge3/envs/mlp/lib/python3.8/site-packages/keras/src/engine/training.py:3000:
UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considere
d legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')`.
    saving_api.save_model(
```

```
56/56 [=====] - 46s 686ms/step - loss: 24.4869 - accuracy: 0.7287 - val_loss: 1
1.1411 - val_accuracy: 0.3248
```

Epoch 2/100

Epoch 2/100

```
56/56 [=====] - ETA: 0s - loss: 18.9917 - accuracy: 0.7383
```

```
Epoch 2: val_accuracy did not improve from 0.32477
```

```
56/56 [=====] - 30s 527ms/step - loss: 18.9917 - accuracy: 0.7383 - val_loss: 1
```

```
0.8736 - val_accuracy: 0.3248
Epoch 3/100
56/56 [=====] - ETA: 0s - loss: 13.8320 - accuracy: 0.7686
Epoch 3: val_accuracy did not improve from 0.32477
56/56 [=====] - 29s 518ms/step - loss: 13.8320 - accuracy: 0.7686 - val_loss:
8.9182 - val_accuracy: 0.3248
Epoch 4/100
56/56 [=====] - ETA: 0s - loss: 11.2904 - accuracy: 0.7715
Epoch 4: val_accuracy improved from 0.32477 to 0.67523, saving model to cnn.hdf5
56/56 [=====] - 29s 517ms/step - loss: 11.2904 - accuracy: 0.7715 - val_loss:
8.6950 - val_accuracy: 0.6752
Epoch 5/100
56/56 [=====] - ETA: 0s - loss: 9.4502 - accuracy: 0.7842
Epoch 5: val_accuracy did not improve from 0.67523
56/56 [=====] - 29s 517ms/step - loss: 9.4502 - accuracy: 0.7842 - val_loss: 6.
4487 - val_accuracy: 0.6604
Epoch 6/100
56/56 [=====] - ETA: 0s - loss: 8.9598 - accuracy: 0.8064
Epoch 6: val_accuracy did not improve from 0.67523
56/56 [=====] - 29s 511ms/step - loss: 8.9598 - accuracy: 0.8064 - val_loss: 1
1.0873 - val_accuracy: 0.3248
Epoch 7/100
56/56 [=====] - ETA: 0s - loss: 7.1940 - accuracy: 0.8477
Epoch 7: val_accuracy did not improve from 0.67523
56/56 [=====] - 28s 511ms/step - loss: 7.1940 - accuracy: 0.8477 - val_loss: 1
0.3647 - val_accuracy: 0.3248
Epoch 8/100
56/56 [=====] - ETA: 0s - loss: 6.1102 - accuracy: 0.8738
Epoch 8: val_accuracy did not improve from 0.67523
56/56 [=====] - 29s 516ms/step - loss: 6.1102 - accuracy: 0.8738 - val_loss: 6.
7128 - val_accuracy: 0.3401
Epoch 9/100
56/56 [=====] - ETA: 0s - loss: 5.7061 - accuracy: 0.8863
Epoch 9: val_accuracy did not improve from 0.67523
56/56 [=====] - 29s 524ms/step - loss: 5.7061 - accuracy: 0.8863 - val_loss: 6.
9720 - val_accuracy: 0.4250
Epoch 10/100
56/56 [=====] - ETA: 0s - loss: 5.6365 - accuracy: 0.8784
Epoch 10: val_accuracy did not improve from 0.67523
56/56 [=====] - 29s 514ms/step - loss: 5.6365 - accuracy: 0.8784 - val_loss: 8.
```

5342 - val_accuracy: 0.3318
Epoch 11/100
56/56 [=====] - ETA: 0s - loss: 5.2914 - accuracy: 0.8886
Epoch 11: val_accuracy improved from 0.67523 to 0.71405, saving model to cnn.hdf5
56/56 [=====] - 30s 538ms/step - loss: 5.2914 - accuracy: 0.8886 - val_loss: 5.
9130 - val_accuracy: 0.7140
Epoch 12/100
56/56 [=====] - ETA: 0s - loss: 4.7072 - accuracy: 0.9200
Epoch 12: val_accuracy improved from 0.71405 to 0.80648, saving model to cnn.hdf5
56/56 [=====] - 29s 523ms/step - loss: 4.7072 - accuracy: 0.9200 - val_loss: 5.
3216 - val_accuracy: 0.8065
Epoch 13/100
56/56 [=====] - ETA: 0s - loss: 4.1866 - accuracy: 0.9439
Epoch 13: val_accuracy did not improve from 0.80648
56/56 [=====] - 29s 525ms/step - loss: 4.1866 - accuracy: 0.9439 - val_loss: 8.
9691 - val_accuracy: 0.4474
Epoch 14/100
56/56 [=====] - ETA: 0s - loss: 3.9907 - accuracy: 0.9464
Epoch 14: val_accuracy did not improve from 0.80648
56/56 [=====] - 30s 530ms/step - loss: 3.9907 - accuracy: 0.9464 - val_loss: 7.
0342 - val_accuracy: 0.5588
Epoch 15/100
56/56 [=====] - ETA: 0s - loss: 3.9393 - accuracy: 0.9390
Epoch 15: val_accuracy improved from 0.80648 to 0.83692, saving model to cnn.hdf5
56/56 [=====] - 29s 527ms/step - loss: 3.9393 - accuracy: 0.9390 - val_loss: 5.
6722 - val_accuracy: 0.8369
Epoch 16/100
56/56 [=====] - ETA: 0s - loss: 3.9065 - accuracy: 0.9390
Epoch 16: val_accuracy improved from 0.83692 to 0.84194, saving model to cnn.hdf5
56/56 [=====] - 29s 516ms/step - loss: 3.9065 - accuracy: 0.9390 - val_loss: 5.
9622 - val_accuracy: 0.8419
Epoch 17/100
56/56 [=====] - ETA: 0s - loss: 3.6299 - accuracy: 0.9501
Epoch 17: val_accuracy improved from 0.84194 to 0.87043, saving model to cnn.hdf5
56/56 [=====] - 29s 525ms/step - loss: 3.6299 - accuracy: 0.9501 - val_loss: 5.
2961 - val_accuracy: 0.8704
Epoch 18/100
56/56 [=====] - ETA: 0s - loss: 3.7580 - accuracy: 0.9456
Epoch 18: val_accuracy did not improve from 0.87043
56/56 [=====] - 29s 513ms/step - loss: 3.7580 - accuracy: 0.9456 - val_loss: 9.

3813 - val_accuracy: 0.6976
Epoch 19/100
56/56 [=====] - ETA: 0s - loss: 3.5810 - accuracy: 0.9473
Epoch 19: val_accuracy did not improve from 0.87043
56/56 [=====] - 29s 520ms/step - loss: 3.5810 - accuracy: 0.9473 - val_loss: 5.
1063 - val_accuracy: 0.8308
Epoch 20/100
56/56 [=====] - ETA: 0s - loss: 3.4212 - accuracy: 0.9586
Epoch 20: val_accuracy did not improve from 0.87043
56/56 [=====] - 29s 518ms/step - loss: 3.4212 - accuracy: 0.9586 - val_loss: 4.
8550 - val_accuracy: 0.8679
Epoch 21/100
56/56 [=====] - ETA: 0s - loss: 3.2332 - accuracy: 0.9716
Epoch 21: val_accuracy did not improve from 0.87043
56/56 [=====] - 29s 519ms/step - loss: 3.2332 - accuracy: 0.9716 - val_loss: 5.
3521 - val_accuracy: 0.8688
Epoch 22/100
56/56 [=====] - ETA: 0s - loss: 3.0405 - accuracy: 0.9824
Epoch 22: val_accuracy did not improve from 0.87043
56/56 [=====] - 30s 534ms/step - loss: 3.0405 - accuracy: 0.9824 - val_loss: 5.
3339 - val_accuracy: 0.8685
Epoch 23/100
56/56 [=====] - ETA: 0s - loss: 2.9151 - accuracy: 0.9870
Epoch 23: val_accuracy improved from 0.87043 to 0.88160, saving model to cnn.hdf5
56/56 [=====] - 29s 518ms/step - loss: 2.9151 - accuracy: 0.9870 - val_loss: 4.
7784 - val_accuracy: 0.8816
Epoch 24/100
56/56 [=====] - ETA: 0s - loss: 2.8836 - accuracy: 0.9836
Epoch 24: val_accuracy did not improve from 0.88160
56/56 [=====] - 28s 509ms/step - loss: 2.8836 - accuracy: 0.9836 - val_loss: 7.
6122 - val_accuracy: 0.8015
Epoch 25/100
56/56 [=====] - ETA: 0s - loss: 2.7683 - accuracy: 0.9912
Epoch 25: val_accuracy did not improve from 0.88160
56/56 [=====] - 29s 513ms/step - loss: 2.7683 - accuracy: 0.9912 - val_loss: 5.
5227 - val_accuracy: 0.8718
Epoch 26/100
56/56 [=====] - ETA: 0s - loss: 2.6667 - accuracy: 0.9940
Epoch 26: val_accuracy did not improve from 0.88160
56/56 [=====] - 29s 517ms/step - loss: 2.6667 - accuracy: 0.9940 - val_loss: 6.

1623 - val_accuracy: 0.8553
Epoch 27/100
56/56 [=====] - ETA: 0s - loss: 2.5987 - accuracy: 0.9952
Epoch 27: val_accuracy improved from 0.88160 to 0.88188, saving model to cnn.hdf5
56/56 [=====] - 30s 536ms/step - loss: 2.5987 - accuracy: 0.9952 - val_loss: 4.
5282 - val_accuracy: 0.8819
Epoch 28/100
56/56 [=====] - ETA: 0s - loss: 2.5279 - accuracy: 0.9957
Epoch 28: val_accuracy did not improve from 0.88188
56/56 [=====] - 29s 513ms/step - loss: 2.5279 - accuracy: 0.9957 - val_loss: 4.
6421 - val_accuracy: 0.8794
Epoch 29/100
56/56 [=====] - ETA: 0s - loss: 2.4640 - accuracy: 0.9977
Epoch 29: val_accuracy improved from 0.88188 to 0.88607, saving model to cnn.hdf5
56/56 [=====] - 29s 514ms/step - loss: 2.4640 - accuracy: 0.9977 - val_loss: 4.
5218 - val_accuracy: 0.8861
Epoch 30/100
56/56 [=====] - ETA: 0s - loss: 2.4006 - accuracy: 0.9989
Epoch 30: val_accuracy did not improve from 0.88607
56/56 [=====] - 28s 506ms/step - loss: 2.4006 - accuracy: 0.9989 - val_loss: 4.
7137 - val_accuracy: 0.8732
Epoch 31/100
56/56 [=====] - ETA: 0s - loss: 2.3901 - accuracy: 0.9926
Epoch 31: val_accuracy did not improve from 0.88607
56/56 [=====] - 29s 511ms/step - loss: 2.3901 - accuracy: 0.9926 - val_loss: 4.
6987 - val_accuracy: 0.7894
Epoch 32/100
56/56 [=====] - ETA: 0s - loss: 2.3771 - accuracy: 0.9892
Epoch 32: val_accuracy did not improve from 0.88607
56/56 [=====] - 28s 509ms/step - loss: 2.3771 - accuracy: 0.9892 - val_loss: 4.
6488 - val_accuracy: 0.8503
Epoch 33/100
56/56 [=====] - ETA: 0s - loss: 2.3256 - accuracy: 0.9918
Epoch 33: val_accuracy did not improve from 0.88607
56/56 [=====] - 29s 515ms/step - loss: 2.3256 - accuracy: 0.9918 - val_loss: 4.
4126 - val_accuracy: 0.8542
Epoch 34/100
56/56 [=====] - ETA: 0s - loss: 2.2725 - accuracy: 0.9943
Epoch 34: val_accuracy did not improve from 0.88607
56/56 [=====] - 29s 511ms/step - loss: 2.2725 - accuracy: 0.9943 - val_loss: 4.

```

4265 - val_accuracy: 0.8799
Epoch 35/100
56/56 [=====] - ETA: 0s - loss: 2.2265 - accuracy: 0.9935
Epoch 35: val_accuracy did not improve from 0.88607
56/56 [=====] - 28s 505ms/step - loss: 2.2265 - accuracy: 0.9935 - val_loss: 4.
5263 - val_accuracy: 0.8796
Epoch 36/100
56/56 [=====] - ETA: 0s - loss: 2.1845 - accuracy: 0.9921
Epoch 36: val_accuracy did not improve from 0.88607
56/56 [=====] - 29s 514ms/step - loss: 2.1845 - accuracy: 0.9921 - val_loss: 4.
0289 - val_accuracy: 0.8634
Epoch 37/100
56/56 [=====] - ETA: 0s - loss: 2.3969 - accuracy: 0.9688
Epoch 37: val_accuracy did not improve from 0.88607
56/56 [=====] - 29s 515ms/step - loss: 2.3969 - accuracy: 0.9688 - val_loss: 4.
0577 - val_accuracy: 0.8715
Epoch 38/100
56/56 [=====] - ETA: 0s - loss: 2.9464 - accuracy: 0.9521
Epoch 38: val_accuracy did not improve from 0.88607
56/56 [=====] - 30s 545ms/step - loss: 2.9464 - accuracy: 0.9521 - val_loss: 5.
7500 - val_accuracy: 0.8782
Epoch 39/100
56/56 [=====] - ETA: 0s - loss: 2.8399 - accuracy: 0.9623
Epoch 39: val_accuracy did not improve from 0.88607
56/56 [=====] - 29s 526ms/step - loss: 2.8399 - accuracy: 0.9623 - val_loss: 7.
2841 - val_accuracy: 0.7939
Epoch 39: early stopping

```

```

In [ ]: accuracy = model.evaluate_generator(generator= test_ds)[1]
print(f"The accuracy of your model is = {accuracy*100} %")

```

```

/var/folders/gf/dq4lmmx17b71h0cbpbkxfj9c0000gn/T/ipykernel_13194/924913612.py:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.

```

```

accuracy = model.evaluate_generator(generator= test_ds)[1]
The accuracy of your model is = 79.39122915267944 %

```

```
In [ ]: import matplotlib.pyplot as plt
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

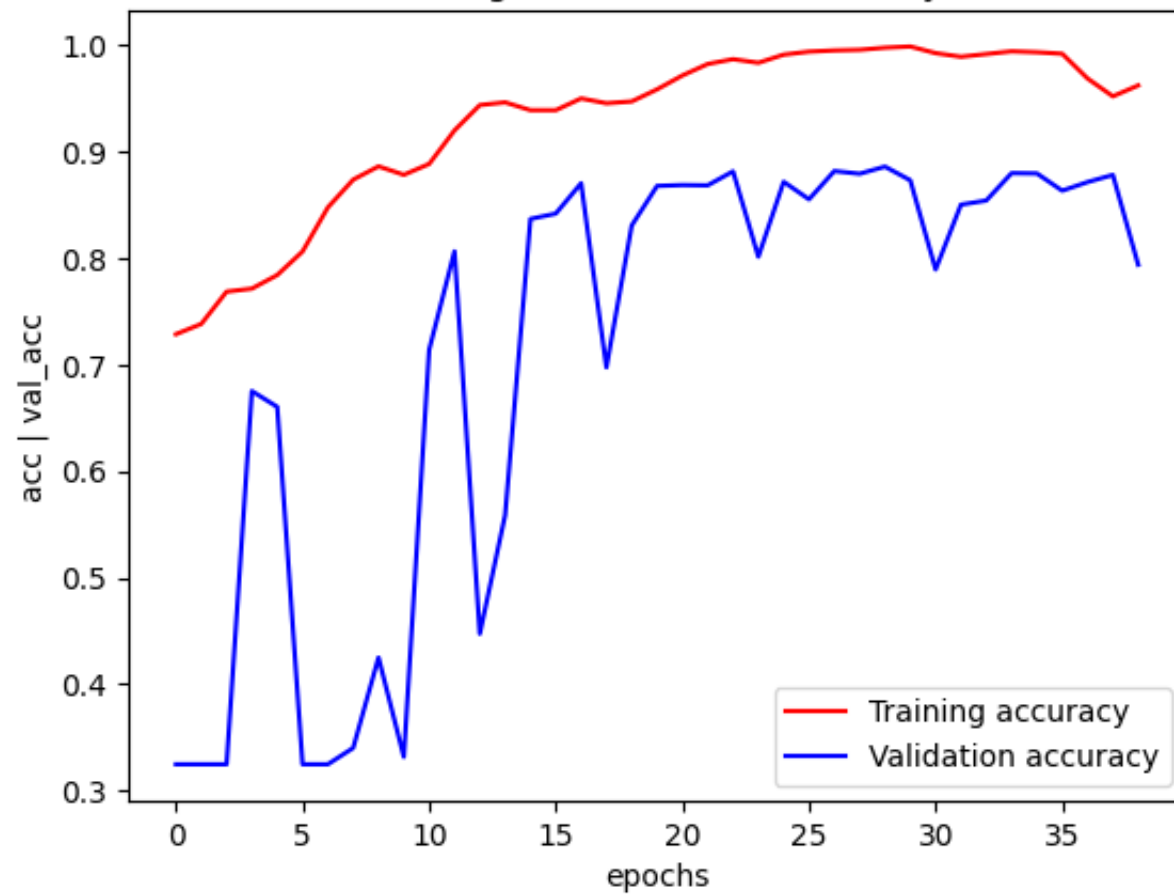
epochs = range(len(acc))

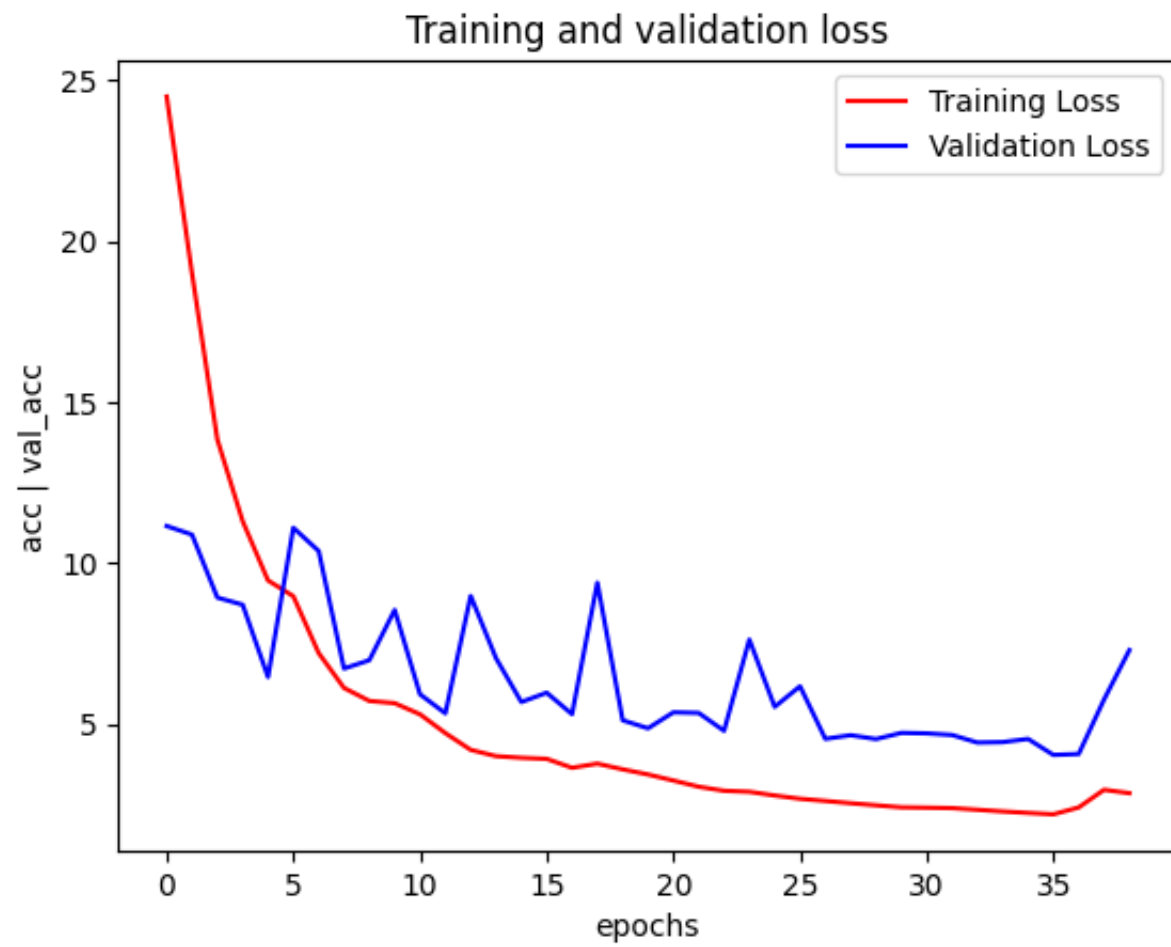
plt.plot(epochs, acc, 'r', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.xlabel("epochs")
plt.ylabel("acc | val_acc")
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()

plt.plot(epochs, loss, 'r', label='Training Loss')
plt.plot(epochs, val_loss, 'b', label='Validation Loss')
plt.xlabel("epochs")
plt.ylabel("acc | val_acc")
plt.title('Training and validation loss')
plt.legend()

plt.show()
```

Training and validation accuracy





Squeezenet

```
In [ ]: from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, concatenate, Dropout, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.layers.experimental.preprocessing import Rescaling
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define input tensor and add Rescaling layer for data normalization
input_tensor = Input(shape=(180, 180, 3))
normalized_input = Rescaling(scale=1./255)(input_tensor) # Normalize input data

# SqueezeNet layers
x = Conv2D(96, kernel_size=7, strides=2, activation='relu', padding='same')(normalized_input)
x = MaxPooling2D(pool_size=3, strides=2)(x)

# Rest of your SqueezeNet architecture...

# Output layers
x = Dropout(0.5)(x)
x = Conv2D(filters=5, kernel_size=1, activation='relu', padding='same')(x)
x = GlobalAveragePooling2D()(x)
output_tensor = Dense(1, activation='sigmoid')(x) # Use 'sigmoid' for binary classification

# Create model
model_squeezenet = Model(inputs=input_tensor, outputs=output_tensor)

# Compile model
model_squeezenet.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Print model summary
model_squeezenet.summary()
```

Model: "model_2"

| Layer (type) | Output Shape | Param # |
|-------------------------|-----------------------|---------|
| ===== | | |
| input_3 (InputLayer) | [(None, 180, 180, 3)] | 0 |
| rescaling_4 (Rescaling) | (None, 180, 180, 3) | 0 |
| conv2d_104 (Conv2D) | (None, 90, 90, 96) | 14208 |

| | | |
|---|--------------------|-----|
| max_pooling2d_13 (MaxPooling2D) | (None, 44, 44, 96) | 0 |
| dropout_6 (Dropout) | (None, 44, 44, 96) | 0 |
| conv2d_105 (Conv2D) | (None, 44, 44, 5) | 485 |
| global_average_pooling2d_2 (GlobalAveragePooling2D) | (None, 5) | 0 |
| dense_8 (Dense) | (None, 1) | 6 |

=====
Total params: 14699 (57.42 KB)
Trainable params: 14699 (57.42 KB)
Non-trainable params: 0 (0.00 Byte)

| Layer (type) | Output Shape | Param # |
|---|-----------------------|---------|
| input_3 (InputLayer) | [(None, 180, 180, 3)] | 0 |
| rescaling_4 (Rescaling) | (None, 180, 180, 3) | 0 |
| conv2d_104 (Conv2D) | (None, 90, 90, 96) | 14208 |
| max_pooling2d_13 (MaxPooling2D) | (None, 44, 44, 96) | 0 |
| dropout_6 (Dropout) | (None, 44, 44, 96) | 0 |
| conv2d_105 (Conv2D) | (None, 44, 44, 5) | 485 |
| global_average_pooling2d_2 (GlobalAveragePooling2D) | (None, 5) | 0 |
| dense_8 (Dense) | (None, 1) | 6 |

=====
Total params: 14699 (57.42 KB)

Trainable params: 14699 (57.42 KB)
Non-trainable params: 0 (0.00 Byte)

```
In [ ]: mc_sqnet = ModelCheckpoint(filepath="squeezenet_net.hdf5", monitor= 'val_accuracy', verbose= 1, save_best_only= 1,
call_back_sqnet = [ mc_sqnet,es ];
```

```
In [ ]: history_sqnet = model_squeezenet.fit(train_ds,
                                             epochs= 150,
                                             validation_data= validation_ds,
                                             validation_steps= 6,
                                             callbacks = call_back_sqnet )
```

Epoch 1/150

2023-10-30 16:32:17.282402: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.

56/56 [=====] - ETA: 0s - loss: 0.6077 - accuracy: 0.6734

2023-10-30 16:32:28.033513: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.

Epoch 1: val_accuracy improved from -inf to 0.68490, saving model to squeezenet_net.hdf5

56/56 [=====] - 13s 154ms/step - loss: 0.6077 - accuracy: 0.6734 - val_loss: 0.5875 - val_accuracy: 0.6849

Epoch 2/150

55/56 [=====>.] - ETA: 0s - loss: 0.5768 - accuracy: 0.6798

Epoch 2: val_accuracy did not improve from 0.68490

56/56 [=====] - 4s 77ms/step - loss: 0.5769 - accuracy: 0.6796 - val_loss: 0.5957 - val_accuracy: 0.6771

Epoch 3/150

55/56 [=====>.] - ETA: 0s - loss: 0.5748 - accuracy: 0.6798

Epoch 3: val_accuracy improved from 0.68490 to 0.69531, saving model to squeezenet_net.hdf5

56/56 [=====] - 4s 74ms/step - loss: 0.5749 - accuracy: 0.6796 - val_loss: 0.5681 - val_accuracy: 0.6953

Epoch 4/150

56/56 [=====] - ETA: 0s - loss: 0.5734 - accuracy: 0.6796

Epoch 4: val_accuracy did not improve from 0.69531

56/56 [=====] - 4s 72ms/step - loss: 0.5734 - accuracy: 0.6796 - val_loss: 0.5978 - val_accuracy: 0.6745

Epoch 5/150

56/56 [=====] - ETA: 0s - loss: 0.5718 - accuracy: 0.6796

Epoch 5: val_accuracy did not improve from 0.69531

56/56 [=====] - 4s 77ms/step - loss: 0.5718 - accuracy: 0.6796 - val_loss: 0.5923 - val_accuracy: 0.6797
Epoch 6/150
55/56 [=====>.] - ETA: 0s - loss: 0.5701 - accuracy: 0.6798
Epoch 6: val_accuracy did not improve from 0.69531
56/56 [=====] - 4s 76ms/step - loss: 0.5701 - accuracy: 0.6796 - val_loss: 0.5820 - val_accuracy: 0.6771
Epoch 7/150
55/56 [=====>.] - ETA: 0s - loss: 0.5684 - accuracy: 0.6798
Epoch 7: val_accuracy did not improve from 0.69531
56/56 [=====] - 4s 70ms/step - loss: 0.5685 - accuracy: 0.6796 - val_loss: 0.5757 - val_accuracy: 0.6901
Epoch 8/150
56/56 [=====] - ETA: 0s - loss: 0.5625 - accuracy: 0.6799
Epoch 8: val_accuracy did not improve from 0.69531
56/56 [=====] - 4s 70ms/step - loss: 0.5625 - accuracy: 0.6799 - val_loss: 0.5850 - val_accuracy: 0.6693
Epoch 9/150
55/56 [=====>.] - ETA: 0s - loss: 0.5468 - accuracy: 0.6895
Epoch 9: val_accuracy improved from 0.69531 to 0.72917, saving model to squeezenet_net.hdf5
56/56 [=====] - 4s 69ms/step - loss: 0.5467 - accuracy: 0.6893 - val_loss: 0.5391 - val_accuracy: 0.7292
Epoch 10/150
56/56 [=====] - ETA: 0s - loss: 0.5311 - accuracy: 0.7193
Epoch 10: val_accuracy improved from 0.72917 to 0.73698, saving model to squeezenet_net.hdf5
56/56 [=====] - 4s 71ms/step - loss: 0.5311 - accuracy: 0.7193 - val_loss: 0.5458 - val_accuracy: 0.7370
Epoch 11/150
55/56 [=====>.] - ETA: 0s - loss: 0.5189 - accuracy: 0.7472
Epoch 11: val_accuracy improved from 0.73698 to 0.75521, saving model to squeezenet_net.hdf5
56/56 [=====] - 4s 69ms/step - loss: 0.5188 - accuracy: 0.7471 - val_loss: 0.5271 - val_accuracy: 0.7552
Epoch 12/150
55/56 [=====>.] - ETA: 0s - loss: 0.5092 - accuracy: 0.7724
Epoch 12: val_accuracy did not improve from 0.75521
56/56 [=====] - 4s 69ms/step - loss: 0.5090 - accuracy: 0.7723 - val_loss: 0.5697 - val_accuracy: 0.7474
Epoch 13/150
55/56 [=====>.] - ETA: 0s - loss: 0.5013 - accuracy: 0.7858
Epoch 13: val_accuracy did not improve from 0.75521

56/56 [=====] - 4s 69ms/step - loss: 0.5011 - accuracy: 0.7857 - val_loss: 0.53
37 - val_accuracy: 0.7500
Epoch 14/150
55/56 [=====>.] - ETA: 0s - loss: 0.4963 - accuracy: 0.7866
Epoch 14: val_accuracy did not improve from 0.75521
56/56 [=====] - 4s 69ms/step - loss: 0.4961 - accuracy: 0.7868 - val_loss: 0.55
58 - val_accuracy: 0.7370
Epoch 15/150
55/56 [=====>.] - ETA: 0s - loss: 0.4933 - accuracy: 0.7886
Epoch 15: val_accuracy did not improve from 0.75521
56/56 [=====] - 4s 69ms/step - loss: 0.4930 - accuracy: 0.7888 - val_loss: 0.54
89 - val_accuracy: 0.7370
Epoch 16/150
55/56 [=====>.] - ETA: 0s - loss: 0.4911 - accuracy: 0.7875
Epoch 16: val_accuracy improved from 0.75521 to 0.75781, saving model to squeezenet_net.hdf5
56/56 [=====] - 4s 70ms/step - loss: 0.4909 - accuracy: 0.7876 - val_loss: 0.51
27 - val_accuracy: 0.7578
Epoch 17/150
55/56 [=====>.] - ETA: 0s - loss: 0.4896 - accuracy: 0.7886
Epoch 17: val_accuracy did not improve from 0.75781
56/56 [=====] - 4s 70ms/step - loss: 0.4893 - accuracy: 0.7888 - val_loss: 0.56
41 - val_accuracy: 0.7292
Epoch 18/150
56/56 [=====] - ETA: 0s - loss: 0.4880 - accuracy: 0.7899
Epoch 18: val_accuracy improved from 0.75781 to 0.76302, saving model to squeezenet_net.hdf5
56/56 [=====] - 4s 70ms/step - loss: 0.4880 - accuracy: 0.7899 - val_loss: 0.51
83 - val_accuracy: 0.7630
Epoch 19/150
55/56 [=====>.] - ETA: 0s - loss: 0.4872 - accuracy: 0.7912
Epoch 19: val_accuracy did not improve from 0.76302
56/56 [=====] - 4s 69ms/step - loss: 0.4869 - accuracy: 0.7913 - val_loss: 0.59
59 - val_accuracy: 0.7266
Epoch 20/150
55/56 [=====>.] - ETA: 0s - loss: 0.4861 - accuracy: 0.7915
Epoch 20: val_accuracy did not improve from 0.76302
56/56 [=====] - 4s 68ms/step - loss: 0.4858 - accuracy: 0.7916 - val_loss: 0.56
02 - val_accuracy: 0.7344
Epoch 21/150
55/56 [=====>.] - ETA: 0s - loss: 0.4850 - accuracy: 0.7918
Epoch 21: val_accuracy did not improve from 0.76302

56/56 [=====] - 4s 68ms/step - loss: 0.4847 - accuracy: 0.7919 - val_loss: 0.5515 - val_accuracy: 0.7396
Epoch 22/150
55/56 [=====>.] - ETA: 0s - loss: 0.4837 - accuracy: 0.7926
Epoch 22: val_accuracy did not improve from 0.76302
56/56 [=====] - 4s 69ms/step - loss: 0.4834 - accuracy: 0.7927 - val_loss: 0.5462 - val_accuracy: 0.7448
Epoch 23/150
55/56 [=====>.] - ETA: 0s - loss: 0.4824 - accuracy: 0.7940
Epoch 23: val_accuracy did not improve from 0.76302
56/56 [=====] - 4s 72ms/step - loss: 0.4821 - accuracy: 0.7942 - val_loss: 0.5670 - val_accuracy: 0.7266
Epoch 24/150
55/56 [=====>.] - ETA: 0s - loss: 0.4810 - accuracy: 0.7963
Epoch 24: val_accuracy improved from 0.76302 to 0.76823, saving model to squeezenet_net.hdf5
56/56 [=====] - 4s 71ms/step - loss: 0.4806 - accuracy: 0.7964 - val_loss: 0.5198 - val_accuracy: 0.7682
Epoch 25/150
55/56 [=====>.] - ETA: 0s - loss: 0.4795 - accuracy: 0.7969
Epoch 25: val_accuracy did not improve from 0.76823
56/56 [=====] - 4s 69ms/step - loss: 0.4791 - accuracy: 0.7970 - val_loss: 0.6169 - val_accuracy: 0.7135
Epoch 26/150
55/56 [=====>.] - ETA: 0s - loss: 0.4778 - accuracy: 0.7980
Epoch 26: val_accuracy did not improve from 0.76823
56/56 [=====] - 4s 69ms/step - loss: 0.4774 - accuracy: 0.7981 - val_loss: 0.5695 - val_accuracy: 0.7214
Epoch 27/150
55/56 [=====>.] - ETA: 0s - loss: 0.4759 - accuracy: 0.7989
Epoch 27: val_accuracy did not improve from 0.76823
56/56 [=====] - 4s 69ms/step - loss: 0.4756 - accuracy: 0.7990 - val_loss: 0.5371 - val_accuracy: 0.7500
Epoch 28/150
55/56 [=====>.] - ETA: 0s - loss: 0.4735 - accuracy: 0.7991
Epoch 28: val_accuracy did not improve from 0.76823
56/56 [=====] - 4s 70ms/step - loss: 0.4731 - accuracy: 0.7993 - val_loss: 0.5772 - val_accuracy: 0.7526
Epoch 29/150
55/56 [=====>.] - ETA: 0s - loss: 0.4712 - accuracy: 0.8009
Epoch 29: val_accuracy did not improve from 0.76823

```

56/56 [=====] - 4s 70ms/step - loss: 0.4709 - accuracy: 0.8010 - val_loss: 0.57
80 - val_accuracy: 0.7578
Epoch 30/150
55/56 [=====>.] - ETA: 0s - loss: 0.4689 - accuracy: 0.8020
Epoch 30: val_accuracy did not improve from 0.76823
56/56 [=====] - 4s 69ms/step - loss: 0.4686 - accuracy: 0.8021 - val_loss: 0.59
46 - val_accuracy: 0.7604
Epoch 31/150
56/56 [=====] - ETA: 0s - loss: 0.4664 - accuracy: 0.8024
Epoch 31: val_accuracy did not improve from 0.76823
56/56 [=====] - 4s 70ms/step - loss: 0.4664 - accuracy: 0.8024 - val_loss: 0.60
72 - val_accuracy: 0.7526
Epoch 32/150
55/56 [=====>.] - ETA: 0s - loss: 0.4647 - accuracy: 0.8031
Epoch 32: val_accuracy did not improve from 0.76823
56/56 [=====] - 4s 69ms/step - loss: 0.4643 - accuracy: 0.8032 - val_loss: 0.64
63 - val_accuracy: 0.7370
Epoch 33/150
55/56 [=====>.] - ETA: 0s - loss: 0.4629 - accuracy: 0.8037
Epoch 33: val_accuracy did not improve from 0.76823
56/56 [=====] - 4s 69ms/step - loss: 0.4626 - accuracy: 0.8038 - val_loss: 0.64
37 - val_accuracy: 0.7526
Epoch 34/150
55/56 [=====>.] - ETA: 0s - loss: 0.4613 - accuracy: 0.8057
Epoch 34: val_accuracy did not improve from 0.76823
56/56 [=====] - 4s 70ms/step - loss: 0.4609 - accuracy: 0.8058 - val_loss: 0.64
59 - val_accuracy: 0.7370
Epoch 34: early stopping

```

```

In [ ]: accuracy_sqnet = model_squeezenet.evaluate_generator(generator= test_ds)[1]
print(f"The accuracy of your model is = {accuracy_sqnet*100} %")

```

```

/var/folders/gf/dq4lmmx17b71h0cbpbkwfj9c0000gn/T/ipykernel_13194/2629914926.py:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.

```

```

accuracy_sqnet = model_squeezenet.evaluate_generator(generator= test_ds)[1]
2023-10-30 16:34:41.130944: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.
The accuracy of your model is = 82.5747013092041 %

```



```
In [ ]: import matplotlib.pyplot as plt
acc = history_sqnet.history['accuracy']
val_acc = history_sqnet.history['val_accuracy']
loss = history_sqnet.history['loss']
val_loss = history_sqnet.history['val_loss']

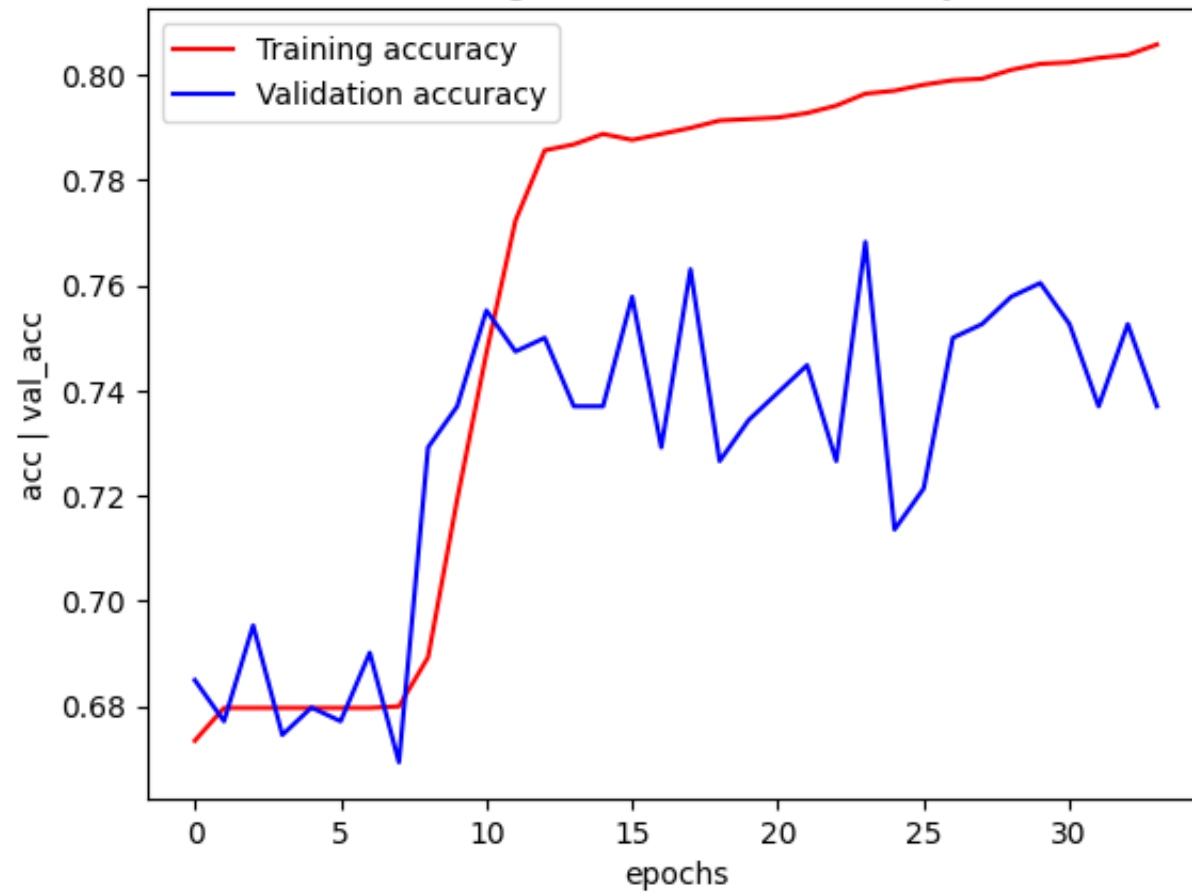
epochs = range(len(acc))

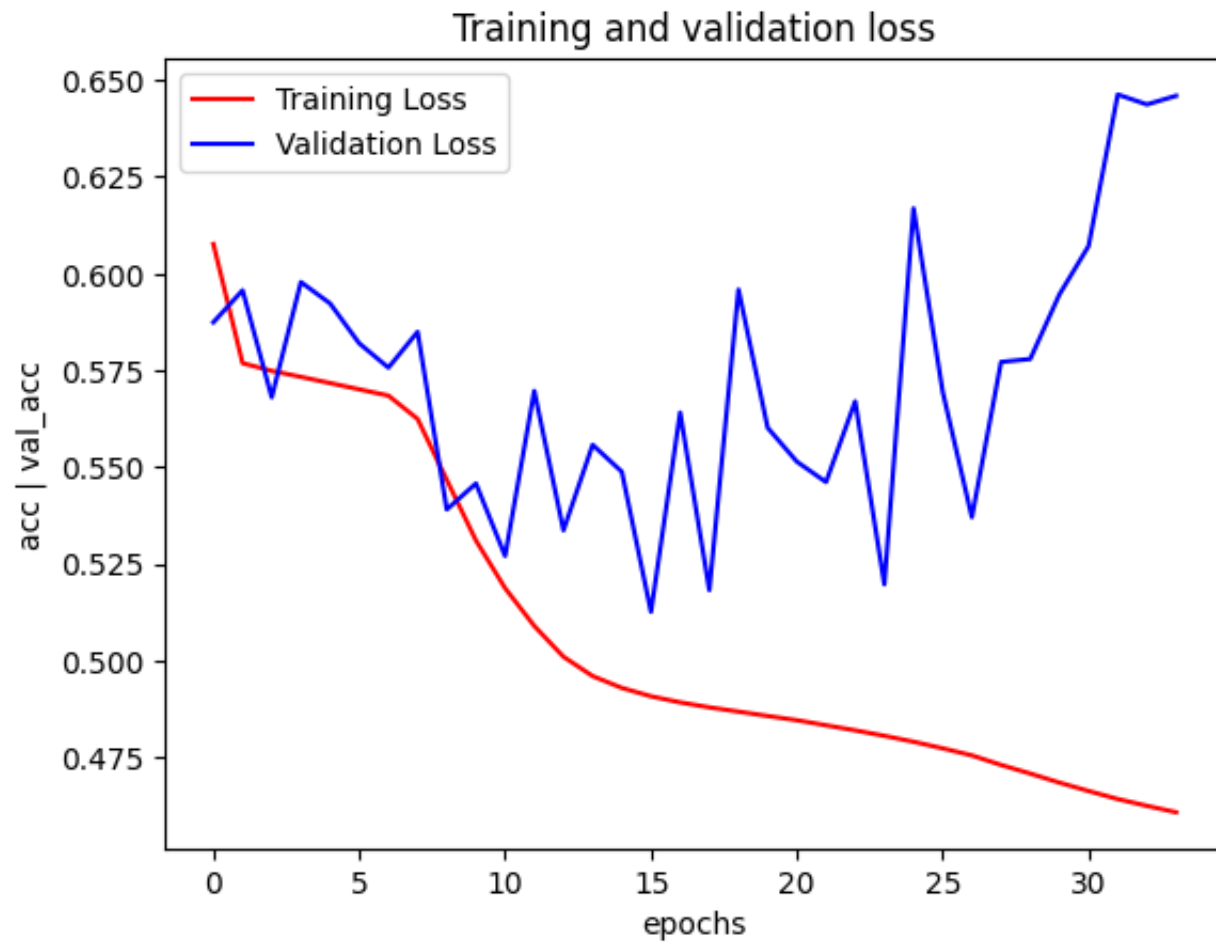
plt.plot(epochs, acc, 'r', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.xlabel("epochs")
plt.ylabel("acc | val_acc")
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()

plt.plot(epochs, loss, 'r', label='Training Loss')
plt.plot(epochs, val_loss, 'b', label='Validation Loss')
plt.xlabel("epochs")
plt.ylabel("acc | val_acc")
plt.title('Training and validation loss')
plt.legend()

plt.show()
```

Training and validation accuracy





GoogleNet (InceptionV3)

```
In [ ]: from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.layers import Input, GlobalAveragePooling2D, Dense, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.layers.experimental.preprocessing import Rescaling

# Define input tensor and add Rescaling layer for data normalization
input_tensor = Input(shape=(180, 180, 3))
normalized_input = Rescaling(scale=1./255)(input_tensor) # Normalize input data

# Load pre-trained InceptionV3 model without top classification layers
base_model = InceptionV3(weights='imagenet', include_top=False, input_tensor=normalized_input)

# Add custom classification layers on top of the pre-trained model
x = GlobalAveragePooling2D()(base_model.output)
x = Dropout(0.5)(x)
output_tensor = Dense(1, activation='sigmoid')(x) # Use 'sigmoid' for binary classification

# Create model
model_inceptionv3 = Model(inputs=base_model.input, outputs=output_tensor)

# Compile model
model_inceptionv3.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Print model summary
model_inceptionv3.summary()
```

Model: "model_3"

| Layer (type) | Output Shape | Param # | Connected to |
|--|-----------------------|---------|-----------------------------------|
| input_4 (InputLayer) | [(None, 180, 180, 3)] | 0 | [] |
| rescaling_5 (Rescaling) | (None, 180, 180, 3) | 0 | ['input_4[0][0]'] |
| conv2d_106 (Conv2D) | (None, 89, 89, 32) | 864 | ['rescaling_5[0][0]'] |
| batch_normalization_102 (BatchNormalization) | (None, 89, 89, 32) | 96 | ['conv2d_106[0][0]'] |
| activation_94 (Activation) | (None, 89, 89, 32) | 0 | ['batch_normalization_102[0][0]'] |

| | | | ']'] |
|--|-----------------------|---------|-----------------------------------|
| conv2d_107 (Conv2D) | (None, 87, 87, 32) | 9216 | ['activation_94[0][0]'] |
| batch_normalization_103 (BatchNormalization) | (None, 87, 87, 32) | 96 | ['conv2d_107[0][0]'] |
| activation_95 (Activation) | (None, 87, 87, 32) | 0 | ['batch_normalization_103[0][0]'] |
| conv2d_108 (Conv2D) | (None, 87, 87, 64) | 18432 | ['activation_95[0][0]'] |
| batch_normalization_104 (BatchNormalization) | (None, 87, 87, 64) | 192 | ['conv2d_108[0][0]'] |
| activation_96 (Activation) | (None, 87, 87, 64) | 0 | ['batch_normalization_104[0][0]'] |
| max_pooling2d_14 (MaxPooling2D) | (None, 43, 43, 64) | 0 | ['activation_96[0][0]'] |
| conv2d_109 (Conv2D) | (None, 43, 43, 80) | 5120 | ['max_pooling2d_14[0][0]'] |
| batch_normalization_105 (BatchNormalization) | (None, 43, 43, 80) | 240 | ['conv2d_109[0][0]'] |
| activation_97 (Activation) | (None, 43, 43, 80) | 0 | ['batch_normalization_105[0][0]'] |
| conv2d_110 (Conv2D) | (None, 41, 41, 192) | 138240 | ['activation_97[0][0]'] |
| batch_normalization_106 (BatchNormalization) | (None, 41, 41, 192) | 576 | ['conv2d_110[0][0]'] |
| Layer (type) | Output Shape | Param # | Connected to |
| input_4 (InputLayer) | [(None, 180, 180, 3)] | 0 | [] |
| rescaling_5 (Rescaling) | (None, 180, 180, 3) | 0 | ['input_4[0][0]'] |
| conv2d_106 (Conv2D) | (None, 89, 89, 32) | 864 | ['rescaling_5[0][0]'] |

| | | | |
|--|---------------------|--------|-----------------------------------|
| batch_normalization_102 (BatchNormalization) | (None, 89, 89, 32) | 96 | ['conv2d_106[0][0]'] |
| activation_94 (Activation) | (None, 89, 89, 32) | 0 | ['batch_normalization_102[0][0]'] |
| conv2d_107 (Conv2D) | (None, 87, 87, 32) | 9216 | ['activation_94[0][0]'] |
| batch_normalization_103 (BatchNormalization) | (None, 87, 87, 32) | 96 | ['conv2d_107[0][0]'] |
| activation_95 (Activation) | (None, 87, 87, 32) | 0 | ['batch_normalization_103[0][0]'] |
| conv2d_108 (Conv2D) | (None, 87, 87, 64) | 18432 | ['activation_95[0][0]'] |
| batch_normalization_104 (BatchNormalization) | (None, 87, 87, 64) | 192 | ['conv2d_108[0][0]'] |
| activation_96 (Activation) | (None, 87, 87, 64) | 0 | ['batch_normalization_104[0][0]'] |
| max_pooling2d_14 (MaxPooling2D) | (None, 43, 43, 64) | 0 | ['activation_96[0][0]'] |
| conv2d_109 (Conv2D) | (None, 43, 43, 80) | 5120 | ['max_pooling2d_14[0][0]'] |
| batch_normalization_105 (BatchNormalization) | (None, 43, 43, 80) | 240 | ['conv2d_109[0][0]'] |
| activation_97 (Activation) | (None, 43, 43, 80) | 0 | ['batch_normalization_105[0][0]'] |
| conv2d_110 (Conv2D) | (None, 41, 41, 192) | 138240 | ['activation_97[0][0]'] |
| batch_normalization_106 (BatchNormalization) | (None, 41, 41, 192) | 576 | ['conv2d_110[0][0]'] |
| activation_98 (Activation) | (None, 41, 41, 192) | 0 | ['batch_normalization_106[0][0]'] |

| | | | |
|--|---------------------|-------|-----------------------------------|
| | | | ']'] |
| max_pooling2d_15 (MaxPooling2D) | (None, 20, 20, 192) | 0 | ['activation_98[0][0]'] |
| conv2d_114 (Conv2D) | (None, 20, 20, 64) | 12288 | ['max_pooling2d_15[0][0]'] |
| batch_normalization_110 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_114[0][0]'] |
| activation_102 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_110[0][0]'] |
| conv2d_112 (Conv2D) | (None, 20, 20, 48) | 9216 | ['max_pooling2d_15[0][0]'] |
| conv2d_115 (Conv2D) | (None, 20, 20, 96) | 55296 | ['activation_102[0][0]'] |
| batch_normalization_108 (BatchNormalization) | (None, 20, 20, 48) | 144 | ['conv2d_112[0][0]'] |
| batch_normalization_111 (BatchNormalization) | (None, 20, 20, 96) | 288 | ['conv2d_115[0][0]'] |
| activation_100 (Activation) | (None, 20, 20, 48) | 0 | ['batch_normalization_108[0][0]'] |
| activation_103 (Activation) | (None, 20, 20, 96) | 0 | ['batch_normalization_111[0][0]'] |
| average_pooling2d_9 (AveragePooling2D) | (None, 20, 20, 192) | 0 | ['max_pooling2d_15[0][0]'] |
| conv2d_111 (Conv2D) | (None, 20, 20, 64) | 12288 | ['max_pooling2d_15[0][0]'] |
| conv2d_113 (Conv2D) | (None, 20, 20, 64) | 76800 | ['activation_100[0][0]'] |
| conv2d_116 (Conv2D) | (None, 20, 20, 96) | 82944 | ['activation_103[0][0]'] |
| conv2d_117 (Conv2D) | (None, 20, 20, 32) | 6144 | ['average_pooling2d_9[0][0]'] |

| | | | |
|--|---------------------|-------|--|
| batch_normalization_107 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_111[0][0]'] |
| batch_normalization_109 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_113[0][0]'] |
| batch_normalization_112 (BatchNormalization) | (None, 20, 20, 96) | 288 | ['conv2d_116[0][0]'] |
| batch_normalization_113 (BatchNormalization) | (None, 20, 20, 32) | 96 | ['conv2d_117[0][0]'] |
| activation_99 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_107[0][0]'] |
| activation_101 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_109[0][0]'] |
| activation_104 (Activation) | (None, 20, 20, 96) | 0 | ['batch_normalization_112[0][0]'] |
| activation_105 (Activation) | (None, 20, 20, 32) | 0 | ['batch_normalization_113[0][0]'] |
| mixed0 (Concatenate) | (None, 20, 20, 256) | 0 | ['activation_99[0][0]', 'activation_101[0][0]', 'activation_104[0][0]', 'activation_105[0][0]'] |
| conv2d_121 (Conv2D) | (None, 20, 20, 64) | 16384 | ['mixed0[0][0]'] |
| batch_normalization_117 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_121[0][0]'] |
| activation_109 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_117[0][0]'] |
| conv2d_119 (Conv2D) | (None, 20, 20, 48) | 12288 | ['mixed0[0][0]'] |
| conv2d_122 (Conv2D) | (None, 20, 20, 96) | 55296 | ['activation_109[0][0]'] |

| | | | |
|--|---------------------|-------|-----------------------------------|
| batch_normalization_115 (BatchNormalization) | (None, 20, 20, 48) | 144 | ['conv2d_119[0][0]'] |
| batch_normalization_118 (BatchNormalization) | (None, 20, 20, 96) | 288 | ['conv2d_122[0][0]'] |
| activation_107 (Activation) | (None, 20, 20, 48) | 0 | ['batch_normalization_115[0][0]'] |
| activation_110 (Activation) | (None, 20, 20, 96) | 0 | ['batch_normalization_118[0][0]'] |
| average_pooling2d_10 (AveragePooling2D) | (None, 20, 20, 256) | 0 | ['mixed0[0][0]'] |
| conv2d_118 (Conv2D) | (None, 20, 20, 64) | 16384 | ['mixed0[0][0]'] |
| conv2d_120 (Conv2D) | (None, 20, 20, 64) | 76800 | ['activation_107[0][0]'] |
| conv2d_123 (Conv2D) | (None, 20, 20, 96) | 82944 | ['activation_110[0][0]'] |
| conv2d_124 (Conv2D) | (None, 20, 20, 64) | 16384 | ['average_pooling2d_10[0][0]'] |
| batch_normalization_114 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_118[0][0]'] |
| batch_normalization_116 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_120[0][0]'] |
| batch_normalization_119 (BatchNormalization) | (None, 20, 20, 96) | 288 | ['conv2d_123[0][0]'] |
| batch_normalization_120 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_124[0][0]'] |
| activation_106 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_114[0][0]'] |
| activation_108 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_116[0][0]'] |

| | | | |
|--|---------------------|-------|---|
|) | | | [''] |
| activation_111 (Activation) | (None, 20, 20, 96) | 0 | ['batch_normalization_119[0][0]'] |
| activation_112 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_120[0][0]'] |
| mixed1 (Concatenate) | (None, 20, 20, 288) | 0 | ['activation_106[0][0]', 'activation_108[0][0]', 'activation_111[0][0]', 'activation_112[0][0]'] |
| conv2d_128 (Conv2D) | (None, 20, 20, 64) | 18432 | ['mixed1[0][0]'] |
| batch_normalization_124 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_128[0][0]'] |
| activation_116 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_124[0][0]'] |
| conv2d_126 (Conv2D) | (None, 20, 20, 48) | 13824 | ['mixed1[0][0]'] |
| conv2d_129 (Conv2D) | (None, 20, 20, 96) | 55296 | ['activation_116[0][0]'] |
| batch_normalization_122 (BatchNormalization) | (None, 20, 20, 48) | 144 | ['conv2d_126[0][0]'] |
| batch_normalization_125 (BatchNormalization) | (None, 20, 20, 96) | 288 | ['conv2d_129[0][0]'] |
| activation_114 (Activation) | (None, 20, 20, 48) | 0 | ['batch_normalization_122[0][0]'] |
| activation_117 (Activation) | (None, 20, 20, 96) | 0 | ['batch_normalization_125[0][0]'] |
| average_pooling2d_11 (AveragePooling2D) | (None, 20, 20, 288) | 0 | ['mixed1[0][0]'] |

| | | | |
|--|---------------------|-------|---|
| conv2d_125 (Conv2D) | (None, 20, 20, 64) | 18432 | ['mixed1[0][0]'] |
| conv2d_127 (Conv2D) | (None, 20, 20, 64) | 76800 | ['activation_114[0][0]'] |
| conv2d_130 (Conv2D) | (None, 20, 20, 96) | 82944 | ['activation_117[0][0]'] |
| conv2d_131 (Conv2D) | (None, 20, 20, 64) | 18432 | ['average_pooling2d_11[0][0]'] |
| batch_normalization_121 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_125[0][0]'] |
| batch_normalization_123 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_127[0][0]'] |
| batch_normalization_126 (BatchNormalization) | (None, 20, 20, 96) | 288 | ['conv2d_130[0][0]'] |
| batch_normalization_127 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_131[0][0]'] |
| activation_113 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_121[0][0]'] |
| activation_115 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_123[0][0]'] |
| activation_118 (Activation) | (None, 20, 20, 96) | 0 | ['batch_normalization_126[0][0]'] |
| activation_119 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_127[0][0]'] |
| mixed2 (Concatenate) | (None, 20, 20, 288) | 0 | ['activation_113[0][0]', 'activation_115[0][0]', 'activation_118[0][0]', 'activation_119[0][0]'] |
| conv2d_133 (Conv2D) | (None, 20, 20, 64) | 18432 | ['mixed2[0][0]'] |
| batch_normalization_129 (BatchNormalization) | (None, 20, 20, 64) | 192 | ['conv2d_133[0][0]'] |

| | | | |
|--|--------------------|--------|--|
| atchNormalization) | | | |
| activation_121 (Activation) | (None, 20, 20, 64) | 0 | ['batch_normalization_129[0][0]'] |
| conv2d_134 (Conv2D) | (None, 20, 20, 96) | 55296 | ['activation_121[0][0]'] |
| batch_normalization_130 (BatchNormalization) | (None, 20, 20, 96) | 288 | ['conv2d_134[0][0]'] |
| activation_122 (Activation) | (None, 20, 20, 96) | 0 | ['batch_normalization_130[0][0]'] |
| conv2d_132 (Conv2D) | (None, 9, 9, 384) | 995328 | ['mixed2[0][0]'] |
| conv2d_135 (Conv2D) | (None, 9, 9, 96) | 82944 | ['activation_122[0][0]'] |
| batch_normalization_128 (BatchNormalization) | (None, 9, 9, 384) | 1152 | ['conv2d_132[0][0]'] |
| batch_normalization_131 (BatchNormalization) | (None, 9, 9, 96) | 288 | ['conv2d_135[0][0]'] |
| activation_120 (Activation) | (None, 9, 9, 384) | 0 | ['batch_normalization_128[0][0]'] |
| activation_123 (Activation) | (None, 9, 9, 96) | 0 | ['batch_normalization_131[0][0]'] |
| max_pooling2d_16 (MaxPooling2D) | (None, 9, 9, 288) | 0 | ['mixed2[0][0]'] |
| mixed3 (Concatenate) | (None, 9, 9, 768) | 0 | ['activation_120[0][0]', 'activation_123[0][0]', 'max_pooling2d_16[0][0]'] |
| conv2d_140 (Conv2D) | (None, 9, 9, 128) | 98304 | ['mixed3[0][0]'] |
| batch_normalization_136 (BatchNormalization) | (None, 9, 9, 128) | 384 | ['conv2d_140[0][0]'] |

| | | | |
|--|-------------------|--------|-----------------------------------|
| activation_128 (Activation) | (None, 9, 9, 128) | 0 | ['batch_normalization_136[0][0]'] |
| conv2d_141 (Conv2D) | (None, 9, 9, 128) | 114688 | ['activation_128[0][0]'] |
| batch_normalization_137 (BatchNormalization) | (None, 9, 9, 128) | 384 | ['conv2d_141[0][0]'] |
| activation_129 (Activation) | (None, 9, 9, 128) | 0 | ['batch_normalization_137[0][0]'] |
| conv2d_137 (Conv2D) | (None, 9, 9, 128) | 98304 | ['mixed3[0][0]'] |
| conv2d_142 (Conv2D) | (None, 9, 9, 128) | 114688 | ['activation_129[0][0]'] |
| batch_normalization_133 (BatchNormalization) | (None, 9, 9, 128) | 384 | ['conv2d_137[0][0]'] |
| batch_normalization_138 (BatchNormalization) | (None, 9, 9, 128) | 384 | ['conv2d_142[0][0]'] |
| activation_125 (Activation) | (None, 9, 9, 128) | 0 | ['batch_normalization_133[0][0]'] |
| activation_130 (Activation) | (None, 9, 9, 128) | 0 | ['batch_normalization_138[0][0]'] |
| conv2d_138 (Conv2D) | (None, 9, 9, 128) | 114688 | ['activation_125[0][0]'] |
| conv2d_143 (Conv2D) | (None, 9, 9, 128) | 114688 | ['activation_130[0][0]'] |
| batch_normalization_134 (BatchNormalization) | (None, 9, 9, 128) | 384 | ['conv2d_138[0][0]'] |
| batch_normalization_139 (BatchNormalization) | (None, 9, 9, 128) | 384 | ['conv2d_143[0][0]'] |
| activation_126 (Activation) | (None, 9, 9, 128) | 0 | ['batch_normalization_134[0][0]'] |

| | | | |
|--|-------------------|--------|-----------------------------------|
| activation_131 (Activation) | (None, 9, 9, 128) | 0 | ['batch_normalization_139[0][0]'] |
| average_pooling2d_12 (AveragePooling2D) | (None, 9, 9, 768) | 0 | ['mixed3[0][0]'] |
| conv2d_136 (Conv2D) | (None, 9, 9, 192) | 147456 | ['mixed3[0][0]'] |
| conv2d_139 (Conv2D) | (None, 9, 9, 192) | 172032 | ['activation_126[0][0]'] |
| conv2d_144 (Conv2D) | (None, 9, 9, 192) | 172032 | ['activation_131[0][0]'] |
| conv2d_145 (Conv2D) | (None, 9, 9, 192) | 147456 | ['average_pooling2d_12[0][0]'] |
| batch_normalization_132 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_136[0][0]'] |
| batch_normalization_135 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_139[0][0]'] |
| batch_normalization_140 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_144[0][0]'] |
| batch_normalization_141 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_145[0][0]'] |
| activation_124 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_132[0][0]'] |
| activation_127 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_135[0][0]'] |
| activation_132 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_140[0][0]'] |
| activation_133 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_141[0][0]'] |
| mixed4 (Concatenate) | (None, 9, 9, 768) | 0 | ['activation_124[0][0]', |

| | | | |
|--|-------------------|--------|---|
| | | | 'activation_127[0][0] ', 'activation_132[0][0] ', 'activation_133[0][0] ' |
| conv2d_150 (Conv2D) | (None, 9, 9, 160) | 122880 | ['mixed4[0][0]'] |
| batch_normalization_146 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_150[0][0]'] |
| activation_138 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_146[0][0]'] |
| conv2d_151 (Conv2D) | (None, 9, 9, 160) | 179200 | ['activation_138[0][0]'] |
| batch_normalization_147 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_151[0][0]'] |
| activation_139 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_147[0][0]'] |
| conv2d_147 (Conv2D) | (None, 9, 9, 160) | 122880 | ['mixed4[0][0]'] |
| conv2d_152 (Conv2D) | (None, 9, 9, 160) | 179200 | ['activation_139[0][0]'] |
| batch_normalization_143 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_147[0][0]'] |
| batch_normalization_148 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_152[0][0]'] |
| activation_135 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_143[0][0]'] |
| activation_140 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_148[0][0]'] |
| conv2d_148 (Conv2D) | (None, 9, 9, 160) | 179200 | ['activation_135[0][0]'] |
| conv2d_153 (Conv2D) | (None, 9, 9, 160) | 179200 | ['activation_140[0][0]'] |

| | | | |
|--|-------------------|--------|-----------------------------------|
| batch_normalization_144 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_148[0][0]'] |
| batch_normalization_149 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_153[0][0]'] |
| activation_136 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_144[0][0]'] |
| activation_141 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_149[0][0]'] |
| average_pooling2d_13 (AveragePooling2D) | (None, 9, 9, 768) | 0 | ['mixed4[0][0]'] |
| conv2d_146 (Conv2D) | (None, 9, 9, 192) | 147456 | ['mixed4[0][0]'] |
| conv2d_149 (Conv2D) | (None, 9, 9, 192) | 215040 | ['activation_136[0][0]'] |
| conv2d_154 (Conv2D) | (None, 9, 9, 192) | 215040 | ['activation_141[0][0]'] |
| conv2d_155 (Conv2D) | (None, 9, 9, 192) | 147456 | ['average_pooling2d_13[0][0]'] |
| batch_normalization_142 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_146[0][0]'] |
| batch_normalization_145 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_149[0][0]'] |
| batch_normalization_150 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_154[0][0]'] |
| batch_normalization_151 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_155[0][0]'] |
| activation_134 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_142[0][0]'] |
| activation_137 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_145[0][0]'] |

| | | | |
|--|-------------------|--------|---|
| activation_142 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_150[0][0]'] |
| activation_143 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_151[0][0]'] |
| mixed5 (Concatenate) | (None, 9, 9, 768) | 0 | ['activation_134[0][0]', 'activation_137[0][0]', 'activation_142[0][0]', 'activation_143[0][0]'] |
| conv2d_160 (Conv2D) | (None, 9, 9, 160) | 122880 | ['mixed5[0][0]'] |
| batch_normalization_156 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_160[0][0]'] |
| activation_148 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_156[0][0]'] |
| conv2d_161 (Conv2D) | (None, 9, 9, 160) | 179200 | ['activation_148[0][0]'] |
| batch_normalization_157 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_161[0][0]'] |
| activation_149 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_157[0][0]'] |
| conv2d_157 (Conv2D) | (None, 9, 9, 160) | 122880 | ['mixed5[0][0]'] |
| conv2d_162 (Conv2D) | (None, 9, 9, 160) | 179200 | ['activation_149[0][0]'] |
| batch_normalization_153 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_157[0][0]'] |
| batch_normalization_158 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_162[0][0]'] |
| activation_145 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_153[0][0]'] |

| | | | |
|--|-------------------|--------|-----------------------------------|
| activation_150 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_158[0][0]'] |
| conv2d_158 (Conv2D) | (None, 9, 9, 160) | 179200 | ['activation_145[0][0]'] |
| conv2d_163 (Conv2D) | (None, 9, 9, 160) | 179200 | ['activation_150[0][0]'] |
| batch_normalization_154 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_158[0][0]'] |
| batch_normalization_159 (BatchNormalization) | (None, 9, 9, 160) | 480 | ['conv2d_163[0][0]'] |
| activation_146 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_154[0][0]'] |
| activation_151 (Activation) | (None, 9, 9, 160) | 0 | ['batch_normalization_159[0][0]'] |
| average_pooling2d_14 (AveragePooling2D) | (None, 9, 9, 768) | 0 | ['mixed5[0][0]'] |
| conv2d_156 (Conv2D) | (None, 9, 9, 192) | 147456 | ['mixed5[0][0]'] |
| conv2d_159 (Conv2D) | (None, 9, 9, 192) | 215040 | ['activation_146[0][0]'] |
| conv2d_164 (Conv2D) | (None, 9, 9, 192) | 215040 | ['activation_151[0][0]'] |
| conv2d_165 (Conv2D) | (None, 9, 9, 192) | 147456 | ['average_pooling2d_14[0][0]'] |
| batch_normalization_152 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_156[0][0]'] |
| batch_normalization_155 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_159[0][0]'] |
| batch_normalization_160 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_164[0][0]'] |

| | | | |
|--|-------------------|--------|---|
| batch_normalization_161 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_165[0][0]'] |
| activation_144 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_152[0][0]'] |
| activation_147 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_155[0][0]'] |
| activation_152 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_160[0][0]'] |
| activation_153 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_161[0][0]'] |
| mixed6 (Concatenate) | (None, 9, 9, 768) | 0 | ['activation_144[0][0]', 'activation_147[0][0]', 'activation_152[0][0]', 'activation_153[0][0]'] |
| conv2d_170 (Conv2D) | (None, 9, 9, 192) | 147456 | ['mixed6[0][0]'] |
| batch_normalization_166 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_170[0][0]'] |
| activation_158 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_166[0][0]'] |
| conv2d_171 (Conv2D) | (None, 9, 9, 192) | 258048 | ['activation_158[0][0]'] |
| batch_normalization_167 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_171[0][0]'] |
| activation_159 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_167[0][0]'] |
| conv2d_167 (Conv2D) | (None, 9, 9, 192) | 147456 | ['mixed6[0][0]'] |
| conv2d_172 (Conv2D) | (None, 9, 9, 192) | 258048 | ['activation_159[0][0]'] |

| | | | |
|--|-------------------|--------|-----------------------------------|
| batch_normalization_163 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_167[0][0]'] |
| batch_normalization_168 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_172[0][0]'] |
| activation_155 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_163[0][0]'] |
| activation_160 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_168[0][0]'] |
| conv2d_168 (Conv2D) | (None, 9, 9, 192) | 258048 | ['activation_155[0][0]'] |
| conv2d_173 (Conv2D) | (None, 9, 9, 192) | 258048 | ['activation_160[0][0]'] |
| batch_normalization_164 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_168[0][0]'] |
| batch_normalization_169 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_173[0][0]'] |
| activation_156 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_164[0][0]'] |
| activation_161 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_169[0][0]'] |
| average_pooling2d_15 (AveragePooling2D) | (None, 9, 9, 768) | 0 | ['mixed6[0][0]'] |
| conv2d_166 (Conv2D) | (None, 9, 9, 192) | 147456 | ['mixed6[0][0]'] |
| conv2d_169 (Conv2D) | (None, 9, 9, 192) | 258048 | ['activation_156[0][0]'] |
| conv2d_174 (Conv2D) | (None, 9, 9, 192) | 258048 | ['activation_161[0][0]'] |
| conv2d_175 (Conv2D) | (None, 9, 9, 192) | 147456 | ['average_pooling2d_15[0][0]'] |
| batch_normalization_162 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_166[0][0]'] |

| | | | |
|---|-------------------|--------|---|
| atchNormalization) | | | |
| batch_normalization_165 (Batch Normalization) | (None, 9, 9, 192) | 576 | ['conv2d_169[0][0]'] |
| batch_normalization_170 (Batch Normalization) | (None, 9, 9, 192) | 576 | ['conv2d_174[0][0]'] |
| batch_normalization_171 (Batch Normalization) | (None, 9, 9, 192) | 576 | ['conv2d_175[0][0]'] |
| activation_154 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_162[0][0]'] |
| activation_157 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_165[0][0]'] |
| activation_162 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_170[0][0]'] |
| activation_163 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_171[0][0]'] |
| mixed7 (Concatenate) | (None, 9, 9, 768) | 0 | ['activation_154[0][0]', 'activation_157[0][0]', 'activation_162[0][0]', 'activation_163[0][0]'] |
| conv2d_178 (Conv2D) | (None, 9, 9, 192) | 147456 | ['mixed7[0][0]'] |
| batch_normalization_174 (Batch Normalization) | (None, 9, 9, 192) | 576 | ['conv2d_178[0][0]'] |
| activation_166 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_174[0][0]'] |
| conv2d_179 (Conv2D) | (None, 9, 9, 192) | 258048 | ['activation_166[0][0]'] |
| batch_normalization_175 (Batch Normalization) | (None, 9, 9, 192) | 576 | ['conv2d_179[0][0]'] |

| | | | |
|--|--------------------|--------|-----------------------------------|
| activation_167 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_175[0][0]'] |
| conv2d_176 (Conv2D) | (None, 9, 9, 192) | 147456 | ['mixed7[0][0]'] |
| conv2d_180 (Conv2D) | (None, 9, 9, 192) | 258048 | ['activation_167[0][0]'] |
| batch_normalization_172 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_176[0][0]'] |
| batch_normalization_176 (BatchNormalization) | (None, 9, 9, 192) | 576 | ['conv2d_180[0][0]'] |
| activation_164 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_172[0][0]'] |
| activation_168 (Activation) | (None, 9, 9, 192) | 0 | ['batch_normalization_176[0][0]'] |
| conv2d_177 (Conv2D) | (None, 4, 4, 320) | 552960 | ['activation_164[0][0]'] |
| conv2d_181 (Conv2D) | (None, 4, 4, 192) | 331776 | ['activation_168[0][0]'] |
| batch_normalization_173 (BatchNormalization) | (None, 4, 4, 320) | 960 | ['conv2d_177[0][0]'] |
| batch_normalization_177 (BatchNormalization) | (None, 4, 4, 192) | 576 | ['conv2d_181[0][0]'] |
| activation_165 (Activation) | (None, 4, 4, 320) | 0 | ['batch_normalization_173[0][0]'] |
| activation_169 (Activation) | (None, 4, 4, 192) | 0 | ['batch_normalization_177[0][0]'] |
| max_pooling2d_17 (MaxPooling2D) | (None, 4, 4, 768) | 0 | ['mixed7[0][0]'] |
| mixed8 (Concatenate) | (None, 4, 4, 1280) | 0 | ['activation_165[0][0]', |

| | | | |
|--|--------------------|---------|---|
| | | | 'activation_169[0][0]', 'max_pooling2d_17[0][0]' |
| conv2d_186 (Conv2D) | (None, 4, 4, 448) | 573440 | ['mixed8[0][0]'] |
| batch_normalization_182 (BatchNormalization) | (None, 4, 4, 448) | 1344 | ['conv2d_186[0][0]'] |
| activation_174 (Activation) | (None, 4, 4, 448) | 0 | ['batch_normalization_182[0][0]'] |
| conv2d_183 (Conv2D) | (None, 4, 4, 384) | 491520 | ['mixed8[0][0]'] |
| conv2d_187 (Conv2D) | (None, 4, 4, 384) | 1548288 | ['activation_174[0][0]'] |
| batch_normalization_179 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_183[0][0]'] |
| batch_normalization_183 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_187[0][0]'] |
| activation_171 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_179[0][0]'] |
| activation_175 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_183[0][0]'] |
| conv2d_184 (Conv2D) | (None, 4, 4, 384) | 442368 | ['activation_171[0][0]'] |
| conv2d_185 (Conv2D) | (None, 4, 4, 384) | 442368 | ['activation_171[0][0]'] |
| conv2d_188 (Conv2D) | (None, 4, 4, 384) | 442368 | ['activation_175[0][0]'] |
| conv2d_189 (Conv2D) | (None, 4, 4, 384) | 442368 | ['activation_175[0][0]'] |
| average_pooling2d_16 (AveragePooling2D) | (None, 4, 4, 1280) | 0 | ['mixed8[0][0]'] |
| conv2d_182 (Conv2D) | (None, 4, 4, 320) | 409600 | ['mixed8[0][0]'] |

| | | | |
|--|-------------------|--------|---|
| batch_normalization_180 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_184[0][0]'] |
| batch_normalization_181 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_185[0][0]'] |
| batch_normalization_184 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_188[0][0]'] |
| batch_normalization_185 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_189[0][0]'] |
| conv2d_190 (Conv2D) | (None, 4, 4, 192) | 245760 | ['average_pooling2d_16[0][0]'] |
| batch_normalization_178 (BatchNormalization) | (None, 4, 4, 320) | 960 | ['conv2d_182[0][0]'] |
| activation_172 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_180[0][0]'] |
| activation_173 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_181[0][0]'] |
| activation_176 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_184[0][0]'] |
| activation_177 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_185[0][0]'] |
| batch_normalization_186 (BatchNormalization) | (None, 4, 4, 192) | 576 | ['conv2d_190[0][0]'] |
| activation_170 (Activation) | (None, 4, 4, 320) | 0 | ['batch_normalization_178[0][0]'] |
| mixed9_0 (Concatenate) | (None, 4, 4, 768) | 0 | ['activation_172[0][0]', 'activation_173[0][0]'] |
| concatenate_2 (Concatenate) | (None, 4, 4, 768) | 0 | ['activation_176[0][0]', 'activation_177[0][0]'] |

| | | | |
|--|--------------------|---------|--|
| activation_178 (Activation) | (None, 4, 4, 192) | 0 | ['batch_normalization_186[0][0]'] |
| mixed9 (Concatenate) | (None, 4, 4, 2048) | 0 | ['activation_170[0][0]', 'mixed9_0[0][0]', 'concatenate_2[0][0]', 'activation_178[0][0]'] |
| conv2d_195 (Conv2D) | (None, 4, 4, 448) | 917504 | ['mixed9[0][0]'] |
| batch_normalization_191 (BatchNormalization) | (None, 4, 4, 448) | 1344 | ['conv2d_195[0][0]'] |
| activation_183 (Activation) | (None, 4, 4, 448) | 0 | ['batch_normalization_191[0][0]'] |
| conv2d_192 (Conv2D) | (None, 4, 4, 384) | 786432 | ['mixed9[0][0]'] |
| conv2d_196 (Conv2D) | (None, 4, 4, 384) | 1548288 | ['activation_183[0][0]'] |
| batch_normalization_188 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_192[0][0]'] |
| batch_normalization_192 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_196[0][0]'] |
| activation_180 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_188[0][0]'] |
| activation_184 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_192[0][0]'] |
| conv2d_193 (Conv2D) | (None, 4, 4, 384) | 442368 | ['activation_180[0][0]'] |
| conv2d_194 (Conv2D) | (None, 4, 4, 384) | 442368 | ['activation_180[0][0]'] |
| conv2d_197 (Conv2D) | (None, 4, 4, 384) | 442368 | ['activation_184[0][0]'] |
| conv2d_198 (Conv2D) | (None, 4, 4, 384) | 442368 | ['activation_184[0][0]'] |

| | | | |
|--|--------------------|--------|-----------------------------------|
| average_pooling2d_17 (AveragePooling2D) | (None, 4, 4, 2048) | 0 | ['mixed9[0][0]'] |
| conv2d_191 (Conv2D) | (None, 4, 4, 320) | 655360 | ['mixed9[0][0]'] |
| batch_normalization_189 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_193[0][0]'] |
| batch_normalization_190 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_194[0][0]'] |
| batch_normalization_193 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_197[0][0]'] |
| batch_normalization_194 (BatchNormalization) | (None, 4, 4, 384) | 1152 | ['conv2d_198[0][0]'] |
| conv2d_199 (Conv2D) | (None, 4, 4, 192) | 393216 | ['average_pooling2d_17[0][0]'] |
| batch_normalization_187 (BatchNormalization) | (None, 4, 4, 320) | 960 | ['conv2d_191[0][0]'] |
| activation_181 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_189[0][0]'] |
| activation_182 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_190[0][0]'] |
| activation_185 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_193[0][0]'] |
| activation_186 (Activation) | (None, 4, 4, 384) | 0 | ['batch_normalization_194[0][0]'] |
| batch_normalization_195 (BatchNormalization) | (None, 4, 4, 192) | 576 | ['conv2d_199[0][0]'] |
| activation_179 (Activation) | (None, 4, 4, 320) | 0 | ['batch_normalization_187[0][0]'] |

| | | | |
|--|--------------------|------|--|
| mixed9_1 (Concatenate) | (None, 4, 4, 768) | 0 | ['activation_181[0][0]', 'activation_182[0][0]'] |
| concatenate_3 (Concatenate) | (None, 4, 4, 768) | 0 | ['activation_185[0][0]', 'activation_186[0][0]'] |
| activation_187 (Activation) | (None, 4, 4, 192) | 0 | ['batch_normalization_195[0][0] '] |
| mixed10 (Concatenate) | (None, 4, 4, 2048) | 0 | ['activation_179[0][0]', 'mixed9_1[0][0]', 'concatenate_3[0][0]', 'activation_187[0][0]'] |
| global_average_pooling2d_3 (GlobalAveragePooling2D) | (None, 2048) | 0 | ['mixed10[0][0]'] |
| dropout_7 (Dropout) | (None, 2048) | 0 | ['global_average_pooling2d_3[0] [0]'] |
| dense_9 (Dense) | (None, 1) | 2049 | ['dropout_7[0][0]'] |

```
=====
Total params: 21804833 (83.18 MB)
Trainable params: 21770401 (83.05 MB)
Non-trainable params: 34432 (134.50 KB)
=====
```

```
In [ ]: mc_V3 = ModelCheckpoint(filepath="v3.hdf5", monitor= 'val_accuracy', verbose= 1, save_best_only= True, mc
call_back_v3 = [ mc_V3,es ];
```

```
In [ ]: history_v3 = model_inceptionv3.fit(train_ds,
epochs= 150,
validation_data= validation_ds,
validation_steps= 6,
callbacks = call_back_v3 )
```

Epoch 1/150
56/56 [=====] - ETA: 0s - loss: 4.3430e-05 - accuracy: 1.0000
Epoch 1: val_accuracy did not improve from 0.85938
56/56 [=====] - 22s 399ms/step - loss: 4.3430e-05 - accuracy: 1.0000 - val_loss: 1.0422 - val_accuracy: 0.8385
Epoch 2/150
56/56 [=====] - ETA: 0s - loss: 3.7290e-05 - accuracy: 1.0000
Epoch 2: val_accuracy did not improve from 0.85938
56/56 [=====] - 22s 398ms/step - loss: 3.7290e-05 - accuracy: 1.0000 - val_loss: 0.9973 - val_accuracy: 0.8411
Epoch 3/150
56/56 [=====] - ETA: 0s - loss: 3.2665e-05 - accuracy: 1.0000
Epoch 3: val_accuracy did not improve from 0.85938
56/56 [=====] - 23s 417ms/step - loss: 3.2665e-05 - accuracy: 1.0000 - val_loss: 1.1300 - val_accuracy: 0.8490
Epoch 4/150
56/56 [=====] - ETA: 0s - loss: 2.8972e-05 - accuracy: 1.0000
Epoch 4: val_accuracy did not improve from 0.85938
56/56 [=====] - 22s 395ms/step - loss: 2.8972e-05 - accuracy: 1.0000 - val_loss: 1.0292 - val_accuracy: 0.8411
Epoch 5/150
56/56 [=====] - ETA: 0s - loss: 2.5940e-05 - accuracy: 1.0000
Epoch 5: val_accuracy did not improve from 0.85938
56/56 [=====] - 22s 389ms/step - loss: 2.5940e-05 - accuracy: 1.0000 - val_loss: 1.2517 - val_accuracy: 0.8359
Epoch 6/150
56/56 [=====] - ETA: 0s - loss: 2.3396e-05 - accuracy: 1.0000
Epoch 6: val_accuracy did not improve from 0.85938
56/56 [=====] - 22s 387ms/step - loss: 2.3396e-05 - accuracy: 1.0000 - val_loss: 1.1802 - val_accuracy: 0.8229
Epoch 7/150
56/56 [=====] - ETA: 0s - loss: 2.1232e-05 - accuracy: 1.0000
Epoch 7: val_accuracy did not improve from 0.85938
56/56 [=====] - 22s 387ms/step - loss: 2.1232e-05 - accuracy: 1.0000 - val_loss: 1.0530 - val_accuracy: 0.8464
Epoch 8/150
56/56 [=====] - ETA: 0s - loss: 1.9363e-05 - accuracy: 1.0000
Epoch 8: val_accuracy improved from 0.85938 to 0.86198, saving model to v3.hdf5

```
/Users/rajitkuthiala/miniforge3/envs/mlp/lib/python3.8/site-packages/keras/src/engine/training.py:3000:
UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considere
d legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')`.
```

```
saving_api.save_model(
56/56 [=====] - 22s 401ms/step - loss: 1.9363e-05 - accuracy: 1.0000 - val_loss: 1.0037 - val_accuracy: 0.8620
Epoch 9/150
Epoch 9/150
56/56 [=====] - ETA: 0s - loss: 1.7736e-05 - accuracy: 1.0000
Epoch 9: val_accuracy did not improve from 0.86198
56/56 [=====] - 22s 387ms/step - loss: 1.7736e-05 - accuracy: 1.0000 - val_loss: 1.2317 - val_accuracy: 0.8464
Epoch 10/150
56/56 [=====] - ETA: 0s - loss: 1.6305e-05 - accuracy: 1.0000
Epoch 10: val_accuracy did not improve from 0.86198
56/56 [=====] - 22s 386ms/step - loss: 1.6305e-05 - accuracy: 1.0000 - val_loss: 1.1223 - val_accuracy: 0.8411
Epoch 11/150
56/56 [=====] - ETA: 0s - loss: 1.5039e-05 - accuracy: 1.0000
Epoch 11: val_accuracy did not improve from 0.86198
56/56 [=====] - 22s 386ms/step - loss: 1.5039e-05 - accuracy: 1.0000 - val_loss: 0.9829 - val_accuracy: 0.8542
Epoch 12/150
56/56 [=====] - ETA: 0s - loss: 1.3911e-05 - accuracy: 1.0000
Epoch 12: val_accuracy did not improve from 0.86198
56/56 [=====] - 22s 385ms/step - loss: 1.3911e-05 - accuracy: 1.0000 - val_loss: 1.3532 - val_accuracy: 0.8177
Epoch 13/150
56/56 [=====] - ETA: 0s - loss: 1.2901e-05 - accuracy: 1.0000
Epoch 13: val_accuracy did not improve from 0.86198
56/56 [=====] - 22s 385ms/step - loss: 1.2901e-05 - accuracy: 1.0000 - val_loss: 1.2157 - val_accuracy: 0.8411
Epoch 14/150
56/56 [=====] - ETA: 0s - loss: 1.1992e-05 - accuracy: 1.0000
Epoch 14: val_accuracy improved from 0.86198 to 0.86458, saving model to v3.hdf5
56/56 [=====] - 22s 397ms/step - loss: 1.1992e-05 - accuracy: 1.0000 - val_loss: 1.0483 - val_accuracy: 0.8646
Epoch 15/150
56/56 [=====] - ETA: 0s - loss: 1.1170e-05 - accuracy: 1.0000
Epoch 15: val_accuracy did not improve from 0.86458
```

56/56 [=====] - 22s 386ms/step - loss: 1.1170e-05 - accuracy: 1.0000 - val_loss: 1.0633 - val_accuracy: 0.8620
Epoch 16/150
56/56 [=====] - ETA: 0s - loss: 1.0423e-05 - accuracy: 1.0000
Epoch 16: val_accuracy did not improve from 0.86458
56/56 [=====] - 22s 389ms/step - loss: 1.0423e-05 - accuracy: 1.0000 - val_loss: 1.2179 - val_accuracy: 0.8464
Epoch 17/150
56/56 [=====] - ETA: 0s - loss: 9.7458e-06 - accuracy: 1.0000
Epoch 17: val_accuracy did not improve from 0.86458
56/56 [=====] - 22s 388ms/step - loss: 9.7458e-06 - accuracy: 1.0000 - val_loss: 1.0099 - val_accuracy: 0.8646
Epoch 18/150
56/56 [=====] - ETA: 0s - loss: 9.1251e-06 - accuracy: 1.0000
Epoch 18: val_accuracy did not improve from 0.86458
56/56 [=====] - 21s 384ms/step - loss: 9.1251e-06 - accuracy: 1.0000 - val_loss: 1.2528 - val_accuracy: 0.8438
Epoch 19/150
56/56 [=====] - ETA: 0s - loss: 8.5537e-06 - accuracy: 1.0000
Epoch 19: val_accuracy did not improve from 0.86458
56/56 [=====] - 21s 384ms/step - loss: 8.5537e-06 - accuracy: 1.0000 - val_loss: 1.1193 - val_accuracy: 0.8490
Epoch 20/150
56/56 [=====] - ETA: 0s - loss: 8.0315e-06 - accuracy: 1.0000
Epoch 20: val_accuracy did not improve from 0.86458
56/56 [=====] - 22s 385ms/step - loss: 8.0315e-06 - accuracy: 1.0000 - val_loss: 1.1467 - val_accuracy: 0.8568
Epoch 21/150
56/56 [=====] - ETA: 0s - loss: 7.5496e-06 - accuracy: 1.0000
Epoch 21: val_accuracy did not improve from 0.86458
56/56 [=====] - 22s 387ms/step - loss: 7.5496e-06 - accuracy: 1.0000 - val_loss: 1.1537 - val_accuracy: 0.8568
Epoch 22/150
56/56 [=====] - ETA: 0s - loss: 7.1039e-06 - accuracy: 1.0000
Epoch 22: val_accuracy did not improve from 0.86458
56/56 [=====] - 22s 389ms/step - loss: 7.1039e-06 - accuracy: 1.0000 - val_loss: 1.1491 - val_accuracy: 0.8464
Epoch 23/150
56/56 [=====] - ETA: 0s - loss: 6.6924e-06 - accuracy: 1.0000
Epoch 23: val_accuracy did not improve from 0.86458

```
56/56 [=====] - 22s 389ms/step - loss: 6.6924e-06 - accuracy: 1.0000 - val_loss: 1.2561 - val_accuracy: 0.8307
Epoch 24/150
56/56 [=====] - ETA: 0s - loss: 6.3119e-06 - accuracy: 1.0000
Epoch 24: val_accuracy did not improve from 0.86458
56/56 [=====] - 22s 385ms/step - loss: 6.3119e-06 - accuracy: 1.0000 - val_loss: 1.3240 - val_accuracy: 0.8307
Epoch 24: early stopping
```

```
In [ ]: accuracy_inceptionv3 = model_inceptionv3.evaluate_generator(generator= test_ds)[1]
print(f"The accuracy of your model is = {accuracy_inceptionv3*100} %")
```

```
/var/folders/gf/dq4lmmx17b71h0cbpbkwfj9c0000gn/T/ipykernel_13194/1764400038.py:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.
```

```
    accuracy_inceptionv3 = model_inceptionv3.evaluate_generator(generator= test_ds)[1]
2023-10-30 17:02:41.651881: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.
The accuracy of your model is = 84.13850665092468 %
```

```
In [ ]: import matplotlib.pyplot as plt
acc = history_v3.history['accuracy']
val_acc = history_v3.history['val_accuracy']
loss = history_v3.history['loss']
val_loss = history_v3.history['val_loss']

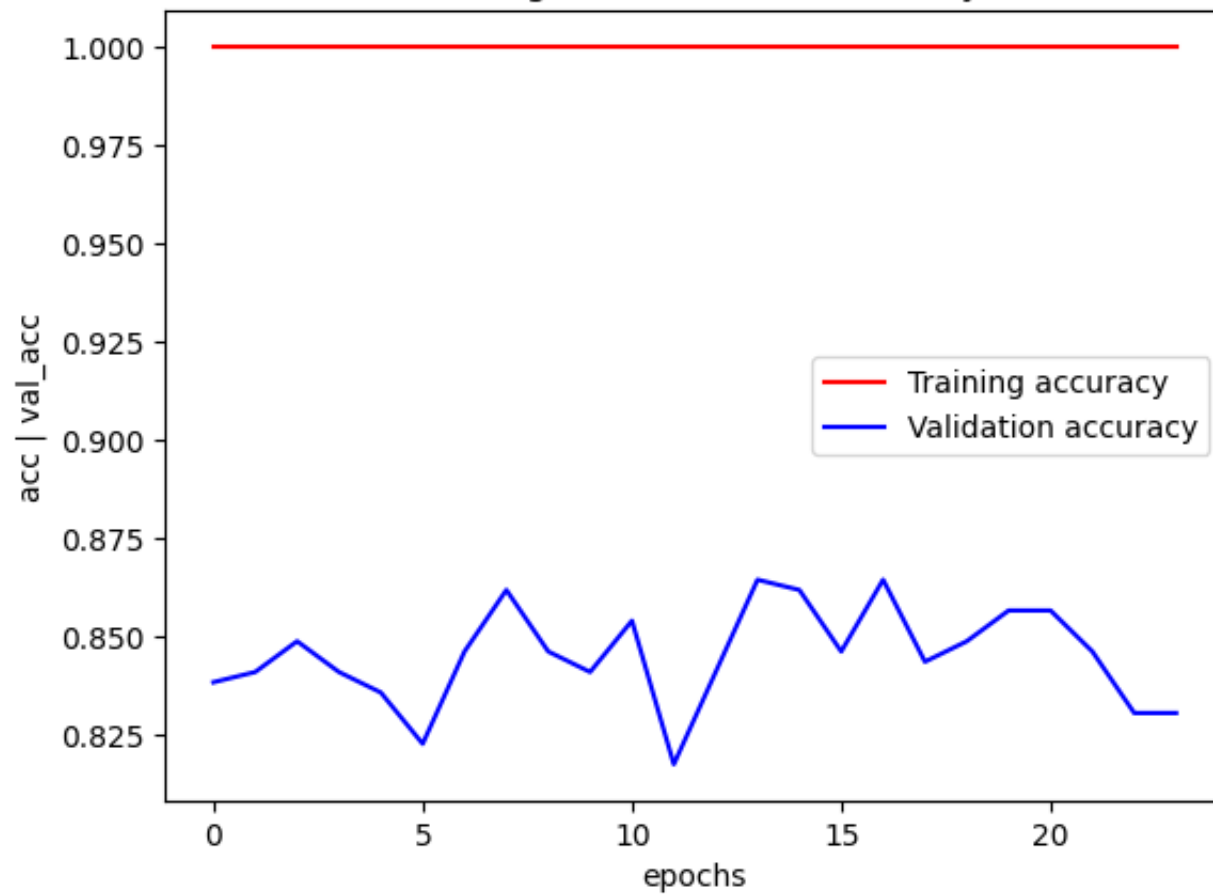
epochs = range(len(acc))

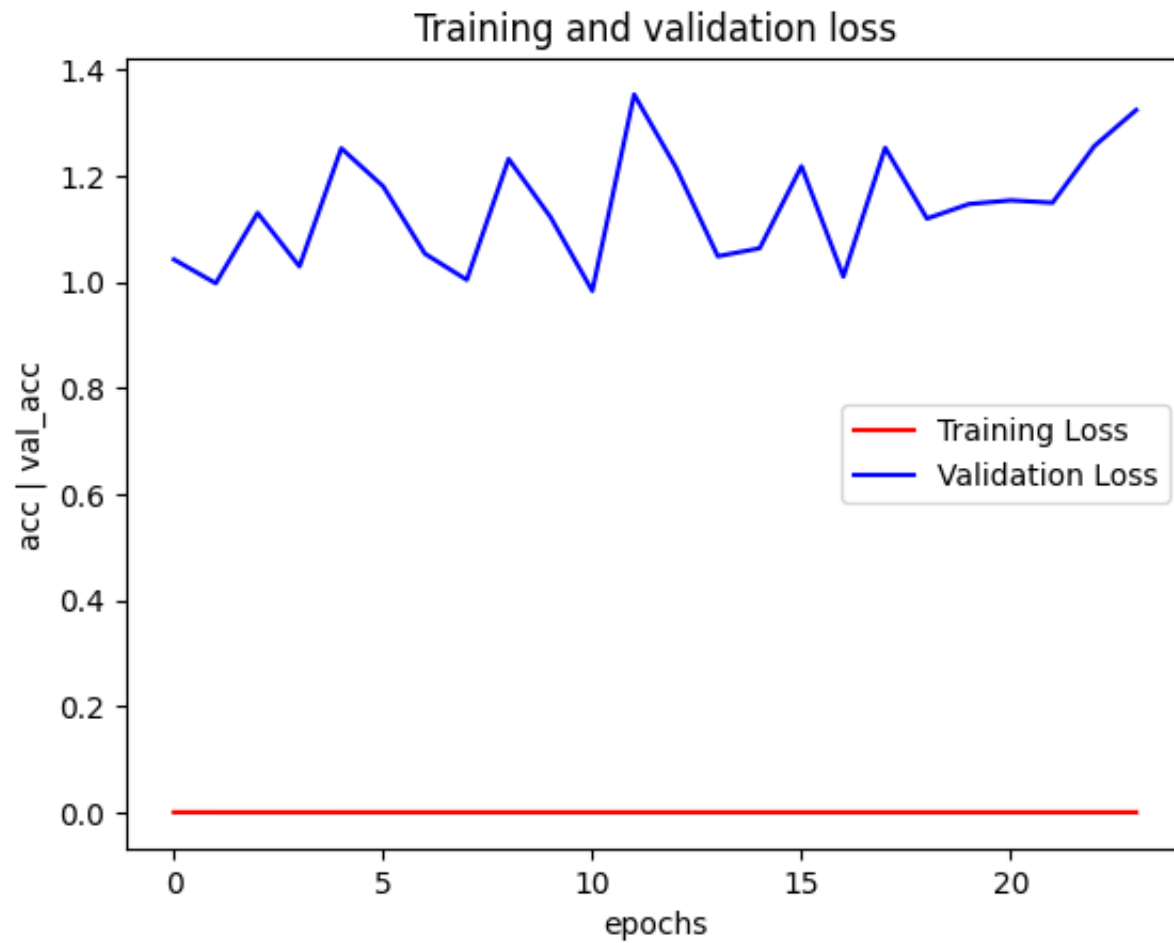
plt.plot(epochs, acc, 'r', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.xlabel("epochs")
plt.ylabel("acc | val_acc")
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()

plt.plot(epochs, loss, 'r', label='Training Loss')
plt.plot(epochs, val_loss, 'b', label='Validation Loss')
plt.xlabel("epochs")
plt.ylabel("acc | val_acc")
plt.title('Training and validation loss')
plt.legend()

plt.show()
```


Training and validation accuracy





Ensemble Learning Model

```
In [ ]: from tensorflow.keras.models import Model, load_model
        from tensorflow.keras.layers import Input, Average
```

```
# Load the individual models
model_1 = load_model('./cnn.hdf5')
model_2 = load_model('./squeezenet_net.hdf5')
model_3 = load_model('./v3.hdf5')

## Define the accuracies of the individual models
# accuracy_1 = accuracy
# accuracy_2 = accuracy_sqnet
# accuracy_3 = accuracy_inceptionv3

## Calculate the weights based on the accuracies
# weight_1 = accuracy_1 / (accuracy_1 + accuracy_2 + accuracy_3)
# weight_2 = accuracy_2 / (accuracy_1 + accuracy_2 + accuracy_3)
# weight_3 = accuracy_3 / (accuracy_1 + accuracy_2 + accuracy_3)

# Create the input layer for the ensemble model
model_input = Input(shape=(180,180,3))

# Get the outputs from each model and apply the weights
# output_1 = model_1(model_input) * weight_1
# output_2 = model_2(model_input) * weight_2
# output_3 = model_3(model_input) * weight_3

output_1 = model_1(model_input)
output_2 = model_2(model_input)
output_3 = model_3(model_input)

# Perform the weighted average
ensemble_output = Average()(output_1, output_2, output_3)

# Create the ensemble model
ensemble_model = Model(inputs=model_input, outputs=ensemble_output)
```

```
In [ ]: mc_ensemble = ModelCheckpoint(filepath="ensemble.hdf5", monitor= 'val_accuracy', verbose= 1, save_best_only=True)
        call_back_ensemble = [ mc_ensemble,es ];
```

```
In [ ]: ensemble_model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

```
In [ ]: history_ensemble = ensemble_model.fit(train_ds,
      epochs= 150,
      validation_data= validation_ds,
      validation_steps= 6,
      callbacks = call_back_v3 )
```

Epoch 1/150

/Users/rajitkuthiala/miniforge3/envs/mlp/lib/python3.8/site-packages/tensorflow/python/util/dispatch.py:1176: SyntaxWarning: In loss categorical_crossentropy, expected y_pred.shape to be (batch_size, num_classes) with num_classes > 1. Received: y_pred.shape=(None, 1). Consider using 'binary_crossentropy' if you only have 2 classes.

```
    return dispatch_target(*args, **kwargs)
```

2023-10-30 17:31:28.911143: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.

56/56 [=====] - ETA: 0s - loss: 0.7931 - accuracy: 0.9456

2023-10-30 17:34:12.490010: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.

Epoch 1: val_accuracy improved from 0.86458 to 0.86719, saving model to v3.hdf5

/Users/rajitkuthiala/miniforge3/envs/mlp/lib/python3.8/site-packages/keras/src/engine/training.py:3000: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')`.

```
    saving_api.save_model(
```

56/56 [=====] - 194s 2s/step - loss: 0.7931 - accuracy: 0.9456 - val_loss: 0.1751 - val_accuracy: 0.8672

Epoch 2/150

Epoch 2/150

56/56 [=====] - ETA: 0s - loss: 0.0610 - accuracy: 1.0000

Epoch 2: val_accuracy did not improve from 0.86719

56/56 [=====] - 54s 949ms/step - loss: 0.0610 - accuracy: 1.0000 - val_loss: 0.0089 - val_accuracy: 0.8672

Epoch 3/150

56/56 [=====] - ETA: 0s - loss: 0.0026 - accuracy: 1.0000

Epoch 3: val_accuracy did not improve from 0.86719

56/56 [=====] - 57s 1s/step - loss: 0.0026 - accuracy: 1.0000 - val_loss: 2.7296e-04 - val_accuracy: 0.8229

Epoch 4/150

56/56 [=====] - ETA: 0s - loss: 1.1474e-04 - accuracy: 1.0000

Epoch 4: val_accuracy improved from 0.86719 to 0.87240, saving model to v3.hdf5
56/56 [=====] - 5453s 99s/step - loss: 1.1474e-04 - accuracy: 1.0000 - val_loss: 6.7235e-05 - val_accuracy: 0.8724
Epoch 5/150
56/56 [=====] - ETA: 0s - loss: 7.5949e-05 - accuracy: 1.0000
Epoch 5: val_accuracy did not improve from 0.87240
56/56 [=====] - 4307s 78s/step - loss: 7.5949e-05 - accuracy: 1.0000 - val_loss: 7.7628e-05 - val_accuracy: 0.8542
Epoch 6/150
56/56 [=====] - ETA: 0s - loss: 8.6309e-05 - accuracy: 1.0000
Epoch 6: val_accuracy did not improve from 0.87240
56/56 [=====] - 64s 1s/step - loss: 8.6309e-05 - accuracy: 1.0000 - val_loss: 9.3027e-05 - val_accuracy: 0.8568
Epoch 7/150
56/56 [=====] - ETA: 0s - loss: 9.5338e-05 - accuracy: 1.0000
Epoch 7: val_accuracy improved from 0.87240 to 0.88542, saving model to v3.hdf5
56/56 [=====] - 59s 1s/step - loss: 9.5338e-05 - accuracy: 1.0000 - val_loss: 9.6478e-05 - val_accuracy: 0.8854
Epoch 8/150
56/56 [=====] - ETA: 0s - loss: 9.5297e-05 - accuracy: 1.0000
Epoch 8: val_accuracy did not improve from 0.88542
56/56 [=====] - 53s 933ms/step - loss: 9.5297e-05 - accuracy: 1.0000 - val_loss: 9.7014e-05 - val_accuracy: 0.8385
Epoch 9/150
56/56 [=====] - ETA: 0s - loss: 9.9968e-05 - accuracy: 1.0000
Epoch 9: val_accuracy did not improve from 0.88542
56/56 [=====] - 52s 937ms/step - loss: 9.9968e-05 - accuracy: 1.0000 - val_loss: 9.9845e-05 - val_accuracy: 0.8464
Epoch 10/150
56/56 [=====] - ETA: 0s - loss: 1.1053e-04 - accuracy: 1.0000
Epoch 10: val_accuracy did not improve from 0.88542
56/56 [=====] - 52s 929ms/step - loss: 1.1053e-04 - accuracy: 1.0000 - val_loss: 9.9843e-05 - val_accuracy: 0.8490
Epoch 11/150
56/56 [=====] - ETA: 0s - loss: 9.8510e-05 - accuracy: 1.0000
Epoch 11: val_accuracy did not improve from 0.88542
56/56 [=====] - 52s 922ms/step - loss: 9.8510e-05 - accuracy: 1.0000 - val_loss: 1.0534e-04 - val_accuracy: 0.8724
Epoch 12/150
56/56 [=====] - ETA: 0s - loss: 1.0691e-04 - accuracy: 1.0000

```

Epoch 12: val_accuracy did not improve from 0.88542
56/56 [=====] - 53s 937ms/step - loss: 1.0691e-04 - accuracy: 1.0000 - val_loss: 1.1454e-04 - val_accuracy: 0.8333
Epoch 13/150
56/56 [=====] - ETA: 0s - loss: 1.1070e-04 - accuracy: 1.0000
Epoch 13: val_accuracy did not improve from 0.88542
56/56 [=====] - 51s 906ms/step - loss: 1.1070e-04 - accuracy: 1.0000 - val_loss: 1.1011e-04 - val_accuracy: 0.8594
Epoch 14/150
56/56 [=====] - ETA: 0s - loss: 1.1006e-04 - accuracy: 1.0000
Epoch 14: val_accuracy did not improve from 0.88542
56/56 [=====] - 52s 927ms/step - loss: 1.1006e-04 - accuracy: 1.0000 - val_loss: 1.1447e-04 - val_accuracy: 0.8620
Epoch 15/150
56/56 [=====] - ETA: 0s - loss: 1.1758e-04 - accuracy: 1.0000
Epoch 15: val_accuracy did not improve from 0.88542
56/56 [=====] - 51s 908ms/step - loss: 1.1758e-04 - accuracy: 1.0000 - val_loss: 1.2312e-04 - val_accuracy: 0.8620
Epoch 16/150
56/56 [=====] - ETA: 0s - loss: 1.2777e-04 - accuracy: 1.0000
Epoch 16: val_accuracy did not improve from 0.88542
56/56 [=====] - 51s 917ms/step - loss: 1.2777e-04 - accuracy: 1.0000 - val_loss: 1.3240e-04 - val_accuracy: 0.8568
Epoch 17/150
56/56 [=====] - ETA: 0s - loss: 1.4076e-04 - accuracy: 1.0000
Epoch 17: val_accuracy did not improve from 0.88542
56/56 [=====] - 52s 918ms/step - loss: 1.4076e-04 - accuracy: 1.0000 - val_loss: 1.4767e-04 - val_accuracy: 0.8620
Epoch 17: early stopping

```

```

In [ ]: accuracy_ensemble = ensemble_model.evaluate_generator(generator= test_ds)[1]
print(f"The accuracy of your model is = {accuracy_ensemble*100} %")

```

```

/var/folders/gf/dq4lmmx17b71h0cbpbkxfj9c0000gn/T/ipykernel_13194/1787813163.py:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.

```

```

    accuracy_ensemble = ensemble_model.evaluate_generator(generator= test_ds)[1]
2023-10-30 20:29:52.803181: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.

```

```

The accuracy of your model is = 86.06534600257874 %

```

MODEL WISE ACCURACY

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

# Sample data for accuracy scores
models = ['CNN', 'Squeezent', 'InceptionV3', 'Ensemble']
accuracy_scores = [accuracy, accuracy_sqnet, accuracy_inceptionv3, accuracy_ensemble]

accuracy_scores_rounded = [round(score, 3) for score in accuracy_scores]

# Plot bar chart
fig, ax = plt.subplots()
ax.bar(models, accuracy_scores_rounded, color='green')
ax.set_xlabel('Models')
ax.set_ylabel('Accuracy Scores')
ax.set_title('Accuracy Scores for Different Models')

# Add values to bars
for i, v in enumerate(accuracy_scores_rounded):
    ax.text(i, v+0.01, str(v), color='black', ha='center')

plt.show()
```

Accuracy Scores for Different Models

