

## Imported Libraries

```
In [1]: import tensorflow as tf
import keras
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, Flatten, Dropout, BatchNormalization, LSTM, Input, Reshape
from tensorflow.keras.applications import InceptionResNetV2
from tensorflow.keras.losses import sparse_categorical_crossentropy
from tensorflow.keras.optimizers import RMSprop
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sys
import random
import cv2
import os
```

## Image Dataset Import

```
In [2]: labels = ['1_normal', '2_cataract', '3_glaucoma', '4_retina_disease']
img_size = 224
def get_data(data_dir):
    data = []

    for label in labels:
        path = os.path.join(data_dir, label)
        class_num = labels.index(label)
        for img in os.listdir(path):
            try:
                img_arr = cv2.imread(os.path.join(path, img))[...::-1] #convert BGR to RGB format
                crop_image= img_arr[0:1728,430:2190]
                resized_arr = cv2.resize(crop_image, (img_size, img_size)) # Reshaping images to preferred size
                data.append([resized_arr, class_num])
            except Exception as e:
                print(e)
    return np.array(data)
```

```
In [3]: #function call to get_data function that takes file path of the dataset.
data= get_data('dataset/dataset_all_equal_size_image/')
```

<ipython-input-2-b08f5e223f84>:17: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray

```
return np.array(data)
```

```
In [4]: data.shape
```

```
Out[4]: (600, 2)
```

```
In [5]: type(data)
```

```
Out[5]: numpy.ndarray
```

## Dividing Data Narray into Normal, Cataract, Glaucoma and Retina diseases.

```
In [6]: normal= data[0:300]
normal.shape
```

```
Out[6]: (300, 2)
```

```
In [7]: cataract=data[300:400]
cataract.shape
```

```
Out[7]: (100, 2)
```

```
In [8]: glaucoma= data[400:500]
glaucoma.shape
```

```
Out[8]: (100, 2)
```

```
In [9]: retina_disease= data[500:600]
retina_disease.shape
```

```
Out[9]: (100, 2)
```

```
In [10]: random.seed(15)
np.random.shuffle(normal)
np.random.shuffle(cataract)
np.random.shuffle(glaucoma)
np.random.shuffle(retina_disease)
```

## Performing Normalization and Resize operation

```
In [11]: def normalize(x_train,x_val,x_test):

    x_train = np.array(x_train) / 255
    x_train.reshape(-1, img_size, img_size, 1)

    x_test= np.array(x_test) / 255
    x_test.reshape(-1, img_size, img_size, 1)

    x_val= np.array(x_val) / 255
    x_val.reshape(-1, img_size, img_size, 1)

    return (x_train,x_val,x_test)
```

## Separating the Images and Labels into Respective Variables

```
In [12]: def image_label_split(train,validation,test):

    x_train = []
    y_train = []
    x_val = []
    y_val = []
    x_test = []
    y_test = []

    for feature, label in train:
        x_train.append(feature)
        y_train.append(label)

    for feature, label in validation:
        x_val.append(feature)
        y_val.append(label)

    for feature, label in test:
        x_test.append(feature)
        y_test.append(label)

    y_train = np.array(y_train)
    y_val = np.array(y_val)
    y_test= np.array(y_test)

    return (x_train,y_train,x_val,y_val,x_test,y_test)
```

## InceptionResNetV2-LSTM MODEL

```

In [13]: def model_build_compile(k):
    baseModel = InceptionResNetV2(weights="imagenet", include_top=False, input_tensor=Input(shape=(224, 224, 3)))
    for layer in baseModel.layers:
        layer.trainable = False

    x = baseModel.output

    # LSTM layer
    x = Reshape((25,1536))(x)
    x = ((LSTM(512, activation="relu", return_sequences=True, trainable=False)))(x)
    x = BatchNormalization()(x)
    #

    # FC layer
    x = Flatten(name="flatten")(x)

    # fc1 layer
    x = Dense(units=4096, activation='relu')(x)
    x = BatchNormalization()(x)
    #

    # fc2 layer
    x = Dense(units=4096, activation='relu')(x)
    x = BatchNormalization()(x)
    #

    # Output Layer
    output = Dense(units=4, activation='softmax')(x)

    model = Model(inputs=baseModel.input, outputs=output)
    opt = RMSprop(learning_rate=0.01, clipvalue=100)
    model.compile(loss='sparse_categorical_crossentropy', optimizer=opt, metrics=["accuracy"])
    k=k+1
    print("model building and compiling for fold",k)
    return model

```

## Model prediction for Test Images and Computation of Sensitivity and Specificity

```

In [14]: def test_pred(x_val,y_val,k):
    predictions = model.predict(x_val)
    predictions = np.argmax(predictions, axis = -1)

    print('-----Test accuracy for',k+1,'fold-----')
    #Confusion matrix, Accuracy, sensitivity and specificity
    cm1 = confusion_matrix(y_val,predictions)
    print('Confusion Matrix : \n', cm1)

    #####from confusion matrix calculate accuracy

    sensitivity_1_normal = (cm1[0,0])/(cm1[0,0]+cm1[0,1]+cm1[0,2]+cm1[0,3])
    #print('Sensitivity_1_normal      : ', sensitivity_1_normal )

    sensitivity_2_cataract = (cm1[1,1])/(cm1[1,0]+cm1[1,1]+cm1[1,2]+cm1[1,3])
    #print('Sensitivity_2_cataract    : ', sensitivity_2_cataract )

    sensitivity_3_glaucoma = (cm1[2,2])/(cm1[2,0]+cm1[2,1]+cm1[2,2]+cm1[2,3])
    #print('Sensitivity_3_glaucoma    : ', sensitivity_3_glaucoma )

    sensitivity_4_retina_disease = (cm1[3,3])/(cm1[3,0]+cm1[3,1]+cm1[3,2]+cm1[3,3])
    #print('Sensitivity_4_retina_disease : ', sensitivity_4_retina_disease )

    specificity_1_normal = (cm1[1,1]+cm1[1,2]+cm1[1,3]+cm1[2,1]+cm1[2,2]+cm1[2,3]+cm1[3,1]+cm1[3,2]+cm1[3,3])/(cm1[1,0]
+cm1[2,0]+cm1[3,0]+cm1[1,1]+cm1[1,2]+cm1[1,3]+cm1[2,1]+cm1[2,2]+cm1[2,3]+cm1[3,1]+cm1[3,2]+cm1[3,3])
    #print('Specificity : ', specificity_1_normal)

    specificity_2_cataract = (cm1[0,0]+cm1[0,2]+cm1[0,3]+cm1[2,0]+cm1[2,2]+cm1[2,3]+cm1[3,0]+cm1[3,2]+cm1[3,3])/(cm1[0
,1]+cm1[2,1]+cm1[3,1]+cm1[0,0]+cm1[0,2]+cm1[0,3]+cm1[2,0]+cm1[2,2]+cm1[2,3]+cm1[3,0]+cm1[3,2]+cm1[3,3])
    #print('Specificity : ', specificity_2_cataract)

    specificity_3_glaucoma = (cm1[0,0]+cm1[0,1]+cm1[0,3]+cm1[1,0]+cm1[1,1]+cm1[1,3]+cm1[3,0]+cm1[3,1]+cm1[3,3])/(cm1[0
,2]+cm1[1,2]+cm1[3,2]+cm1[0,0]+cm1[0,1]+cm1[0,3]+cm1[1,0]+cm1[1,1]+cm1[1,3]+cm1[3,0]+cm1[3,1]+cm1[3,3])
    #print('Specificity : ', specificity_3_glaucoma)

    specificity_4_retina_disease= (cm1[0,0]+cm1[0,1]+cm1[0,2]+cm1[1,0]+cm1[1,1]+cm1[1,2]+cm1[2,0]+cm1[2,1]+cm1[2,2])/(
cm1[0,3]+cm1[1,3]+cm1[2,3]+cm1[0,0]+cm1[0,1]+cm1[0,2]+cm1[1,0]+cm1[1,1]+cm1[1,2]+cm1[2,0]+cm1[2,1]+cm1[2,2])
    #print('Specificity : ', specificity_4_retina_disease)

    Sensitivity= (sensitivity_1_normal + sensitivity_2_cataract + sensitivity_3_glaucoma + sensitivity_4_retina_diseas
e)/4
    #print(Sensitivity)

    Specificity= (specificity_1_normal + specificity_2_cataract + specificity_3_glaucoma + specificity_4_retina_diseas
e)/4
    #print(Specificity)

    total1=sum(sum(cm1))
    test_accuracy=(cm1[0,0]+cm1[1,1]+cm1[2,2]+cm1[3,3])/total1

    print ('Accuracy      : ', test_accuracy)
    print ('Specificity : ', Specificity)
    print ('Sensitivity : ', Sensitivity)
    print('-----End of',k+1,'Fold-----')
    return test_accuracy,Specificity,Sensitivity,cm1

```

```

In [15]: CM= []
    test_accuracy=[]
    test_sensitivity=[]
    test_specificity=[]
    train_acc = []
    val_acc = []
    train_loss = []
    val_loss = []

```

## InceptionResNetV2-LSTM 5 Fold Cross Validation

```

In [16]: for k in range (5): # for loop to run 5 folds
        n_normal=30 # specifying the number of images for normal class in test phase,calulated as per 10% of total normal class images 300.
        n_rest=10 # specifying the number of images for disease classes in test phase,calulated as per 10% of total normal class images 100.

        # Adding the images in normal validation set by using k*n_normal to (k+1)*n_normal as index values for normal data set divided in cell 6.
        test_normal= normal[k*n_normal:(k+1)*n_normal]
        print('-----Start of',k+1,'Fold-----')
        print('test images for normal class from',k*n_normal,(k+1)*n_normal)

        # Adding the images in cataract validation set by using k*n_rest to (k+1)*n_rest as index values for cataract data set divided in cell 7.
        test_cataract= cataract[k*n_rest:(k+1)*n_rest]
        print('test images for cataract class from',k*n_rest,(k+1)*n_rest)

        # Adding the images in glaucoma validation set by using k*n_rest to (k+1)*n_rest as index values for glaucoma data set divided in cell 8.
        test_glaucoma= glaucoma[k*n_rest:(k+1)*n_rest]
        print('test images for glaucoma class from',k*n_rest,(k+1)*n_rest)

        # Adding the images in retina disease validation set by using k*n_rest to (k+1)*n_rest as index values for retina disease dataset divided in cell 9.
        test_retina= retina_disease[k*n_rest:(k+1)*n_rest]
        print('test images for retina disease class from',k*n_rest,(k+1)*n_rest)

        # Now for train and validation set of Normal images first adding 0 to k*n_normal images and then adding all the images from (k+1)*n_normal till last image.

        train_validation_normal= normal[:k*n_normal]
        train_validation_normal= np.append(train_validation_normal,normal[(k+1)*n_normal:],axis=0)
        print('train_validation images for normal class from 0 to',k*n_normal,'and',(k+1)*n_normal,'to 300')

        # Now for train and validation set of cataract images first adding 0 to k*n_rest images and then adding all the images from (k+1)*n_rest till last image.

        train_validation_cataract= cataract[:k*n_rest]
        train_validation_cataract= np.append(train_validation_cataract,cataract[(k+1)*n_rest:],axis=0)
        print('train_validation images for cataract class from 0 to',k*n_rest,'and',(k+1)*n_rest,'to 100')

        # Now for train and validation set of glaucoma images first adding 0 to k*n_rest images and then adding all the images from (k+1)*n_rest till last image.
        train_validation_glaucoma= glaucoma[:k*n_rest]
        train_validation_glaucoma= np.append(train_validation_glaucoma,glaucoma[(k+1)*n_rest:],axis=0)
        print('train_validation images for glaucoma class from 0',k*n_rest,'and',(k+1)*n_rest,'to 100')

        # Now for train and validation set of retina disease images first adding 0 to k*n_rest images and then adding all the images from (k+1)*n_rest till last image.
        train_validation_retina= retina_disease[:k*n_rest]
        train_validation_retina= np.append(train_validation_retina,retina_disease[(k+1)*n_rest:],axis=0)
        print('train_validation images for retina disease class from 0 to',k*n_rest,'and',(k+1)*n_rest,'to 100')

        # Splitting the train validation datasets in 80:20 ratio which would eventually give us 70% images in train and 20% images in validation and 10% in test.
        normal_train, normal_validation = train_test_split(train_validation_normal, test_size=0.20, random_state=14,shuffle=True)
        cataract_train, cataract_validation = train_test_split(train_validation_cataract, test_size=0.20, random_state=14,shuffle=True)
        glaucoma_train, glaucoma_validation = train_test_split(train_validation_glaucoma, test_size=0.20, random_state=14,shuffle=True)
        retina_disease_train, retina_disease_validation = train_test_split(train_validation_retina, test_size=0.20, random_state=14,shuffle=True)

        # Appending all train set images for all classes
        train= np.append(normal_train,cataract_train,axis=0)
        train= np.append(train,glaucoma_train,axis=0)
        train= np.append(train,retina_disease_train,axis=0)

        # Appending all validation set images for all classes
        validation= np.append(normal_validation,cataract_validation,axis=0)
        validation= np.append(validation,glaucoma_validation,axis=0)
        validation= np.append(validation,retina_disease_validation,axis=0)

        # Appending all test set images for all classes
        test= np.append(test_normal,test_cataract,axis=0)
        test= np.append(test,test_glaucoma,axis=0)
        test= np.append(test,test_retina,axis=0)

        # Shuffling the train validation and test set as they are added sequentially.
        random.seed(6)
        np.random.shuffle(train)
        np.random.shuffle(validation)
        np.random.shuffle(test)

        # Passing the train validation test as argument for image_label_split function that return features and labels separated.

```

```
x_train,y_train,x_val,y_val,x_test,y_test = image_label_split(train,validation,test)

# Passing the x_Train x_val and x_test as a argument for normalize function that returns the normalized and reshaped sets.
x_train,x_val,x_test = normalize(x_train,x_val,x_test)

# model building and model compile is done using a model_build_compile().

model = model_build_compile(k)
history = model.fit(x_train,y_train,epochs =50, validation_data = (x_val,y_val))

train_acc = np.append(train_acc,history.history['accuracy'])
val_acc = np.append(val_acc,history.history['val_accuracy'])

train_loss = np.append(train_loss,history.history['loss'])
val_loss = np.append(val_loss,history.history['val_loss'])

x,y,z,c = test_pred(x_test,y_test,k)

CM.append([c])
test_accuracy.append(x)
test_specificity.append(y)
test_sensitivity.append(z)
```

```
-----Start of 1 Fold-----
test images for normal class from 0 30
test images for cataract class from 0 10
test images for glaucoma class from 0 10
test images for retina disease class from 0 10
train_validation images for normal class from 0 to 0 and 30 to 300
train_validation images for cataract class from 0 to 0 and 10 to 100
train_validation images for glaucoma class from 0 0 and 10 to 100
train_validation images for retina disease class from 0 to 0 and 10 to 100
model building and compiling for fold 1
Epoch 1/50
14/14 [=====] - 58s 3s/step - loss: 17.2841 - accuracy: 0.4190 - val_loss: 153.5402 - val_ac
curacy: 0.2685
Epoch 2/50
14/14 [=====] - 51s 4s/step - loss: 9.7481 - accuracy: 0.4722 - val_loss: 55.7080 - val_accu
racy: 0.5833
Epoch 3/50
14/14 [=====] - 54s 4s/step - loss: 6.4178 - accuracy: 0.5463 - val_loss: 19.1684 - val_accu
racy: 0.6111
Epoch 4/50
14/14 [=====] - 49s 4s/step - loss: 3.9386 - accuracy: 0.5995 - val_loss: 6.3904 - val_accu
racy: 0.5185
Epoch 5/50
14/14 [=====] - 48s 3s/step - loss: 3.1486 - accuracy: 0.6319 - val_loss: 5.8248 - val_accu
racy: 0.4907
Epoch 6/50
14/14 [=====] - 48s 3s/step - loss: 4.2591 - accuracy: 0.6088 - val_loss: 7.9830 - val_accu
racy: 0.5741
Epoch 7/50
14/14 [=====] - 50s 4s/step - loss: 1.6033 - accuracy: 0.7083 - val_loss: 10.1425 - val_accu
racy: 0.3426
Epoch 8/50
14/14 [=====] - 50s 4s/step - loss: 1.3120 - accuracy: 0.7963 - val_loss: 4.6583 - val_accu
racy: 0.2685
Epoch 9/50
14/14 [=====] - 50s 4s/step - loss: 1.3342 - accuracy: 0.7639 - val_loss: 10.6681 - val_accu
racy: 0.2778
Epoch 10/50
14/14 [=====] - 49s 4s/step - loss: 1.0925 - accuracy: 0.8125 - val_loss: 12.6004 - val_accu
racy: 0.4722
Epoch 11/50
14/14 [=====] - 49s 4s/step - loss: 2.1442 - accuracy: 0.8171 - val_loss: 25.2041 - val_accu
racy: 0.1944
Epoch 12/50
14/14 [=====] - 50s 4s/step - loss: 0.4936 - accuracy: 0.9074 - val_loss: 9.2103 - val_accu
racy: 0.4815
Epoch 13/50
14/14 [=====] - 50s 4s/step - loss: 0.8820 - accuracy: 0.8819 - val_loss: 23.3810 - val_accu
racy: 0.2500
Epoch 14/50
14/14 [=====] - 51s 4s/step - loss: 0.7937 - accuracy: 0.8819 - val_loss: 12.5981 - val_accu
racy: 0.3426
Epoch 15/50
14/14 [=====] - 53s 4s/step - loss: 0.7759 - accuracy: 0.8935 - val_loss: 20.2755 - val_accu
racy: 0.2870
Epoch 16/50
14/14 [=====] - 55s 4s/step - loss: 0.6177 - accuracy: 0.9190 - val_loss: 7.1625 - val_accu
racy: 0.4167
Epoch 17/50
14/14 [=====] - 52s 4s/step - loss: 0.3939 - accuracy: 0.9468 - val_loss: 10.6047 - val_accu
racy: 0.3889
Epoch 18/50
14/14 [=====] - 52s 4s/step - loss: 0.6567 - accuracy: 0.9051 - val_loss: 7.2956 - val_accu
racy: 0.3889
Epoch 19/50
14/14 [=====] - 59s 4s/step - loss: 0.5889 - accuracy: 0.9190 - val_loss: 9.4716 - val_accu
racy: 0.5185
Epoch 20/50
14/14 [=====] - 58s 4s/step - loss: 0.5198 - accuracy: 0.9306 - val_loss: 8.7194 - val_accu
racy: 0.4074
Epoch 21/50
14/14 [=====] - 58s 4s/step - loss: 0.3503 - accuracy: 0.9491 - val_loss: 8.2896 - val_accu
racy: 0.4444
Epoch 22/50
14/14 [=====] - 58s 4s/step - loss: 0.5877 - accuracy: 0.9352 - val_loss: 19.2183 - val_accu
racy: 0.2778
Epoch 23/50
14/14 [=====] - 58s 4s/step - loss: 0.6053 - accuracy: 0.9329 - val_loss: 12.8700 - val_accu
racy: 0.3611
Epoch 24/50
14/14 [=====] - 59s 4s/step - loss: 0.2909 - accuracy: 0.9421 - val_loss: 8.5596 - val_accu
racy: 0.5741
Epoch 25/50
14/14 [=====] - 59s 4s/step - loss: 0.2906 - accuracy: 0.9444 - val_loss: 18.3331 - val_accu
racy: 0.4259
Epoch 26/50
14/14 [=====] - 60s 4s/step - loss: 0.5941 - accuracy: 0.9398 - val_loss: 7.5231 - val_accu
racy: 0.5370
```

```

Epoch 27/50
14/14 [=====] - 59s 4s/step - loss: 0.2347 - accuracy: 0.9491 - val_loss: 6.3569 - val_accuracy: 0.5556
Epoch 28/50
14/14 [=====] - 60s 4s/step - loss: 0.5233 - accuracy: 0.9537 - val_loss: 6.8908 - val_accuracy: 0.5648
Epoch 29/50
14/14 [=====] - 60s 4s/step - loss: 0.0664 - accuracy: 0.9884 - val_loss: 5.7945 - val_accuracy: 0.5833
Epoch 30/50
14/14 [=====] - 60s 4s/step - loss: 0.2285 - accuracy: 0.9630 - val_loss: 8.8942 - val_accuracy: 0.4815
Epoch 31/50
14/14 [=====] - 60s 4s/step - loss: 0.0919 - accuracy: 0.9745 - val_loss: 9.2472 - val_accuracy: 0.5463
Epoch 32/50
14/14 [=====] - 60s 4s/step - loss: 0.4916 - accuracy: 0.9468 - val_loss: 11.9573 - val_accuracy: 0.6296
Epoch 33/50
14/14 [=====] - 60s 4s/step - loss: 0.1381 - accuracy: 0.9722 - val_loss: 17.5858 - val_accuracy: 0.5370
Epoch 34/50
14/14 [=====] - 60s 4s/step - loss: 0.2403 - accuracy: 0.9676 - val_loss: 38.7305 - val_accuracy: 0.4815
Epoch 35/50
14/14 [=====] - 60s 4s/step - loss: 0.3489 - accuracy: 0.9560 - val_loss: 21.6260 - val_accuracy: 0.5278
Epoch 36/50
14/14 [=====] - 60s 4s/step - loss: 0.3140 - accuracy: 0.9769 - val_loss: 11.1485 - val_accuracy: 0.6759
Epoch 37/50
14/14 [=====] - 60s 4s/step - loss: 0.3455 - accuracy: 0.9560 - val_loss: 10.0031 - val_accuracy: 0.6574
Epoch 38/50
14/14 [=====] - 60s 4s/step - loss: 0.4400 - accuracy: 0.9676 - val_loss: 9.7232 - val_accuracy: 0.7130
Epoch 39/50
14/14 [=====] - 61s 4s/step - loss: 0.3808 - accuracy: 0.9514 - val_loss: 7.1395 - val_accuracy: 0.7130
Epoch 40/50
14/14 [=====] - 60s 4s/step - loss: 0.1886 - accuracy: 0.9745 - val_loss: 14.3204 - val_accuracy: 0.6204
Epoch 41/50
14/14 [=====] - 60s 4s/step - loss: 0.3144 - accuracy: 0.9537 - val_loss: 13.4373 - val_accuracy: 0.6296
Epoch 42/50
14/14 [=====] - 60s 4s/step - loss: 0.1421 - accuracy: 0.9722 - val_loss: 8.5680 - val_accuracy: 0.6389
Epoch 43/50
14/14 [=====] - 60s 4s/step - loss: 0.0045 - accuracy: 0.9954 - val_loss: 11.1715 - val_accuracy: 0.6296
Epoch 44/50
14/14 [=====] - 61s 4s/step - loss: 0.2271 - accuracy: 0.9630 - val_loss: 9.1211 - val_accuracy: 0.6204
Epoch 45/50
14/14 [=====] - 61s 4s/step - loss: 0.4277 - accuracy: 0.9514 - val_loss: 16.2611 - val_accuracy: 0.4167
Epoch 46/50
14/14 [=====] - 60s 4s/step - loss: 0.0310 - accuracy: 0.9861 - val_loss: 11.2167 - val_accuracy: 0.5093
Epoch 47/50
14/14 [=====] - 61s 4s/step - loss: 0.0846 - accuracy: 0.9861 - val_loss: 9.0745 - val_accuracy: 0.6204
Epoch 48/50
14/14 [=====] - 61s 4s/step - loss: 0.1746 - accuracy: 0.9699 - val_loss: 16.1745 - val_accuracy: 0.4907
Epoch 49/50
14/14 [=====] - 61s 4s/step - loss: 0.1140 - accuracy: 0.9769 - val_loss: 18.3632 - val_accuracy: 0.5926
Epoch 50/50
14/14 [=====] - 62s 4s/step - loss: 0.1612 - accuracy: 0.9699 - val_loss: 10.7261 - val_accuracy: 0.6019
-----Test accuracy for 1 fold-----
Confusion Matrix :
[[18  2  7  3]
 [ 1  8  0  1]
 [ 1  4  5  0]
 [ 1  3  3  3]]
Accuracy      : 0.5666666666666667
Specificity   : 0.8035666088297667
Sensitivity   : 0.5499999999999999
-----End of 1 Fold-----
-----Start of 2 Fold-----
test images for normal class from 30 60
test images for cataract class from 10 20
test images for glaucoma class from 10 20
test images for retina disease class from 10 20
train_validation images for normal class from 0 to 30 and 60 to 300

```



train\_validation images for cataract class from 0 to 10 and 20 to 100  
train\_validation images for glaucoma class from 0 10 and 20 to 100  
train\_validation images for retina disease class from 0 to 10 and 20 to 100  
model building and compiling for fold 2  
Epoch 1/50  
14/14 [=====] - 73s 5s/step - loss: 17.2635 - accuracy: 0.3889 - val\_loss: 171.5703 - val\_accuracy: 0.2778  
Epoch 2/50  
14/14 [=====] - 60s 4s/step - loss: 8.0821 - accuracy: 0.5046 - val\_loss: 29.0900 - val\_accuracy: 0.3519  
Epoch 3/50  
14/14 [=====] - 61s 4s/step - loss: 7.6368 - accuracy: 0.4722 - val\_loss: 13.2008 - val\_accuracy: 0.5833  
Epoch 4/50  
14/14 [=====] - 61s 4s/step - loss: 5.7092 - accuracy: 0.5579 - val\_loss: 20.1447 - val\_accuracy: 0.3426  
Epoch 5/50  
14/14 [=====] - 61s 4s/step - loss: 4.6409 - accuracy: 0.5833 - val\_loss: 23.1926 - val\_accuracy: 0.4259  
Epoch 6/50  
14/14 [=====] - 60s 4s/step - loss: 2.1570 - accuracy: 0.6065 - val\_loss: 6.6577 - val\_accuracy: 0.5185  
Epoch 7/50  
14/14 [=====] - 61s 4s/step - loss: 2.8875 - accuracy: 0.6597 - val\_loss: 3.1054 - val\_accuracy: 0.5000  
Epoch 8/50  
14/14 [=====] - 61s 4s/step - loss: 0.9731 - accuracy: 0.7685 - val\_loss: 5.8374 - val\_accuracy: 0.4815  
Epoch 9/50  
14/14 [=====] - 61s 4s/step - loss: 1.2638 - accuracy: 0.7593 - val\_loss: 4.8641 - val\_accuracy: 0.3796  
Epoch 10/50  
14/14 [=====] - 61s 4s/step - loss: 1.7394 - accuracy: 0.8032 - val\_loss: 17.5394 - val\_accuracy: 0.2963  
Epoch 11/50  
14/14 [=====] - 60s 4s/step - loss: 1.7026 - accuracy: 0.8171 - val\_loss: 4.4923 - val\_accuracy: 0.4815  
Epoch 12/50  
14/14 [=====] - 60s 4s/step - loss: 0.7558 - accuracy: 0.8380 - val\_loss: 6.1202 - val\_accuracy: 0.3704  
Epoch 13/50  
14/14 [=====] - 60s 4s/step - loss: 0.3690 - accuracy: 0.9213 - val\_loss: 5.0065 - val\_accuracy: 0.4722  
Epoch 14/50  
14/14 [=====] - 61s 4s/step - loss: 1.3181 - accuracy: 0.8079 - val\_loss: 5.5068 - val\_accuracy: 0.5278  
Epoch 15/50  
14/14 [=====] - 65s 5s/step - loss: 0.3178 - accuracy: 0.9236 - val\_loss: 4.4705 - val\_accuracy: 0.5370  
Epoch 16/50  
14/14 [=====] - 64s 5s/step - loss: 1.1185 - accuracy: 0.9097 - val\_loss: 5.6700 - val\_accuracy: 0.4907  
Epoch 17/50  
14/14 [=====] - 66s 5s/step - loss: 0.4341 - accuracy: 0.9028 - val\_loss: 4.7146 - val\_accuracy: 0.4722  
Epoch 18/50  
14/14 [=====] - 61s 4s/step - loss: 0.4934 - accuracy: 0.9097 - val\_loss: 3.7380 - val\_accuracy: 0.4722  
Epoch 19/50  
14/14 [=====] - 68s 5s/step - loss: 0.4021 - accuracy: 0.9144 - val\_loss: 4.2503 - val\_accuracy: 0.4630  
Epoch 20/50  
14/14 [=====] - 65s 5s/step - loss: 0.3810 - accuracy: 0.9491 - val\_loss: 3.9182 - val\_accuracy: 0.5556  
Epoch 21/50  
14/14 [=====] - 62s 4s/step - loss: 0.3444 - accuracy: 0.9421 - val\_loss: 8.5742 - val\_accuracy: 0.3426  
Epoch 22/50  
14/14 [=====] - 65s 5s/step - loss: 0.3397 - accuracy: 0.9306 - val\_loss: 4.5662 - val\_accuracy: 0.5556  
Epoch 23/50  
14/14 [=====] - 65s 5s/step - loss: 0.3106 - accuracy: 0.9421 - val\_loss: 5.0240 - val\_accuracy: 0.5833  
Epoch 24/50  
14/14 [=====] - 64s 5s/step - loss: 0.3315 - accuracy: 0.9352 - val\_loss: 7.2425 - val\_accuracy: 0.4074  
Epoch 25/50  
14/14 [=====] - 64s 5s/step - loss: 0.1749 - accuracy: 0.9606 - val\_loss: 5.5917 - val\_accuracy: 0.5000  
Epoch 26/50  
14/14 [=====] - 66s 5s/step - loss: 0.4309 - accuracy: 0.9259 - val\_loss: 6.7109 - val\_accuracy: 0.6296  
Epoch 27/50  
14/14 [=====] - 64s 5s/step - loss: 0.1608 - accuracy: 0.9630 - val\_loss: 4.1455 - val\_accuracy: 0.6019  
Epoch 28/50  
14/14 [=====] - 68s 5s/step - loss: 0.2221 - accuracy: 0.9537 - val\_loss: 7.0408 - val\_accuracy: 0.5833

```

Epoch 29/50
14/14 [=====] - 65s 5s/step - loss: 0.0793 - accuracy: 0.9699 - val_loss: 7.2442 - val_accuracy: 0.6019
Epoch 30/50
14/14 [=====] - 65s 5s/step - loss: 0.2758 - accuracy: 0.9514 - val_loss: 5.6187 - val_accuracy: 0.5278
Epoch 31/50
14/14 [=====] - 61s 4s/step - loss: 0.2250 - accuracy: 0.9745 - val_loss: 6.6003 - val_accuracy: 0.6111
Epoch 32/50
14/14 [=====] - 61s 4s/step - loss: 0.2225 - accuracy: 0.9560 - val_loss: 5.8900 - val_accuracy: 0.5463
Epoch 33/50
14/14 [=====] - 61s 4s/step - loss: 0.2433 - accuracy: 0.9421 - val_loss: 5.0730 - val_accuracy: 0.6019
Epoch 34/50
14/14 [=====] - 61s 4s/step - loss: 0.0050 - accuracy: 0.9977 - val_loss: 5.4300 - val_accuracy: 0.6296
Epoch 35/50
14/14 [=====] - 61s 4s/step - loss: 0.2054 - accuracy: 0.9653 - val_loss: 6.8518 - val_accuracy: 0.5000
Epoch 36/50
14/14 [=====] - 61s 4s/step - loss: 0.0676 - accuracy: 0.9838 - val_loss: 5.5392 - val_accuracy: 0.6019
Epoch 37/50
14/14 [=====] - 62s 4s/step - loss: 0.0510 - accuracy: 0.9884 - val_loss: 8.6842 - val_accuracy: 0.5741
Epoch 38/50
14/14 [=====] - 65s 5s/step - loss: 0.1194 - accuracy: 0.9653 - val_loss: 12.9881 - val_accuracy: 0.5833
Epoch 39/50
14/14 [=====] - 73s 5s/step - loss: 0.1671 - accuracy: 0.9583 - val_loss: 8.5792 - val_accuracy: 0.5000
Epoch 40/50
14/14 [=====] - 65s 5s/step - loss: 0.2219 - accuracy: 0.9653 - val_loss: 14.1004 - val_accuracy: 0.6019
Epoch 41/50
14/14 [=====] - 67s 5s/step - loss: 0.0683 - accuracy: 0.9769 - val_loss: 7.1637 - val_accuracy: 0.6111
Epoch 42/50
14/14 [=====] - 68s 5s/step - loss: 0.0518 - accuracy: 0.9861 - val_loss: 14.2329 - val_accuracy: 0.4444
Epoch 43/50
14/14 [=====] - 65s 5s/step - loss: 0.0899 - accuracy: 0.9722 - val_loss: 11.3022 - val_accuracy: 0.5833
Epoch 44/50
14/14 [=====] - 61s 4s/step - loss: 0.1074 - accuracy: 0.9722 - val_loss: 15.2516 - val_accuracy: 0.5648
Epoch 45/50
14/14 [=====] - 62s 4s/step - loss: 0.0247 - accuracy: 0.9931 - val_loss: 13.4750 - val_accuracy: 0.5278
Epoch 46/50
14/14 [=====] - 62s 4s/step - loss: 0.2198 - accuracy: 0.9653 - val_loss: 5.9763 - val_accuracy: 0.5833
Epoch 47/50
14/14 [=====] - 64s 5s/step - loss: 0.0329 - accuracy: 0.9861 - val_loss: 6.4527 - val_accuracy: 0.6204
Epoch 48/50
14/14 [=====] - 62s 4s/step - loss: 0.1778 - accuracy: 0.9722 - val_loss: 6.1013 - val_accuracy: 0.5648
Epoch 49/50
14/14 [=====] - 62s 4s/step - loss: 0.1454 - accuracy: 0.9722 - val_loss: 6.1945 - val_accuracy: 0.5556
Epoch 50/50
14/14 [=====] - 63s 5s/step - loss: 0.0094 - accuracy: 0.9977 - val_loss: 5.6420 - val_accuracy: 0.5278
-----Test accuracy for 2 fold-----
Confusion Matrix :
[[18  0  2 10]
 [ 1  6  0  3]
 [ 4  0  3  3]
 [ 2  1  1  6]]
Accuracy      : 0.55
Specificity   : 0.7957754454847479
Sensitivity   : 0.525
-----End of 2 Fold-----
-----Start of 3 Fold-----
test images for normal class from 60 90
test images for cataract class from 20 30
test images for glaucoma class from 20 30
test images for retina disease class from 20 30
train_validation images for normal class from 0 to 60 and 90 to 300
train_validation images for cataract class from 0 to 20 and 30 to 100
train_validation images for glaucoma class from 0 20 and 30 to 100
train_validation images for retina disease class from 0 to 20 and 30 to 100
model building and compiling for fold 3
Epoch 1/50
14/14 [=====] - 83s 5s/step - loss: 16.5114 - accuracy: 0.3889 - val_loss: 79.3620 - val_acc

```

uracy: 0.5000  
Epoch 2/50  
14/14 [=====] - 65s 5s/step - loss: 7.3891 - accuracy: 0.5116 - val\_loss: 66.8311 - val\_accuracy: 0.5278  
Epoch 3/50  
14/14 [=====] - 63s 5s/step - loss: 8.0075 - accuracy: 0.5162 - val\_loss: 19.2802 - val\_accuracy: 0.4352  
Epoch 4/50  
14/14 [=====] - 66s 5s/step - loss: 3.1340 - accuracy: 0.6435 - val\_loss: 5.3265 - val\_accuracy: 0.5833  
Epoch 5/50  
14/14 [=====] - 69s 5s/step - loss: 4.8871 - accuracy: 0.6111 - val\_loss: 11.5907 - val\_accuracy: 0.3889  
Epoch 6/50  
14/14 [=====] - 67s 5s/step - loss: 2.5160 - accuracy: 0.6829 - val\_loss: 5.8496 - val\_accuracy: 0.4352  
Epoch 7/50  
14/14 [=====] - 65s 5s/step - loss: 1.2175 - accuracy: 0.7361 - val\_loss: 4.3941 - val\_accuracy: 0.4167  
Epoch 8/50  
14/14 [=====] - 62s 4s/step - loss: 1.4838 - accuracy: 0.7593 - val\_loss: 8.0205 - val\_accuracy: 0.3796  
Epoch 9/50  
14/14 [=====] - 65s 5s/step - loss: 1.3021 - accuracy: 0.7801 - val\_loss: 8.0694 - val\_accuracy: 0.3333  
Epoch 10/50  
14/14 [=====] - 65s 5s/step - loss: 0.8655 - accuracy: 0.8333 - val\_loss: 7.9074 - val\_accuracy: 0.5741  
Epoch 11/50  
14/14 [=====] - 64s 5s/step - loss: 0.9060 - accuracy: 0.8449 - val\_loss: 3.4046 - val\_accuracy: 0.5648  
Epoch 12/50  
14/14 [=====] - 63s 5s/step - loss: 1.3232 - accuracy: 0.8287 - val\_loss: 7.6374 - val\_accuracy: 0.4444  
Epoch 13/50  
14/14 [=====] - 64s 5s/step - loss: 0.5952 - accuracy: 0.8935 - val\_loss: 8.9877 - val\_accuracy: 0.3611  
Epoch 14/50  
14/14 [=====] - 63s 5s/step - loss: 0.6465 - accuracy: 0.9282 - val\_loss: 7.6898 - val\_accuracy: 0.4352  
Epoch 15/50  
14/14 [=====] - 63s 5s/step - loss: 0.7981 - accuracy: 0.8542 - val\_loss: 6.1425 - val\_accuracy: 0.4722  
Epoch 16/50  
14/14 [=====] - 62s 5s/step - loss: 0.6795 - accuracy: 0.9074 - val\_loss: 3.4878 - val\_accuracy: 0.5833  
Epoch 17/50  
14/14 [=====] - 61s 4s/step - loss: 0.2819 - accuracy: 0.9306 - val\_loss: 7.4443 - val\_accuracy: 0.4167  
Epoch 18/50  
14/14 [=====] - 61s 4s/step - loss: 0.4279 - accuracy: 0.9259 - val\_loss: 3.6674 - val\_accuracy: 0.5185  
Epoch 19/50  
14/14 [=====] - 64s 5s/step - loss: 0.1980 - accuracy: 0.9421 - val\_loss: 4.7647 - val\_accuracy: 0.5833  
Epoch 20/50  
14/14 [=====] - 63s 5s/step - loss: 0.5650 - accuracy: 0.9213 - val\_loss: 5.9497 - val\_accuracy: 0.5093  
Epoch 21/50  
14/14 [=====] - 63s 5s/step - loss: 0.6435 - accuracy: 0.9074 - val\_loss: 7.2355 - val\_accuracy: 0.5463  
Epoch 22/50  
14/14 [=====] - 64s 5s/step - loss: 0.3479 - accuracy: 0.9514 - val\_loss: 6.0272 - val\_accuracy: 0.6111  
Epoch 23/50  
14/14 [=====] - 63s 5s/step - loss: 0.2187 - accuracy: 0.9537 - val\_loss: 13.8781 - val\_accuracy: 0.5556  
Epoch 24/50  
14/14 [=====] - 66s 5s/step - loss: 0.8365 - accuracy: 0.9259 - val\_loss: 7.9400 - val\_accuracy: 0.6389  
Epoch 25/50  
14/14 [=====] - 67s 5s/step - loss: 0.1523 - accuracy: 0.9630 - val\_loss: 5.4361 - val\_accuracy: 0.5463  
Epoch 26/50  
14/14 [=====] - 67s 5s/step - loss: 0.0646 - accuracy: 0.9769 - val\_loss: 5.9451 - val\_accuracy: 0.5463  
Epoch 27/50  
14/14 [=====] - 65s 5s/step - loss: 0.3333 - accuracy: 0.9514 - val\_loss: 8.1208 - val\_accuracy: 0.4630  
Epoch 28/50  
14/14 [=====] - 64s 5s/step - loss: 0.2162 - accuracy: 0.9514 - val\_loss: 7.8425 - val\_accuracy: 0.4722  
Epoch 29/50  
14/14 [=====] - 69s 5s/step - loss: 0.5221 - accuracy: 0.9560 - val\_loss: 7.5759 - val\_accuracy: 0.5463  
Epoch 30/50  
14/14 [=====] - 69s 5s/step - loss: 0.1046 - accuracy: 0.9769 - val\_loss: 5.9304 - val\_accuracy: 0.5278

```

Epoch 31/50
14/14 [=====] - 67s 5s/step - loss: 0.1100 - accuracy: 0.9815 - val_loss: 19.5170 - val_accuracy: 0.6296
Epoch 32/50
14/14 [=====] - 71s 5s/step - loss: 0.5302 - accuracy: 0.9120 - val_loss: 3.9329 - val_accuracy: 0.6852
Epoch 33/50
14/14 [=====] - 67s 5s/step - loss: 0.1028 - accuracy: 0.9792 - val_loss: 11.1048 - val_accuracy: 0.6574
Epoch 34/50
14/14 [=====] - 70s 5s/step - loss: 0.0189 - accuracy: 0.9861 - val_loss: 15.4818 - val_accuracy: 0.6481
Epoch 35/50
14/14 [=====] - 66s 5s/step - loss: 0.1544 - accuracy: 0.9653 - val_loss: 13.1234 - val_accuracy: 0.5741
Epoch 36/50
14/14 [=====] - 67s 5s/step - loss: 0.2266 - accuracy: 0.9630 - val_loss: 9.1231 - val_accuracy: 0.5741
Epoch 37/50
14/14 [=====] - 66s 5s/step - loss: 0.1814 - accuracy: 0.9745 - val_loss: 13.4963 - val_accuracy: 0.6944
Epoch 38/50
14/14 [=====] - 71s 5s/step - loss: 0.2765 - accuracy: 0.9769 - val_loss: 9.4926 - val_accuracy: 0.5556
Epoch 39/50
14/14 [=====] - 68s 5s/step - loss: 0.3268 - accuracy: 0.9537 - val_loss: 6.7609 - val_accuracy: 0.6389
Epoch 40/50
14/14 [=====] - 71s 5s/step - loss: 0.2084 - accuracy: 0.9722 - val_loss: 6.2360 - val_accuracy: 0.6481
Epoch 41/50
14/14 [=====] - 69s 5s/step - loss: 0.0519 - accuracy: 0.9907 - val_loss: 7.0247 - val_accuracy: 0.5926
Epoch 42/50
14/14 [=====] - 68s 5s/step - loss: 0.3308 - accuracy: 0.9676 - val_loss: 11.9977 - val_accuracy: 0.5463
Epoch 43/50
14/14 [=====] - 62s 5s/step - loss: 0.1759 - accuracy: 0.9699 - val_loss: 16.6241 - val_accuracy: 0.6296
Epoch 44/50
14/14 [=====] - 63s 5s/step - loss: 0.1025 - accuracy: 0.9838 - val_loss: 17.6387 - val_accuracy: 0.6667
Epoch 45/50
14/14 [=====] - 70s 5s/step - loss: 0.1233 - accuracy: 0.9745 - val_loss: 19.7569 - val_accuracy: 0.6481
Epoch 46/50
14/14 [=====] - 69s 5s/step - loss: 0.0413 - accuracy: 0.9907 - val_loss: 18.2880 - val_accuracy: 0.5648
Epoch 47/50
14/14 [=====] - 68s 5s/step - loss: 0.2399 - accuracy: 0.9653 - val_loss: 32.8459 - val_accuracy: 0.6019
Epoch 48/50
14/14 [=====] - 67s 5s/step - loss: 0.3890 - accuracy: 0.9491 - val_loss: 59.6093 - val_accuracy: 0.5370
Epoch 49/50
14/14 [=====] - 66s 5s/step - loss: 0.5853 - accuracy: 0.9792 - val_loss: 7.1689 - val_accuracy: 0.6204
Epoch 50/50
14/14 [=====] - 68s 5s/step - loss: 0.0246 - accuracy: 0.9931 - val_loss: 9.0464 - val_accuracy: 0.6667
-----Test accuracy for 3 fold-----
Confusion Matrix :
[[22  0  2  6]
 [ 2  3  2  3]
 [ 4  1  5  0]
 [ 5  0  0  5]]
Accuracy      : 0.5833333333333334
Specificity   : 0.790736836692719
Sensitivity   : 0.5083333333333333
-----End of 3 Fold-----
-----Start of 4 Fold-----
test images for normal class from 90 120
test images for cataract class from 30 40
test images for glaucoma class from 30 40
test images for retina disease class from 30 40
train_validation images for normal class from 0 to 90 and 120 to 300
train_validation images for cataract class from 0 to 30 and 40 to 100
train_validation images for glaucoma class from 0 30 and 40 to 100
train_validation images for retina disease class from 0 to 30 and 40 to 100
model building and compiling for fold 4
Epoch 1/50
14/14 [=====] - 84s 5s/step - loss: 16.4083 - accuracy: 0.4421 - val_loss: 111.7088 - val_accuracy: 0.5278
Epoch 2/50
14/14 [=====] - 64s 5s/step - loss: 7.5717 - accuracy: 0.4861 - val_loss: 33.1182 - val_accuracy: 0.4722
Epoch 3/50
14/14 [=====] - 68s 5s/step - loss: 5.9330 - accuracy: 0.5324 - val_loss: 11.6992 - val_accuracy: 0.5324

```

racy: 0.4259  
Epoch 4/50  
14/14 [=====] - 64s 5s/step - loss: 4.4009 - accuracy: 0.5440 - val\_loss: 12.0992 - val\_accu  
racy: 0.5185  
Epoch 5/50  
14/14 [=====] - 65s 5s/step - loss: 6.3400 - accuracy: 0.5903 - val\_loss: 22.1179 - val\_accu  
racy: 0.2870  
Epoch 6/50  
14/14 [=====] - 65s 5s/step - loss: 2.5884 - accuracy: 0.6458 - val\_loss: 5.7328 - val\_accu  
acy: 0.4722  
Epoch 7/50  
14/14 [=====] - 65s 5s/step - loss: 1.1794 - accuracy: 0.7708 - val\_loss: 8.2184 - val\_accu  
acy: 0.4167  
Epoch 8/50  
14/14 [=====] - 64s 5s/step - loss: 1.9006 - accuracy: 0.7037 - val\_loss: 7.2428 - val\_accu  
acy: 0.4630  
Epoch 9/50  
14/14 [=====] - 65s 5s/step - loss: 1.7541 - accuracy: 0.7616 - val\_loss: 7.9216 - val\_accu  
acy: 0.3889  
Epoch 10/50  
14/14 [=====] - 72s 5s/step - loss: 1.4971 - accuracy: 0.8264 - val\_loss: 4.8575 - val\_accu  
acy: 0.5000  
Epoch 11/50  
14/14 [=====] - 73s 5s/step - loss: 1.3741 - accuracy: 0.8171 - val\_loss: 6.2237 - val\_accu  
acy: 0.5278  
Epoch 12/50  
14/14 [=====] - 72s 5s/step - loss: 1.3626 - accuracy: 0.8009 - val\_loss: 9.0697 - val\_accu  
acy: 0.5463  
Epoch 13/50  
14/14 [=====] - 71s 5s/step - loss: 0.4423 - accuracy: 0.9051 - val\_loss: 7.2322 - val\_accu  
acy: 0.3241  
Epoch 14/50  
14/14 [=====] - 72s 5s/step - loss: 0.6848 - accuracy: 0.8796 - val\_loss: 16.0549 - val\_accu  
racy: 0.2315  
Epoch 15/50  
14/14 [=====] - 71s 5s/step - loss: 0.5686 - accuracy: 0.9120 - val\_loss: 6.8447 - val\_accu  
acy: 0.3519  
Epoch 16/50  
14/14 [=====] - 72s 5s/step - loss: 0.2579 - accuracy: 0.9444 - val\_loss: 7.7335 - val\_accu  
acy: 0.3241  
Epoch 17/50  
14/14 [=====] - 68s 5s/step - loss: 0.5176 - accuracy: 0.9005 - val\_loss: 12.3681 - val\_accu  
racy: 0.2685  
Epoch 18/50  
14/14 [=====] - 65s 5s/step - loss: 0.6572 - accuracy: 0.8866 - val\_loss: 8.1734 - val\_accu  
acy: 0.3796  
Epoch 19/50  
14/14 [=====] - 65s 5s/step - loss: 0.4088 - accuracy: 0.9352 - val\_loss: 6.4734 - val\_accu  
acy: 0.4907  
Epoch 20/50  
14/14 [=====] - 65s 5s/step - loss: 0.3619 - accuracy: 0.9421 - val\_loss: 8.9399 - val\_accu  
acy: 0.5000  
Epoch 21/50  
14/14 [=====] - 71s 5s/step - loss: 0.3089 - accuracy: 0.9444 - val\_loss: 17.7810 - val\_accu  
racy: 0.2778  
Epoch 22/50  
14/14 [=====] - 70s 5s/step - loss: 0.2450 - accuracy: 0.9583 - val\_loss: 8.9018 - val\_accu  
acy: 0.5370  
Epoch 23/50  
14/14 [=====] - 70s 5s/step - loss: 0.3007 - accuracy: 0.9491 - val\_loss: 6.0992 - val\_accu  
acy: 0.4630  
Epoch 24/50  
14/14 [=====] - 71s 5s/step - loss: 0.4689 - accuracy: 0.9329 - val\_loss: 4.7129 - val\_accu  
acy: 0.6019  
Epoch 25/50  
14/14 [=====] - 69s 5s/step - loss: 0.1980 - accuracy: 0.9606 - val\_loss: 9.5807 - val\_accu  
acy: 0.4259  
Epoch 26/50  
14/14 [=====] - 70s 5s/step - loss: 0.0737 - accuracy: 0.9838 - val\_loss: 7.8474 - val\_accu  
acy: 0.5278  
Epoch 27/50  
14/14 [=====] - 66s 5s/step - loss: 0.0845 - accuracy: 0.9792 - val\_loss: 11.8902 - val\_accu  
racy: 0.3704  
Epoch 28/50  
14/14 [=====] - 67s 5s/step - loss: 0.4671 - accuracy: 0.9375 - val\_loss: 6.8091 - val\_accu  
acy: 0.5926  
Epoch 29/50  
14/14 [=====] - 68s 5s/step - loss: 0.2269 - accuracy: 0.9699 - val\_loss: 9.9267 - val\_accu  
acy: 0.5185  
Epoch 30/50  
14/14 [=====] - 67s 5s/step - loss: 0.3394 - accuracy: 0.9468 - val\_loss: 9.9964 - val\_accu  
acy: 0.4815  
Epoch 31/50  
14/14 [=====] - 68s 5s/step - loss: 0.5923 - accuracy: 0.9630 - val\_loss: 11.3837 - val\_accu  
racy: 0.4259  
Epoch 32/50  
14/14 [=====] - 68s 5s/step - loss: 0.1558 - accuracy: 0.9722 - val\_loss: 9.2577 - val\_accu  
acy: 0.5926

```

Epoch 33/50
14/14 [=====] - 69s 5s/step - loss: 0.1139 - accuracy: 0.9699 - val_loss: 8.3192 - val_accuracy: 0.6204
Epoch 34/50
14/14 [=====] - 66s 5s/step - loss: 0.1832 - accuracy: 0.9583 - val_loss: 22.6201 - val_accuracy: 0.5463
Epoch 35/50
14/14 [=====] - 64s 5s/step - loss: 0.3482 - accuracy: 0.9560 - val_loss: 31.9727 - val_accuracy: 0.4444
Epoch 36/50
14/14 [=====] - 65s 5s/step - loss: 0.4106 - accuracy: 0.9560 - val_loss: 30.2404 - val_accuracy: 0.5556
Epoch 37/50
14/14 [=====] - 64s 5s/step - loss: 0.2450 - accuracy: 0.9606 - val_loss: 26.6628 - val_accuracy: 0.6389
Epoch 38/50
14/14 [=====] - 65s 5s/step - loss: 0.0084 - accuracy: 0.9954 - val_loss: 27.1072 - val_accuracy: 0.6111
Epoch 39/50
14/14 [=====] - 64s 5s/step - loss: 0.0917 - accuracy: 0.9838 - val_loss: 66.8255 - val_accuracy: 0.5370
Epoch 40/50
14/14 [=====] - 65s 5s/step - loss: 0.1079 - accuracy: 0.9792 - val_loss: 14.3534 - val_accuracy: 0.5185
Epoch 41/50
14/14 [=====] - 65s 5s/step - loss: 0.3082 - accuracy: 0.9630 - val_loss: 7.0812 - val_accuracy: 0.6389
Epoch 42/50
14/14 [=====] - 69s 5s/step - loss: 0.1860 - accuracy: 0.9861 - val_loss: 8.0221 - val_accuracy: 0.5556
Epoch 43/50
14/14 [=====] - 66s 5s/step - loss: 0.4348 - accuracy: 0.9514 - val_loss: 20.2448 - val_accuracy: 0.6111
Epoch 44/50
14/14 [=====] - 66s 5s/step - loss: 0.1545 - accuracy: 0.9722 - val_loss: 28.4735 - val_accuracy: 0.5648
Epoch 45/50
14/14 [=====] - 65s 5s/step - loss: 0.2233 - accuracy: 0.9653 - val_loss: 30.1475 - val_accuracy: 0.5833
Epoch 46/50
14/14 [=====] - 64s 5s/step - loss: 0.0186 - accuracy: 0.9931 - val_loss: 25.0335 - val_accuracy: 0.5741
Epoch 47/50
14/14 [=====] - 65s 5s/step - loss: 0.2050 - accuracy: 0.9653 - val_loss: 9.0410 - val_accuracy: 0.5370
Epoch 48/50
14/14 [=====] - 64s 5s/step - loss: 0.2750 - accuracy: 0.9722 - val_loss: 14.4980 - val_accuracy: 0.6481
Epoch 49/50
14/14 [=====] - 65s 5s/step - loss: 0.3710 - accuracy: 0.9676 - val_loss: 12.9928 - val_accuracy: 0.5370
Epoch 50/50
14/14 [=====] - 67s 5s/step - loss: 0.0593 - accuracy: 0.9815 - val_loss: 9.9789 - val_accuracy: 0.5833
-----Test accuracy for 4 fold-----
Confusion Matrix :
[[25  0  2  3]
 [ 5  5  0  0]
 [ 6  0  2  2]
 [ 7  1  1  1]]
Accuracy      : 0.55
Specificity   : 0.7624597799547089
Sensitivity   : 0.40833333333333334
-----End of 4 Fold-----
-----Start of 5 Fold-----
test images for normal class from 120 150
test images for cataract class from 40 50
test images for glaucoma class from 40 50
test images for retina disease class from 40 50
train_validation images for normal class from 0 to 120 and 150 to 300
train_validation images for cataract class from 0 to 40 and 50 to 100
train_validation images for glaucoma class from 0 40 and 50 to 100
train_validation images for retina disease class from 0 to 40 and 50 to 100
model building and compiling for fold 5
Epoch 1/50
14/14 [=====] - 84s 5s/step - loss: 16.8061 - accuracy: 0.4329 - val_loss: 100.5702 - val_accuracy: 0.3611
Epoch 2/50
14/14 [=====] - 74s 5s/step - loss: 8.6403 - accuracy: 0.4630 - val_loss: 63.2929 - val_accuracy: 0.2130
Epoch 3/50
14/14 [=====] - 72s 5s/step - loss: 7.8459 - accuracy: 0.5116 - val_loss: 12.7316 - val_accuracy: 0.5278
Epoch 4/50
14/14 [=====] - 73s 5s/step - loss: 3.3636 - accuracy: 0.5949 - val_loss: 42.3234 - val_accuracy: 0.1574
Epoch 5/50
14/14 [=====] - 74s 5s/step - loss: 5.3458 - accuracy: 0.5648 - val_loss: 9.1764 - val_accuracy:

```

acy: 0.5185  
Epoch 6/50  
14/14 [=====] - 72s 5s/step - loss: 3.6388 - accuracy: 0.6505 - val\_loss: 6.8789 - val\_accuracy: 0.5093  
Epoch 7/50  
14/14 [=====] - 74s 5s/step - loss: 2.2885 - accuracy: 0.7106 - val\_loss: 4.0748 - val\_accuracy: 0.4167  
Epoch 8/50  
14/14 [=====] - 71s 5s/step - loss: 1.0823 - accuracy: 0.7685 - val\_loss: 10.0831 - val\_accuracy: 0.2963  
Epoch 9/50  
14/14 [=====] - 75s 5s/step - loss: 0.9378 - accuracy: 0.8194 - val\_loss: 11.5755 - val\_accuracy: 0.3519  
Epoch 10/50  
14/14 [=====] - 79s 6s/step - loss: 1.0131 - accuracy: 0.7986 - val\_loss: 4.7080 - val\_accuracy: 0.3148  
Epoch 11/50  
14/14 [=====] - 77s 6s/step - loss: 1.1128 - accuracy: 0.8495 - val\_loss: 7.1734 - val\_accuracy: 0.3611  
Epoch 12/50  
14/14 [=====] - 77s 6s/step - loss: 0.6837 - accuracy: 0.8727 - val\_loss: 11.3418 - val\_accuracy: 0.2870  
Epoch 13/50  
14/14 [=====] - 74s 5s/step - loss: 0.6888 - accuracy: 0.8657 - val\_loss: 13.1919 - val\_accuracy: 0.2870  
Epoch 14/50  
14/14 [=====] - 76s 6s/step - loss: 0.7118 - accuracy: 0.8750 - val\_loss: 7.4913 - val\_accuracy: 0.4167  
Epoch 15/50  
14/14 [=====] - 77s 6s/step - loss: 0.7340 - accuracy: 0.9144 - val\_loss: 9.0704 - val\_accuracy: 0.3056  
Epoch 16/50  
14/14 [=====] - 72s 5s/step - loss: 0.3166 - accuracy: 0.9306 - val\_loss: 8.5901 - val\_accuracy: 0.3611  
Epoch 17/50  
14/14 [=====] - 71s 5s/step - loss: 0.4969 - accuracy: 0.9329 - val\_loss: 9.9101 - val\_accuracy: 0.4630  
Epoch 18/50  
14/14 [=====] - 70s 5s/step - loss: 0.3557 - accuracy: 0.9306 - val\_loss: 5.4460 - val\_accuracy: 0.4722  
Epoch 19/50  
14/14 [=====] - 81s 6s/step - loss: 0.4224 - accuracy: 0.9352 - val\_loss: 9.8275 - val\_accuracy: 0.3981  
Epoch 20/50  
14/14 [=====] - 72s 5s/step - loss: 0.3909 - accuracy: 0.9236 - val\_loss: 13.0086 - val\_accuracy: 0.3981  
Epoch 21/50  
14/14 [=====] - 76s 5s/step - loss: 0.2573 - accuracy: 0.9606 - val\_loss: 11.9194 - val\_accuracy: 0.4167  
Epoch 22/50  
14/14 [=====] - 72s 5s/step - loss: 0.5935 - accuracy: 0.9259 - val\_loss: 6.8054 - val\_accuracy: 0.3611  
Epoch 23/50  
14/14 [=====] - 76s 6s/step - loss: 0.7078 - accuracy: 0.9190 - val\_loss: 7.3068 - val\_accuracy: 0.4815  
Epoch 24/50  
14/14 [=====] - 74s 5s/step - loss: 0.2119 - accuracy: 0.9537 - val\_loss: 7.4345 - val\_accuracy: 0.5556  
Epoch 25/50  
14/14 [=====] - 77s 6s/step - loss: 0.4142 - accuracy: 0.9583 - val\_loss: 11.3159 - val\_accuracy: 0.3796  
Epoch 26/50  
14/14 [=====] - 71s 5s/step - loss: 0.4054 - accuracy: 0.9421 - val\_loss: 8.4829 - val\_accuracy: 0.5833  
Epoch 27/50  
14/14 [=====] - 70s 5s/step - loss: 0.1901 - accuracy: 0.9676 - val\_loss: 7.7634 - val\_accuracy: 0.5463  
Epoch 28/50  
14/14 [=====] - 77s 6s/step - loss: 0.4358 - accuracy: 0.9306 - val\_loss: 12.0577 - val\_accuracy: 0.4537  
Epoch 29/50  
14/14 [=====] - 73s 5s/step - loss: 0.2386 - accuracy: 0.9560 - val\_loss: 6.1322 - val\_accuracy: 0.5556  
Epoch 30/50  
14/14 [=====] - 70s 5s/step - loss: 0.0764 - accuracy: 0.9815 - val\_loss: 6.2925 - val\_accuracy: 0.5093  
Epoch 31/50  
14/14 [=====] - 72s 5s/step - loss: 0.0892 - accuracy: 0.9676 - val\_loss: 8.7189 - val\_accuracy: 0.5185  
Epoch 32/50  
14/14 [=====] - 75s 5s/step - loss: 0.1591 - accuracy: 0.9722 - val\_loss: 11.7770 - val\_accuracy: 0.4907  
Epoch 33/50  
14/14 [=====] - 72s 5s/step - loss: 0.1831 - accuracy: 0.9653 - val\_loss: 10.4377 - val\_accuracy: 0.4630  
Epoch 34/50  
14/14 [=====] - 71s 5s/step - loss: 0.2619 - accuracy: 0.9606 - val\_loss: 9.3999 - val\_accuracy: 0.5741

```

Epoch 35/50
14/14 [=====] - 70s 5s/step - loss: 0.2613 - accuracy: 0.9606 - val_loss: 9.3236 - val_accuracy: 0.5185
Epoch 36/50
14/14 [=====] - 74s 5s/step - loss: 0.1023 - accuracy: 0.9676 - val_loss: 12.3150 - val_accuracy: 0.4444
Epoch 37/50
14/14 [=====] - 77s 6s/step - loss: 0.1345 - accuracy: 0.9699 - val_loss: 14.3579 - val_accuracy: 0.4815
Epoch 38/50
14/14 [=====] - 76s 5s/step - loss: 0.0671 - accuracy: 0.9838 - val_loss: 16.6811 - val_accuracy: 0.4537
Epoch 39/50
14/14 [=====] - 71s 5s/step - loss: 0.3616 - accuracy: 0.9468 - val_loss: 6.7016 - val_accuracy: 0.5648
Epoch 40/50
14/14 [=====] - 71s 5s/step - loss: 0.1247 - accuracy: 0.9815 - val_loss: 6.8449 - val_accuracy: 0.5648
Epoch 41/50
14/14 [=====] - 71s 5s/step - loss: 0.0048 - accuracy: 0.9977 - val_loss: 7.9764 - val_accuracy: 0.5370
Epoch 42/50
14/14 [=====] - 71s 5s/step - loss: 0.0167 - accuracy: 0.9907 - val_loss: 11.0307 - val_accuracy: 0.5093
Epoch 43/50
14/14 [=====] - 71s 5s/step - loss: 0.0858 - accuracy: 0.9907 - val_loss: 14.3441 - val_accuracy: 0.5000
Epoch 44/50
14/14 [=====] - 73s 5s/step - loss: 0.2303 - accuracy: 0.9653 - val_loss: 13.4991 - val_accuracy: 0.6019
Epoch 45/50
14/14 [=====] - 78s 6s/step - loss: 0.2730 - accuracy: 0.9583 - val_loss: 17.6591 - val_accuracy: 0.5648
Epoch 46/50
14/14 [=====] - 75s 5s/step - loss: 0.1083 - accuracy: 0.9722 - val_loss: 15.5467 - val_accuracy: 0.5278
Epoch 47/50
14/14 [=====] - 77s 6s/step - loss: 0.1408 - accuracy: 0.9722 - val_loss: 21.2682 - val_accuracy: 0.5370
Epoch 48/50
14/14 [=====] - 78s 6s/step - loss: 0.0167 - accuracy: 0.9931 - val_loss: 20.0615 - val_accuracy: 0.5741
Epoch 49/50
14/14 [=====] - 80s 6s/step - loss: 0.0160 - accuracy: 0.9954 - val_loss: 19.1696 - val_accuracy: 0.6296
Epoch 50/50
14/14 [=====] - 79s 6s/step - loss: 0.0351 - accuracy: 0.9907 - val_loss: 22.9709 - val_accuracy: 0.6019
WARNING:tensorflow:5 out of the last 9 calls to <function Model.make_predict_function.<locals>.predict_function at 0x000001826AB85A60> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.
-----Test accuracy for 5 fold-----
Confusion Matrix :
[[24  0  3  3]
 [ 2  6  2  0]
 [ 8  0  1  1]
 [ 7  1  1  1]]
Accuracy      : 0.5333333333333333
Specificity   : 0.7516287716287716
Sensitivity   : 0.4
-----End of 5 Fold-----

```

## Test Evaluation Results

```
In [17]: test_accuracy
```

```
Out[17]: [0.5666666666666667, 0.55, 0.5833333333333334, 0.55, 0.5333333333333333]
```

```
In [18]: mean_test_accuracy=np.mean(test_accuracy)
mean_test_accuracy
```

```
Out[18]: 0.5566666666666666
```

```
In [19]: test_sensitivity
```

```
Out[19]: [0.5499999999999999, 0.525, 0.5083333333333333, 0.4083333333333334, 0.4]
```



```
In [20]: mean_test_sensitivity= np.mean(test_sensitivity)
mean_test_sensitivity
```

```
Out[20]: 0.4783333333333333
```

```
In [21]: test_specificity
```

```
Out[21]: [0.8035666088297667,
0.7957754454847479,
0.790736836692719,
0.7624597799547089,
0.7516287716287716]
```

```
In [22]: mean_test_specificity= np.mean(test_specificity)
mean_test_specificity
```

```
Out[22]: 0.7808334885181429
```

## Training and Validation Evaluation Results

```
In [23]: train_acc
```

```
Out[23]: array([0.41898149, 0.47222221, 0.5462963 , 0.59953701, 0.63194442,
0.6087963 , 0.70833331, 0.7962963 , 0.7638889 , 0.8125 ,
0.81712961, 0.9074074 , 0.88194442, 0.88194442, 0.89351851,
0.91898149, 0.94675928, 0.9050926 , 0.91898149, 0.93055558,
0.94907409, 0.93518519, 0.93287039, 0.94212961, 0.94444442,
0.93981481, 0.94907409, 0.9537037 , 0.98842591, 0.96296299,
0.97453701, 0.94675928, 0.97222221, 0.9675926 , 0.95601851,
0.97685188, 0.95601851, 0.9675926 , 0.9513889 , 0.97453701,
0.9537037 , 0.97222221, 0.99537039, 0.96296299, 0.9513889 ,
0.9861111 , 0.9861111 , 0.9699074 , 0.97685188, 0.9699074 ,
0.3888889 , 0.50462961, 0.47222221, 0.55787039, 0.58333331,
0.60648149, 0.65972221, 0.76851851, 0.75925928, 0.80324072,
0.81712961, 0.83796299, 0.9212963 , 0.80787039, 0.9236111 ,
0.90972221, 0.90277779, 0.90972221, 0.91435188, 0.94907409,
0.94212961, 0.93055558, 0.94212961, 0.93518519, 0.96064812,
0.92592591, 0.96296299, 0.9537037 , 0.9699074 , 0.9513889 ,
0.97453701, 0.95601851, 0.94212961, 0.99768519, 0.96527779,
0.9837963 , 0.98842591, 0.96527779, 0.95833331, 0.96527779,
0.97685188, 0.9861111 , 0.97222221, 0.97222221, 0.99305558,
0.96527779, 0.9861111 , 0.97222221, 0.97222221, 0.99768519,
0.3888889 , 0.51157409, 0.5162037 , 0.64351851, 0.6111111 ,
0.68287039, 0.7361111 , 0.75925928, 0.7800926 , 0.83333331,
0.8449074 , 0.8287037 , 0.89351851, 0.92824072, 0.85416669,
0.9074074 , 0.93055558, 0.92592591, 0.94212961, 0.9212963 ,
0.9074074 , 0.9513889 , 0.9537037 , 0.92592591, 0.96296299,
0.97685188, 0.9513889 , 0.9513889 , 0.95601851, 0.97685188,
0.98148149, 0.91203701, 0.97916669, 0.9861111 , 0.96527779,
0.96296299, 0.97453701, 0.97685188, 0.9537037 , 0.97222221,
0.99074072, 0.9675926 , 0.9699074 , 0.9837963 , 0.97453701,
0.99074072, 0.96527779, 0.94907409, 0.97916669, 0.99305558,
0.44212964, 0.4861111 , 0.5324074 , 0.54398149, 0.59027779,
0.64583331, 0.77083331, 0.7037037 , 0.76157409, 0.8263889 ,
0.81712961, 0.80092591, 0.9050926 , 0.87962961, 0.91203701,
0.94444442, 0.90046299, 0.88657409, 0.93518519, 0.94212961,
0.94444442, 0.95833331, 0.94907409, 0.93287039, 0.96064812,
0.9837963 , 0.97916669, 0.9375 , 0.9699074 , 0.94675928,
0.96296299, 0.97222221, 0.9699074 , 0.95833331, 0.95601851,
0.95601851, 0.96064812, 0.99537039, 0.9837963 , 0.97916669,
0.96296299, 0.9861111 , 0.9513889 , 0.97222221, 0.96527779,
0.99305558, 0.96527779, 0.97222221, 0.9675926 , 0.98148149,
0.43287036, 0.46296296, 0.51157409, 0.5949074 , 0.56481481,
0.65046299, 0.71064812, 0.76851851, 0.81944442, 0.7986111 ,
0.84953701, 0.87268519, 0.86574072, 0.875 , 0.91435188,
0.93055558, 0.93287039, 0.93055558, 0.93518519, 0.9236111 ,
0.96064812, 0.92592591, 0.91898149, 0.9537037 , 0.95833331,
0.94212961, 0.9675926 , 0.93055558, 0.95601851, 0.98148149,
0.9675926 , 0.97222221, 0.96527779, 0.96064812, 0.96064812,
0.9675926 , 0.9699074 , 0.9837963 , 0.94675928, 0.98148149,
0.99768519, 0.99074072, 0.99074072, 0.96527779, 0.95833331,
0.97222221, 0.97222221, 0.99305558, 0.99537039, 0.99074072])
```

```
In [24]: mean_train_accuracy=np.mean(train_acc)
mean_train_accuracy
```

```
Out[24]: 0.8844629625082016
```

```
In [25]: val_acc
```

```
Out[25]: array([0.26851851, 0.58333331, 0.61111111, 0.51851851, 0.49074075,
0.57407409, 0.3425926 , 0.26851851, 0.27777779, 0.47222221,
0.19444445, 0.48148149, 0.25      , 0.3425926 , 0.28703704,
0.41666666, 0.38888889 , 0.38888889 , 0.51851851, 0.4074074 ,
0.44444445, 0.27777779, 0.36111111 , 0.57407409, 0.42592594,
0.53703701, 0.55555558, 0.56481481, 0.58333331, 0.48148149,
0.5462963 , 0.62962961, 0.53703701, 0.48148149, 0.52777779,
0.67592591, 0.6574074 , 0.71296299, 0.71296299, 0.62037039,
0.62962961, 0.63888889 , 0.62962961, 0.62037039, 0.41666666,
0.50925928, 0.62037039, 0.49074075, 0.5925926 , 0.60185188,
0.27777779, 0.35185185, 0.58333331, 0.3425926 , 0.42592594,
0.51851851, 0.5      , 0.48148149, 0.37962964, 0.2962963 ,
0.48148149, 0.37037036, 0.47222221, 0.52777779, 0.53703701,
0.49074075, 0.47222221, 0.47222221, 0.46296296, 0.55555558,
0.3425926 , 0.55555558, 0.58333331, 0.4074074 , 0.5      ,
0.62962961, 0.60185188, 0.58333331, 0.60185188, 0.52777779,
0.61111111 , 0.5462963 , 0.60185188, 0.62962961, 0.5      ,
0.60185188, 0.57407409, 0.58333331, 0.5      , 0.60185188,
0.61111111 , 0.44444445, 0.58333331, 0.56481481, 0.52777779,
0.58333331, 0.62037039, 0.56481481, 0.55555558, 0.52777779,
0.5      , 0.52777779, 0.43518519, 0.58333331, 0.38888889 ,
0.43518519, 0.41666666, 0.37962964, 0.33333334, 0.57407409,
0.56481481, 0.44444445, 0.36111111 , 0.43518519, 0.47222221,
0.58333331, 0.41666666, 0.51851851, 0.58333331, 0.50925928,
0.5462963 , 0.61111111 , 0.55555558, 0.63888889 , 0.5462963 ,
0.5462963 , 0.46296296, 0.47222221, 0.5462963 , 0.52777779,
0.62962961, 0.68518519, 0.6574074 , 0.64814812, 0.57407409,
0.57407409, 0.69444442, 0.55555558, 0.63888889 , 0.64814812,
0.5925926 , 0.5462963 , 0.62962961, 0.66666669, 0.64814812,
0.56481481, 0.60185188, 0.53703701, 0.62037039, 0.66666669,
0.52777779, 0.47222221, 0.42592594, 0.51851851, 0.28703704,
0.47222221, 0.41666666, 0.46296296, 0.38888889 , 0.5      ,
0.52777779, 0.5462963 , 0.32407406, 0.23148148, 0.35185185,
0.32407406, 0.26851851, 0.37962964, 0.49074075, 0.5      ,
0.27777779, 0.53703701, 0.46296296, 0.60185188, 0.42592594,
0.52777779, 0.37037036, 0.5925926 , 0.51851851, 0.48148149,
0.42592594, 0.5925926 , 0.62037039, 0.5462963 , 0.44444445,
0.55555558, 0.63888889 , 0.61111111 , 0.53703701, 0.51851851,
0.63888889 , 0.55555558, 0.61111111 , 0.56481481, 0.58333331,
0.57407409, 0.53703701, 0.64814812, 0.53703701, 0.58333331,
0.36111111 , 0.21296297, 0.52777779, 0.1574074 , 0.51851851,
0.50925928, 0.41666666, 0.2962963 , 0.35185185, 0.31481481,
0.36111111 , 0.28703704, 0.28703704, 0.41666666, 0.30555555,
0.36111111 , 0.46296296, 0.47222221, 0.39814815, 0.39814815,
0.41666666, 0.36111111 , 0.48148149, 0.55555558, 0.37962964,
0.58333331, 0.5462963 , 0.4537037 , 0.55555558, 0.50925928,
0.51851851, 0.49074075, 0.46296296, 0.57407409, 0.51851851,
0.44444445, 0.48148149, 0.4537037 , 0.56481481, 0.56481481,
0.53703701, 0.50925928, 0.5      , 0.60185188, 0.56481481,
0.52777779, 0.53703701, 0.57407409, 0.62962961, 0.60185188])
```

```
In [26]: mean_val_accuracy=np.mean(val_acc)
mean_val_accuracy
```

```
Out[26]: 0.5006296305656434
```

```
In [27]: train_loss
```

```
Out[27]: array([1.72841301e+01, 9.74805164e+00, 6.41780710e+00, 3.93864870e+00,
 3.14864564e+00, 4.25907993e+00, 1.60331726e+00, 1.31203234e+00,
 1.33422756e+00, 1.09247172e+00, 2.14421582e+00, 4.93645161e-01,
 8.81963491e-01, 7.93705463e-01, 7.75893748e-01, 6.17669344e-01,
 3.93884420e-01, 6.56697512e-01, 5.88859618e-01, 5.19827962e-01,
 3.50265235e-01, 5.87748945e-01, 6.05326176e-01, 2.90918261e-01,
 2.90612429e-01, 5.94127536e-01, 2.34718740e-01, 5.23282766e-01,
 6.63762018e-02, 2.28535295e-01, 9.19221416e-02, 4.91625637e-01,
 1.38091639e-01, 2.40258530e-01, 3.48894536e-01, 3.13991487e-01,
 3.45467627e-01, 4.39981073e-01, 3.80849808e-01, 1.88644975e-01,
 3.14365119e-01, 1.42126426e-01, 4.48983349e-03, 2.27064118e-01,
 4.27701503e-01, 3.09979171e-02, 8.45943317e-02, 1.74587190e-01,
 1.14005566e-01, 1.61222577e-01, 1.72635193e+01, 8.08212757e+00,
 7.63679457e+00, 5.70920897e+00, 4.64085054e+00, 2.15698242e+00,
 2.88747096e+00, 9.73084986e-01, 1.26382422e+00, 1.73940384e+00,
 1.70256710e+00, 7.55769193e-01, 3.69001746e-01, 1.31812871e+00,
 3.17754626e-01, 1.11846077e+00, 4.34098512e-01, 4.93405133e-01,
 4.02087629e-01, 3.80952597e-01, 3.44380707e-01, 3.39705229e-01,
 3.10606420e-01, 3.31500679e-01, 1.74861029e-01, 4.30903107e-01,
 1.60815179e-01, 2.22083375e-01, 7.93077275e-02, 2.75829047e-01,
 2.25031257e-01, 2.22507313e-01, 2.43299738e-01, 4.99932887e-03,
 2.05400065e-01, 6.76057190e-02, 5.10083921e-02, 1.19410694e-01,
 1.67144284e-01, 2.21941352e-01, 6.83116093e-02, 5.17660230e-02,
 8.99073929e-02, 1.07434034e-01, 2.46936046e-02, 2.19789922e-01,
 3.29356529e-02, 1.77779362e-01, 1.45397037e-01, 9.43156332e-03,
 1.65113716e+01, 7.38908291e+00, 8.00753403e+00, 3.13403535e+00,
 4.88714552e+00, 2.51596165e+00, 1.21750808e+00, 1.48380280e+00,
 1.30205464e+00, 8.65472734e-01, 9.05965567e-01, 1.32322228e+00,
 5.95164180e-01, 6.46503925e-01, 7.98051119e-01, 6.79475725e-01,
 2.81944871e-01, 4.27944630e-01, 1.98033690e-01, 5.65043569e-01,
 6.43512964e-01, 3.47882718e-01, 2.18735427e-01, 8.36467505e-01,
 1.52345344e-01, 6.46168664e-02, 3.33334476e-01, 2.16207981e-01,
 5.22091031e-01, 1.04628421e-01, 1.10038228e-01, 5.30211866e-01,
 1.02813415e-01, 1.89061537e-02, 1.54395223e-01, 2.26583317e-01,
 1.81382552e-01, 2.76523381e-01, 3.26830298e-01, 2.08391890e-01,
 5.19049503e-02, 3.30849379e-01, 1.75859004e-01, 1.02461278e-01,
 1.23276420e-01, 4.13187519e-02, 2.39867628e-01, 3.88980716e-01,
 5.85266531e-01, 2.46180966e-02, 1.64082680e+01, 7.57170200e+00,
 5.93296909e+00, 4.40094805e+00, 6.33999348e+00, 2.58836865e+00,
 1.17936361e+00, 1.90062547e+00, 1.75410450e+00, 1.49710488e+00,
 1.37411416e+00, 1.36259615e+00, 4.42275465e-01, 6.84846878e-01,
 5.68554640e-01, 2.57916123e-01, 5.17606854e-01, 6.57231450e-01,
 4.08758670e-01, 3.61874998e-01, 3.08928668e-01, 2.44956240e-01,
 3.00695390e-01, 4.68946040e-01, 1.97995156e-01, 7.36775696e-02,
 8.44841525e-02, 4.67119008e-01, 2.26890564e-01, 3.39379340e-01,
 5.92347801e-01, 1.55777961e-01, 1.13882765e-01, 1.83190450e-01,
 3.48159522e-01, 4.10579175e-01, 2.44985119e-01, 8.42879340e-03,
 9.17000100e-02, 1.07945748e-01, 3.08213741e-01, 1.86003760e-01,
 4.34812158e-01, 1.54506236e-01, 2.23271206e-01, 1.85675733e-02,
 2.04980776e-01, 2.75028378e-01, 3.71010005e-01, 5.92988580e-02,
 1.68061047e+01, 8.64034843e+00, 7.84587622e+00, 3.36362171e+00,
 5.34575701e+00, 3.63882709e+00, 2.28851056e+00, 1.08230507e+00,
 9.37759101e-01, 1.01312768e+00, 1.11278903e+00, 6.83694601e-01,
 6.88834608e-01, 7.11833954e-01, 7.34040499e-01, 3.16647261e-01,
 4.96850014e-01, 3.55678350e-01, 4.22447234e-01, 3.90920699e-01,
 2.57310569e-01, 5.93487918e-01, 7.07791984e-01, 2.11937234e-01,
 4.14170802e-01, 4.05430108e-01, 1.90124646e-01, 4.35797304e-01,
 2.38627657e-01, 7.64419958e-02, 8.92417505e-02, 1.59095854e-01,
 1.83127508e-01, 2.61918962e-01, 2.61306256e-01, 1.02260873e-01,
 1.34456500e-01, 6.71050549e-02, 3.61649573e-01, 1.24677591e-01,
 4.78008715e-03, 1.66804474e-02, 8.58087614e-02, 2.30331972e-01,
 2.73000866e-01, 1.08253755e-01, 1.40805915e-01, 1.67160258e-02,
 1.60035025e-02, 3.51244584e-02])
```

```
In [28]: mean_train_loss=np.mean(train_loss)
mean_train_loss
```

```
Out[28]: 1.2763155761882663
```

```
In [29]: val_loss
```

```
Out[29]: array([[153.54016113, 55.70801544, 19.16837502, 6.39042139,
 5.82477427, 7.98298073, 10.14249802, 4.65828228,
10.66814709, 12.60039902, 25.20410156, 9.21026993,
23.38103104, 12.59810829, 20.27552223, 7.16247845,
10.60472775, 7.29560328, 9.47160244, 8.71938705,
8.28960133, 19.21827888, 12.86996841, 8.55958748,
18.33307838, 7.52307796, 6.35692215, 6.89080572,
5.79446793, 8.89416885, 9.24718761, 11.95727444,
17.58582115, 38.73047256, 21.6259613 , 11.14853096,
10.00309658, 9.72321892, 7.13947105, 14.32043934,
13.43727589, 8.56797123, 11.17147827, 9.12106514,
16.26109505, 11.21669388, 9.07446957, 16.17450142,
18.36318588, 10.72614384, 171.57029724, 29.09004402,
13.20075607, 20.14474297, 23.19255066, 6.65774965,
3.10536385, 5.83740187, 4.8640666 , 17.53942108,
4.4922967 , 6.12019539, 5.0065093 , 5.5067873 ,
4.47045231, 5.66999197, 4.71460867, 3.73795962,
4.25027895, 3.91816425, 8.57417583, 4.56618547,
5.02399063, 7.2425456 , 5.59167862, 6.71085262,
4.14545059, 7.04081917, 7.24420404, 5.61866856,
6.6003089 , 5.88995171, 5.07297659, 5.43002081,
6.85182381, 5.53919792, 8.68419933, 12.98806572,
8.57916641, 14.1003809 , 7.16374922, 14.23286533,
11.30220127, 15.25160313, 13.47504139, 5.97625494,
6.4526825 , 6.10128546, 6.19452858, 5.64203882,
79.36199951, 66.83113861, 19.28015518, 5.32647753,
11.59071636, 5.84959364, 4.39410257, 8.02045727,
8.06939125, 7.90744829, 3.40456223, 7.63743496,
8.98768902, 7.68977118, 6.14253139, 3.48782516,
7.44427061, 3.66739917, 4.76471281, 5.94974041,
7.2354641 , 6.02715492, 13.87805939, 7.93997478,
5.43612576, 5.94505501, 8.12081814, 7.84245729,
7.57592201, 5.93043947, 19.51704407, 3.93292689,
11.10476494, 15.48176193, 13.12343311, 9.12312889,
13.49632454, 9.49261951, 6.76086664, 6.23600578,
7.0246911 , 11.99772167, 16.6240921 , 17.63868523,
19.75691795, 18.28801346, 32.84589005, 59.60928726,
7.16886663, 9.04643726, 111.70879364, 33.11818695,
11.69917107, 12.09919071, 22.1178875 , 5.73284817,
8.21843624, 7.24278784, 7.92156267, 4.85751915,
6.223701 , 9.06967258, 7.23220873, 16.0548687 ,
6.84468937, 7.73346806, 12.36810875, 8.17343712,
6.47341013, 8.93990326, 17.78100204, 8.90176487,
6.09923792, 4.71289349, 9.58065987, 7.84739161,
11.89021397, 6.80913115, 9.92665195, 9.99639225,
11.38365555, 9.25769043, 8.31919956, 22.62007141,
31.97268105, 30.24036026, 26.66275215, 27.1071701 ,
66.82551575, 14.35342789, 7.08117485, 8.02213955,
20.24476814, 28.47348404, 30.14750099, 25.03346825,
9.04101181, 14.49799252, 12.99283218, 9.9788866 ,
100.57019806, 63.29291534, 12.73159122, 42.32344055,
9.17635441, 6.87893915, 4.07479811, 10.08311653,
11.57548237, 4.70801497, 7.17337942, 11.34183979,
13.1918993 , 7.49126816, 9.07042694, 8.59011936,
9.91007042, 5.44595575, 9.82749462, 13.00856113,
11.91942883, 6.80544806, 7.30679846, 7.4345336 ,
11.31587696, 8.48287201, 7.76342154, 12.05768013,
6.13223553, 6.2924571 , 8.7189064 , 11.77698803,
10.43774223, 9.39993477, 9.32359409, 12.31500435,
14.3578577 , 16.68113899, 6.70155287, 6.84491062,
7.97642183, 11.03070068, 14.34405994, 13.49910927,
17.65910149, 15.5466671 , 21.26820374, 20.06152725,
19.16956329, 22.97093582]])
```

```
In [30]: mean_val_loss=np.mean(val_loss)
mean_val_loss
```

```
Out[30]: 14.292034516334533
```

## Plot to Visualize the Number of Images in Each Label of Trainig Dataset

```
In [31]: l = []
for i in train:
    if(i[1] == 0):
        l.append("1_normal")

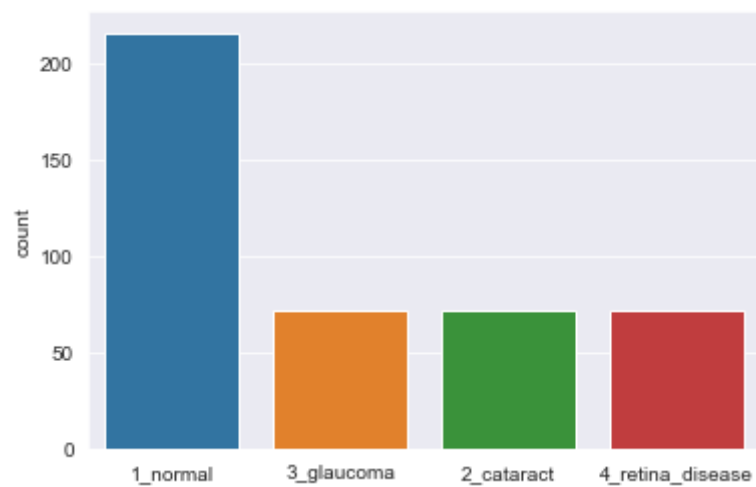
    elif (i[1] == 1):
        l.append("2_cataract")

    elif (i[1] == 2):
        l.append("3_glaucoma")

    else :
        l.append("4_retina_disease")

sns.set_style('darkgrid')
sns.countplot(l)
```

Out[31]: <matplotlib.axes.\_subplots.AxesSubplot at 0x1820e43a520>



## Plot to Visualize the Number of Images in Each Label of Test Dataset.

```
In [32]: l = []
for i in test:
    if(i[1] == 0):
        l.append("1_normal")

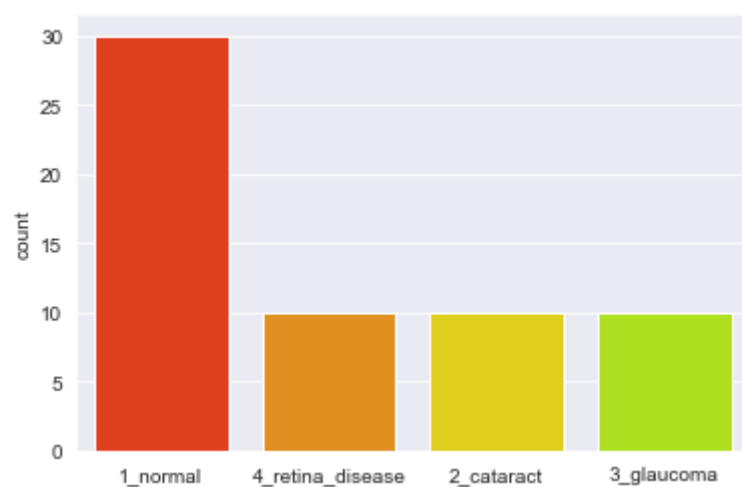
    elif (i[1] == 1):
        l.append("2_cataract")

    elif (i[1] == 2):
        l.append("3_glaucoma")

    else :
        l.append("4_retina_disease")

sns.set_style('darkgrid')
sns.countplot(l,palette='prism')
```

Out[32]: <matplotlib.axes.\_subplots.AxesSubplot at 0x1820f4fce20>



## Plot to Visualize the Number of Images in Each Label of Validation Dataset.

```
In [33]: l = []
for i in validation:
    if(i[1] == 0):
        l.append("1_normal")

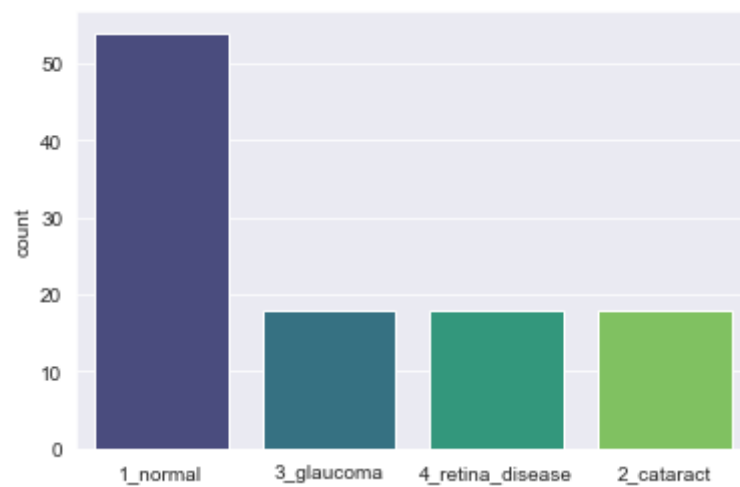
    elif (i[1] == 1):
        l.append("2_cataract")

    elif (i[1] == 2):
        l.append("3_glaucoma")

    else :
        l.append("4_retina_disease")

sns.set_style('darkgrid')
sns.countplot(l,palette='viridis')
```

Out[33]: <matplotlib.axes.\_subplots.AxesSubplot at 0x182652e5fd0>



## Training, Validation Accuracy and Loss Plot for 50 Epochs

```
In [34]: def plot_print(i,j):
epochs_range = range(50)

plt.figure(figsize=(15, 15))
plt.subplot(2, 2, 1)
plt.plot(epochs_range, train_acc[i:j], label='Training Accuracy')
plt.plot(epochs_range, val_acc[i:j], label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(2, 2, 2)
plt.plot(epochs_range, train_loss[i:j], label='Training Loss')
plt.plot(epochs_range, val_loss[i:j], label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')

return plt.show()
```

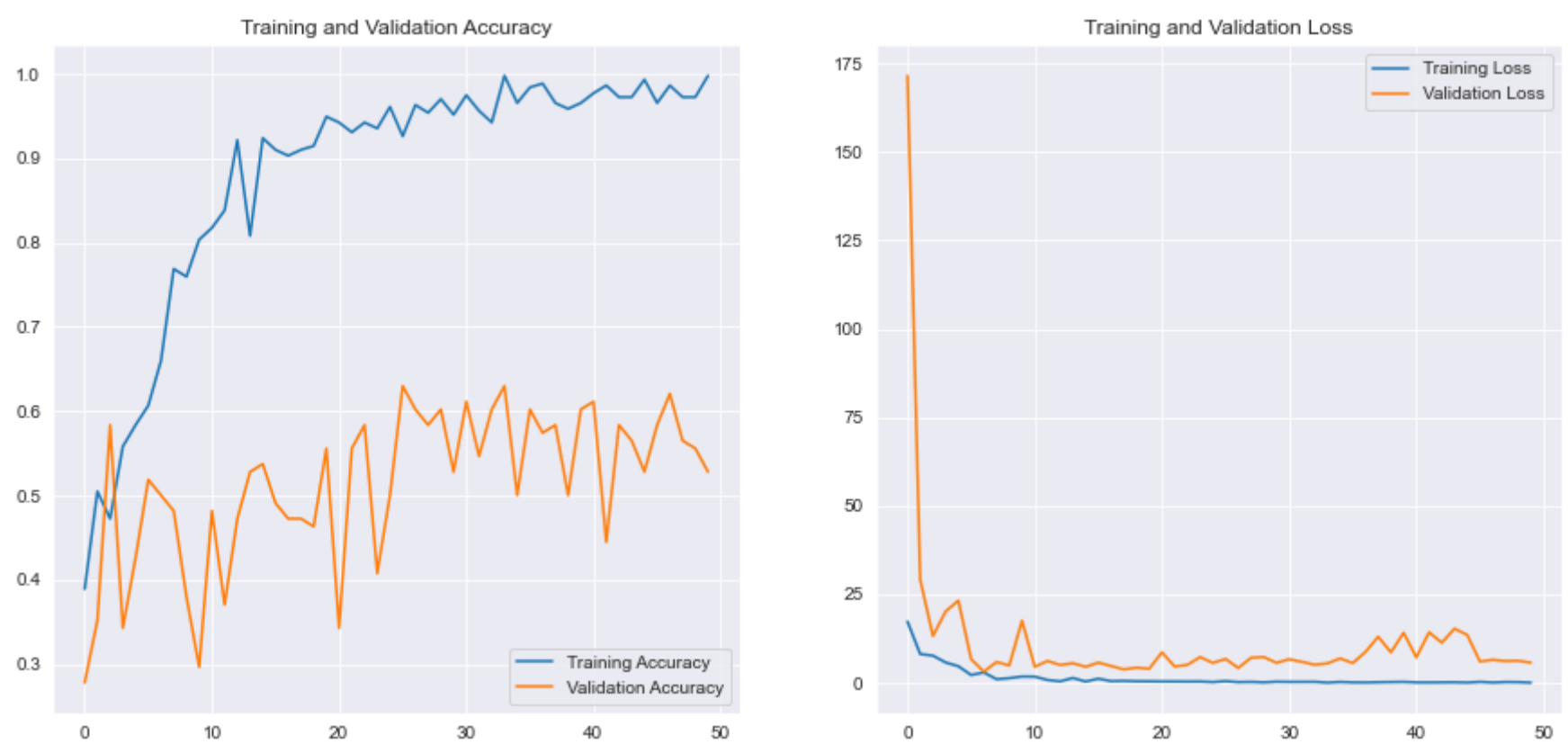
```
In [35]: k=1
j=0
for i in range(0,250,50):
    j +=50
    print('Plot for ',k,'cross validation accuracy and loss for Training and Validation phase')
    k +=1
    plot_print(i,j)
```



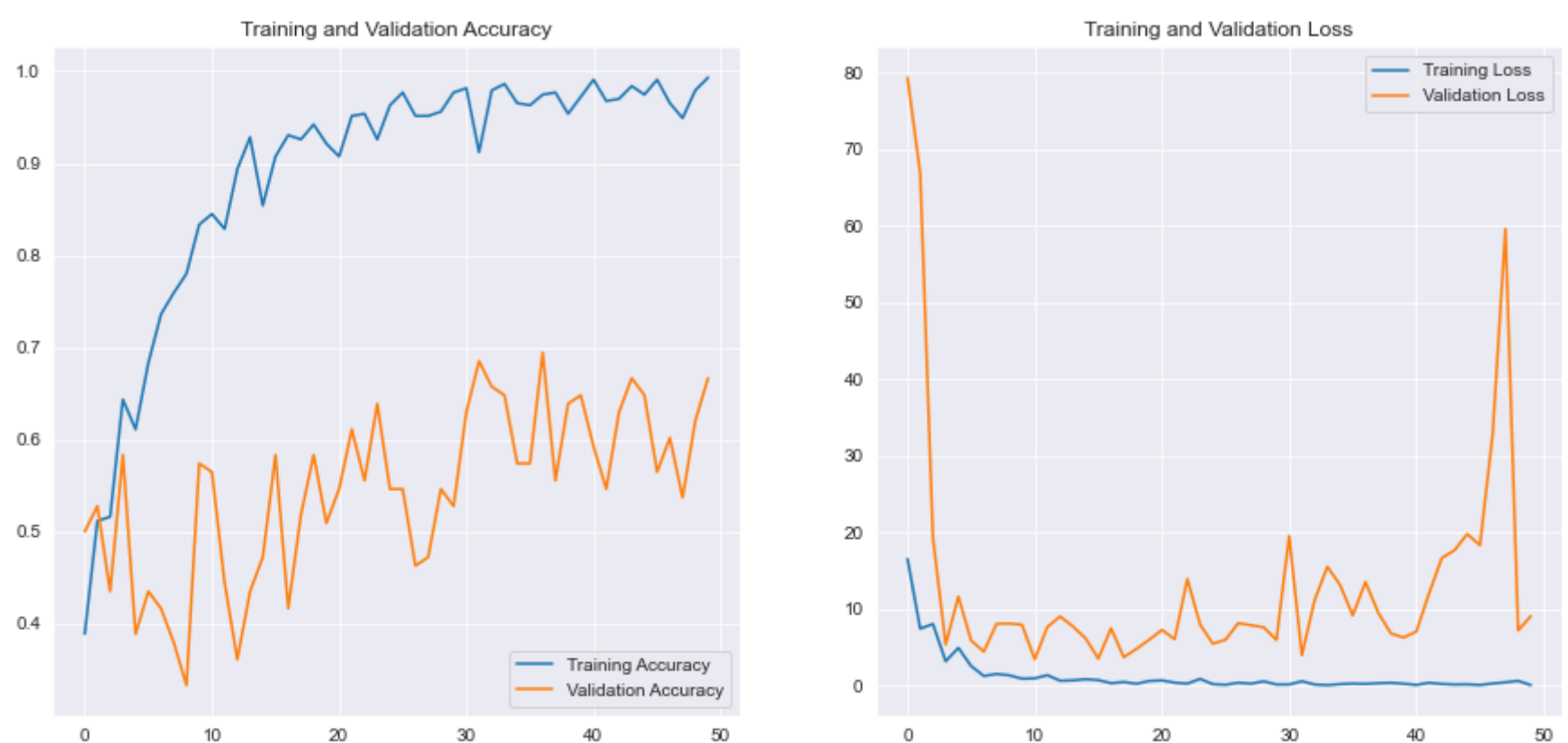
Plot for 1 cross validation accuracy and loss for Training and Validation phase



Plot for 2 cross validation accuracy and loss for Training and Validation phase

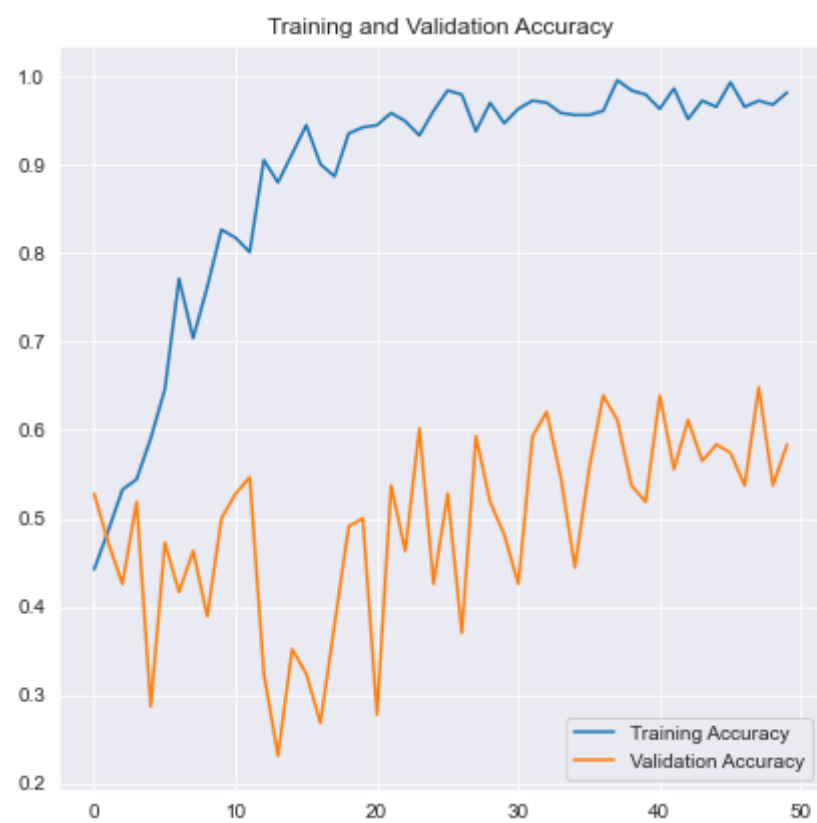


Plot for 3 cross validation accuracy and loss for Training and Validation phase

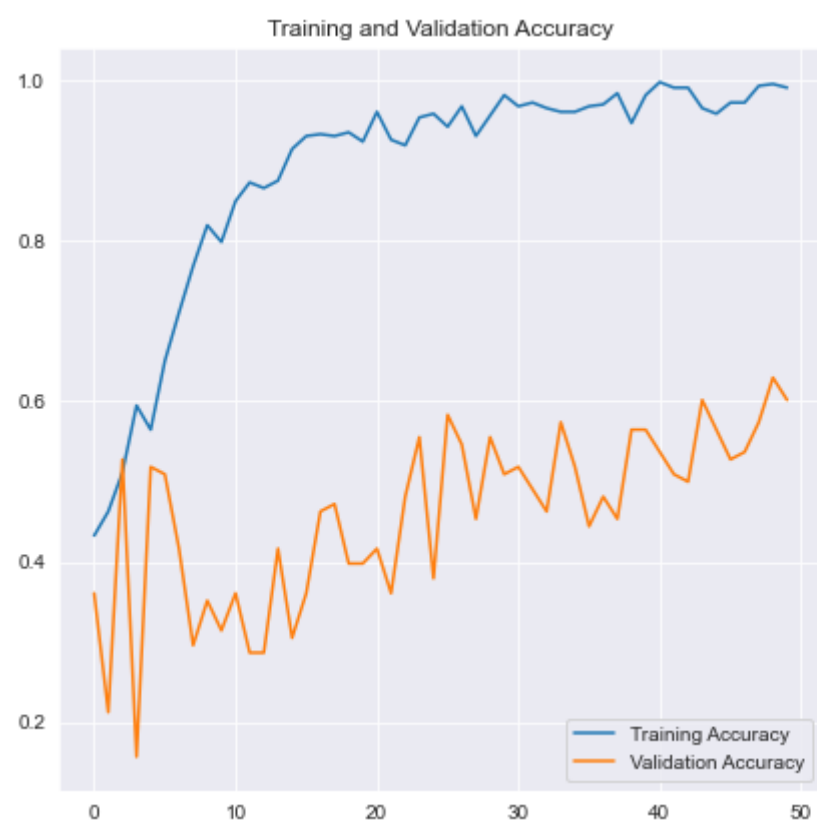


Plot for 4 cross validation accuracy and loss for Training and Validation phase





Plot for 5 cross validation accuracy and loss for Training and Validation phase



## Visualizing Confusion Matrix for Each Fold

```
In [36]: CM= np.array(CM)
          CM.resize(5,4,4)
```

```
In [37]: def confusionmatrix_vis(i):

          yticklabels=['1_normal', '2_cataract', '3_glaucoma', '4_retina_disease']
          xticklabels=['1_normal', '2_cataract', '3_glaucoma', '4_retina_disease']
          plt.figure(figsize=(8, 8))
          hm =sns.heatmap(CM[i], annot=True,annot_kws={"size": 20}, cbar=False,cmap="YlGnBu",yticklabels=yticklabels,xti
          cklabels=xticklabels)

          hm.set_xticklabels(hm.get_xticklabels(), rotation=0, fontsize = 12, )
          hm.set_yticklabels(hm.get_yticklabels(), rotation=0, fontsize = 12)

          plt.ylabel("Actual", fontsize = 18)
          plt.xlabel("Predicted",fontsize = 18)

          return plt.show()
```

```
In [38]: k=1
for i in range(5):
    print('Confusion Matrix for ',k,'Cross Validation Test phase')
    k +=1
    confusionmatrix_vis(i)
```

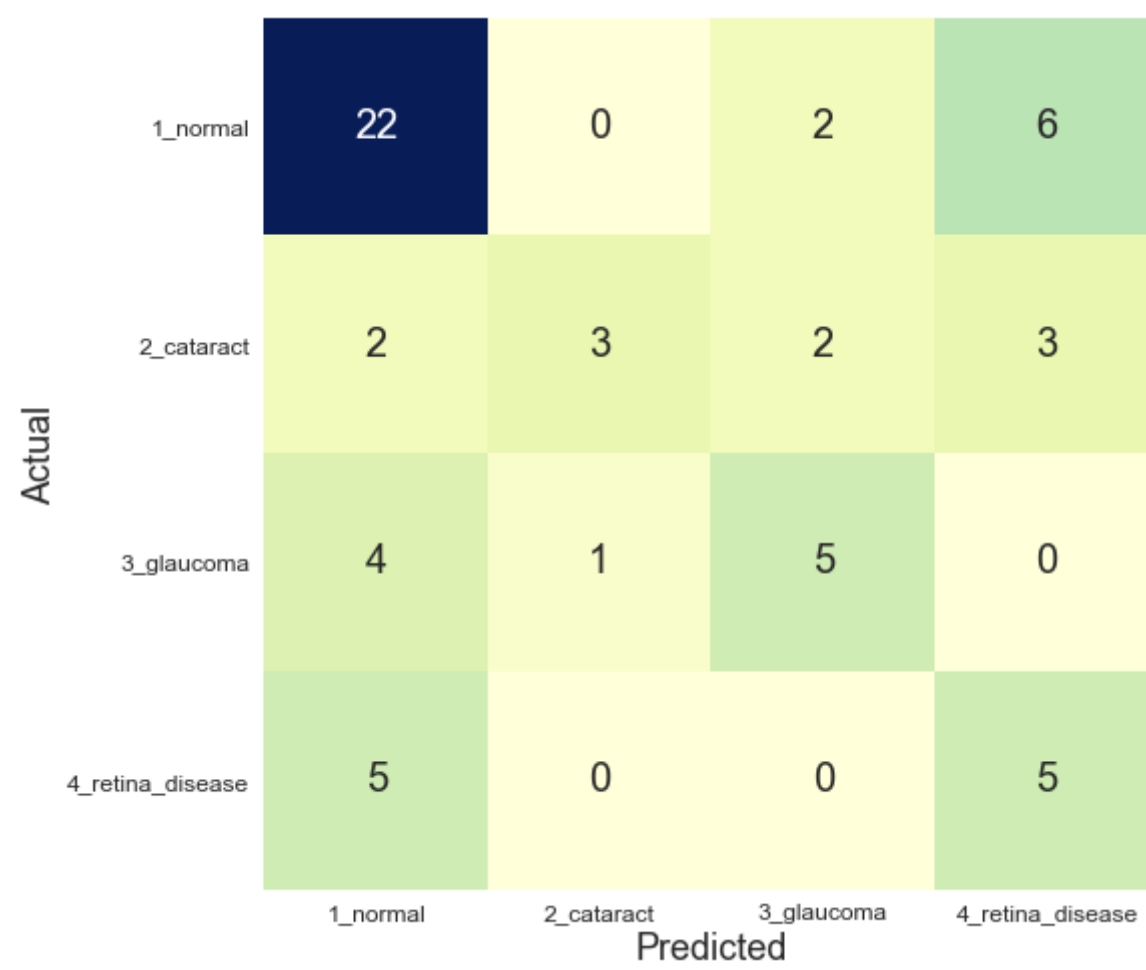
Confusion Matrix for 1 Cross Validation Test phase

Actual	1_normal	18	2	7	3
	2_cataract	1	8	0	1
	3_glaucoma	1	4	5	0
	4_retina_disease	1	3	3	3
		1_normal	2_cataract	3_glaucoma	4_retina_disease
		Predicted			

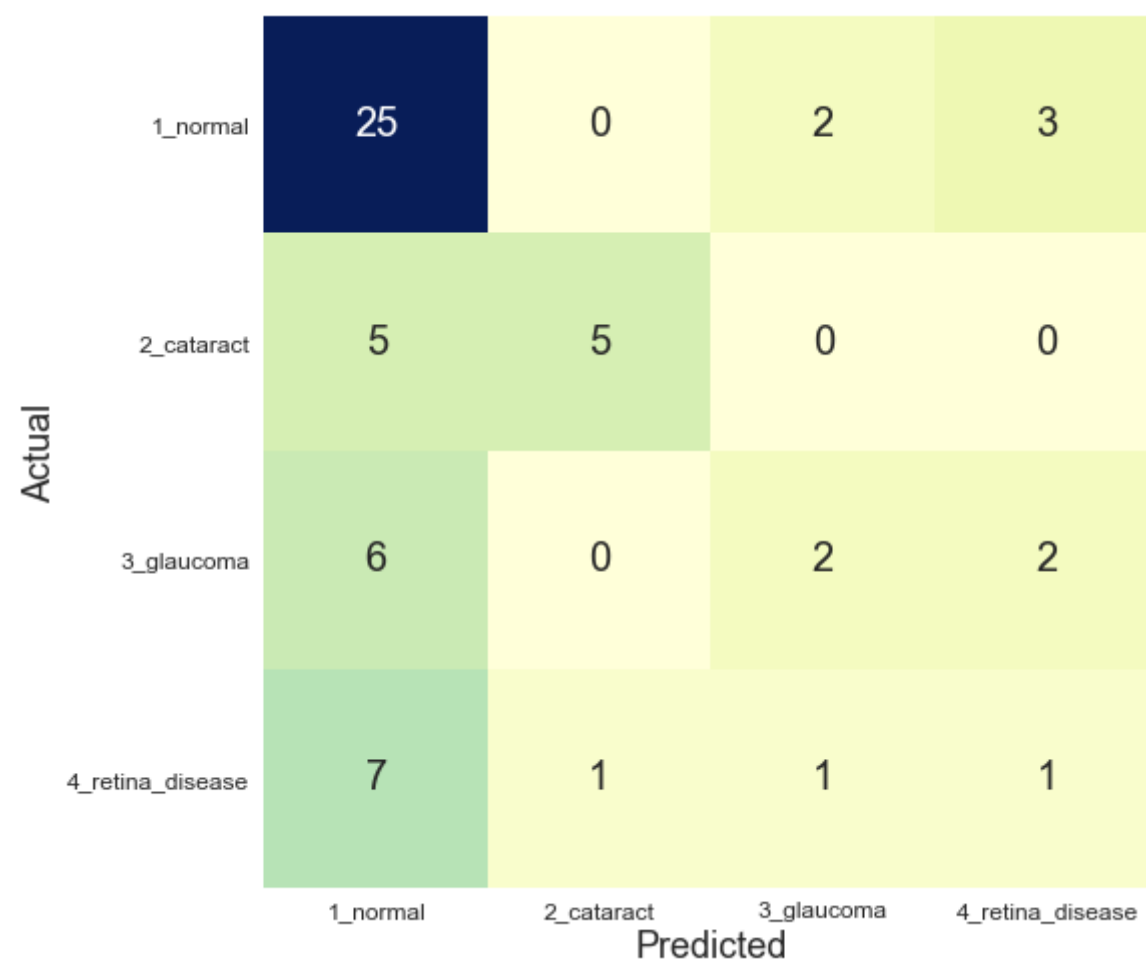
Confusion Matrix for 2 Cross Validation Test phase

Actual	1_normal	18	0	2	10
	2_cataract	1	6	0	3
	3_glaucoma	4	0	3	3
	4_retina_disease	2	1	1	6
		1_normal	2_cataract	3_glaucoma	4_retina_disease
		Predicted			

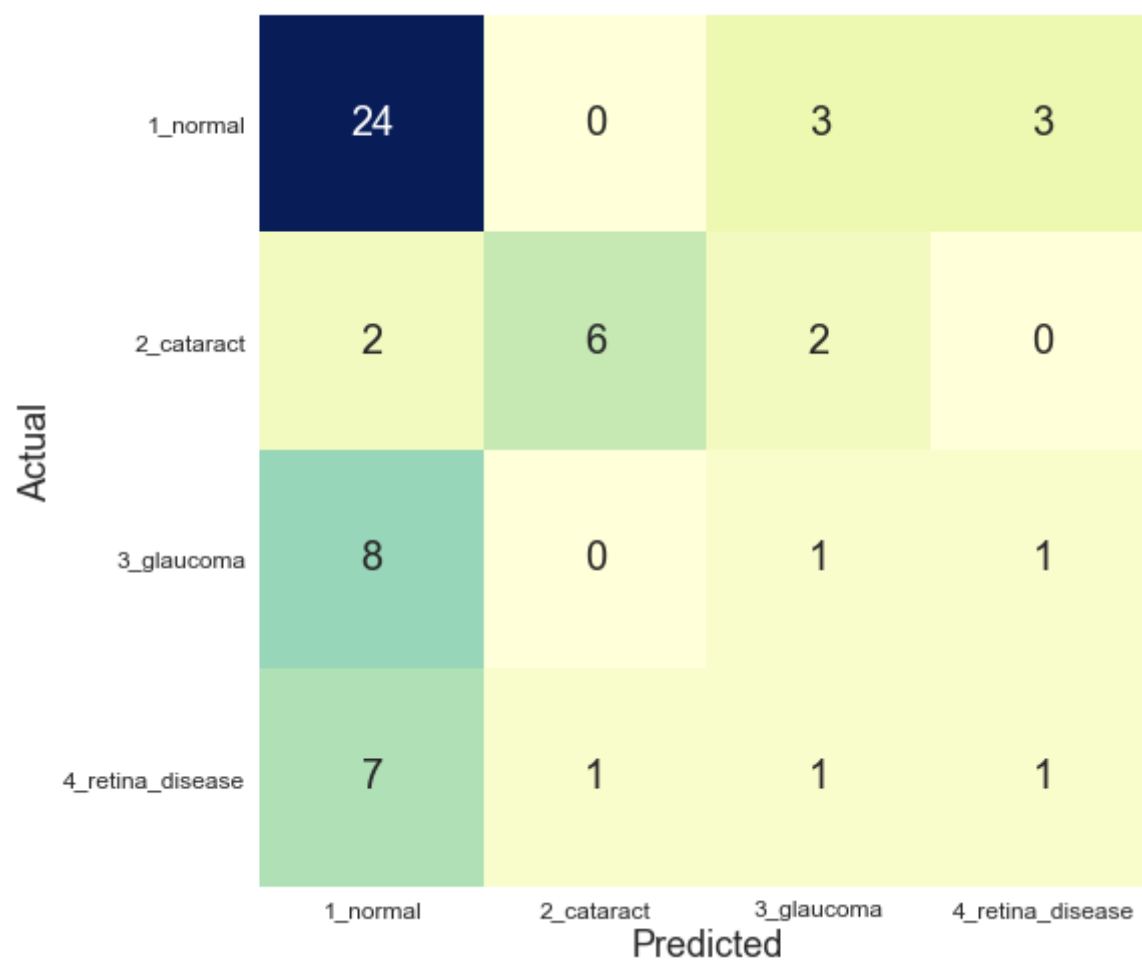
Confusion Matrix for 3 Cross Validation Test phase



Confusion Matrix for 4 Cross Validation Test phase



Confusion Matrix for 5 Cross Validation Test phase



## Visualizing Summarized Confusion Matrix of all 5 folds

```
In [39]: CM_sum = CM[0]+CM[1]+CM[2]+CM[3]+CM[4]
          CM_sum
```

```
Out[39]: array([[107,  2, 16, 25],
                [ 11, 28,  4,  7],
                [ 23,  5, 16,  6],
                [ 22,  6,  6, 16]], dtype=int64)
```

```
In [40]: yticklabels=['1_normal', '2_cataract','3_glaucoma','4_retina_disease']
          xticklabels=['1_normal', '2_cataract','3_glaucoma','4_retina_disease']
          plt.figure(figsize=(8, 8))
          hm =sns.heatmap(CM_sum, annot=True,annot_kws={"size": 20},fmt='g', cbar=False,cmap="YlGnBu",yticklabels=yticklabels,xt
          icklabels=xticklabels)

          hm.set_xticklabels(hm.get_xticklabels(), rotation=0, fontsize = 12, )
          hm.set_yticklabels(hm.get_yticklabels(), rotation=0, fontsize = 12)

          plt.ylabel("Actual", fontsize = 18)
          plt.xlabel("Predicted",fontsize = 18)

          plt.show()
```



## Reconfirming the values of Accuracy,Sensitivity and Specificity

```
In [41]: sensitivity_1_normal = (CM_sum[0,0])/(CM_sum[0,0]+CM_sum[0,1]+CM_sum[0,2]+CM_sum[0,3])
#print('Sensitivity_1_normal      : ', sensitivity_1_normal )

sensitivity_2_cataract = (CM_sum[1,1])/(CM_sum[1,0]+CM_sum[1,1]+CM_sum[1,2]+CM_sum[1,3])
#print('Sensitivity_2_cataract    : ', sensitivity_2_cataract )

sensitivity_3_glaucoma = (CM_sum[2,2])/(CM_sum[2,0]+CM_sum[2,1]+CM_sum[2,2]+CM_sum[2,3])
#print('Sensitivity_3_glaucoma   : ', sensitivity_3_glaucoma )

sensitivity_4_retina_disease = (CM_sum[3,3])/(CM_sum[3,0]+CM_sum[3,1]+CM_sum[3,2]+CM_sum[3,3])
#print('Sensitivity_4_retina_disease : ', sensitivity_4_retina_disease )

specificity_1_normal = (CM_sum[1,1]+CM_sum[1,2]+CM_sum[1,3]+CM_sum[2,1]+CM_sum[2,2]+CM_sum[2,3]+CM_sum[3,1]+CM_sum
[3,2]+CM_sum[3,3])/(CM_sum[1,0]+CM_sum[2,0]+CM_sum[3,0]+CM_sum[1,1]+CM_sum[1,2]+CM_sum[1,3]+CM_sum[2,1]+CM_sum[2,2]+CM
_sum[2,3]+CM_sum[3,1]+CM_sum[3,2]+CM_sum[3,3])
#print('Specificity : ', specificity_1_normal)

specificity_2_cataract = (CM_sum[0,0]+CM_sum[0,2]+CM_sum[0,3]+CM_sum[2,0]+CM_sum[2,2]+CM_sum[2,3]+CM_sum[3,0]+CM_s
um[3,2]+CM_sum[3,3])/(CM_sum[0,1]+CM_sum[2,1]+CM_sum[3,1]+CM_sum[0,0]+CM_sum[0,2]+CM_sum[0,3]+CM_sum[2,0]+CM_sum[2,2]+
CM_sum[2,3]+CM_sum[3,0]+CM_sum[3,2]+CM_sum[3,3])
#print('Specificity : ', specificity_2_cataract)

specificity_3_glaucoma = (CM_sum[0,0]+CM_sum[0,1]+CM_sum[0,3]+CM_sum[1,0]+CM_sum[1,1]+CM_sum[1,3]+CM_sum[3,0]+CM_s
um[3,1]+CM_sum[3,3])/(CM_sum[0,2]+CM_sum[1,2]+CM_sum[3,2]+CM_sum[0,0]+CM_sum[0,1]+CM_sum[0,3]+CM_sum[1,0]+CM_sum[1,1]+
CM_sum[1,3]+CM_sum[3,0]+CM_sum[3,1]+CM_sum[3,3])
#print('Specificity : ', specificity_3_glaucoma)

specificity_4_retina_disease= (CM_sum[0,0]+CM_sum[0,1]+CM_sum[0,2]+CM_sum[1,0]+CM_sum[1,1]+CM_sum[1,2]+CM_sum[2,0]
+CM_sum[2,1]+CM_sum[2,2])/(CM_sum[0,3]+CM_sum[1,3]+CM_sum[2,3]+CM_sum[0,0]+CM_sum[0,1]+CM_sum[0,2]+CM_sum[1,0]+CM_sum[
1,1]+CM_sum[1,2]+CM_sum[2,0]+CM_sum[2,1]+CM_sum[2,2])
#print('Specificity : ', specificity_4_retina_disease)

Sensitivity= (sensitivity_1_normal + sensitivity_2_cataract + sensitivity_3_glaucoma + sensitivity_4_retina_diseas
e)/4
#print(Sensitivity)

Specificity= (specificity_1_normal + specificity_2_cataract + specificity_3_glaucoma + specificity_4_retina_diseas
e)/4
#print(Specificity)

total1=sum(sum(CM_sum))
test_accuracy=(CM_sum[0,0]+CM_sum[1,1]+CM_sum[2,2]+CM_sum[3,3])/total1

print ('Accuracy      : ', test_accuracy)
print ('Specificity   : ', Specificity)
print ('Sensitivity   : ', Sensitivity)
```

```
Accuracy      :  0.5566666666666666
Specificity   :  0.7709410517738597
Sensitivity   :  0.4783333333333334
```

## Model Summary

```
In [42]: model_build_compile(k)
```

```
model building and compiling for fold 7
```

```
Out[42]: <tensorflow.python.keras.engine.functional.Functional at 0x182c8924d00>
```

In [43]: `model.summary()`

Model: "model\_4"

Layer (type)	Output Shape	Param #	Connected to
=====			
input_5 (InputLayer)	[(None, 224, 224, 3)]	0	
conv2d_812 (Conv2D)	(None, 111, 111, 32)	864	input_5[0][0]
batch_normalization_824 (Batch Normalization)	(None, 111, 111, 32)	96	conv2d_812[0][0]
activation_812 (Activation)	(None, 111, 111, 32)	0	batch_normalization_824[0][0]
conv2d_813 (Conv2D)	(None, 109, 109, 32)	9216	activation_812[0][0]
batch_normalization_825 (Batch Normalization)	(None, 109, 109, 32)	96	conv2d_813[0][0]
activation_813 (Activation)	(None, 109, 109, 32)	0	batch_normalization_825[0][0]
conv2d_814 (Conv2D)	(None, 109, 109, 64)	18432	activation_813[0][0]
batch_normalization_826 (Batch Normalization)	(None, 109, 109, 64)	192	conv2d_814[0][0]
activation_814 (Activation)	(None, 109, 109, 64)	0	batch_normalization_826[0][0]
max_pooling2d_16 (MaxPooling2D)	(None, 54, 54, 64)	0	activation_814[0][0]
conv2d_815 (Conv2D)	(None, 54, 54, 80)	5120	max_pooling2d_16[0][0]
batch_normalization_827 (Batch Normalization)	(None, 54, 54, 80)	240	conv2d_815[0][0]
activation_815 (Activation)	(None, 54, 54, 80)	0	batch_normalization_827[0][0]
conv2d_816 (Conv2D)	(None, 52, 52, 192)	138240	activation_815[0][0]
batch_normalization_828 (Batch Normalization)	(None, 52, 52, 192)	576	conv2d_816[0][0]
activation_816 (Activation)	(None, 52, 52, 192)	0	batch_normalization_828[0][0]
max_pooling2d_17 (MaxPooling2D)	(None, 25, 25, 192)	0	activation_816[0][0]
conv2d_820 (Conv2D)	(None, 25, 25, 64)	12288	max_pooling2d_17[0][0]
batch_normalization_832 (Batch Normalization)	(None, 25, 25, 64)	192	conv2d_820[0][0]
activation_820 (Activation)	(None, 25, 25, 64)	0	batch_normalization_832[0][0]
conv2d_818 (Conv2D)	(None, 25, 25, 48)	9216	max_pooling2d_17[0][0]
conv2d_821 (Conv2D)	(None, 25, 25, 96)	55296	activation_820[0][0]
batch_normalization_830 (Batch Normalization)	(None, 25, 25, 48)	144	conv2d_818[0][0]
batch_normalization_833 (Batch Normalization)	(None, 25, 25, 96)	288	conv2d_821[0][0]
activation_818 (Activation)	(None, 25, 25, 48)	0	batch_normalization_830[0][0]
activation_821 (Activation)	(None, 25, 25, 96)	0	batch_normalization_833[0][0]
average_pooling2d_4 (AveragePooling2D)	(None, 25, 25, 192)	0	max_pooling2d_17[0][0]
conv2d_817 (Conv2D)	(None, 25, 25, 96)	18432	max_pooling2d_17[0][0]
conv2d_819 (Conv2D)	(None, 25, 25, 64)	76800	activation_818[0][0]
conv2d_822 (Conv2D)	(None, 25, 25, 96)	82944	activation_821[0][0]
conv2d_823 (Conv2D)	(None, 25, 25, 64)	12288	average_pooling2d_4[0][0]
batch_normalization_829 (Batch Normalization)	(None, 25, 25, 96)	288	conv2d_817[0][0]
batch_normalization_831 (Batch Normalization)	(None, 25, 25, 64)	192	conv2d_819[0][0]
batch_normalization_834 (Batch Normalization)	(None, 25, 25, 96)	288	conv2d_822[0][0]
batch_normalization_835 (Batch Normalization)	(None, 25, 25, 64)	192	conv2d_823[0][0]
activation_817 (Activation)	(None, 25, 25, 96)	0	batch_normalization_829[0][0]
activation_819 (Activation)	(None, 25, 25, 64)	0	batch_normalization_831[0][0]
activation_822 (Activation)	(None, 25, 25, 96)	0	batch_normalization_834[0][0]
activation_823 (Activation)	(None, 25, 25, 64)	0	batch_normalization_835[0][0]
mixed_5b (Concatenate)	(None, 25, 25, 320)	0	activation_817[0][0] activation_819[0][0] activation_822[0][0] activation_823[0][0]



conv2d_827 (Conv2D)	(None, 25, 25, 32)	10240	mixed_5b[0][0]
batch_normalization_839 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_827[0][0]
activation_827 (Activation)	(None, 25, 25, 32)	0	batch_normalization_839[0][0]
conv2d_825 (Conv2D)	(None, 25, 25, 32)	10240	mixed_5b[0][0]
conv2d_828 (Conv2D)	(None, 25, 25, 48)	13824	activation_827[0][0]
batch_normalization_837 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_825[0][0]
batch_normalization_840 (Batch Normalization)	(None, 25, 25, 48)	144	conv2d_828[0][0]
activation_825 (Activation)	(None, 25, 25, 32)	0	batch_normalization_837[0][0]
activation_828 (Activation)	(None, 25, 25, 48)	0	batch_normalization_840[0][0]
conv2d_824 (Conv2D)	(None, 25, 25, 32)	10240	mixed_5b[0][0]
conv2d_826 (Conv2D)	(None, 25, 25, 32)	9216	activation_825[0][0]
conv2d_829 (Conv2D)	(None, 25, 25, 64)	27648	activation_828[0][0]
batch_normalization_836 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_824[0][0]
batch_normalization_838 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_826[0][0]
batch_normalization_841 (Batch Normalization)	(None, 25, 25, 64)	192	conv2d_829[0][0]
activation_824 (Activation)	(None, 25, 25, 32)	0	batch_normalization_836[0][0]
activation_826 (Activation)	(None, 25, 25, 32)	0	batch_normalization_838[0][0]
activation_829 (Activation)	(None, 25, 25, 64)	0	batch_normalization_841[0][0]
block35_1_mixed (Concatenate)	(None, 25, 25, 128)	0	activation_824[0][0] activation_826[0][0] activation_829[0][0]
block35_1_conv (Conv2D)	(None, 25, 25, 320)	41280	block35_1_mixed[0][0]
block35_1 (Lambda)	(None, 25, 25, 320)	0	mixed_5b[0][0] block35_1_conv[0][0]
block35_1_ac (Activation)	(None, 25, 25, 320)	0	block35_1[0][0]
conv2d_833 (Conv2D)	(None, 25, 25, 32)	10240	block35_1_ac[0][0]
batch_normalization_845 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_833[0][0]
activation_833 (Activation)	(None, 25, 25, 32)	0	batch_normalization_845[0][0]
conv2d_831 (Conv2D)	(None, 25, 25, 32)	10240	block35_1_ac[0][0]
conv2d_834 (Conv2D)	(None, 25, 25, 48)	13824	activation_833[0][0]
batch_normalization_843 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_831[0][0]
batch_normalization_846 (Batch Normalization)	(None, 25, 25, 48)	144	conv2d_834[0][0]
activation_831 (Activation)	(None, 25, 25, 32)	0	batch_normalization_843[0][0]
activation_834 (Activation)	(None, 25, 25, 48)	0	batch_normalization_846[0][0]
conv2d_830 (Conv2D)	(None, 25, 25, 32)	10240	block35_1_ac[0][0]
conv2d_832 (Conv2D)	(None, 25, 25, 32)	9216	activation_831[0][0]
conv2d_835 (Conv2D)	(None, 25, 25, 64)	27648	activation_834[0][0]
batch_normalization_842 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_830[0][0]
batch_normalization_844 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_832[0][0]
batch_normalization_847 (Batch Normalization)	(None, 25, 25, 64)	192	conv2d_835[0][0]
activation_830 (Activation)	(None, 25, 25, 32)	0	batch_normalization_842[0][0]
activation_832 (Activation)	(None, 25, 25, 32)	0	batch_normalization_844[0][0]
activation_835 (Activation)	(None, 25, 25, 64)	0	batch_normalization_847[0][0]
block35_2_mixed (Concatenate)	(None, 25, 25, 128)	0	activation_830[0][0] activation_832[0][0] activation_835[0][0]

block35_2_conv (Conv2D)	(None, 25, 25, 320)	41280	block35_2_mixed[0][0]
block35_2 (Lambda)	(None, 25, 25, 320)	0	block35_1_ac[0][0] block35_2_conv[0][0]
block35_2_ac (Activation)	(None, 25, 25, 320)	0	block35_2[0][0]
conv2d_839 (Conv2D)	(None, 25, 25, 32)	10240	block35_2_ac[0][0]
batch_normalization_851 (BatchN	(None, 25, 25, 32)	96	conv2d_839[0][0]
activation_839 (Activation)	(None, 25, 25, 32)	0	batch_normalization_851[0][0]
conv2d_837 (Conv2D)	(None, 25, 25, 32)	10240	block35_2_ac[0][0]
conv2d_840 (Conv2D)	(None, 25, 25, 48)	13824	activation_839[0][0]
batch_normalization_849 (BatchN	(None, 25, 25, 32)	96	conv2d_837[0][0]
batch_normalization_852 (BatchN	(None, 25, 25, 48)	144	conv2d_840[0][0]
activation_837 (Activation)	(None, 25, 25, 32)	0	batch_normalization_849[0][0]
activation_840 (Activation)	(None, 25, 25, 48)	0	batch_normalization_852[0][0]
conv2d_836 (Conv2D)	(None, 25, 25, 32)	10240	block35_2_ac[0][0]
conv2d_838 (Conv2D)	(None, 25, 25, 32)	9216	activation_837[0][0]
conv2d_841 (Conv2D)	(None, 25, 25, 64)	27648	activation_840[0][0]
batch_normalization_848 (BatchN	(None, 25, 25, 32)	96	conv2d_836[0][0]
batch_normalization_850 (BatchN	(None, 25, 25, 32)	96	conv2d_838[0][0]
batch_normalization_853 (BatchN	(None, 25, 25, 64)	192	conv2d_841[0][0]
activation_836 (Activation)	(None, 25, 25, 32)	0	batch_normalization_848[0][0]
activation_838 (Activation)	(None, 25, 25, 32)	0	batch_normalization_850[0][0]
activation_841 (Activation)	(None, 25, 25, 64)	0	batch_normalization_853[0][0]
block35_3_mixed (Concatenate)	(None, 25, 25, 128)	0	activation_836[0][0] activation_838[0][0] activation_841[0][0]
block35_3_conv (Conv2D)	(None, 25, 25, 320)	41280	block35_3_mixed[0][0]
block35_3 (Lambda)	(None, 25, 25, 320)	0	block35_2_ac[0][0] block35_3_conv[0][0]
block35_3_ac (Activation)	(None, 25, 25, 320)	0	block35_3[0][0]
conv2d_845 (Conv2D)	(None, 25, 25, 32)	10240	block35_3_ac[0][0]
batch_normalization_857 (BatchN	(None, 25, 25, 32)	96	conv2d_845[0][0]
activation_845 (Activation)	(None, 25, 25, 32)	0	batch_normalization_857[0][0]
conv2d_843 (Conv2D)	(None, 25, 25, 32)	10240	block35_3_ac[0][0]
conv2d_846 (Conv2D)	(None, 25, 25, 48)	13824	activation_845[0][0]
batch_normalization_855 (BatchN	(None, 25, 25, 32)	96	conv2d_843[0][0]
batch_normalization_858 (BatchN	(None, 25, 25, 48)	144	conv2d_846[0][0]
activation_843 (Activation)	(None, 25, 25, 32)	0	batch_normalization_855[0][0]
activation_846 (Activation)	(None, 25, 25, 48)	0	batch_normalization_858[0][0]
conv2d_842 (Conv2D)	(None, 25, 25, 32)	10240	block35_3_ac[0][0]
conv2d_844 (Conv2D)	(None, 25, 25, 32)	9216	activation_843[0][0]
conv2d_847 (Conv2D)	(None, 25, 25, 64)	27648	activation_846[0][0]
batch_normalization_854 (BatchN	(None, 25, 25, 32)	96	conv2d_842[0][0]
batch_normalization_856 (BatchN	(None, 25, 25, 32)	96	conv2d_844[0][0]
batch_normalization_859 (BatchN	(None, 25, 25, 64)	192	conv2d_847[0][0]
activation_842 (Activation)	(None, 25, 25, 32)	0	batch_normalization_854[0][0]
activation_844 (Activation)	(None, 25, 25, 32)	0	batch_normalization_856[0][0]

activation_847 (Activation)	(None, 25, 25, 64)	0	batch_normalization_859[0][0]
block35_4_mixed (Concatenate)	(None, 25, 25, 128)	0	activation_842[0][0] activation_844[0][0] activation_847[0][0]
block35_4_conv (Conv2D)	(None, 25, 25, 320)	41280	block35_4_mixed[0][0]
block35_4 (Lambda)	(None, 25, 25, 320)	0	block35_3_ac[0][0] block35_4_conv[0][0]
block35_4_ac (Activation)	(None, 25, 25, 320)	0	block35_4[0][0]
conv2d_851 (Conv2D)	(None, 25, 25, 32)	10240	block35_4_ac[0][0]
batch_normalization_863 (BatchN	(None, 25, 25, 32)	96	conv2d_851[0][0]
activation_851 (Activation)	(None, 25, 25, 32)	0	batch_normalization_863[0][0]
conv2d_849 (Conv2D)	(None, 25, 25, 32)	10240	block35_4_ac[0][0]
conv2d_852 (Conv2D)	(None, 25, 25, 48)	13824	activation_851[0][0]
batch_normalization_861 (BatchN	(None, 25, 25, 32)	96	conv2d_849[0][0]
batch_normalization_864 (BatchN	(None, 25, 25, 48)	144	conv2d_852[0][0]
activation_849 (Activation)	(None, 25, 25, 32)	0	batch_normalization_861[0][0]
activation_852 (Activation)	(None, 25, 25, 48)	0	batch_normalization_864[0][0]
conv2d_848 (Conv2D)	(None, 25, 25, 32)	10240	block35_4_ac[0][0]
conv2d_850 (Conv2D)	(None, 25, 25, 32)	9216	activation_849[0][0]
conv2d_853 (Conv2D)	(None, 25, 25, 64)	27648	activation_852[0][0]
batch_normalization_860 (BatchN	(None, 25, 25, 32)	96	conv2d_848[0][0]
batch_normalization_862 (BatchN	(None, 25, 25, 32)	96	conv2d_850[0][0]
batch_normalization_865 (BatchN	(None, 25, 25, 64)	192	conv2d_853[0][0]
activation_848 (Activation)	(None, 25, 25, 32)	0	batch_normalization_860[0][0]
activation_850 (Activation)	(None, 25, 25, 32)	0	batch_normalization_862[0][0]
activation_853 (Activation)	(None, 25, 25, 64)	0	batch_normalization_865[0][0]
block35_5_mixed (Concatenate)	(None, 25, 25, 128)	0	activation_848[0][0] activation_850[0][0] activation_853[0][0]
block35_5_conv (Conv2D)	(None, 25, 25, 320)	41280	block35_5_mixed[0][0]
block35_5 (Lambda)	(None, 25, 25, 320)	0	block35_4_ac[0][0] block35_5_conv[0][0]
block35_5_ac (Activation)	(None, 25, 25, 320)	0	block35_5[0][0]
conv2d_857 (Conv2D)	(None, 25, 25, 32)	10240	block35_5_ac[0][0]
batch_normalization_869 (BatchN	(None, 25, 25, 32)	96	conv2d_857[0][0]
activation_857 (Activation)	(None, 25, 25, 32)	0	batch_normalization_869[0][0]
conv2d_855 (Conv2D)	(None, 25, 25, 32)	10240	block35_5_ac[0][0]
conv2d_858 (Conv2D)	(None, 25, 25, 48)	13824	activation_857[0][0]
batch_normalization_867 (BatchN	(None, 25, 25, 32)	96	conv2d_855[0][0]
batch_normalization_870 (BatchN	(None, 25, 25, 48)	144	conv2d_858[0][0]
activation_855 (Activation)	(None, 25, 25, 32)	0	batch_normalization_867[0][0]
activation_858 (Activation)	(None, 25, 25, 48)	0	batch_normalization_870[0][0]
conv2d_854 (Conv2D)	(None, 25, 25, 32)	10240	block35_5_ac[0][0]
conv2d_856 (Conv2D)	(None, 25, 25, 32)	9216	activation_855[0][0]
conv2d_859 (Conv2D)	(None, 25, 25, 64)	27648	activation_858[0][0]
batch_normalization_866 (BatchN	(None, 25, 25, 32)	96	conv2d_854[0][0]
batch_normalization_868 (BatchN	(None, 25, 25, 32)	96	conv2d_856[0][0]

batch_normalization_871 (BatchN	(None, 25, 25, 64)	192	conv2d_859[0][0]
activation_854 (Activation)	(None, 25, 25, 32)	0	batch_normalization_866[0][0]
activation_856 (Activation)	(None, 25, 25, 32)	0	batch_normalization_868[0][0]
activation_859 (Activation)	(None, 25, 25, 64)	0	batch_normalization_871[0][0]
block35_6_mixed (Concatenate)	(None, 25, 25, 128)	0	activation_854[0][0] activation_856[0][0] activation_859[0][0]
block35_6_conv (Conv2D)	(None, 25, 25, 320)	41280	block35_6_mixed[0][0]
block35_6 (Lambda)	(None, 25, 25, 320)	0	block35_5_ac[0][0] block35_6_conv[0][0]
block35_6_ac (Activation)	(None, 25, 25, 320)	0	block35_6[0][0]
conv2d_863 (Conv2D)	(None, 25, 25, 32)	10240	block35_6_ac[0][0]
batch_normalization_875 (BatchN	(None, 25, 25, 32)	96	conv2d_863[0][0]
activation_863 (Activation)	(None, 25, 25, 32)	0	batch_normalization_875[0][0]
conv2d_861 (Conv2D)	(None, 25, 25, 32)	10240	block35_6_ac[0][0]
conv2d_864 (Conv2D)	(None, 25, 25, 48)	13824	activation_863[0][0]
batch_normalization_873 (BatchN	(None, 25, 25, 32)	96	conv2d_861[0][0]
batch_normalization_876 (BatchN	(None, 25, 25, 48)	144	conv2d_864[0][0]
activation_861 (Activation)	(None, 25, 25, 32)	0	batch_normalization_873[0][0]
activation_864 (Activation)	(None, 25, 25, 48)	0	batch_normalization_876[0][0]
conv2d_860 (Conv2D)	(None, 25, 25, 32)	10240	block35_6_ac[0][0]
conv2d_862 (Conv2D)	(None, 25, 25, 32)	9216	activation_861[0][0]
conv2d_865 (Conv2D)	(None, 25, 25, 64)	27648	activation_864[0][0]
batch_normalization_872 (BatchN	(None, 25, 25, 32)	96	conv2d_860[0][0]
batch_normalization_874 (BatchN	(None, 25, 25, 32)	96	conv2d_862[0][0]
batch_normalization_877 (BatchN	(None, 25, 25, 64)	192	conv2d_865[0][0]
activation_860 (Activation)	(None, 25, 25, 32)	0	batch_normalization_872[0][0]
activation_862 (Activation)	(None, 25, 25, 32)	0	batch_normalization_874[0][0]
activation_865 (Activation)	(None, 25, 25, 64)	0	batch_normalization_877[0][0]
block35_7_mixed (Concatenate)	(None, 25, 25, 128)	0	activation_860[0][0] activation_862[0][0] activation_865[0][0]
block35_7_conv (Conv2D)	(None, 25, 25, 320)	41280	block35_7_mixed[0][0]
block35_7 (Lambda)	(None, 25, 25, 320)	0	block35_6_ac[0][0] block35_7_conv[0][0]
block35_7_ac (Activation)	(None, 25, 25, 320)	0	block35_7[0][0]
conv2d_869 (Conv2D)	(None, 25, 25, 32)	10240	block35_7_ac[0][0]
batch_normalization_881 (BatchN	(None, 25, 25, 32)	96	conv2d_869[0][0]
activation_869 (Activation)	(None, 25, 25, 32)	0	batch_normalization_881[0][0]
conv2d_867 (Conv2D)	(None, 25, 25, 32)	10240	block35_7_ac[0][0]
conv2d_870 (Conv2D)	(None, 25, 25, 48)	13824	activation_869[0][0]
batch_normalization_879 (BatchN	(None, 25, 25, 32)	96	conv2d_867[0][0]
batch_normalization_882 (BatchN	(None, 25, 25, 48)	144	conv2d_870[0][0]
activation_867 (Activation)	(None, 25, 25, 32)	0	batch_normalization_879[0][0]
activation_870 (Activation)	(None, 25, 25, 48)	0	batch_normalization_882[0][0]
conv2d_866 (Conv2D)	(None, 25, 25, 32)	10240	block35_7_ac[0][0]
conv2d_868 (Conv2D)	(None, 25, 25, 32)	9216	activation_867[0][0]

conv2d_871 (Conv2D)	(None, 25, 25, 64)	27648	activation_870[0][0]
batch_normalization_878 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_866[0][0]
batch_normalization_880 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_868[0][0]
batch_normalization_883 (Batch Normalization)	(None, 25, 25, 64)	192	conv2d_871[0][0]
activation_866 (Activation)	(None, 25, 25, 32)	0	batch_normalization_878[0][0]
activation_868 (Activation)	(None, 25, 25, 32)	0	batch_normalization_880[0][0]
activation_871 (Activation)	(None, 25, 25, 64)	0	batch_normalization_883[0][0]
block35_8_mixed (Concatenate)	(None, 25, 25, 128)	0	activation_866[0][0] activation_868[0][0] activation_871[0][0]
block35_8_conv (Conv2D)	(None, 25, 25, 320)	41280	block35_8_mixed[0][0]
block35_8 (Lambda)	(None, 25, 25, 320)	0	block35_7_ac[0][0] block35_8_conv[0][0]
block35_8_ac (Activation)	(None, 25, 25, 320)	0	block35_8[0][0]
conv2d_875 (Conv2D)	(None, 25, 25, 32)	10240	block35_8_ac[0][0]
batch_normalization_887 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_875[0][0]
activation_875 (Activation)	(None, 25, 25, 32)	0	batch_normalization_887[0][0]
conv2d_873 (Conv2D)	(None, 25, 25, 32)	10240	block35_8_ac[0][0]
conv2d_876 (Conv2D)	(None, 25, 25, 48)	13824	activation_875[0][0]
batch_normalization_885 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_873[0][0]
batch_normalization_888 (Batch Normalization)	(None, 25, 25, 48)	144	conv2d_876[0][0]
activation_873 (Activation)	(None, 25, 25, 32)	0	batch_normalization_885[0][0]
activation_876 (Activation)	(None, 25, 25, 48)	0	batch_normalization_888[0][0]
conv2d_872 (Conv2D)	(None, 25, 25, 32)	10240	block35_8_ac[0][0]
conv2d_874 (Conv2D)	(None, 25, 25, 32)	9216	activation_873[0][0]
conv2d_877 (Conv2D)	(None, 25, 25, 64)	27648	activation_876[0][0]
batch_normalization_884 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_872[0][0]
batch_normalization_886 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_874[0][0]
batch_normalization_889 (Batch Normalization)	(None, 25, 25, 64)	192	conv2d_877[0][0]
activation_872 (Activation)	(None, 25, 25, 32)	0	batch_normalization_884[0][0]
activation_874 (Activation)	(None, 25, 25, 32)	0	batch_normalization_886[0][0]
activation_877 (Activation)	(None, 25, 25, 64)	0	batch_normalization_889[0][0]
block35_9_mixed (Concatenate)	(None, 25, 25, 128)	0	activation_872[0][0] activation_874[0][0] activation_877[0][0]
block35_9_conv (Conv2D)	(None, 25, 25, 320)	41280	block35_9_mixed[0][0]
block35_9 (Lambda)	(None, 25, 25, 320)	0	block35_8_ac[0][0] block35_9_conv[0][0]
block35_9_ac (Activation)	(None, 25, 25, 320)	0	block35_9[0][0]
conv2d_881 (Conv2D)	(None, 25, 25, 32)	10240	block35_9_ac[0][0]
batch_normalization_893 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_881[0][0]
activation_881 (Activation)	(None, 25, 25, 32)	0	batch_normalization_893[0][0]
conv2d_879 (Conv2D)	(None, 25, 25, 32)	10240	block35_9_ac[0][0]
conv2d_882 (Conv2D)	(None, 25, 25, 48)	13824	activation_881[0][0]
batch_normalization_891 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_879[0][0]
batch_normalization_894 (Batch Normalization)	(None, 25, 25, 48)	144	conv2d_882[0][0]
activation_879 (Activation)	(None, 25, 25, 32)	0	batch_normalization_891[0][0]

activation_882 (Activation)	(None, 25, 25, 48)	0	batch_normalization_894[0][0]
conv2d_878 (Conv2D)	(None, 25, 25, 32)	10240	block35_9_ac[0][0]
conv2d_880 (Conv2D)	(None, 25, 25, 32)	9216	activation_879[0][0]
conv2d_883 (Conv2D)	(None, 25, 25, 64)	27648	activation_882[0][0]
batch_normalization_890 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_878[0][0]
batch_normalization_892 (Batch Normalization)	(None, 25, 25, 32)	96	conv2d_880[0][0]
batch_normalization_895 (Batch Normalization)	(None, 25, 25, 64)	192	conv2d_883[0][0]
activation_878 (Activation)	(None, 25, 25, 32)	0	batch_normalization_890[0][0]
activation_880 (Activation)	(None, 25, 25, 32)	0	batch_normalization_892[0][0]
activation_883 (Activation)	(None, 25, 25, 64)	0	batch_normalization_895[0][0]
block35_10_mixed (Concatenate)	(None, 25, 25, 128)	0	activation_878[0][0] activation_880[0][0] activation_883[0][0]
block35_10_conv (Conv2D)	(None, 25, 25, 320)	41280	block35_10_mixed[0][0]
block35_10 (Lambda)	(None, 25, 25, 320)	0	block35_9_ac[0][0] block35_10_conv[0][0]
block35_10_ac (Activation)	(None, 25, 25, 320)	0	block35_10[0][0]
conv2d_885 (Conv2D)	(None, 25, 25, 256)	81920	block35_10_ac[0][0]
batch_normalization_897 (Batch Normalization)	(None, 25, 25, 256)	768	conv2d_885[0][0]
activation_885 (Activation)	(None, 25, 25, 256)	0	batch_normalization_897[0][0]
conv2d_886 (Conv2D)	(None, 25, 25, 256)	589824	activation_885[0][0]
batch_normalization_898 (Batch Normalization)	(None, 25, 25, 256)	768	conv2d_886[0][0]
activation_886 (Activation)	(None, 25, 25, 256)	0	batch_normalization_898[0][0]
conv2d_884 (Conv2D)	(None, 12, 12, 384)	1105920	block35_10_ac[0][0]
conv2d_887 (Conv2D)	(None, 12, 12, 384)	884736	activation_886[0][0]
batch_normalization_896 (Batch Normalization)	(None, 12, 12, 384)	1152	conv2d_884[0][0]
batch_normalization_899 (Batch Normalization)	(None, 12, 12, 384)	1152	conv2d_887[0][0]
activation_884 (Activation)	(None, 12, 12, 384)	0	batch_normalization_896[0][0]
activation_887 (Activation)	(None, 12, 12, 384)	0	batch_normalization_899[0][0]
max_pooling2d_18 (MaxPooling2D)	(None, 12, 12, 320)	0	block35_10_ac[0][0]
mixed_6a (Concatenate)	(None, 12, 12, 1088)	0	activation_884[0][0] activation_887[0][0] max_pooling2d_18[0][0]
conv2d_889 (Conv2D)	(None, 12, 12, 128)	139264	mixed_6a[0][0]
batch_normalization_901 (Batch Normalization)	(None, 12, 12, 128)	384	conv2d_889[0][0]
activation_889 (Activation)	(None, 12, 12, 128)	0	batch_normalization_901[0][0]
conv2d_890 (Conv2D)	(None, 12, 12, 160)	143360	activation_889[0][0]
batch_normalization_902 (Batch Normalization)	(None, 12, 12, 160)	480	conv2d_890[0][0]
activation_890 (Activation)	(None, 12, 12, 160)	0	batch_normalization_902[0][0]
conv2d_888 (Conv2D)	(None, 12, 12, 192)	208896	mixed_6a[0][0]
conv2d_891 (Conv2D)	(None, 12, 12, 192)	215040	activation_890[0][0]
batch_normalization_900 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_888[0][0]
batch_normalization_903 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_891[0][0]
activation_888 (Activation)	(None, 12, 12, 192)	0	batch_normalization_900[0][0]
activation_891 (Activation)	(None, 12, 12, 192)	0	batch_normalization_903[0][0]
block17_1_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_888[0][0] activation_891[0][0]

block17_1_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_1_mixed[0][0]
block17_1 (Lambda)	(None, 12, 12, 1088)	0	mixed_6a[0][0] block17_1_conv[0][0]
block17_1_ac (Activation)	(None, 12, 12, 1088)	0	block17_1[0][0]
conv2d_893 (Conv2D)	(None, 12, 12, 128)	139264	block17_1_ac[0][0]
batch_normalization_905 (BatchN	(None, 12, 12, 128)	384	conv2d_893[0][0]
activation_893 (Activation)	(None, 12, 12, 128)	0	batch_normalization_905[0][0]
conv2d_894 (Conv2D)	(None, 12, 12, 160)	143360	activation_893[0][0]
batch_normalization_906 (BatchN	(None, 12, 12, 160)	480	conv2d_894[0][0]
activation_894 (Activation)	(None, 12, 12, 160)	0	batch_normalization_906[0][0]
conv2d_892 (Conv2D)	(None, 12, 12, 192)	208896	block17_1_ac[0][0]
conv2d_895 (Conv2D)	(None, 12, 12, 192)	215040	activation_894[0][0]
batch_normalization_904 (BatchN	(None, 12, 12, 192)	576	conv2d_892[0][0]
batch_normalization_907 (BatchN	(None, 12, 12, 192)	576	conv2d_895[0][0]
activation_892 (Activation)	(None, 12, 12, 192)	0	batch_normalization_904[0][0]
activation_895 (Activation)	(None, 12, 12, 192)	0	batch_normalization_907[0][0]
block17_2_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_892[0][0] activation_895[0][0]
block17_2_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_2_mixed[0][0]
block17_2 (Lambda)	(None, 12, 12, 1088)	0	block17_1_ac[0][0] block17_2_conv[0][0]
block17_2_ac (Activation)	(None, 12, 12, 1088)	0	block17_2[0][0]
conv2d_897 (Conv2D)	(None, 12, 12, 128)	139264	block17_2_ac[0][0]
batch_normalization_909 (BatchN	(None, 12, 12, 128)	384	conv2d_897[0][0]
activation_897 (Activation)	(None, 12, 12, 128)	0	batch_normalization_909[0][0]
conv2d_898 (Conv2D)	(None, 12, 12, 160)	143360	activation_897[0][0]
batch_normalization_910 (BatchN	(None, 12, 12, 160)	480	conv2d_898[0][0]
activation_898 (Activation)	(None, 12, 12, 160)	0	batch_normalization_910[0][0]
conv2d_896 (Conv2D)	(None, 12, 12, 192)	208896	block17_2_ac[0][0]
conv2d_899 (Conv2D)	(None, 12, 12, 192)	215040	activation_898[0][0]
batch_normalization_908 (BatchN	(None, 12, 12, 192)	576	conv2d_896[0][0]
batch_normalization_911 (BatchN	(None, 12, 12, 192)	576	conv2d_899[0][0]
activation_896 (Activation)	(None, 12, 12, 192)	0	batch_normalization_908[0][0]
activation_899 (Activation)	(None, 12, 12, 192)	0	batch_normalization_911[0][0]
block17_3_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_896[0][0] activation_899[0][0]
block17_3_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_3_mixed[0][0]
block17_3 (Lambda)	(None, 12, 12, 1088)	0	block17_2_ac[0][0] block17_3_conv[0][0]
block17_3_ac (Activation)	(None, 12, 12, 1088)	0	block17_3[0][0]
conv2d_901 (Conv2D)	(None, 12, 12, 128)	139264	block17_3_ac[0][0]
batch_normalization_913 (BatchN	(None, 12, 12, 128)	384	conv2d_901[0][0]
activation_901 (Activation)	(None, 12, 12, 128)	0	batch_normalization_913[0][0]
conv2d_902 (Conv2D)	(None, 12, 12, 160)	143360	activation_901[0][0]
batch_normalization_914 (BatchN	(None, 12, 12, 160)	480	conv2d_902[0][0]
activation_902 (Activation)	(None, 12, 12, 160)	0	batch_normalization_914[0][0]
conv2d_900 (Conv2D)	(None, 12, 12, 192)	208896	block17_3_ac[0][0]

conv2d_903 (Conv2D)	(None, 12, 12, 192)	215040	activation_902[0][0]
batch_normalization_912 (BatchN	(None, 12, 12, 192)	576	conv2d_900[0][0]
batch_normalization_915 (BatchN	(None, 12, 12, 192)	576	conv2d_903[0][0]
activation_900 (Activation)	(None, 12, 12, 192)	0	batch_normalization_912[0][0]
activation_903 (Activation)	(None, 12, 12, 192)	0	batch_normalization_915[0][0]
block17_4_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_900[0][0] activation_903[0][0]
block17_4_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_4_mixed[0][0]
block17_4 (Lambda)	(None, 12, 12, 1088)	0	block17_3_ac[0][0] block17_4_conv[0][0]
block17_4_ac (Activation)	(None, 12, 12, 1088)	0	block17_4[0][0]
conv2d_905 (Conv2D)	(None, 12, 12, 128)	139264	block17_4_ac[0][0]
batch_normalization_917 (BatchN	(None, 12, 12, 128)	384	conv2d_905[0][0]
activation_905 (Activation)	(None, 12, 12, 128)	0	batch_normalization_917[0][0]
conv2d_906 (Conv2D)	(None, 12, 12, 160)	143360	activation_905[0][0]
batch_normalization_918 (BatchN	(None, 12, 12, 160)	480	conv2d_906[0][0]
activation_906 (Activation)	(None, 12, 12, 160)	0	batch_normalization_918[0][0]
conv2d_904 (Conv2D)	(None, 12, 12, 192)	208896	block17_4_ac[0][0]
conv2d_907 (Conv2D)	(None, 12, 12, 192)	215040	activation_906[0][0]
batch_normalization_916 (BatchN	(None, 12, 12, 192)	576	conv2d_904[0][0]
batch_normalization_919 (BatchN	(None, 12, 12, 192)	576	conv2d_907[0][0]
activation_904 (Activation)	(None, 12, 12, 192)	0	batch_normalization_916[0][0]
activation_907 (Activation)	(None, 12, 12, 192)	0	batch_normalization_919[0][0]
block17_5_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_904[0][0] activation_907[0][0]
block17_5_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_5_mixed[0][0]
block17_5 (Lambda)	(None, 12, 12, 1088)	0	block17_4_ac[0][0] block17_5_conv[0][0]
block17_5_ac (Activation)	(None, 12, 12, 1088)	0	block17_5[0][0]
conv2d_909 (Conv2D)	(None, 12, 12, 128)	139264	block17_5_ac[0][0]
batch_normalization_921 (BatchN	(None, 12, 12, 128)	384	conv2d_909[0][0]
activation_909 (Activation)	(None, 12, 12, 128)	0	batch_normalization_921[0][0]
conv2d_910 (Conv2D)	(None, 12, 12, 160)	143360	activation_909[0][0]
batch_normalization_922 (BatchN	(None, 12, 12, 160)	480	conv2d_910[0][0]
activation_910 (Activation)	(None, 12, 12, 160)	0	batch_normalization_922[0][0]
conv2d_908 (Conv2D)	(None, 12, 12, 192)	208896	block17_5_ac[0][0]
conv2d_911 (Conv2D)	(None, 12, 12, 192)	215040	activation_910[0][0]
batch_normalization_920 (BatchN	(None, 12, 12, 192)	576	conv2d_908[0][0]
batch_normalization_923 (BatchN	(None, 12, 12, 192)	576	conv2d_911[0][0]
activation_908 (Activation)	(None, 12, 12, 192)	0	batch_normalization_920[0][0]
activation_911 (Activation)	(None, 12, 12, 192)	0	batch_normalization_923[0][0]
block17_6_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_908[0][0] activation_911[0][0]
block17_6_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_6_mixed[0][0]
block17_6 (Lambda)	(None, 12, 12, 1088)	0	block17_5_ac[0][0] block17_6_conv[0][0]
block17_6_ac (Activation)	(None, 12, 12, 1088)	0	block17_6[0][0]



conv2d_913 (Conv2D)	(None, 12, 12, 128)	139264	block17_6_ac[0][0]
batch_normalization_925 (Batch Normalization)	(None, 12, 12, 128)	384	conv2d_913[0][0]
activation_913 (Activation)	(None, 12, 12, 128)	0	batch_normalization_925[0][0]
conv2d_914 (Conv2D)	(None, 12, 12, 160)	143360	activation_913[0][0]
batch_normalization_926 (Batch Normalization)	(None, 12, 12, 160)	480	conv2d_914[0][0]
activation_914 (Activation)	(None, 12, 12, 160)	0	batch_normalization_926[0][0]
conv2d_912 (Conv2D)	(None, 12, 12, 192)	208896	block17_6_ac[0][0]
conv2d_915 (Conv2D)	(None, 12, 12, 192)	215040	activation_914[0][0]
batch_normalization_924 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_912[0][0]
batch_normalization_927 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_915[0][0]
activation_912 (Activation)	(None, 12, 12, 192)	0	batch_normalization_924[0][0]
activation_915 (Activation)	(None, 12, 12, 192)	0	batch_normalization_927[0][0]
block17_7_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_912[0][0] activation_915[0][0]
block17_7_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_7_mixed[0][0]
block17_7 (Lambda)	(None, 12, 12, 1088)	0	block17_6_ac[0][0] block17_7_conv[0][0]
block17_7_ac (Activation)	(None, 12, 12, 1088)	0	block17_7[0][0]
conv2d_917 (Conv2D)	(None, 12, 12, 128)	139264	block17_7_ac[0][0]
batch_normalization_929 (Batch Normalization)	(None, 12, 12, 128)	384	conv2d_917[0][0]
activation_917 (Activation)	(None, 12, 12, 128)	0	batch_normalization_929[0][0]
conv2d_918 (Conv2D)	(None, 12, 12, 160)	143360	activation_917[0][0]
batch_normalization_930 (Batch Normalization)	(None, 12, 12, 160)	480	conv2d_918[0][0]
activation_918 (Activation)	(None, 12, 12, 160)	0	batch_normalization_930[0][0]
conv2d_916 (Conv2D)	(None, 12, 12, 192)	208896	block17_7_ac[0][0]
conv2d_919 (Conv2D)	(None, 12, 12, 192)	215040	activation_918[0][0]
batch_normalization_928 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_916[0][0]
batch_normalization_931 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_919[0][0]
activation_916 (Activation)	(None, 12, 12, 192)	0	batch_normalization_928[0][0]
activation_919 (Activation)	(None, 12, 12, 192)	0	batch_normalization_931[0][0]
block17_8_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_916[0][0] activation_919[0][0]
block17_8_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_8_mixed[0][0]
block17_8 (Lambda)	(None, 12, 12, 1088)	0	block17_7_ac[0][0] block17_8_conv[0][0]
block17_8_ac (Activation)	(None, 12, 12, 1088)	0	block17_8[0][0]
conv2d_921 (Conv2D)	(None, 12, 12, 128)	139264	block17_8_ac[0][0]
batch_normalization_933 (Batch Normalization)	(None, 12, 12, 128)	384	conv2d_921[0][0]
activation_921 (Activation)	(None, 12, 12, 128)	0	batch_normalization_933[0][0]
conv2d_922 (Conv2D)	(None, 12, 12, 160)	143360	activation_921[0][0]
batch_normalization_934 (Batch Normalization)	(None, 12, 12, 160)	480	conv2d_922[0][0]
activation_922 (Activation)	(None, 12, 12, 160)	0	batch_normalization_934[0][0]
conv2d_920 (Conv2D)	(None, 12, 12, 192)	208896	block17_8_ac[0][0]
conv2d_923 (Conv2D)	(None, 12, 12, 192)	215040	activation_922[0][0]
batch_normalization_932 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_920[0][0]
batch_normalization_935 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_923[0][0]

activation_920 (Activation)	(None, 12, 12, 192)	0	batch_normalization_932[0][0]
activation_923 (Activation)	(None, 12, 12, 192)	0	batch_normalization_935[0][0]
block17_9_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_920[0][0] activation_923[0][0]
block17_9_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_9_mixed[0][0]
block17_9 (Lambda)	(None, 12, 12, 1088)	0	block17_8_ac[0][0] block17_9_conv[0][0]
block17_9_ac (Activation)	(None, 12, 12, 1088)	0	block17_9[0][0]
conv2d_925 (Conv2D)	(None, 12, 12, 128)	139264	block17_9_ac[0][0]
batch_normalization_937 (BatchN	(None, 12, 12, 128)	384	conv2d_925[0][0]
activation_925 (Activation)	(None, 12, 12, 128)	0	batch_normalization_937[0][0]
conv2d_926 (Conv2D)	(None, 12, 12, 160)	143360	activation_925[0][0]
batch_normalization_938 (BatchN	(None, 12, 12, 160)	480	conv2d_926[0][0]
activation_926 (Activation)	(None, 12, 12, 160)	0	batch_normalization_938[0][0]
conv2d_924 (Conv2D)	(None, 12, 12, 192)	208896	block17_9_ac[0][0]
conv2d_927 (Conv2D)	(None, 12, 12, 192)	215040	activation_926[0][0]
batch_normalization_936 (BatchN	(None, 12, 12, 192)	576	conv2d_924[0][0]
batch_normalization_939 (BatchN	(None, 12, 12, 192)	576	conv2d_927[0][0]
activation_924 (Activation)	(None, 12, 12, 192)	0	batch_normalization_936[0][0]
activation_927 (Activation)	(None, 12, 12, 192)	0	batch_normalization_939[0][0]
block17_10_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_924[0][0] activation_927[0][0]
block17_10_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_10_mixed[0][0]
block17_10 (Lambda)	(None, 12, 12, 1088)	0	block17_9_ac[0][0] block17_10_conv[0][0]
block17_10_ac (Activation)	(None, 12, 12, 1088)	0	block17_10[0][0]
conv2d_929 (Conv2D)	(None, 12, 12, 128)	139264	block17_10_ac[0][0]
batch_normalization_941 (BatchN	(None, 12, 12, 128)	384	conv2d_929[0][0]
activation_929 (Activation)	(None, 12, 12, 128)	0	batch_normalization_941[0][0]
conv2d_930 (Conv2D)	(None, 12, 12, 160)	143360	activation_929[0][0]
batch_normalization_942 (BatchN	(None, 12, 12, 160)	480	conv2d_930[0][0]
activation_930 (Activation)	(None, 12, 12, 160)	0	batch_normalization_942[0][0]
conv2d_928 (Conv2D)	(None, 12, 12, 192)	208896	block17_10_ac[0][0]
conv2d_931 (Conv2D)	(None, 12, 12, 192)	215040	activation_930[0][0]
batch_normalization_940 (BatchN	(None, 12, 12, 192)	576	conv2d_928[0][0]
batch_normalization_943 (BatchN	(None, 12, 12, 192)	576	conv2d_931[0][0]
activation_928 (Activation)	(None, 12, 12, 192)	0	batch_normalization_940[0][0]
activation_931 (Activation)	(None, 12, 12, 192)	0	batch_normalization_943[0][0]
block17_11_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_928[0][0] activation_931[0][0]
block17_11_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_11_mixed[0][0]
block17_11 (Lambda)	(None, 12, 12, 1088)	0	block17_10_ac[0][0] block17_11_conv[0][0]
block17_11_ac (Activation)	(None, 12, 12, 1088)	0	block17_11[0][0]
conv2d_933 (Conv2D)	(None, 12, 12, 128)	139264	block17_11_ac[0][0]
batch_normalization_945 (BatchN	(None, 12, 12, 128)	384	conv2d_933[0][0]
activation_933 (Activation)	(None, 12, 12, 128)	0	batch_normalization_945[0][0]

conv2d_934 (Conv2D)	(None, 12, 12, 160)	143360	activation_933[0][0]
batch_normalization_946 (Batch Normalization)	(None, 12, 12, 160)	480	conv2d_934[0][0]
activation_934 (Activation)	(None, 12, 12, 160)	0	batch_normalization_946[0][0]
conv2d_932 (Conv2D)	(None, 12, 12, 192)	208896	block17_11_ac[0][0]
conv2d_935 (Conv2D)	(None, 12, 12, 192)	215040	activation_934[0][0]
batch_normalization_944 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_932[0][0]
batch_normalization_947 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_935[0][0]
activation_932 (Activation)	(None, 12, 12, 192)	0	batch_normalization_944[0][0]
activation_935 (Activation)	(None, 12, 12, 192)	0	batch_normalization_947[0][0]
block17_12_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_932[0][0] activation_935[0][0]
block17_12_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_12_mixed[0][0]
block17_12 (Lambda)	(None, 12, 12, 1088)	0	block17_11_ac[0][0] block17_12_conv[0][0]
block17_12_ac (Activation)	(None, 12, 12, 1088)	0	block17_12[0][0]
conv2d_937 (Conv2D)	(None, 12, 12, 128)	139264	block17_12_ac[0][0]
batch_normalization_949 (Batch Normalization)	(None, 12, 12, 128)	384	conv2d_937[0][0]
activation_937 (Activation)	(None, 12, 12, 128)	0	batch_normalization_949[0][0]
conv2d_938 (Conv2D)	(None, 12, 12, 160)	143360	activation_937[0][0]
batch_normalization_950 (Batch Normalization)	(None, 12, 12, 160)	480	conv2d_938[0][0]
activation_938 (Activation)	(None, 12, 12, 160)	0	batch_normalization_950[0][0]
conv2d_936 (Conv2D)	(None, 12, 12, 192)	208896	block17_12_ac[0][0]
conv2d_939 (Conv2D)	(None, 12, 12, 192)	215040	activation_938[0][0]
batch_normalization_948 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_936[0][0]
batch_normalization_951 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_939[0][0]
activation_936 (Activation)	(None, 12, 12, 192)	0	batch_normalization_948[0][0]
activation_939 (Activation)	(None, 12, 12, 192)	0	batch_normalization_951[0][0]
block17_13_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_936[0][0] activation_939[0][0]
block17_13_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_13_mixed[0][0]
block17_13 (Lambda)	(None, 12, 12, 1088)	0	block17_12_ac[0][0] block17_13_conv[0][0]
block17_13_ac (Activation)	(None, 12, 12, 1088)	0	block17_13[0][0]
conv2d_941 (Conv2D)	(None, 12, 12, 128)	139264	block17_13_ac[0][0]
batch_normalization_953 (Batch Normalization)	(None, 12, 12, 128)	384	conv2d_941[0][0]
activation_941 (Activation)	(None, 12, 12, 128)	0	batch_normalization_953[0][0]
conv2d_942 (Conv2D)	(None, 12, 12, 160)	143360	activation_941[0][0]
batch_normalization_954 (Batch Normalization)	(None, 12, 12, 160)	480	conv2d_942[0][0]
activation_942 (Activation)	(None, 12, 12, 160)	0	batch_normalization_954[0][0]
conv2d_940 (Conv2D)	(None, 12, 12, 192)	208896	block17_13_ac[0][0]
conv2d_943 (Conv2D)	(None, 12, 12, 192)	215040	activation_942[0][0]
batch_normalization_952 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_940[0][0]
batch_normalization_955 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_943[0][0]
activation_940 (Activation)	(None, 12, 12, 192)	0	batch_normalization_952[0][0]
activation_943 (Activation)	(None, 12, 12, 192)	0	batch_normalization_955[0][0]
block17_14_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_940[0][0]

			activation_943[0][0]
block17_14_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_14_mixed[0][0]
block17_14 (Lambda)	(None, 12, 12, 1088)	0	block17_13_ac[0][0] block17_14_conv[0][0]
block17_14_ac (Activation)	(None, 12, 12, 1088)	0	block17_14[0][0]
conv2d_945 (Conv2D)	(None, 12, 12, 128)	139264	block17_14_ac[0][0]
batch_normalization_957 (BatchN	(None, 12, 12, 128)	384	conv2d_945[0][0]
activation_945 (Activation)	(None, 12, 12, 128)	0	batch_normalization_957[0][0]
conv2d_946 (Conv2D)	(None, 12, 12, 160)	143360	activation_945[0][0]
batch_normalization_958 (BatchN	(None, 12, 12, 160)	480	conv2d_946[0][0]
activation_946 (Activation)	(None, 12, 12, 160)	0	batch_normalization_958[0][0]
conv2d_944 (Conv2D)	(None, 12, 12, 192)	208896	block17_14_ac[0][0]
conv2d_947 (Conv2D)	(None, 12, 12, 192)	215040	activation_946[0][0]
batch_normalization_956 (BatchN	(None, 12, 12, 192)	576	conv2d_944[0][0]
batch_normalization_959 (BatchN	(None, 12, 12, 192)	576	conv2d_947[0][0]
activation_944 (Activation)	(None, 12, 12, 192)	0	batch_normalization_956[0][0]
activation_947 (Activation)	(None, 12, 12, 192)	0	batch_normalization_959[0][0]
block17_15_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_944[0][0] activation_947[0][0]
block17_15_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_15_mixed[0][0]
block17_15 (Lambda)	(None, 12, 12, 1088)	0	block17_14_ac[0][0] block17_15_conv[0][0]
block17_15_ac (Activation)	(None, 12, 12, 1088)	0	block17_15[0][0]
conv2d_949 (Conv2D)	(None, 12, 12, 128)	139264	block17_15_ac[0][0]
batch_normalization_961 (BatchN	(None, 12, 12, 128)	384	conv2d_949[0][0]
activation_949 (Activation)	(None, 12, 12, 128)	0	batch_normalization_961[0][0]
conv2d_950 (Conv2D)	(None, 12, 12, 160)	143360	activation_949[0][0]
batch_normalization_962 (BatchN	(None, 12, 12, 160)	480	conv2d_950[0][0]
activation_950 (Activation)	(None, 12, 12, 160)	0	batch_normalization_962[0][0]
conv2d_948 (Conv2D)	(None, 12, 12, 192)	208896	block17_15_ac[0][0]
conv2d_951 (Conv2D)	(None, 12, 12, 192)	215040	activation_950[0][0]
batch_normalization_960 (BatchN	(None, 12, 12, 192)	576	conv2d_948[0][0]
batch_normalization_963 (BatchN	(None, 12, 12, 192)	576	conv2d_951[0][0]
activation_948 (Activation)	(None, 12, 12, 192)	0	batch_normalization_960[0][0]
activation_951 (Activation)	(None, 12, 12, 192)	0	batch_normalization_963[0][0]
block17_16_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_948[0][0] activation_951[0][0]
block17_16_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_16_mixed[0][0]
block17_16 (Lambda)	(None, 12, 12, 1088)	0	block17_15_ac[0][0] block17_16_conv[0][0]
block17_16_ac (Activation)	(None, 12, 12, 1088)	0	block17_16[0][0]
conv2d_953 (Conv2D)	(None, 12, 12, 128)	139264	block17_16_ac[0][0]
batch_normalization_965 (BatchN	(None, 12, 12, 128)	384	conv2d_953[0][0]
activation_953 (Activation)	(None, 12, 12, 128)	0	batch_normalization_965[0][0]
conv2d_954 (Conv2D)	(None, 12, 12, 160)	143360	activation_953[0][0]
batch_normalization_966 (BatchN	(None, 12, 12, 160)	480	conv2d_954[0][0]
activation_954 (Activation)	(None, 12, 12, 160)	0	batch_normalization_966[0][0]

conv2d_952 (Conv2D)	(None, 12, 12, 192)	208896	block17_16_ac[0][0]
conv2d_955 (Conv2D)	(None, 12, 12, 192)	215040	activation_954[0][0]
batch_normalization_964 (BatchN	(None, 12, 12, 192)	576	conv2d_952[0][0]
batch_normalization_967 (BatchN	(None, 12, 12, 192)	576	conv2d_955[0][0]
activation_952 (Activation)	(None, 12, 12, 192)	0	batch_normalization_964[0][0]
activation_955 (Activation)	(None, 12, 12, 192)	0	batch_normalization_967[0][0]
block17_17_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_952[0][0] activation_955[0][0]
block17_17_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_17_mixed[0][0]
block17_17 (Lambda)	(None, 12, 12, 1088)	0	block17_16_ac[0][0] block17_17_conv[0][0]
block17_17_ac (Activation)	(None, 12, 12, 1088)	0	block17_17[0][0]
conv2d_957 (Conv2D)	(None, 12, 12, 128)	139264	block17_17_ac[0][0]
batch_normalization_969 (BatchN	(None, 12, 12, 128)	384	conv2d_957[0][0]
activation_957 (Activation)	(None, 12, 12, 128)	0	batch_normalization_969[0][0]
conv2d_958 (Conv2D)	(None, 12, 12, 160)	143360	activation_957[0][0]
batch_normalization_970 (BatchN	(None, 12, 12, 160)	480	conv2d_958[0][0]
activation_958 (Activation)	(None, 12, 12, 160)	0	batch_normalization_970[0][0]
conv2d_956 (Conv2D)	(None, 12, 12, 192)	208896	block17_17_ac[0][0]
conv2d_959 (Conv2D)	(None, 12, 12, 192)	215040	activation_958[0][0]
batch_normalization_968 (BatchN	(None, 12, 12, 192)	576	conv2d_956[0][0]
batch_normalization_971 (BatchN	(None, 12, 12, 192)	576	conv2d_959[0][0]
activation_956 (Activation)	(None, 12, 12, 192)	0	batch_normalization_968[0][0]
activation_959 (Activation)	(None, 12, 12, 192)	0	batch_normalization_971[0][0]
block17_18_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_956[0][0] activation_959[0][0]
block17_18_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_18_mixed[0][0]
block17_18 (Lambda)	(None, 12, 12, 1088)	0	block17_17_ac[0][0] block17_18_conv[0][0]
block17_18_ac (Activation)	(None, 12, 12, 1088)	0	block17_18[0][0]
conv2d_961 (Conv2D)	(None, 12, 12, 128)	139264	block17_18_ac[0][0]
batch_normalization_973 (BatchN	(None, 12, 12, 128)	384	conv2d_961[0][0]
activation_961 (Activation)	(None, 12, 12, 128)	0	batch_normalization_973[0][0]
conv2d_962 (Conv2D)	(None, 12, 12, 160)	143360	activation_961[0][0]
batch_normalization_974 (BatchN	(None, 12, 12, 160)	480	conv2d_962[0][0]
activation_962 (Activation)	(None, 12, 12, 160)	0	batch_normalization_974[0][0]
conv2d_960 (Conv2D)	(None, 12, 12, 192)	208896	block17_18_ac[0][0]
conv2d_963 (Conv2D)	(None, 12, 12, 192)	215040	activation_962[0][0]
batch_normalization_972 (BatchN	(None, 12, 12, 192)	576	conv2d_960[0][0]
batch_normalization_975 (BatchN	(None, 12, 12, 192)	576	conv2d_963[0][0]
activation_960 (Activation)	(None, 12, 12, 192)	0	batch_normalization_972[0][0]
activation_963 (Activation)	(None, 12, 12, 192)	0	batch_normalization_975[0][0]
block17_19_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_960[0][0] activation_963[0][0]
block17_19_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_19_mixed[0][0]
block17_19 (Lambda)	(None, 12, 12, 1088)	0	block17_18_ac[0][0] block17_19_conv[0][0]

block17_19_ac (Activation)	(None, 12, 12, 1088)	0	block17_19[0][0]
conv2d_965 (Conv2D)	(None, 12, 12, 128)	139264	block17_19_ac[0][0]
batch_normalization_977 (Batch Normalization)	(None, 12, 12, 128)	384	conv2d_965[0][0]
activation_965 (Activation)	(None, 12, 12, 128)	0	batch_normalization_977[0][0]
conv2d_966 (Conv2D)	(None, 12, 12, 160)	143360	activation_965[0][0]
batch_normalization_978 (Batch Normalization)	(None, 12, 12, 160)	480	conv2d_966[0][0]
activation_966 (Activation)	(None, 12, 12, 160)	0	batch_normalization_978[0][0]
conv2d_964 (Conv2D)	(None, 12, 12, 192)	208896	block17_19_ac[0][0]
conv2d_967 (Conv2D)	(None, 12, 12, 192)	215040	activation_966[0][0]
batch_normalization_976 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_964[0][0]
batch_normalization_979 (Batch Normalization)	(None, 12, 12, 192)	576	conv2d_967[0][0]
activation_964 (Activation)	(None, 12, 12, 192)	0	batch_normalization_976[0][0]
activation_967 (Activation)	(None, 12, 12, 192)	0	batch_normalization_979[0][0]
block17_20_mixed (Concatenate)	(None, 12, 12, 384)	0	activation_964[0][0] activation_967[0][0]
block17_20_conv (Conv2D)	(None, 12, 12, 1088)	418880	block17_20_mixed[0][0]
block17_20 (Lambda)	(None, 12, 12, 1088)	0	block17_19_ac[0][0] block17_20_conv[0][0]
block17_20_ac (Activation)	(None, 12, 12, 1088)	0	block17_20[0][0]
conv2d_972 (Conv2D)	(None, 12, 12, 256)	278528	block17_20_ac[0][0]
batch_normalization_984 (Batch Normalization)	(None, 12, 12, 256)	768	conv2d_972[0][0]
activation_972 (Activation)	(None, 12, 12, 256)	0	batch_normalization_984[0][0]
conv2d_968 (Conv2D)	(None, 12, 12, 256)	278528	block17_20_ac[0][0]
conv2d_970 (Conv2D)	(None, 12, 12, 256)	278528	block17_20_ac[0][0]
conv2d_973 (Conv2D)	(None, 12, 12, 288)	663552	activation_972[0][0]
batch_normalization_980 (Batch Normalization)	(None, 12, 12, 256)	768	conv2d_968[0][0]
batch_normalization_982 (Batch Normalization)	(None, 12, 12, 256)	768	conv2d_970[0][0]
batch_normalization_985 (Batch Normalization)	(None, 12, 12, 288)	864	conv2d_973[0][0]
activation_968 (Activation)	(None, 12, 12, 256)	0	batch_normalization_980[0][0]
activation_970 (Activation)	(None, 12, 12, 256)	0	batch_normalization_982[0][0]
activation_973 (Activation)	(None, 12, 12, 288)	0	batch_normalization_985[0][0]
conv2d_969 (Conv2D)	(None, 5, 5, 384)	884736	activation_968[0][0]
conv2d_971 (Conv2D)	(None, 5, 5, 288)	663552	activation_970[0][0]
conv2d_974 (Conv2D)	(None, 5, 5, 320)	829440	activation_973[0][0]
batch_normalization_981 (Batch Normalization)	(None, 5, 5, 384)	1152	conv2d_969[0][0]
batch_normalization_983 (Batch Normalization)	(None, 5, 5, 288)	864	conv2d_971[0][0]
batch_normalization_986 (Batch Normalization)	(None, 5, 5, 320)	960	conv2d_974[0][0]
activation_969 (Activation)	(None, 5, 5, 384)	0	batch_normalization_981[0][0]
activation_971 (Activation)	(None, 5, 5, 288)	0	batch_normalization_983[0][0]
activation_974 (Activation)	(None, 5, 5, 320)	0	batch_normalization_986[0][0]
max_pooling2d_19 (MaxPooling2D)	(None, 5, 5, 1088)	0	block17_20_ac[0][0]
mixed_7a (Concatenate)	(None, 5, 5, 2080)	0	activation_969[0][0] activation_971[0][0] activation_974[0][0] max_pooling2d_19[0][0]
conv2d_976 (Conv2D)	(None, 5, 5, 192)	399360	mixed_7a[0][0]

batch_normalization_988 (BatchN	(None, 5, 5, 192)	576	conv2d_976[0][0]
activation_976 (Activation)	(None, 5, 5, 192)	0	batch_normalization_988[0][0]
conv2d_977 (Conv2D)	(None, 5, 5, 224)	129024	activation_976[0][0]
batch_normalization_989 (BatchN	(None, 5, 5, 224)	672	conv2d_977[0][0]
activation_977 (Activation)	(None, 5, 5, 224)	0	batch_normalization_989[0][0]
conv2d_975 (Conv2D)	(None, 5, 5, 192)	399360	mixed_7a[0][0]
conv2d_978 (Conv2D)	(None, 5, 5, 256)	172032	activation_977[0][0]
batch_normalization_987 (BatchN	(None, 5, 5, 192)	576	conv2d_975[0][0]
batch_normalization_990 (BatchN	(None, 5, 5, 256)	768	conv2d_978[0][0]
activation_975 (Activation)	(None, 5, 5, 192)	0	batch_normalization_987[0][0]
activation_978 (Activation)	(None, 5, 5, 256)	0	batch_normalization_990[0][0]
block8_1_mixed (Concatenate)	(None, 5, 5, 448)	0	activation_975[0][0] activation_978[0][0]
block8_1_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_1_mixed[0][0]
block8_1 (Lambda)	(None, 5, 5, 2080)	0	mixed_7a[0][0] block8_1_conv[0][0]
block8_1_ac (Activation)	(None, 5, 5, 2080)	0	block8_1[0][0]
conv2d_980 (Conv2D)	(None, 5, 5, 192)	399360	block8_1_ac[0][0]
batch_normalization_992 (BatchN	(None, 5, 5, 192)	576	conv2d_980[0][0]
activation_980 (Activation)	(None, 5, 5, 192)	0	batch_normalization_992[0][0]
conv2d_981 (Conv2D)	(None, 5, 5, 224)	129024	activation_980[0][0]
batch_normalization_993 (BatchN	(None, 5, 5, 224)	672	conv2d_981[0][0]
activation_981 (Activation)	(None, 5, 5, 224)	0	batch_normalization_993[0][0]
conv2d_979 (Conv2D)	(None, 5, 5, 192)	399360	block8_1_ac[0][0]
conv2d_982 (Conv2D)	(None, 5, 5, 256)	172032	activation_981[0][0]
batch_normalization_991 (BatchN	(None, 5, 5, 192)	576	conv2d_979[0][0]
batch_normalization_994 (BatchN	(None, 5, 5, 256)	768	conv2d_982[0][0]
activation_979 (Activation)	(None, 5, 5, 192)	0	batch_normalization_991[0][0]
activation_982 (Activation)	(None, 5, 5, 256)	0	batch_normalization_994[0][0]
block8_2_mixed (Concatenate)	(None, 5, 5, 448)	0	activation_979[0][0] activation_982[0][0]
block8_2_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_2_mixed[0][0]
block8_2 (Lambda)	(None, 5, 5, 2080)	0	block8_1_ac[0][0] block8_2_conv[0][0]
block8_2_ac (Activation)	(None, 5, 5, 2080)	0	block8_2[0][0]
conv2d_984 (Conv2D)	(None, 5, 5, 192)	399360	block8_2_ac[0][0]
batch_normalization_996 (BatchN	(None, 5, 5, 192)	576	conv2d_984[0][0]
activation_984 (Activation)	(None, 5, 5, 192)	0	batch_normalization_996[0][0]
conv2d_985 (Conv2D)	(None, 5, 5, 224)	129024	activation_984[0][0]
batch_normalization_997 (BatchN	(None, 5, 5, 224)	672	conv2d_985[0][0]
activation_985 (Activation)	(None, 5, 5, 224)	0	batch_normalization_997[0][0]
conv2d_983 (Conv2D)	(None, 5, 5, 192)	399360	block8_2_ac[0][0]
conv2d_986 (Conv2D)	(None, 5, 5, 256)	172032	activation_985[0][0]
batch_normalization_995 (BatchN	(None, 5, 5, 192)	576	conv2d_983[0][0]
batch_normalization_998 (BatchN	(None, 5, 5, 256)	768	conv2d_986[0][0]
activation_983 (Activation)	(None, 5, 5, 192)	0	batch_normalization_995[0][0]

activation_986 (Activation)	(None, 5, 5, 256)	0	batch_normalization_998[0][0]
block8_3_mixed (Concatenate)	(None, 5, 5, 448)	0	activation_983[0][0] activation_986[0][0]
block8_3_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_3_mixed[0][0]
block8_3 (Lambda)	(None, 5, 5, 2080)	0	block8_2_ac[0][0] block8_3_conv[0][0]
block8_3_ac (Activation)	(None, 5, 5, 2080)	0	block8_3[0][0]
conv2d_988 (Conv2D)	(None, 5, 5, 192)	399360	block8_3_ac[0][0]
batch_normalization_1000 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_988[0][0]
activation_988 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1000[0][0]
conv2d_989 (Conv2D)	(None, 5, 5, 224)	129024	activation_988[0][0]
batch_normalization_1001 (Batch Normalization)	(None, 5, 5, 224)	672	conv2d_989[0][0]
activation_989 (Activation)	(None, 5, 5, 224)	0	batch_normalization_1001[0][0]
conv2d_987 (Conv2D)	(None, 5, 5, 192)	399360	block8_3_ac[0][0]
conv2d_990 (Conv2D)	(None, 5, 5, 256)	172032	activation_989[0][0]
batch_normalization_999 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_987[0][0]
batch_normalization_1002 (Batch Normalization)	(None, 5, 5, 256)	768	conv2d_990[0][0]
activation_987 (Activation)	(None, 5, 5, 192)	0	batch_normalization_999[0][0]
activation_990 (Activation)	(None, 5, 5, 256)	0	batch_normalization_1002[0][0]
block8_4_mixed (Concatenate)	(None, 5, 5, 448)	0	activation_987[0][0] activation_990[0][0]
block8_4_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_4_mixed[0][0]
block8_4 (Lambda)	(None, 5, 5, 2080)	0	block8_3_ac[0][0] block8_4_conv[0][0]
block8_4_ac (Activation)	(None, 5, 5, 2080)	0	block8_4[0][0]
conv2d_992 (Conv2D)	(None, 5, 5, 192)	399360	block8_4_ac[0][0]
batch_normalization_1004 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_992[0][0]
activation_992 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1004[0][0]
conv2d_993 (Conv2D)	(None, 5, 5, 224)	129024	activation_992[0][0]
batch_normalization_1005 (Batch Normalization)	(None, 5, 5, 224)	672	conv2d_993[0][0]
activation_993 (Activation)	(None, 5, 5, 224)	0	batch_normalization_1005[0][0]
conv2d_991 (Conv2D)	(None, 5, 5, 192)	399360	block8_4_ac[0][0]
conv2d_994 (Conv2D)	(None, 5, 5, 256)	172032	activation_993[0][0]
batch_normalization_1003 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_991[0][0]
batch_normalization_1006 (Batch Normalization)	(None, 5, 5, 256)	768	conv2d_994[0][0]
activation_991 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1003[0][0]
activation_994 (Activation)	(None, 5, 5, 256)	0	batch_normalization_1006[0][0]
block8_5_mixed (Concatenate)	(None, 5, 5, 448)	0	activation_991[0][0] activation_994[0][0]
block8_5_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_5_mixed[0][0]
block8_5 (Lambda)	(None, 5, 5, 2080)	0	block8_4_ac[0][0] block8_5_conv[0][0]
block8_5_ac (Activation)	(None, 5, 5, 2080)	0	block8_5[0][0]
conv2d_996 (Conv2D)	(None, 5, 5, 192)	399360	block8_5_ac[0][0]
batch_normalization_1008 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_996[0][0]
activation_996 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1008[0][0]
conv2d_997 (Conv2D)	(None, 5, 5, 224)	129024	activation_996[0][0]



batch_normalization_1009 (Batch Normalization)	(None, 5, 5, 224)	672	conv2d_997[0][0]
activation_997 (Activation)	(None, 5, 5, 224)	0	batch_normalization_1009[0][0]
conv2d_995 (Conv2D)	(None, 5, 5, 192)	399360	block8_5_ac[0][0]
conv2d_998 (Conv2D)	(None, 5, 5, 256)	172032	activation_997[0][0]
batch_normalization_1007 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_995[0][0]
batch_normalization_1010 (Batch Normalization)	(None, 5, 5, 256)	768	conv2d_998[0][0]
activation_995 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1007[0][0]
activation_998 (Activation)	(None, 5, 5, 256)	0	batch_normalization_1010[0][0]
block8_6_mixed (Concatenate)	(None, 5, 5, 448)	0	activation_995[0][0] activation_998[0][0]
block8_6_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_6_mixed[0][0]
block8_6 (Lambda)	(None, 5, 5, 2080)	0	block8_5_ac[0][0] block8_6_conv[0][0]
block8_6_ac (Activation)	(None, 5, 5, 2080)	0	block8_6[0][0]
conv2d_1000 (Conv2D)	(None, 5, 5, 192)	399360	block8_6_ac[0][0]
batch_normalization_1012 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_1000[0][0]
activation_1000 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1012[0][0]
conv2d_1001 (Conv2D)	(None, 5, 5, 224)	129024	activation_1000[0][0]
batch_normalization_1013 (Batch Normalization)	(None, 5, 5, 224)	672	conv2d_1001[0][0]
activation_1001 (Activation)	(None, 5, 5, 224)	0	batch_normalization_1013[0][0]
conv2d_999 (Conv2D)	(None, 5, 5, 192)	399360	block8_6_ac[0][0]
conv2d_1002 (Conv2D)	(None, 5, 5, 256)	172032	activation_1001[0][0]
batch_normalization_1011 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_999[0][0]
batch_normalization_1014 (Batch Normalization)	(None, 5, 5, 256)	768	conv2d_1002[0][0]
activation_999 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1011[0][0]
activation_1002 (Activation)	(None, 5, 5, 256)	0	batch_normalization_1014[0][0]
block8_7_mixed (Concatenate)	(None, 5, 5, 448)	0	activation_999[0][0] activation_1002[0][0]
block8_7_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_7_mixed[0][0]
block8_7 (Lambda)	(None, 5, 5, 2080)	0	block8_6_ac[0][0] block8_7_conv[0][0]
block8_7_ac (Activation)	(None, 5, 5, 2080)	0	block8_7[0][0]
conv2d_1004 (Conv2D)	(None, 5, 5, 192)	399360	block8_7_ac[0][0]
batch_normalization_1016 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_1004[0][0]
activation_1004 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1016[0][0]
conv2d_1005 (Conv2D)	(None, 5, 5, 224)	129024	activation_1004[0][0]
batch_normalization_1017 (Batch Normalization)	(None, 5, 5, 224)	672	conv2d_1005[0][0]
activation_1005 (Activation)	(None, 5, 5, 224)	0	batch_normalization_1017[0][0]
conv2d_1003 (Conv2D)	(None, 5, 5, 192)	399360	block8_7_ac[0][0]
conv2d_1006 (Conv2D)	(None, 5, 5, 256)	172032	activation_1005[0][0]
batch_normalization_1015 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_1003[0][0]
batch_normalization_1018 (Batch Normalization)	(None, 5, 5, 256)	768	conv2d_1006[0][0]
activation_1003 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1015[0][0]
activation_1006 (Activation)	(None, 5, 5, 256)	0	batch_normalization_1018[0][0]
block8_8_mixed (Concatenate)	(None, 5, 5, 448)	0	activation_1003[0][0] activation_1006[0][0]
block8_8_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_8_mixed[0][0]

block8_8 (Lambda)	(None, 5, 5, 2080)	0	block8_7_ac[0][0] block8_8_conv[0][0]
block8_8_ac (Activation)	(None, 5, 5, 2080)	0	block8_8[0][0]
conv2d_1008 (Conv2D)	(None, 5, 5, 192)	399360	block8_8_ac[0][0]
batch_normalization_1020 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_1008[0][0]
activation_1008 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1020[0][0]
conv2d_1009 (Conv2D)	(None, 5, 5, 224)	129024	activation_1008[0][0]
batch_normalization_1021 (Batch Normalization)	(None, 5, 5, 224)	672	conv2d_1009[0][0]
activation_1009 (Activation)	(None, 5, 5, 224)	0	batch_normalization_1021[0][0]
conv2d_1007 (Conv2D)	(None, 5, 5, 192)	399360	block8_8_ac[0][0]
conv2d_1010 (Conv2D)	(None, 5, 5, 256)	172032	activation_1009[0][0]
batch_normalization_1019 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_1007[0][0]
batch_normalization_1022 (Batch Normalization)	(None, 5, 5, 256)	768	conv2d_1010[0][0]
activation_1007 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1019[0][0]
activation_1010 (Activation)	(None, 5, 5, 256)	0	batch_normalization_1022[0][0]
block8_9_mixed (Concatenate)	(None, 5, 5, 448)	0	activation_1007[0][0] activation_1010[0][0]
block8_9_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_9_mixed[0][0]
block8_9 (Lambda)	(None, 5, 5, 2080)	0	block8_8_ac[0][0] block8_9_conv[0][0]
block8_9_ac (Activation)	(None, 5, 5, 2080)	0	block8_9[0][0]
conv2d_1012 (Conv2D)	(None, 5, 5, 192)	399360	block8_9_ac[0][0]
batch_normalization_1024 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_1012[0][0]
activation_1012 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1024[0][0]
conv2d_1013 (Conv2D)	(None, 5, 5, 224)	129024	activation_1012[0][0]
batch_normalization_1025 (Batch Normalization)	(None, 5, 5, 224)	672	conv2d_1013[0][0]
activation_1013 (Activation)	(None, 5, 5, 224)	0	batch_normalization_1025[0][0]
conv2d_1011 (Conv2D)	(None, 5, 5, 192)	399360	block8_9_ac[0][0]
conv2d_1014 (Conv2D)	(None, 5, 5, 256)	172032	activation_1013[0][0]
batch_normalization_1023 (Batch Normalization)	(None, 5, 5, 192)	576	conv2d_1011[0][0]
batch_normalization_1026 (Batch Normalization)	(None, 5, 5, 256)	768	conv2d_1014[0][0]
activation_1011 (Activation)	(None, 5, 5, 192)	0	batch_normalization_1023[0][0]
activation_1014 (Activation)	(None, 5, 5, 256)	0	batch_normalization_1026[0][0]
block8_10_mixed (Concatenate)	(None, 5, 5, 448)	0	activation_1011[0][0] activation_1014[0][0]
block8_10_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_10_mixed[0][0]
block8_10 (Lambda)	(None, 5, 5, 2080)	0	block8_9_ac[0][0] block8_10_conv[0][0]
conv_7b (Conv2D)	(None, 5, 5, 1536)	3194880	block8_10[0][0]
conv_7b_bn (Batch Normalization)	(None, 5, 5, 1536)	4608	conv_7b[0][0]
conv_7b_ac (Activation)	(None, 5, 5, 1536)	0	conv_7b_bn[0][0]
reshape_4 (Reshape)	(None, 25, 1536)	0	conv_7b_ac[0][0]
lstm_4 (LSTM)	(None, 25, 512)	4196352	reshape_4[0][0]
batch_normalization_1027 (Batch Normalization)	(None, 25, 512)	2048	lstm_4[0][0]
flatten (Flatten)	(None, 12800)	0	batch_normalization_1027[0][0]
dense_12 (Dense)	(None, 4096)	52432896	flatten[0][0]

batch_normalization_1028 (Batch (None, 4096)	16384	dense_12[0][0]
dense_13 (Dense)	(None, 4096)	16781312 batch_normalization_1028[0][0]
batch_normalization_1029 (Batch (None, 4096)	16384	dense_13[0][0]
dense_14 (Dense)	(None, 4)	16388 batch_normalization_1029[0][0]
=====		
Total params: 127,798,500		
Trainable params: 69,248,004		
Non-trainable params: 58,550,496		
=====		