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, Re
shape
from tensorflow.keras.applications import DenseNet169
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 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 Ndarray 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(20)
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)
```

DenseNet169-LSTM MODEL

In [13]:
```
def model_build_compile(k):
    baseModel = DenseNet169(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((49,1664))(x)
    x = ((LSTM(1664, 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 = []
```

DenseNet169-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 no
rmal 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 gluacoma validation set by using k*n_rest to (k+1)*n_rest as index values for gluacoma 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 im
ages 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 im
ages 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 im
ages 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 2
0% 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, rand
om_state=14,shuffle=True)
    glaucoma_train, glaucoma_validation            = train_test_split(train_validation_glaucoma, test_size=0.20, rand
om_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 sep
arated.
```

```
    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 reshap
ed 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 [==============================] - 119s 8s/step - loss: 19.6142 - accuracy: 0.4190 - val_loss: 51.2408 - val_ac
curacy: 0.4259
Epoch 2/50
14/14 [==============================] - 108s 8s/step - loss: 7.4453 - accuracy: 0.5486 - val_loss: 45.4294 - val_acc
uracy: 0.4722
Epoch 3/50
14/14 [==============================] - 103s 7s/step - loss: 8.9957 - accuracy: 0.5463 - val_loss: 19.3660 - val_acc
uracy: 0.4630
Epoch 4/50
14/14 [==============================] - 105s 8s/step - loss: 4.4261 - accuracy: 0.6296 - val_loss: 13.7295 - val_acc
uracy: 0.3889
Epoch 5/50
14/14 [==============================] - 102s 7s/step - loss: 3.7341 - accuracy: 0.6412 - val_loss: 8.6881 - val_accu
racy: 0.5370
Epoch 6/50
14/14 [==============================] - 105s 8s/step - loss: 2.4661 - accuracy: 0.6968 - val_loss: 7.4903 - val_accu
racy: 0.2870
Epoch 7/50
14/14 [==============================] - 107s 8s/step - loss: 1.6290 - accuracy: 0.7454 - val_loss: 6.7891 - val_accu
racy: 0.2593
Epoch 8/50
14/14 [==============================] - 112s 8s/step - loss: 2.1105 - accuracy: 0.6968 - val_loss: 5.9112 - val_accu
racy: 0.4630
Epoch 9/50
14/14 [==============================] - 102s 7s/step - loss: 1.1446 - accuracy: 0.8194 - val_loss: 16.7728 - val_acc
uracy: 0.1574
Epoch 10/50
14/14 [==============================] - 103s 7s/step - loss: 1.4188 - accuracy: 0.7755 - val_loss: 6.7945 - val_accu
racy: 0.4074
Epoch 11/50
14/14 [==============================] - 102s 7s/step - loss: 0.6320 - accuracy: 0.8611 - val_loss: 13.5503 - val_acc
uracy: 0.2963
Epoch 12/50
14/14 [==============================] - 103s 7s/step - loss: 0.5872 - accuracy: 0.8681 - val_loss: 14.4499 - val_acc
uracy: 0.2130
Epoch 13/50
14/14 [==============================] - 103s 7s/step - loss: 0.3462 - accuracy: 0.9213 - val_loss: 20.7879 - val_acc
uracy: 0.1667
Epoch 14/50
14/14 [==============================] - 103s 7s/step - loss: 0.7360 - accuracy: 0.8681 - val_loss: 21.5582 - val_acc
uracy: 0.1852
Epoch 15/50
14/14 [==============================] - 103s 7s/step - loss: 0.6273 - accuracy: 0.8843 - val_loss: 19.1663 - val_acc
uracy: 0.1759
Epoch 16/50
14/14 [==============================] - 102s 7s/step - loss: 0.8340 - accuracy: 0.8773 - val_loss: 4.1809 - val_accu
racy: 0.5000
Epoch 17/50
14/14 [==============================] - 102s 7s/step - loss: 0.1584 - accuracy: 0.9606 - val_loss: 10.6138 - val_acc
uracy: 0.3241
Epoch 18/50
14/14 [==============================] - 103s 7s/step - loss: 0.5361 - accuracy: 0.8935 - val_loss: 8.4215 - val_accu
racy: 0.3796
Epoch 19/50
14/14 [==============================] - 103s 7s/step - loss: 0.1910 - accuracy: 0.9630 - val_loss: 11.4896 - val_acc
uracy: 0.3426
Epoch 20/50
14/14 [==============================] - 101s 7s/step - loss: 0.0840 - accuracy: 0.9792 - val_loss: 6.0928 - val_accu
racy: 0.5093
Epoch 21/50
14/14 [==============================] - 105s 8s/step - loss: 0.5733 - accuracy: 0.9352 - val_loss: 24.1382 - val_acc
uracy: 0.1944
Epoch 22/50
14/14 [==============================] - 103s 7s/step - loss: 0.2023 - accuracy: 0.9514 - val_loss: 16.6201 - val_acc
uracy: 0.2222
Epoch 23/50
14/14 [==============================] - 104s 7s/step - loss: 0.1686 - accuracy: 0.9653 - val_loss: 4.2661 - val_accu
racy: 0.6296
Epoch 24/50
14/14 [==============================] - 103s 7s/step - loss: 0.1118 - accuracy: 0.9653 - val_loss: 3.7324 - val_accu
racy: 0.6574
Epoch 25/50
14/14 [==============================] - 103s 7s/step - loss: 0.3491 - accuracy: 0.9421 - val_loss: 3.5406 - val_accu
racy: 0.6481
Epoch 26/50
14/14 [==============================] - 105s 8s/step - loss: 0.0505 - accuracy: 0.9792 - val_loss: 3.7888 - val_accu
racy: 0.6759
```

```
Epoch 27/50
14/14 [==============================] - 104s 7s/step - loss: 0.1815 - accuracy: 0.9560 - val_loss: 5.6150 - val_accu
racy: 0.5648
Epoch 28/50
14/14 [==============================] - 108s 8s/step - loss: 0.2744 - accuracy: 0.9583 - val_loss: 6.5071 - val_accu
racy: 0.5000
Epoch 29/50
14/14 [==============================] - 103s 7s/step - loss: 0.0481 - accuracy: 0.9769 - val_loss: 5.7111 - val_accu
racy: 0.5185
Epoch 30/50
14/14 [==============================] - 104s 8s/step - loss: 0.2471 - accuracy: 0.9537 - val_loss: 6.0646 - val_accu
racy: 0.5741
Epoch 31/50
14/14 [==============================] - 104s 7s/step - loss: 0.1051 - accuracy: 0.9722 - val_loss: 6.1058 - val_accu
racy: 0.3981
Epoch 32/50
14/14 [==============================] - 103s 7s/step - loss: 0.0689 - accuracy: 0.9838 - val_loss: 4.4550 - val_accu
racy: 0.6296
Epoch 33/50
14/14 [==============================] - 104s 7s/step - loss: 0.0360 - accuracy: 0.9884 - val_loss: 11.1949 - val_acc
uracy: 0.4537
Epoch 34/50
14/14 [==============================] - 106s 8s/step - loss: 0.2521 - accuracy: 0.9537 - val_loss: 4.5878 - val_accu
racy: 0.6389
Epoch 35/50
14/14 [==============================] - 104s 7s/step - loss: 0.0050 - accuracy: 0.9977 - val_loss: 4.3455 - val_accu
racy: 0.6019
Epoch 36/50
14/14 [==============================] - 106s 8s/step - loss: 0.2535 - accuracy: 0.9560 - val_loss: 5.6172 - val_accu
racy: 0.6481
Epoch 37/50
14/14 [==============================] - 111s 8s/step - loss: 0.0618 - accuracy: 0.9884 - val_loss: 6.3743 - val_accu
racy: 0.5093
Epoch 38/50
14/14 [==============================] - 107s 8s/step - loss: 0.1157 - accuracy: 0.9653 - val_loss: 5.4469 - val_accu
racy: 0.6111
Epoch 39/50
14/14 [==============================] - 108s 8s/step - loss: 0.2887 - accuracy: 0.9606 - val_loss: 4.1783 - val_accu
racy: 0.7130
Epoch 40/50
14/14 [==============================] - 110s 8s/step - loss: 0.0359 - accuracy: 0.9884 - val_loss: 4.5588 - val_accu
racy: 0.5833
Epoch 41/50
14/14 [==============================] - 108s 8s/step - loss: 0.0994 - accuracy: 0.9769 - val_loss: 7.9604 - val_accu
racy: 0.6019
Epoch 42/50
14/14 [==============================] - 110s 8s/step - loss: 0.0507 - accuracy: 0.9907 - val_loss: 5.0212 - val_accu
racy: 0.6389
Epoch 43/50
14/14 [==============================] - 108s 8s/step - loss: 0.0570 - accuracy: 0.9861 - val_loss: 5.4071 - val_accu
racy: 0.5463
Epoch 44/50
14/14 [==============================] - 111s 8s/step - loss: 0.1083 - accuracy: 0.9745 - val_loss: 5.0123 - val_accu
racy: 0.6019
Epoch 45/50
14/14 [==============================] - 108s 8s/step - loss: 0.0779 - accuracy: 0.9838 - val_loss: 4.4291 - val_accu
racy: 0.6296
Epoch 46/50
14/14 [==============================] - 110s 8s/step - loss: 0.2308 - accuracy: 0.9560 - val_loss: 4.3835 - val_accu
racy: 0.6667
Epoch 47/50
14/14 [==============================] - 110s 8s/step - loss: 0.0013 - accuracy: 1.0000 - val_loss: 4.0925 - val_accu
racy: 0.6759
Epoch 48/50
14/14 [==============================] - 110s 8s/step - loss: 0.0709 - accuracy: 0.9792 - val_loss: 5.6711 - val_accu
racy: 0.6296
Epoch 49/50
14/14 [==============================] - 110s 8s/step - loss: 0.0210 - accuracy: 0.9931 - val_loss: 5.1912 - val_accu
racy: 0.7037
Epoch 50/50
14/14 [==============================] - 109s 8s/step - loss: 0.1394 - accuracy: 0.9792 - val_loss: 5.9227 - val_accu
racy: 0.6574
----------------Test accuracy for 1 fold----------------
Confusion Matrix :
 [[22  0  6  2]
 [ 1  7  1  1]
 [ 3  0  6  1]
 [ 3  1  4  2]]
Accuracy    :  0.6166666666666667
Specificity :  0.821272813208297
Sensitivity :  0.5583333333333333
----------------------------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 [==============================] - 136s 9s/step - loss: 19.2123 - accuracy: 0.4120 - val_loss: 88.9085 - val_ac
curacy: 0.5648
Epoch 2/50
14/14 [==============================] - 114s 8s/step - loss: 7.4640 - accuracy: 0.5324 - val_loss: 50.6491 - val_acc
uracy: 0.4815
Epoch 3/50
14/14 [==============================] - 119s 9s/step - loss: 6.4589 - accuracy: 0.5579 - val_loss: 14.6420 - val_acc
uracy: 0.4167
Epoch 4/50
14/14 [==============================] - 121s 9s/step - loss: 6.6755 - accuracy: 0.5718 - val_loss: 13.1836 - val_acc
uracy: 0.4352
Epoch 5/50
14/14 [==============================] - 122s 9s/step - loss: 2.8604 - accuracy: 0.6829 - val_loss: 13.9169 - val_acc
uracy: 0.5278
Epoch 6/50
14/14 [==============================] - 117s 8s/step - loss: 3.0463 - accuracy: 0.6505 - val_loss: 15.3263 - val_acc
uracy: 0.4074
Epoch 7/50
14/14 [==============================] - 118s 8s/step - loss: 1.8698 - accuracy: 0.6782 - val_loss: 3.5099 - val_accu
racy: 0.5833
Epoch 8/50
14/14 [==============================] - 114s 8s/step - loss: 2.3297 - accuracy: 0.6852 - val_loss: 10.3166 - val_acc
uracy: 0.4630
Epoch 9/50
14/14 [==============================] - 110s 8s/step - loss: 1.3198 - accuracy: 0.7731 - val_loss: 6.1202 - val_accu
racy: 0.5463
Epoch 10/50
14/14 [==============================] - 111s 8s/step - loss: 0.8572 - accuracy: 0.8218 - val_loss: 3.8769 - val_accu
racy: 0.5000
Epoch 11/50
14/14 [==============================] - 111s 8s/step - loss: 1.2475 - accuracy: 0.7847 - val_loss: 4.1754 - val_accu
racy: 0.4537
Epoch 12/50
14/14 [==============================] - 112s 8s/step - loss: 0.3110 - accuracy: 0.9144 - val_loss: 3.4755 - val_accu
racy: 0.6389
Epoch 13/50
14/14 [==============================] - 112s 8s/step - loss: 0.9840 - accuracy: 0.8356 - val_loss: 3.6801 - val_accu
racy: 0.3889
Epoch 14/50
14/14 [==============================] - 111s 8s/step - loss: 0.9533 - accuracy: 0.8796 - val_loss: 3.7217 - val_accu
racy: 0.5093
Epoch 15/50
14/14 [==============================] - 111s 8s/step - loss: 0.5524 - accuracy: 0.8981 - val_loss: 2.9090 - val_accu
racy: 0.5833
Epoch 16/50
14/14 [==============================] - 112s 8s/step - loss: 0.6169 - accuracy: 0.9259 - val_loss: 3.5565 - val_accu
racy: 0.5463
Epoch 17/50
14/14 [==============================] - 111s 8s/step - loss: 0.4393 - accuracy: 0.9259 - val_loss: 2.6278 - val_accu
racy: 0.6019
Epoch 18/50
14/14 [==============================] - 112s 8s/step - loss: 0.2385 - accuracy: 0.9398 - val_loss: 3.5166 - val_accu
racy: 0.5463
Epoch 19/50
14/14 [==============================] - 112s 8s/step - loss: 0.5852 - accuracy: 0.8935 - val_loss: 5.8075 - val_accu
racy: 0.5648
Epoch 20/50
14/14 [==============================] - 111s 8s/step - loss: 0.1368 - accuracy: 0.9606 - val_loss: 4.9039 - val_accu
racy: 0.5185
Epoch 21/50
14/14 [==============================] - 112s 8s/step - loss: 0.2815 - accuracy: 0.9398 - val_loss: 6.7060 - val_accu
racy: 0.6019
Epoch 22/50
14/14 [==============================] - 111s 8s/step - loss: 0.6639 - accuracy: 0.9097 - val_loss: 6.5767 - val_accu
racy: 0.6204
Epoch 23/50
14/14 [==============================] - 113s 8s/step - loss: 0.2455 - accuracy: 0.9491 - val_loss: 8.6185 - val_accu
racy: 0.6111
Epoch 24/50
14/14 [==============================] - 112s 8s/step - loss: 0.3513 - accuracy: 0.9491 - val_loss: 4.2082 - val_accu
racy: 0.6111
Epoch 25/50
14/14 [==============================] - 112s 8s/step - loss: 0.1447 - accuracy: 0.9676 - val_loss: 5.6313 - val_accu
racy: 0.5370
Epoch 26/50
14/14 [==============================] - 112s 8s/step - loss: 0.1217 - accuracy: 0.9745 - val_loss: 4.2440 - val_accu
racy: 0.6204
Epoch 27/50
14/14 [==============================] - 112s 8s/step - loss: 0.2469 - accuracy: 0.9537 - val_loss: 7.8113 - val_accu
racy: 0.4907
Epoch 28/50
14/14 [==============================] - 112s 8s/step - loss: 0.4100 - accuracy: 0.9306 - val_loss: 5.3496 - val_accu
racy: 0.6019
```

```
Epoch 29/50
14/14 [==============================] - 112s 8s/step - loss: 0.1757 - accuracy: 0.9653 - val_loss: 3.6624 - val_accu
racy: 0.6389
Epoch 30/50
14/14 [==============================] - 111s 8s/step - loss: 0.0275 - accuracy: 0.9907 - val_loss: 4.6382 - val_accu
racy: 0.5926
Epoch 31/50
14/14 [==============================] - 115s 8s/step - loss: 0.2773 - accuracy: 0.9514 - val_loss: 5.4130 - val_accu
racy: 0.5093
Epoch 32/50
14/14 [==============================] - 114s 8s/step - loss: 0.0423 - accuracy: 0.9861 - val_loss: 3.9356 - val_accu
racy: 0.6296
Epoch 33/50
14/14 [==============================] - 112s 8s/step - loss: 0.0570 - accuracy: 0.9884 - val_loss: 6.4003 - val_accu
racy: 0.5370
Epoch 34/50
14/14 [==============================] - 113s 8s/step - loss: 0.2859 - accuracy: 0.9421 - val_loss: 6.6293 - val_accu
racy: 0.5278
Epoch 35/50
14/14 [==============================] - 113s 8s/step - loss: 0.2920 - accuracy: 0.9537 - val_loss: 4.5580 - val_accu
racy: 0.6389
Epoch 36/50
14/14 [==============================] - 112s 8s/step - loss: 0.1186 - accuracy: 0.9745 - val_loss: 5.8873 - val_accu
racy: 0.6111
Epoch 37/50
14/14 [==============================] - 115s 8s/step - loss: 0.0506 - accuracy: 0.9884 - val_loss: 5.5166 - val_accu
racy: 0.5926
Epoch 38/50
14/14 [==============================] - 111s 8s/step - loss: 0.1577 - accuracy: 0.9722 - val_loss: 5.7810 - val_accu
racy: 0.6481
Epoch 39/50
14/14 [==============================] - 112s 8s/step - loss: 0.0484 - accuracy: 0.9838 - val_loss: 8.5847 - val_accu
racy: 0.6111
Epoch 40/50
14/14 [==============================] - 112s 8s/step - loss: 0.1719 - accuracy: 0.9630 - val_loss: 8.4987 - val_accu
racy: 0.6296
Epoch 41/50
14/14 [==============================] - 112s 8s/step - loss: 0.1133 - accuracy: 0.9769 - val_loss: 11.4574 - val_acc
uracy: 0.4815
Epoch 42/50
14/14 [==============================] - 112s 8s/step - loss: 0.1443 - accuracy: 0.9838 - val_loss: 5.9056 - val_accu
racy: 0.5833
Epoch 43/50
14/14 [==============================] - 116s 8s/step - loss: 0.0022 - accuracy: 0.9977 - val_loss: 5.5218 - val_accu
racy: 0.6296
Epoch 44/50
14/14 [==============================] - 122s 9s/step - loss: 0.0223 - accuracy: 0.9931 - val_loss: 6.0823 - val_accu
racy: 0.6296
Epoch 45/50
14/14 [==============================] - 112s 8s/step - loss: 0.1548 - accuracy: 0.9676 - val_loss: 7.6996 - val_accu
racy: 0.6481
Epoch 46/50
14/14 [==============================] - 113s 8s/step - loss: 0.1041 - accuracy: 0.9792 - val_loss: 10.9499 - val_acc
uracy: 0.5278
Epoch 47/50
14/14 [==============================] - 113s 8s/step - loss: 0.2251 - accuracy: 0.9653 - val_loss: 7.5719 - val_accu
racy: 0.6204
Epoch 48/50
14/14 [==============================] - 112s 8s/step - loss: 0.0349 - accuracy: 0.9884 - val_loss: 7.0891 - val_accu
racy: 0.6296
Epoch 49/50
14/14 [==============================] - 112s 8s/step - loss: 0.1152 - accuracy: 0.9792 - val_loss: 7.3951 - val_accu
racy: 0.6481
Epoch 50/50
14/14 [==============================] - 112s 8s/step - loss: 0.1236 - accuracy: 0.9838 - val_loss: 6.4268 - val_accu
racy: 0.6296
----------------Test accuracy for 2 fold----------------
Confusion Matrix :
 [[24  2  2  2]
 [ 1  8  1  0]
 [ 4  1  5  0]
 [ 3  1  3  3]]
Accuracy    :  0.6666666666666666
Specificity :  0.8394830102147175
Sensitivity :  0.6
----------------------------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 [==============================] - 147s 10s/step - loss: 17.3792 - accuracy: 0.3981 - val_loss: 100.7148 - val_
```

```
accuracy: 0.5093
Epoch 2/50
14/14 [==============================] - 114s 8s/step - loss: 6.7905 - accuracy: 0.5255 - val_loss: 56.8900 - val_acc
uracy: 0.4907
Epoch 3/50
14/14 [==============================] - 113s 8s/step - loss: 7.0493 - accuracy: 0.5694 - val_loss: 17.8336 - val_acc
uracy: 0.3611
Epoch 4/50
14/14 [==============================] - 114s 8s/step - loss: 3.9897 - accuracy: 0.5926 - val_loss: 27.1339 - val_acc
uracy: 0.4907
Epoch 5/50
14/14 [==============================] - 114s 8s/step - loss: 4.8464 - accuracy: 0.5856 - val_loss: 7.7560 - val_accu
racy: 0.3981
Epoch 6/50
14/14 [==============================] - 113s 8s/step - loss: 1.8717 - accuracy: 0.6759 - val_loss: 8.9662 - val_accu
racy: 0.2500
Epoch 7/50
14/14 [==============================] - 113s 8s/step - loss: 1.4406 - accuracy: 0.7245 - val_loss: 11.8587 - val_acc
uracy: 0.3519
Epoch 8/50
14/14 [==============================] - 113s 8s/step - loss: 1.3601 - accuracy: 0.7523 - val_loss: 12.9869 - val_acc
uracy: 0.2037
Epoch 9/50
14/14 [==============================] - 113s 8s/step - loss: 1.2253 - accuracy: 0.7662 - val_loss: 5.6314 - val_accu
racy: 0.6111
Epoch 10/50
14/14 [==============================] - 114s 8s/step - loss: 1.1452 - accuracy: 0.7731 - val_loss: 9.0379 - val_accu
racy: 0.4259
Epoch 11/50
14/14 [==============================] - 113s 8s/step - loss: 0.9108 - accuracy: 0.8426 - val_loss: 4.8572 - val_accu
racy: 0.4815
Epoch 12/50
14/14 [==============================] - 113s 8s/step - loss: 0.4701 - accuracy: 0.9005 - val_loss: 7.5270 - val_accu
racy: 0.3519
Epoch 13/50
14/14 [==============================] - 113s 8s/step - loss: 0.9767 - accuracy: 0.8588 - val_loss: 7.2977 - val_accu
racy: 0.4352
Epoch 14/50
14/14 [==============================] - 113s 8s/step - loss: 0.6001 - accuracy: 0.8773 - val_loss: 7.1348 - val_accu
racy: 0.2870
Epoch 15/50
14/14 [==============================] - 113s 8s/step - loss: 0.6448 - accuracy: 0.8727 - val_loss: 6.5287 - val_accu
racy: 0.5000
Epoch 16/50
14/14 [==============================] - 113s 8s/step - loss: 0.4554 - accuracy: 0.9352 - val_loss: 9.7186 - val_accu
racy: 0.4815
Epoch 17/50
14/14 [==============================] - 114s 8s/step - loss: 0.4114 - accuracy: 0.9329 - val_loss: 9.9129 - val_accu
racy: 0.2778
Epoch 18/50
14/14 [==============================] - 113s 8s/step - loss: 0.2992 - accuracy: 0.9306 - val_loss: 6.6670 - val_accu
racy: 0.3981
Epoch 19/50
14/14 [==============================] - 112s 8s/step - loss: 0.7447 - accuracy: 0.9005 - val_loss: 6.8069 - val_accu
racy: 0.5556
Epoch 20/50
14/14 [==============================] - 113s 8s/step - loss: 0.2323 - accuracy: 0.9329 - val_loss: 4.3706 - val_accu
racy: 0.5556
Epoch 21/50
14/14 [==============================] - 113s 8s/step - loss: 0.2278 - accuracy: 0.9468 - val_loss: 7.6577 - val_accu
racy: 0.3519
Epoch 22/50
14/14 [==============================] - 113s 8s/step - loss: 0.4060 - accuracy: 0.9514 - val_loss: 4.1583 - val_accu
racy: 0.6204
Epoch 23/50
14/14 [==============================] - 113s 8s/step - loss: 0.3368 - accuracy: 0.9514 - val_loss: 3.8651 - val_accu
racy: 0.6389
Epoch 24/50
14/14 [==============================] - 113s 8s/step - loss: 0.1487 - accuracy: 0.9583 - val_loss: 10.0664 - val_acc
uracy: 0.3056
Epoch 25/50
14/14 [==============================] - 112s 8s/step - loss: 0.1885 - accuracy: 0.9676 - val_loss: 4.6094 - val_accu
racy: 0.6204
Epoch 26/50
14/14 [==============================] - 114s 8s/step - loss: 0.1147 - accuracy: 0.9745 - val_loss: 10.3350 - val_acc
uracy: 0.3056
Epoch 27/50
14/14 [==============================] - 114s 8s/step - loss: 0.4135 - accuracy: 0.9352 - val_loss: 7.4677 - val_accu
racy: 0.3796
Epoch 28/50
14/14 [==============================] - 114s 8s/step - loss: 0.1635 - accuracy: 0.9722 - val_loss: 7.5920 - val_accu
racy: 0.5648
Epoch 29/50
14/14 [==============================] - 114s 8s/step - loss: 0.3054 - accuracy: 0.9491 - val_loss: 7.1513 - val_accu
racy: 0.5000
Epoch 30/50
14/14 [==============================] - 113s 8s/step - loss: 0.0385 - accuracy: 0.9931 - val_loss: 4.5989 - val_accu
racy: 0.6296
```

```
Epoch 31/50
14/14 [==============================] - 113s 8s/step - loss: 0.2590 - accuracy: 0.9514 - val_loss: 6.3590 - val_accu
racy: 0.5278
Epoch 32/50
14/14 [==============================] - 112s 8s/step - loss: 0.3072 - accuracy: 0.9329 - val_loss: 9.6221 - val_accu
racy: 0.4630
Epoch 33/50
14/14 [==============================] - 112s 8s/step - loss: 0.0361 - accuracy: 0.9861 - val_loss: 5.6672 - val_accu
racy: 0.5648
Epoch 34/50
14/14 [==============================] - 113s 8s/step - loss: 0.2448 - accuracy: 0.9722 - val_loss: 36.1429 - val_acc
uracy: 0.5093
Epoch 35/50
14/14 [==============================] - 109s 8s/step - loss: 0.7485 - accuracy: 0.9282 - val_loss: 5.2059 - val_accu
racy: 0.6111
Epoch 36/50
14/14 [==============================] - 111s 8s/step - loss: 0.1194 - accuracy: 0.9815 - val_loss: 6.6993 - val_accu
racy: 0.4907
Epoch 37/50
14/14 [==============================] - 113s 8s/step - loss: 0.1674 - accuracy: 0.9676 - val_loss: 6.3149 - val_accu
racy: 0.5370
Epoch 38/50
14/14 [==============================] - 109s 8s/step - loss: 0.1114 - accuracy: 0.9699 - val_loss: 7.2948 - val_accu
racy: 0.6204
Epoch 39/50
14/14 [==============================] - 110s 8s/step - loss: 0.0945 - accuracy: 0.9676 - val_loss: 5.3579 - val_accu
racy: 0.6389
Epoch 40/50
14/14 [==============================] - 107s 8s/step - loss: 0.0501 - accuracy: 0.9861 - val_loss: 5.6308 - val_accu
racy: 0.6481
Epoch 41/50
14/14 [==============================] - 113s 8s/step - loss: 0.0035 - accuracy: 1.0000 - val_loss: 6.0927 - val_accu
racy: 0.6574
Epoch 42/50
14/14 [==============================] - 110s 8s/step - loss: 0.0054 - accuracy: 1.0000 - val_loss: 6.8629 - val_accu
racy: 0.6944
Epoch 43/50
14/14 [==============================] - 108s 8s/step - loss: 0.1932 - accuracy: 0.9630 - val_loss: 6.2774 - val_accu
racy: 0.6111
Epoch 44/50
14/14 [==============================] - 108s 8s/step - loss: 0.1087 - accuracy: 0.9769 - val_loss: 5.6580 - val_accu
racy: 0.6667
Epoch 45/50
14/14 [==============================] - 108s 8s/step - loss: 0.0174 - accuracy: 0.9954 - val_loss: 7.2881 - val_accu
racy: 0.6019
Epoch 46/50
14/14 [==============================] - 106s 8s/step - loss: 0.0575 - accuracy: 0.9884 - val_loss: 14.3391 - val_acc
uracy: 0.3889
Epoch 47/50
14/14 [==============================] - 107s 8s/step - loss: 0.1234 - accuracy: 0.9792 - val_loss: 8.5741 - val_accu
racy: 0.6019
Epoch 48/50
14/14 [==============================] - 109s 8s/step - loss: 0.0932 - accuracy: 0.9815 - val_loss: 7.8272 - val_accu
racy: 0.6204
Epoch 49/50
14/14 [==============================] - 110s 8s/step - loss: 0.1215 - accuracy: 0.9792 - val_loss: 10.9553 - val_acc
uracy: 0.5000
Epoch 50/50
14/14 [==============================] - 112s 8s/step - loss: 0.0481 - accuracy: 0.9884 - val_loss: 6.7980 - val_accu
racy: 0.6204
----------------Test accuracy for 3 fold----------------
Confusion Matrix :
 [[15  0  9  6]
 [ 2  2  5  1]
 [ 2  0  7  1]
 [ 7  1  1  1]]
Accuracy    :  0.4166666666666667
Specificity :  0.6824945887445888
Sensitivity :  0.375
-----------------------------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 [==============================] - 143s 9s/step - loss: 18.7327 - accuracy: 0.3889 - val_loss: 132.1708 - val_a
ccuracy: 0.4907
Epoch 2/50
14/14 [==============================] - 120s 8s/step - loss: 8.6955 - accuracy: 0.5301 - val_loss: 20.6556 - val_acc
uracy: 0.5463
Epoch 3/50
14/14 [==============================] - 115s 8s/step - loss: 5.4894 - accuracy: 0.6042 - val_loss: 12.8990 - val_acc
```

```
uracy: 0.3333
Epoch 4/50
14/14 [==============================] - 114s 8s/step - loss: 4.9625 - accuracy: 0.6319 - val_loss: 12.4099 - val_acc
uracy: 0.4907
Epoch 5/50
14/14 [==============================] - 114s 8s/step - loss: 3.3453 - accuracy: 0.6458 - val_loss: 14.2004 - val_acc
uracy: 0.2222
Epoch 6/50
14/14 [==============================] - 114s 8s/step - loss: 2.0479 - accuracy: 0.7176 - val_loss: 16.3184 - val_acc
uracy: 0.1944
Epoch 7/50
14/14 [==============================] - 115s 8s/step - loss: 1.4248 - accuracy: 0.7546 - val_loss: 28.8525 - val_acc
uracy: 0.1759
Epoch 8/50
14/14 [==============================] - 115s 8s/step - loss: 2.0713 - accuracy: 0.7870 - val_loss: 22.2815 - val_acc
uracy: 0.1759
Epoch 9/50
14/14 [==============================] - 114s 8s/step - loss: 1.2346 - accuracy: 0.8009 - val_loss: 15.9119 - val_acc
uracy: 0.2130
Epoch 10/50
14/14 [==============================] - 114s 8s/step - loss: 1.2751 - accuracy: 0.8056 - val_loss: 38.7317 - val_acc
uracy: 0.1667
Epoch 11/50
14/14 [==============================] - 115s 8s/step - loss: 1.4835 - accuracy: 0.8032 - val_loss: 14.1036 - val_acc
uracy: 0.1667
Epoch 12/50
14/14 [==============================] - 115s 8s/step - loss: 1.2172 - accuracy: 0.8819 - val_loss: 17.4378 - val_acc
uracy: 0.1667
Epoch 13/50
14/14 [==============================] - 116s 8s/step - loss: 0.5677 - accuracy: 0.8727 - val_loss: 27.3761 - val_acc
uracy: 0.1667
Epoch 14/50
14/14 [==============================] - 115s 8s/step - loss: 0.6133 - accuracy: 0.8981 - val_loss: 17.4635 - val_acc
uracy: 0.1667
Epoch 15/50
14/14 [==============================] - 115s 8s/step - loss: 0.4409 - accuracy: 0.9167 - val_loss: 21.1938 - val_acc
uracy: 0.1667
Epoch 16/50
14/14 [==============================] - 114s 8s/step - loss: 0.3630 - accuracy: 0.9306 - val_loss: 10.3342 - val_acc
uracy: 0.3333
Epoch 17/50
14/14 [==============================] - 115s 8s/step - loss: 0.7789 - accuracy: 0.8981 - val_loss: 13.8745 - val_acc
uracy: 0.2407
Epoch 18/50
14/14 [==============================] - 115s 8s/step - loss: 0.5584 - accuracy: 0.9144 - val_loss: 14.1543 - val_acc
uracy: 0.1852
Epoch 19/50
14/14 [==============================] - 115s 8s/step - loss: 0.0504 - accuracy: 0.9838 - val_loss: 16.6924 - val_acc
uracy: 0.2315
Epoch 20/50
14/14 [==============================] - 114s 8s/step - loss: 0.1573 - accuracy: 0.9676 - val_loss: 15.6967 - val_acc
uracy: 0.2870
Epoch 21/50
14/14 [==============================] - 115s 8s/step - loss: 0.5450 - accuracy: 0.9097 - val_loss: 13.4428 - val_acc
uracy: 0.3704
Epoch 22/50
14/14 [==============================] - 114s 8s/step - loss: 0.3568 - accuracy: 0.9537 - val_loss: 13.9705 - val_acc
uracy: 0.3333
Epoch 23/50
14/14 [==============================] - 115s 8s/step - loss: 0.0834 - accuracy: 0.9745 - val_loss: 18.2842 - val_acc
uracy: 0.3333
Epoch 24/50
14/14 [==============================] - 114s 8s/step - loss: 0.4601 - accuracy: 0.9514 - val_loss: 12.1834 - val_acc
uracy: 0.3056
Epoch 25/50
14/14 [==============================] - 114s 8s/step - loss: 0.2395 - accuracy: 0.9606 - val_loss: 28.6205 - val_acc
uracy: 0.2593
Epoch 26/50
14/14 [==============================] - 118s 8s/step - loss: 0.3868 - accuracy: 0.9491 - val_loss: 11.6926 - val_acc
uracy: 0.4352
Epoch 27/50
14/14 [==============================] - 114s 8s/step - loss: 0.1889 - accuracy: 0.9606 - val_loss: 7.6331 - val_accu
racy: 0.4630
Epoch 28/50
14/14 [==============================] - 116s 8s/step - loss: 0.2676 - accuracy: 0.9699 - val_loss: 12.5993 - val_acc
uracy: 0.4537
Epoch 29/50
14/14 [==============================] - 114s 8s/step - loss: 0.3500 - accuracy: 0.9375 - val_loss: 9.3722 - val_accu
racy: 0.6389
Epoch 30/50
14/14 [==============================] - 115s 8s/step - loss: 0.1350 - accuracy: 0.9745 - val_loss: 7.4182 - val_accu
racy: 0.5648
Epoch 31/50
14/14 [==============================] - 115s 8s/step - loss: 0.1923 - accuracy: 0.9606 - val_loss: 7.2702 - val_accu
racy: 0.5556
Epoch 32/50
14/14 [==============================] - 114s 8s/step - loss: 0.1121 - accuracy: 0.9769 - val_loss: 6.8969 - val_accu
racy: 0.5370
```

```
Epoch 33/50
14/14 [==============================] - 114s 8s/step - loss: 0.0524 - accuracy: 0.9884 - val_loss: 7.8566 - val_accu
racy: 0.5648
Epoch 34/50
14/14 [==============================] - 114s 8s/step - loss: 0.2795 - accuracy: 0.9653 - val_loss: 8.9224 - val_accu
racy: 0.4907
Epoch 35/50
14/14 [==============================] - 114s 8s/step - loss: 0.2162 - accuracy: 0.9653 - val_loss: 8.5659 - val_accu
racy: 0.5278
Epoch 36/50
14/14 [==============================] - 114s 8s/step - loss: 0.0478 - accuracy: 0.9838 - val_loss: 8.4045 - val_accu
racy: 0.5185
Epoch 37/50
14/14 [==============================] - 114s 8s/step - loss: 0.3267 - accuracy: 0.9560 - val_loss: 11.0007 - val_acc
uracy: 0.4537
Epoch 38/50
14/14 [==============================] - 115s 8s/step - loss: 0.4375 - accuracy: 0.9560 - val_loss: 9.6377 - val_accu
racy: 0.4444
Epoch 39/50
14/14 [==============================] - 115s 8s/step - loss: 0.0143 - accuracy: 0.9977 - val_loss: 8.7234 - val_accu
racy: 0.4907
Epoch 40/50
14/14 [==============================] - 114s 8s/step - loss: 0.0045 - accuracy: 0.9977 - val_loss: 7.1651 - val_accu
racy: 0.5926
Epoch 41/50
14/14 [==============================] - 114s 8s/step - loss: 0.1194 - accuracy: 0.9815 - val_loss: 7.5061 - val_accu
racy: 0.6852
Epoch 42/50
14/14 [==============================] - 113s 8s/step - loss: 0.1232 - accuracy: 0.9699 - val_loss: 7.7904 - val_accu
racy: 0.6019
Epoch 43/50
14/14 [==============================] - 114s 8s/step - loss: 0.1163 - accuracy: 0.9792 - val_loss: 12.6312 - val_acc
uracy: 0.5278
Epoch 44/50
14/14 [==============================] - 115s 8s/step - loss: 0.1020 - accuracy: 0.9838 - val_loss: 7.4323 - val_accu
racy: 0.6204
Epoch 45/50
14/14 [==============================] - 114s 8s/step - loss: 0.0133 - accuracy: 0.9977 - val_loss: 5.4362 - val_accu
racy: 0.6574
Epoch 46/50
14/14 [==============================] - 114s 8s/step - loss: 0.0126 - accuracy: 0.9931 - val_loss: 8.7891 - val_accu
racy: 0.5648
Epoch 47/50
14/14 [==============================] - 115s 8s/step - loss: 0.6858 - accuracy: 0.9375 - val_loss: 7.1655 - val_accu
racy: 0.6852
Epoch 48/50
14/14 [==============================] - 114s 8s/step - loss: 0.1176 - accuracy: 0.9769 - val_loss: 6.9797 - val_accu
racy: 0.6389
Epoch 49/50
14/14 [==============================] - 115s 8s/step - loss: 0.1478 - accuracy: 0.9815 - val_loss: 6.8214 - val_accu
racy: 0.6667
Epoch 50/50
14/14 [==============================] - 114s 8s/step - loss: 0.0602 - accuracy: 0.9861 - val_loss: 7.4329 - val_accu
racy: 0.6019
----------------Test accuracy for 4 fold----------------
Confusion Matrix :
 [[16  0 11  3]
 [ 3  6  1  0]
 [ 1  1  8  0]
 [ 1  1  6  2]]
Accuracy    :  0.5333333333333333
Specificity :  0.7927489177489178
Sensitivity :  0.5333333333333333
----------------------------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 [==============================] - 155s 10s/step - loss: 17.4498 - accuracy: 0.4120 - val_loss: 103.6041 - val_
accuracy: 0.5093
Epoch 2/50
14/14 [==============================] - 126s 9s/step - loss: 8.3501 - accuracy: 0.5093 - val_loss: 25.4944 - val_acc
uracy: 0.3426
Epoch 3/50
14/14 [==============================] - 121s 9s/step - loss: 5.3001 - accuracy: 0.5579 - val_loss: 80.2771 - val_acc
uracy: 0.5000
Epoch 4/50
14/14 [==============================] - 122s 9s/step - loss: 3.9604 - accuracy: 0.6111 - val_loss: 30.5585 - val_acc
uracy: 0.4722
Epoch 5/50
14/14 [==============================] - 121s 9s/step - loss: 3.5646 - accuracy: 0.6574 - val_loss: 24.3165 - val_acc
```

```
uracy: 0.2778
Epoch 6/50
14/14 [==============================] - 121s 9s/step - loss: 3.2737 - accuracy: 0.7014 - val_loss: 16.2896 - val_acc
uracy: 0.1944
Epoch 7/50
14/14 [==============================] - 122s 9s/step - loss: 2.4147 - accuracy: 0.7222 - val_loss: 27.2036 - val_acc
uracy: 0.1759
Epoch 8/50
14/14 [==============================] - 123s 9s/step - loss: 1.9375 - accuracy: 0.7222 - val_loss: 19.3063 - val_acc
uracy: 0.2130
Epoch 9/50
14/14 [==============================] - 122s 9s/step - loss: 1.0005 - accuracy: 0.8426 - val_loss: 20.7363 - val_acc
uracy: 0.2315
Epoch 10/50
14/14 [==============================] - 121s 9s/step - loss: 0.8430 - accuracy: 0.8403 - val_loss: 23.6089 - val_acc
uracy: 0.2037
Epoch 11/50
14/14 [==============================] - 123s 9s/step - loss: 1.0542 - accuracy: 0.8356 - val_loss: 30.3548 - val_acc
uracy: 0.1759
Epoch 12/50
14/14 [==============================] - 121s 9s/step - loss: 0.8829 - accuracy: 0.8472 - val_loss: 10.5305 - val_acc
uracy: 0.2407
Epoch 13/50
14/14 [==============================] - 122s 9s/step - loss: 0.7316 - accuracy: 0.8727 - val_loss: 19.8594 - val_acc
uracy: 0.1667
Epoch 14/50
14/14 [==============================] - 123s 9s/step - loss: 0.6224 - accuracy: 0.9074 - val_loss: 20.2181 - val_acc
uracy: 0.2315
Epoch 15/50
14/14 [==============================] - 121s 9s/step - loss: 0.6535 - accuracy: 0.8796 - val_loss: 13.2218 - val_acc
uracy: 0.2407
Epoch 16/50
14/14 [==============================] - 122s 9s/step - loss: 0.5936 - accuracy: 0.8935 - val_loss: 12.6147 - val_acc
uracy: 0.2407
Epoch 17/50
14/14 [==============================] - 120s 9s/step - loss: 0.3968 - accuracy: 0.9120 - val_loss: 12.8842 - val_acc
uracy: 0.2315
Epoch 18/50
14/14 [==============================] - 122s 9s/step - loss: 0.2755 - accuracy: 0.9375 - val_loss: 17.6620 - val_acc
uracy: 0.2407
Epoch 19/50
14/14 [==============================] - 122s 9s/step - loss: 0.4597 - accuracy: 0.9213 - val_loss: 15.2148 - val_acc
uracy: 0.2870
Epoch 20/50
14/14 [==============================] - 122s 9s/step - loss: 0.4552 - accuracy: 0.9421 - val_loss: 5.9475 - val_accu
racy: 0.5185
Epoch 21/50
14/14 [==============================] - 124s 9s/step - loss: 0.1767 - accuracy: 0.9491 - val_loss: 17.4296 - val_acc
uracy: 0.2130
Epoch 22/50
14/14 [==============================] - 122s 9s/step - loss: 0.2659 - accuracy: 0.9583 - val_loss: 9.3660 - val_accu
racy: 0.4537
Epoch 23/50
14/14 [==============================] - 121s 9s/step - loss: 0.2141 - accuracy: 0.9514 - val_loss: 6.6478 - val_accu
racy: 0.5370
Epoch 24/50
14/14 [==============================] - 122s 9s/step - loss: 0.3913 - accuracy: 0.9352 - val_loss: 9.3356 - val_accu
racy: 0.4444
Epoch 25/50
14/14 [==============================] - 120s 9s/step - loss: 0.2923 - accuracy: 0.9537 - val_loss: 10.7075 - val_acc
uracy: 0.4259
Epoch 26/50
14/14 [==============================] - 121s 9s/step - loss: 0.1009 - accuracy: 0.9792 - val_loss: 7.2353 - val_accu
racy: 0.4815
Epoch 27/50
14/14 [==============================] - 120s 9s/step - loss: 0.3069 - accuracy: 0.9583 - val_loss: 17.0711 - val_acc
uracy: 0.2963
Epoch 28/50
14/14 [==============================] - 122s 9s/step - loss: 0.2850 - accuracy: 0.9444 - val_loss: 12.3945 - val_acc
uracy: 0.3056
Epoch 29/50
14/14 [==============================] - 121s 9s/step - loss: 0.1425 - accuracy: 0.9722 - val_loss: 9.6208 - val_accu
racy: 0.4259
Epoch 30/50
14/14 [==============================] - 121s 9s/step - loss: 0.1062 - accuracy: 0.9699 - val_loss: 7.1373 - val_accu
racy: 0.5000
Epoch 31/50
14/14 [==============================] - 122s 9s/step - loss: 0.1924 - accuracy: 0.9745 - val_loss: 6.4111 - val_accu
racy: 0.6667
Epoch 32/50
14/14 [==============================] - 121s 9s/step - loss: 0.3567 - accuracy: 0.9468 - val_loss: 6.2013 - val_accu
racy: 0.6296
Epoch 33/50
14/14 [==============================] - 121s 9s/step - loss: 0.0959 - accuracy: 0.9838 - val_loss: 8.0453 - val_accu
racy: 0.4907
Epoch 34/50
14/14 [==============================] - 122s 9s/step - loss: 0.1894 - accuracy: 0.9630 - val_loss: 5.9983 - val_accu
racy: 0.6574
```

```
Epoch 35/50
14/14 [==============================] - 120s 9s/step - loss: 0.1418 - accuracy: 0.9653 - val_loss: 6.1168 - val_accu
racy: 0.6389
Epoch 36/50
14/14 [==============================] - 120s 9s/step - loss: 0.2210 - accuracy: 0.9676 - val_loss: 6.3609 - val_accu
racy: 0.5648
Epoch 37/50
14/14 [==============================] - 117s 8s/step - loss: 0.0756 - accuracy: 0.9884 - val_loss: 7.7039 - val_accu
racy: 0.5093
Epoch 38/50
14/14 [==============================] - 119s 9s/step - loss: 0.0809 - accuracy: 0.9792 - val_loss: 7.0673 - val_accu
racy: 0.6204
Epoch 39/50
14/14 [==============================] - 115s 8s/step - loss: 0.1723 - accuracy: 0.9792 - val_loss: 28.6780 - val_acc
uracy: 0.2685
Epoch 40/50
14/14 [==============================] - 116s 8s/step - loss: 0.5408 - accuracy: 0.9491 - val_loss: 7.4804 - val_accu
racy: 0.5833
Epoch 41/50
14/14 [==============================] - 116s 8s/step - loss: 0.0190 - accuracy: 0.9907 - val_loss: 7.5089 - val_accu
racy: 0.5926
Epoch 42/50
14/14 [==============================] - 114s 8s/step - loss: 0.3844 - accuracy: 0.9606 - val_loss: 7.7274 - val_accu
racy: 0.5463
Epoch 43/50
14/14 [==============================] - 114s 8s/step - loss: 0.1397 - accuracy: 0.9884 - val_loss: 12.2679 - val_acc
uracy: 0.3981
Epoch 44/50
14/14 [==============================] - 114s 8s/step - loss: 0.0246 - accuracy: 0.9884 - val_loss: 6.7990 - val_accu
racy: 0.5463
Epoch 45/50
14/14 [==============================] - 115s 8s/step - loss: 0.1320 - accuracy: 0.9792 - val_loss: 9.0884 - val_accu
racy: 0.5556
Epoch 46/50
14/14 [==============================] - 114s 8s/step - loss: 0.0476 - accuracy: 0.9838 - val_loss: 8.0413 - val_accu
racy: 0.5648
Epoch 47/50
14/14 [==============================] - 113s 8s/step - loss: 0.2138 - accuracy: 0.9792 - val_loss: 6.9369 - val_accu
racy: 0.6111
Epoch 48/50
14/14 [==============================] - 114s 8s/step - loss: 0.0871 - accuracy: 0.9907 - val_loss: 7.4215 - val_accu
racy: 0.6389
Epoch 49/50
14/14 [==============================] - 112s 8s/step - loss: 0.2432 - accuracy: 0.9491 - val_loss: 7.5117 - val_accu
racy: 0.5556
Epoch 50/50
14/14 [==============================] - 114s 8s/step - loss: 0.0608 - accuracy: 0.9931 - val_loss: 7.4005 - val_accu
racy: 0.5556
WARNING:tensorflow:5 out of the last 9 calls to <function Model.make_predict_function.<locals>.predict_function at 0x
0000015D2528C5E0> 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 e
xperimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), plea
se refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/p
ython/tf/function for  more details.
----------------Test accuracy for 5 fold----------------
Confusion Matrix :
 [[19  0  6  5]
 [ 1  5  3  1]
 [ 2  0  8  0]
 [ 4  1  3  2]]
Accuracy    :  0.5666666666666667
Specificity :  0.7937001594896331
Sensitivity :  0.5333333333333333
----------------------------End of 5 Fold-------------------------
```

Test Evaluation Results

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

Out[17]:
```
[0.6166666666666667,
 0.6666666666666666,
 0.4166666666666667,
 0.5333333333333333,
 0.5666666666666667]
```

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

Out[18]:
```
0.5599999999999999
```

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

Out[19]:
```
[0.5583333333333333, 0.6, 0.375, 0.5333333333333333, 0.5333333333333333]
```

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

Out[20]:
```
0.5199999999999999
```

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

Out[21]:
```
[0.821272813208297,
 0.8394830102147175,
 0.6824945887445888,
 0.7927489177489178,
 0.7937001594896331]
```

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

Out[22]:
```
0.7859398978812309
```

Training and Validation Evaluation Results

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

Out[23]:
```
array([0.41898149, 0.5486111 , 0.5462963 , 0.62962961, 0.6412037 ,
       0.69675928, 0.74537039, 0.69675928, 0.81944442, 0.77546299,
       0.8611111 , 0.86805558, 0.9212963 , 0.86805558, 0.88425928,
0.87731481, 0.96064812, 0.89351851, 0.96296299, 0.97916669,
0.93518519, 0.9513889 , 0.96527779, 0.96527779, 0.94212961,
0.97916669, 0.95601851, 0.95833331, 0.97685188, 0.9537037 ,
0.97222221, 0.9837963 , 0.98842591, 0.9537037 , 0.99768519,
0.95601851, 0.98842591, 0.96527779, 0.96064812, 0.98842591,
0.97685188, 0.99074072, 0.9861111 , 0.97453701, 0.9837963 ,
0.95601851, 1.        , 0.97916669, 0.99305558, 0.97916669,
0.41203704, 0.5324074 , 0.55787039, 0.57175928, 0.68287039,
0.65046299, 0.67824072, 0.68518519, 0.77314812, 0.82175928,
0.78472221, 0.91435188, 0.83564812, 0.87962961, 0.89814812,
0.92592591, 0.92592591, 0.93981481, 0.89351851, 0.96064812,
0.93981481, 0.90972221, 0.94907409, 0.94907409, 0.9675926 ,
0.97453701, 0.9537037 , 0.93055558, 0.96527779, 0.99074072,
0.9513889 , 0.9861111 , 0.98842591, 0.94212961, 0.9537037 ,
0.97453701, 0.98842591, 0.97222221, 0.9837963 , 0.96296299,
0.97685188, 0.9837963 , 0.99768519, 0.99305558, 0.9675926 ,
0.97916669, 0.96527779, 0.98842591, 0.97916669, 0.9837963 ,
0.39814815, 0.52546299, 0.56944442, 0.5925926 , 0.58564812,
0.67592591, 0.72453701, 0.75231481, 0.7662037 , 0.77314812,
0.8425926 , 0.90046299, 0.8587963 , 0.87731481, 0.87268519,
0.93518519, 0.93287039, 0.93055558, 0.90046299, 0.93287039,
0.94675928, 0.9513889 , 0.9513889 , 0.95833331, 0.9675926 ,
0.97453701, 0.93518519, 0.97222221, 0.94907409, 0.99305558,
0.9513889 , 0.93287039, 0.9861111 , 0.97222221, 0.92824072,
0.98148149, 0.9675926 , 0.9699074 , 0.9675926 , 0.9861111 ,
1.        , 1.        , 0.96296299, 0.97685188, 0.99537039,
0.98842591, 0.97916669, 0.98148149, 0.97916669, 0.98842591,
0.3888889 , 0.5300926 , 0.60416669, 0.63194442, 0.64583331,
0.7175926 , 0.75462961, 0.78703701, 0.80092591, 0.80555558,
0.80324072, 0.88194442, 0.87268519, 0.89814812, 0.91666669,
0.93055558, 0.89814812, 0.91435188, 0.9837963 , 0.9675926 ,
0.90972221, 0.9537037 , 0.97453701, 0.9513889 , 0.96064812,
0.94907409, 0.96064812, 0.9699074 , 0.9375    , 0.97453701,
0.96064812, 0.97685188, 0.98842591, 0.96527779, 0.96527779,
0.9837963 , 0.95601851, 0.95601851, 0.99768519, 0.99768519,
0.98148149, 0.9699074 , 0.97916669, 0.9837963 , 0.99768519,
0.99305558, 0.9375    , 0.97685188, 0.98148149, 0.9861111 ,
0.41203704, 0.50925928, 0.55787039, 0.6111111 , 0.6574074 ,
0.7013889 , 0.72222221, 0.72222221, 0.8425926 , 0.84027779,
0.83564812, 0.84722221, 0.87268519, 0.9074074 , 0.87962961,
0.89351851, 0.91203701, 0.9375    , 0.9212963 , 0.94212961,
0.94907409, 0.95833331, 0.9513889 , 0.93518519, 0.9537037 ,
0.97916669, 0.95833331, 0.94444442, 0.97222221, 0.9699074 ,
0.97453701, 0.94675928, 0.9837963 , 0.96296299, 0.96527779,
0.9675926 , 0.98842591, 0.97916669, 0.97916669, 0.94907409,
0.99074072, 0.96064812, 0.98842591, 0.98842591, 0.97916669,
0.9837963 , 0.97916669, 0.99074072, 0.94907409, 0.99305558])
```

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

Out[24]:
```
0.8918981486558915
```

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

Out[25]:
```
array([0.42592594, 0.47222221, 0.46296296, 0.3888889 , 0.53703701,
       0.28703704, 0.25925925, 0.46296296, 0.1574074 , 0.4074074 ,
       0.2962963 , 0.21296297, 0.16666667, 0.18518518, 0.17592593,
       0.5       , 0.32407406, 0.37962964, 0.3425926 , 0.50925928,
       0.19444445, 0.22222222, 0.62962961, 0.6574074 , 0.64814812,
       0.67592591, 0.56481481, 0.5       , 0.51851851, 0.57407409,
       0.39814815, 0.62962961, 0.4537037 , 0.6388889 , 0.60185188,
       0.64814812, 0.50925928, 0.6111111 , 0.71296299, 0.58333331,
       0.60185188, 0.6388889 , 0.5462963 , 0.60185188, 0.62962961,
       0.66666669, 0.67592591, 0.62962961, 0.7037037 , 0.6574074 ,
       0.56481481, 0.48148149, 0.41666666, 0.43518519, 0.52777779,
       0.4074074 , 0.58333331, 0.46296296, 0.5462963 , 0.5       ,
       0.4537037 , 0.6388889 , 0.3888889 , 0.50925928, 0.58333331,
       0.5462963 , 0.60185188, 0.5462963 , 0.56481481, 0.51851851,
       0.60185188, 0.62037039, 0.6111111 , 0.6111111 , 0.53703701,
       0.62037039, 0.49074075, 0.60185188, 0.6388889 , 0.5925926 ,
       0.50925928, 0.62962961, 0.53703701, 0.52777779, 0.6388889 ,
       0.6111111 , 0.5925926 , 0.64814812, 0.6111111 , 0.62962961,
       0.48148149, 0.58333331, 0.62962961, 0.62962961, 0.64814812,
       0.52777779, 0.62037039, 0.62962961, 0.64814812, 0.62962961,
       0.50925928, 0.49074075, 0.3611111 , 0.49074075, 0.39814815,
       0.25      , 0.35185185, 0.2037037 , 0.6111111 , 0.42592594,
       0.48148149, 0.35185185, 0.43518519, 0.28703704, 0.5       ,
       0.48148149, 0.27777779, 0.39814815, 0.55555558, 0.55555558,
       0.35185185, 0.62037039, 0.6388889 , 0.30555555, 0.62037039,
       0.30555555, 0.37962964, 0.56481481, 0.5       , 0.62962961,
       0.52777779, 0.46296296, 0.56481481, 0.50925928, 0.6111111 ,
       0.49074075, 0.53703701, 0.62037039, 0.6388889 , 0.64814812,
       0.6574074 , 0.69444442, 0.6111111 , 0.66666669, 0.60185188,
       0.3888889 , 0.60185188, 0.62037039, 0.5       , 0.62037039,
       0.49074075, 0.5462963 , 0.33333334, 0.49074075, 0.22222222,
       0.19444445, 0.17592593, 0.17592593, 0.21296297, 0.16666667,
       0.16666667, 0.16666667, 0.16666667, 0.16666667, 0.16666667,
       0.33333334, 0.24074075, 0.18518518, 0.23148148, 0.28703704,
       0.37037036, 0.33333334, 0.33333334, 0.30555555, 0.25925925,
       0.43518519, 0.46296296, 0.4537037 , 0.6388889 , 0.56481481,
       0.55555558, 0.53703701, 0.56481481, 0.49074075, 0.52777779,
       0.51851851, 0.4537037 , 0.44444445, 0.49074075, 0.5925926 ,
       0.68518519, 0.60185188, 0.52777779, 0.62037039, 0.6574074 ,
       0.56481481, 0.68518519, 0.6388889 , 0.66666669, 0.60185188,
       0.50925928, 0.3425926 , 0.5       , 0.47222221, 0.27777779,
       0.19444445, 0.17592593, 0.21296297, 0.23148148, 0.2037037 ,
       0.17592593, 0.24074075, 0.16666667, 0.23148148, 0.24074075,
       0.24074075, 0.23148148, 0.24074075, 0.28703704, 0.51851851,
       0.21296297, 0.4537037 , 0.53703701, 0.44444445, 0.42592594,
       0.48148149, 0.2962963 , 0.30555555, 0.42592594, 0.5       ,
       0.66666669, 0.62962961, 0.49074075, 0.6574074 , 0.6388889 ,
       0.56481481, 0.50925928, 0.62037039, 0.26851851, 0.58333331,
       0.5925926 , 0.5462963 , 0.39814815, 0.5462963 , 0.55555558,
       0.56481481, 0.6111111 , 0.6388889 , 0.55555558, 0.55555558])
```

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

Out[26]:
```
0.4761111131310463
```

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

Out[27]:
```
array([1.96142254e+01, 7.44530392e+00, 8.99572468e+00, 4.42612791e+00,
       3.73409319e+00, 2.46614385e+00, 1.62895644e+00, 2.11051440e+00,
       1.14456928e+00, 1.41880238e+00, 6.32037401e-01, 5.87155461e-01,
       3.46188545e-01, 7.35989690e-01, 6.27283394e-01, 8.33966792e-01,
       1.58424377e-01, 5.36053300e-01, 1.90983832e-01, 8.40443894e-02,
       5.73346674e-01, 2.02327117e-01, 1.68602765e-01, 1.11793347e-01,
       3.49108070e-01, 5.04739545e-02, 1.81497961e-01, 2.74424493e-01,
       4.80578914e-02, 2.47148454e-01, 1.05077967e-01, 6.88998327e-02,
       3.60322781e-02, 2.52100110e-01, 5.02019748e-03, 2.53487051e-01,
       6.17871955e-02, 1.15656428e-01, 2.88707495e-01, 3.59094180e-02,
       9.94160548e-02, 5.06915264e-02, 5.70172556e-02, 1.08251043e-01,
       7.78625533e-02, 2.30826244e-01, 1.29609636e-03, 7.09229410e-02,
       2.10378934e-02, 1.39437303e-01, 1.92123318e+01, 7.46403503e+00,
       6.45887518e+00, 6.67546701e+00, 2.86042356e+00, 3.04631925e+00,
       1.86982548e+00, 2.32970238e+00, 1.31980526e+00, 8.57158184e-01,
       1.24748182e+00, 3.10961187e-01, 9.83970225e-01, 9.53348458e-01,
       5.52413642e-01, 6.16903722e-01, 4.39346910e-01, 2.38537773e-01,
       5.85197866e-01, 1.36842713e-01, 2.81471401e-01, 6.63889945e-01,
       2.45480239e-01, 3.51261288e-01, 1.44669279e-01, 1.21690802e-01,
       2.46862054e-01, 4.10027176e-01, 1.75730497e-01, 2.75222994e-02,
       2.77336866e-01, 4.23182510e-02, 5.70253208e-02, 2.85946459e-01,
       2.92017758e-01, 1.18631080e-01, 5.05564176e-02, 1.57674387e-01,
       4.84344475e-02, 1.71903864e-01, 1.13341264e-01, 1.44302502e-01,
       2.24016001e-03, 2.23401766e-02, 1.54777169e-01, 1.04055032e-01,
       2.25118056e-01, 3.49179208e-02, 1.15155332e-01, 1.23567544e-01,
       1.73791580e+01, 6.79053450e+00, 7.04926872e+00, 3.98966217e+00,
       4.84637356e+00, 1.87168431e+00, 1.44055605e+00, 1.36010146e+00,
       1.22529423e+00, 1.14515257e+00, 9.10821795e-01, 4.70139503e-01,
       9.76698577e-01, 6.00050449e-01, 6.44768298e-01, 4.55419928e-01,
       4.11412686e-01, 2.99215078e-01, 7.44681180e-01, 2.32293516e-01,
       2.27789685e-01, 4.06037182e-01, 3.36758852e-01, 1.48653790e-01,
       1.88498214e-01, 1.14726335e-01, 4.13503855e-01, 1.63522065e-01,
       3.05415720e-01, 3.85076068e-02, 2.58986682e-01, 3.07154119e-01,
       3.61319520e-02, 2.44804755e-01, 7.48543978e-01, 1.19447060e-01,
       1.67408377e-01, 1.11433506e-01, 9.44992453e-02, 5.01428843e-02,
       3.54317762e-03, 5.43956086e-03, 1.93177253e-01, 1.08678870e-01,
       1.73964836e-02, 5.74710593e-02, 1.23424083e-01, 9.31822211e-02,
       1.21533237e-01, 4.81104515e-02, 1.87326565e+01, 8.69554520e+00,
       5.48939514e+00, 4.96246004e+00, 3.34533787e+00, 2.04790998e+00,
       1.42475522e+00, 2.07128429e+00, 1.23461103e+00, 1.27508080e+00,
       1.48345041e+00, 1.21717215e+00, 5.67738116e-01, 6.13260150e-01,
       4.40880388e-01, 3.62954974e-01, 7.78932512e-01, 5.58383882e-01,
       5.04378043e-02, 1.57300457e-01, 5.45023024e-01, 3.56804252e-01,
       8.34190398e-02, 4.60057765e-01, 2.39493057e-01, 3.86778325e-01,
       1.88856304e-01, 2.67606288e-01, 3.50034714e-01, 1.35003477e-01,
       1.92339703e-01, 1.12063617e-01, 5.24274595e-02, 2.79504985e-01,
       2.16225043e-01, 4.78041731e-02, 3.26690644e-01, 4.37475592e-01,
       1.43287824e-02, 4.47540917e-03, 1.19448729e-01, 1.23152032e-01,
       1.16312884e-01, 1.02041341e-01, 1.33183300e-02, 1.26116835e-02,
       6.85806572e-01, 1.17589660e-01, 1.47772133e-01, 6.02433160e-02,
       1.74497871e+01, 8.35005093e+00, 5.30010653e+00, 3.96040320e+00,
       3.56459522e+00, 3.27370644e+00, 2.41472650e+00, 1.93749416e+00,
       1.00047517e+00, 8.42954636e-01, 1.05421293e+00, 8.82867396e-01,
       7.31582701e-01, 6.22436404e-01, 6.53523624e-01, 5.93577147e-01,
       3.96795779e-01, 2.75503874e-01, 4.59705353e-01, 4.55196887e-01,
       1.76695138e-01, 2.65866071e-01, 2.14124396e-01, 3.91342133e-01,
       2.92268276e-01, 1.00880757e-01, 3.06896329e-01, 2.84970850e-01,
       1.42453611e-01, 1.06235608e-01, 1.92366213e-01, 3.56650621e-01,
       9.58685726e-02, 1.89404130e-01, 1.41777426e-01, 2.21023977e-01,
       7.56363198e-02, 8.08699280e-02, 1.72273770e-01, 5.40813863e-01,
       1.89736746e-02, 3.84425193e-01, 1.39702633e-01, 2.45526470e-02,
       1.32007286e-01, 4.75876294e-02, 2.13829771e-01, 8.71437937e-02,
       2.43241668e-01, 6.07762821e-02])
```

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

Out[28]:
```
1.220375373212155
```

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

Out[29]:
```
array([ 51.24076843,  45.42938995,  19.36598778,  13.72946358,
         8.68806076,   7.49027061,   6.78911114,   5.91124153,
        16.77279282,   6.79450893,  13.55027485,  14.44989204,
        20.78791046,  21.55817413,  19.16628075,   4.18089676,
        10.6137743 ,   8.42154217,  11.48964119,   6.0927763 ,
        24.13824463,  16.62010002,   4.26607323,   3.73241329,
         3.54057622,   3.78877831,   5.61496878,   6.50707626,
         5.71111107,   6.06456041,   6.10582495,   4.45497561,
        11.19492054,   4.58782911,   4.3454752 ,   5.61723042,
         6.37429714,   5.44692898,   4.17829847,   4.5588131 ,
         7.96037197,   5.02119207,   5.40711403,   5.01226139,
         4.42912436,   4.38349199,   4.09252119,   5.67112064,
         5.19115067,   5.92265606,  88.9084549 ,  50.64912796,
        14.64200401,  13.18357944,  13.91689205,  15.32631588,
         3.50989103,  10.31655312,   6.12020922,   3.87688541,
         4.17542315,   3.47546816,   3.68009329,   3.72169971,
         2.90901279,   3.55647707,   2.627774  ,   3.51656938,
         5.80749416,   4.90388823,   6.70597839,   6.576684  ,
         8.61854362,   4.20816135,   5.63132286,   4.24398232,
         7.81131887,   5.34959126,   3.66236424,   4.63822508,
         5.41297483,   3.93561292,   6.40031719,   6.6293292 ,
         4.55795622,   5.88733578,   5.51656866,   5.78103304,
         8.58465576,   8.49874115,  11.4574461 ,   5.90557528,
         5.52178621,   6.08229733,   7.69960737,  10.94992828,
         7.57189369,   7.08910275,   7.39506721,   6.42678595,
       100.7148056 ,  56.89001083,  17.83356476,  27.1339283 ,
         7.75595093,   8.96618462,  11.85874462,  12.98691368,
         5.6314435 ,   9.03792381,   4.85716772,   7.52698898,
         7.29772568,   7.13481855,   6.52868557,   9.71862125,
         9.91291618,   6.66695929,   6.80694246,   4.37059879,
         7.65772629,   4.15830135,   3.86507154,  10.06643772,
         4.6094451 ,  10.33496189,   7.46774292,   7.59201908,
         7.15129471,   4.59893608,   6.35904598,   9.62214279,
         5.66719866,  36.14288712,   5.20593691,   6.6992836 ,
         6.31490231,   7.29477835,   5.35794163,   5.63075829,
         6.09269428,   6.86292219,   6.27742386,   5.65803623,
         7.28807449,  14.33911324,   8.57409096,   7.82723045,
        10.95527458,   6.79799271, 132.1708374 ,  20.65561485,
        12.89903927,  12.40991592,  14.20038414,  16.31844902,
        28.85245705,  22.28149033,  15.91189003,  38.73167038,
        14.10355091,  17.43777084,  27.37608528,  17.46351624,
        21.19377708,  10.33416176,  13.87447548,  14.15425873,
        16.69242668,  15.69665623,  13.4428215 ,  13.97052193,
        18.28422356,  12.18340588,  28.62045097,  11.69261837,
         7.63310242,  12.59927559,   9.37216854,   7.41819715,
         7.27020597,   6.89688492,   7.85655737,   8.92238998,
         8.56593895,   8.40445614,  11.00074291,   9.6376524 ,
         8.72341537,   7.16514969,   7.50606203,   7.79038668,
        12.63121891,   7.43228579,   5.43616629,   8.78907967,
         7.16554308,   6.97968102,   6.82141399,   7.43290472,
       103.60407257,  25.4944191 ,  80.27713013,  30.55854034,
        24.31646538,  16.28961563,  27.20355988,  19.30633926,
        20.73628807,  23.60894585,  30.35480118,  10.53050613,
        19.85937691,  20.2181263 ,  13.22178173,  12.61470413,
        12.8842268 ,  17.66197205,  15.21482182,   5.94751406,
        17.42962837,   9.36602688,   6.64782047,   9.33557987,
        10.70754147,   7.235322  ,  17.07108498,  12.39445972,
         9.62081337,   7.13728523,   6.41111517,   6.20127249,
         8.04527092,   5.9982686 ,   6.11678743,   6.36093998,
         7.70387077,   7.06734037,  28.67798615,   7.48039865,
         7.50886631,   7.72735929,  12.26794624,   6.79899311,
         9.08835983,   8.04127312,   6.93692446,   7.42149639,
         7.51168013,   7.40047789])
```

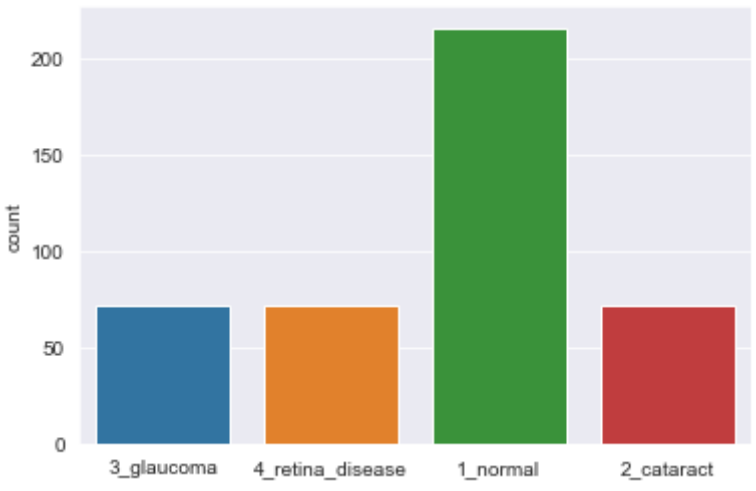
In [30]:
```
mean_val_loss= np.mean(val_loss)
mean_val_loss
```

Out[30]:
```
12.464622261047364
```

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)
```

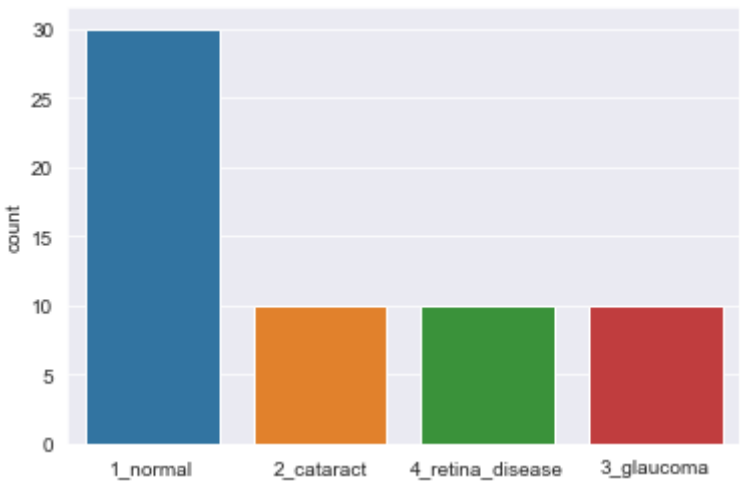
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x15d15cae0d0>



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)
```

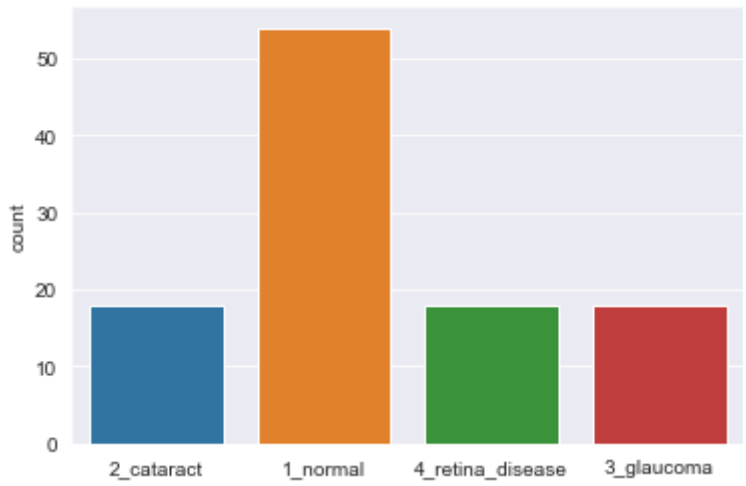
Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x15d29208a60>



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)
```

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x15d292517c0>

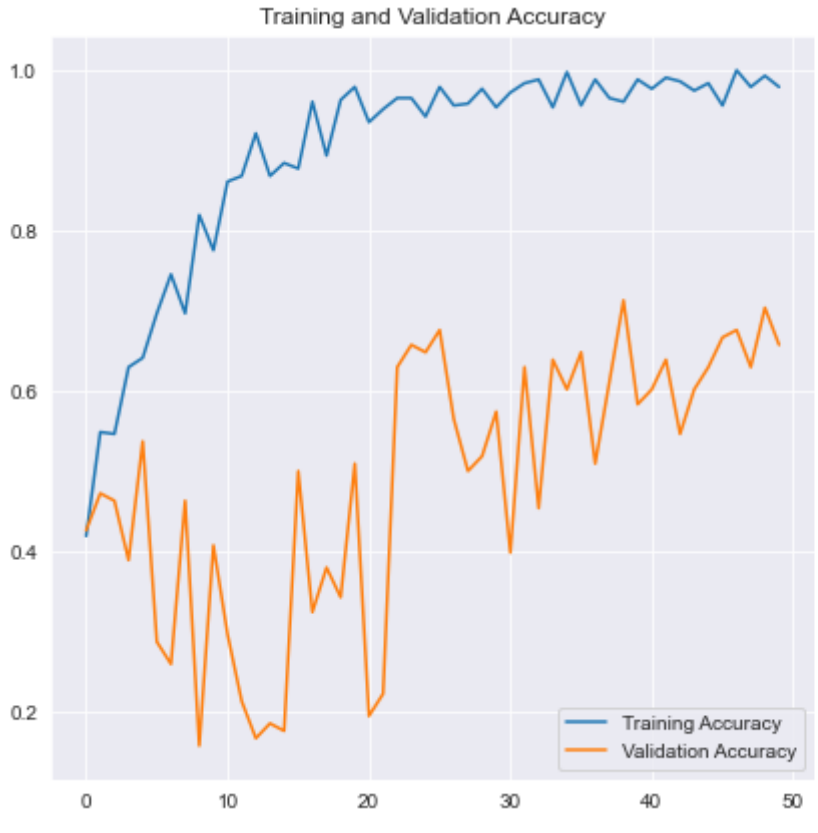


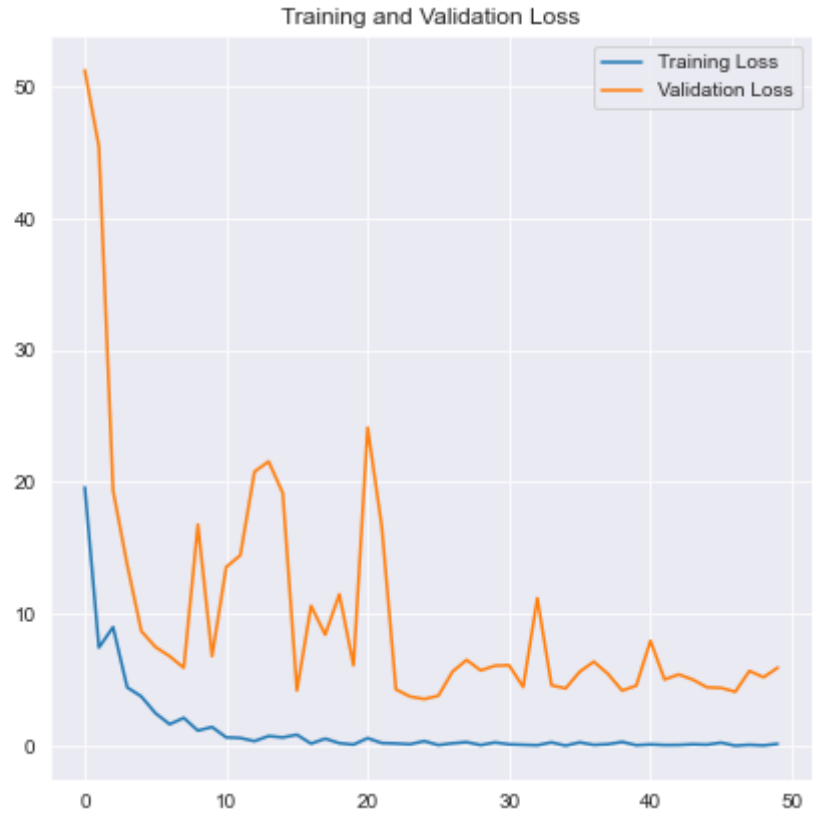
Training,Validation Accuracy and Loss Plot for 50 Epochs

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()
```

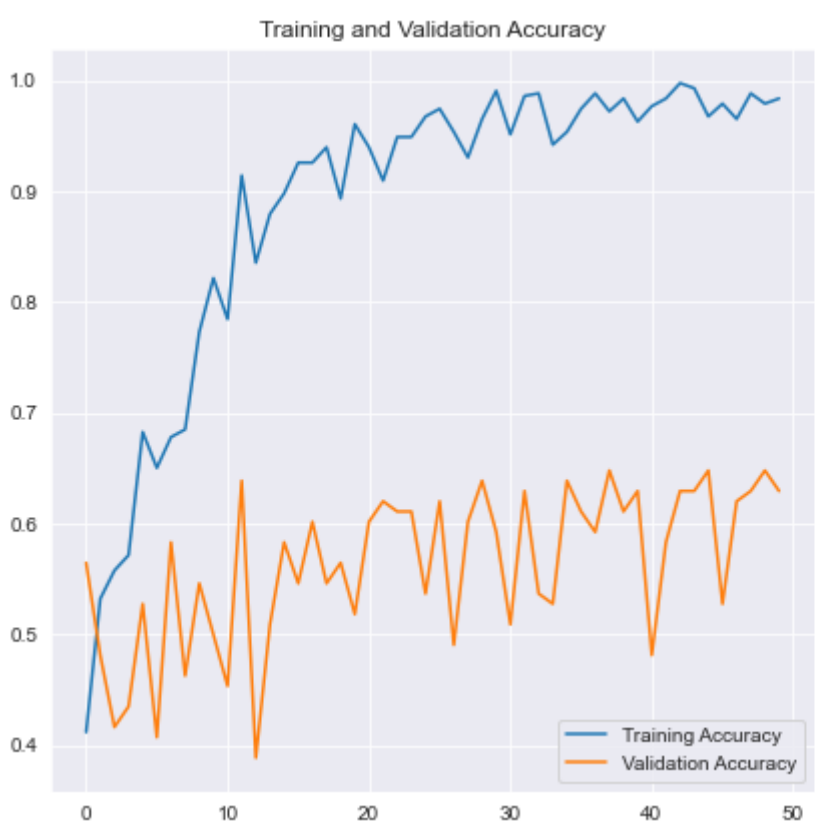
In [35]:
```
k=1
j=0
for i in range(0,250,50):
    j +=50
    print('Plot for ',k,'cross validation accuracy and loss for Training and Validation phase')
    k +=1
    plot_print(i,j)
```

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

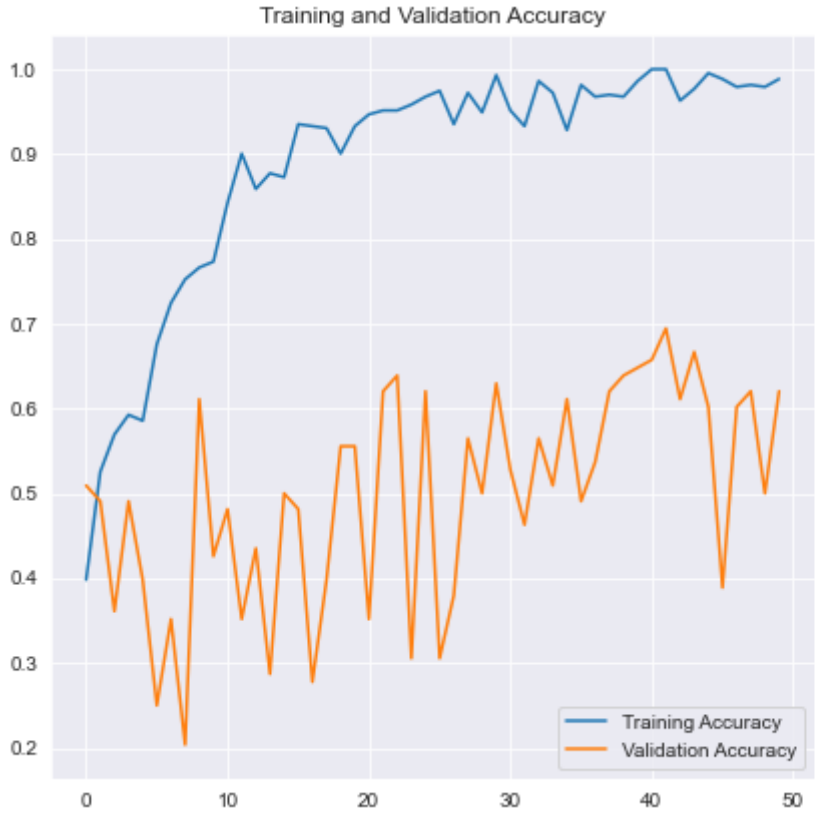




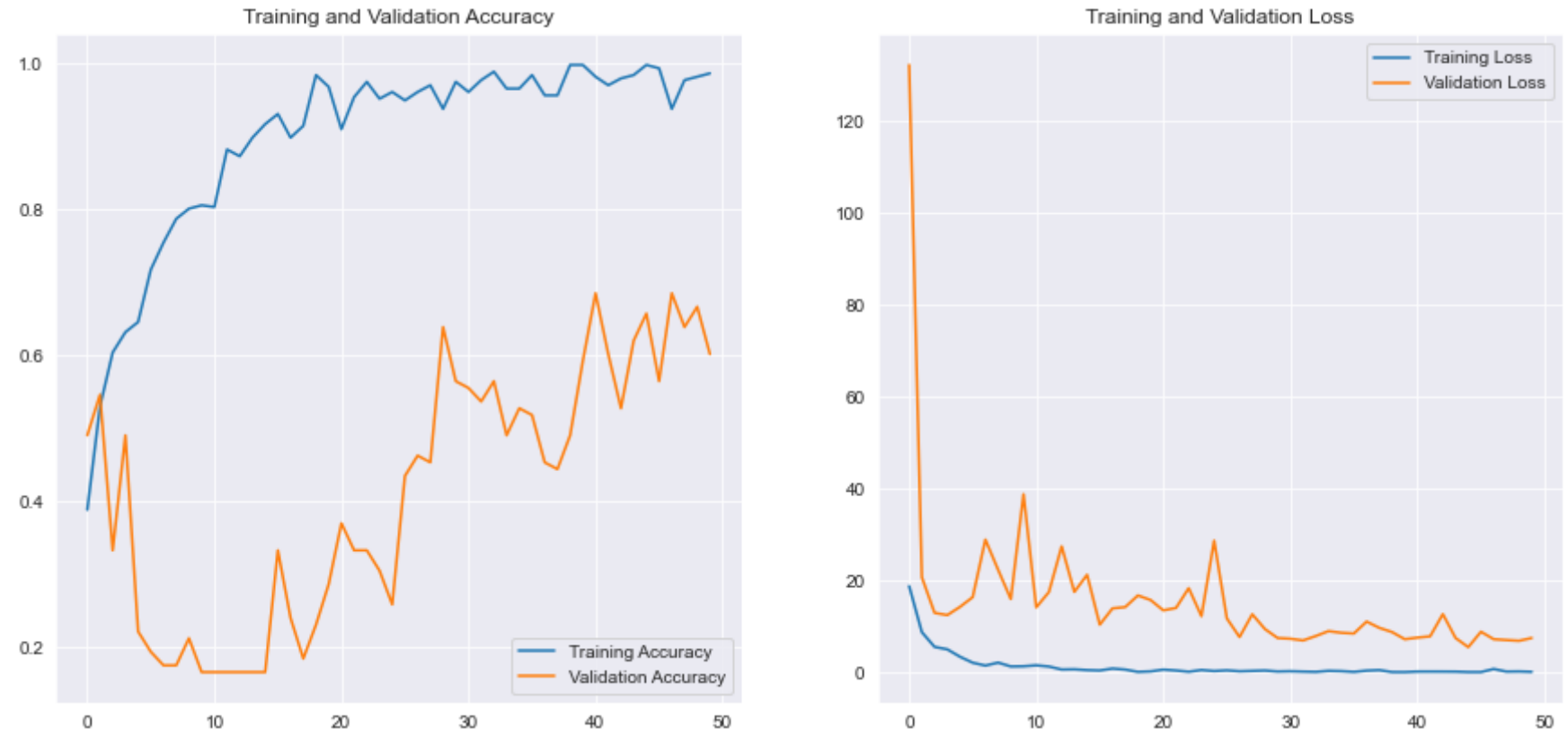
Plot for 2 cross validation accuracy and loss for Training and Validation phase



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

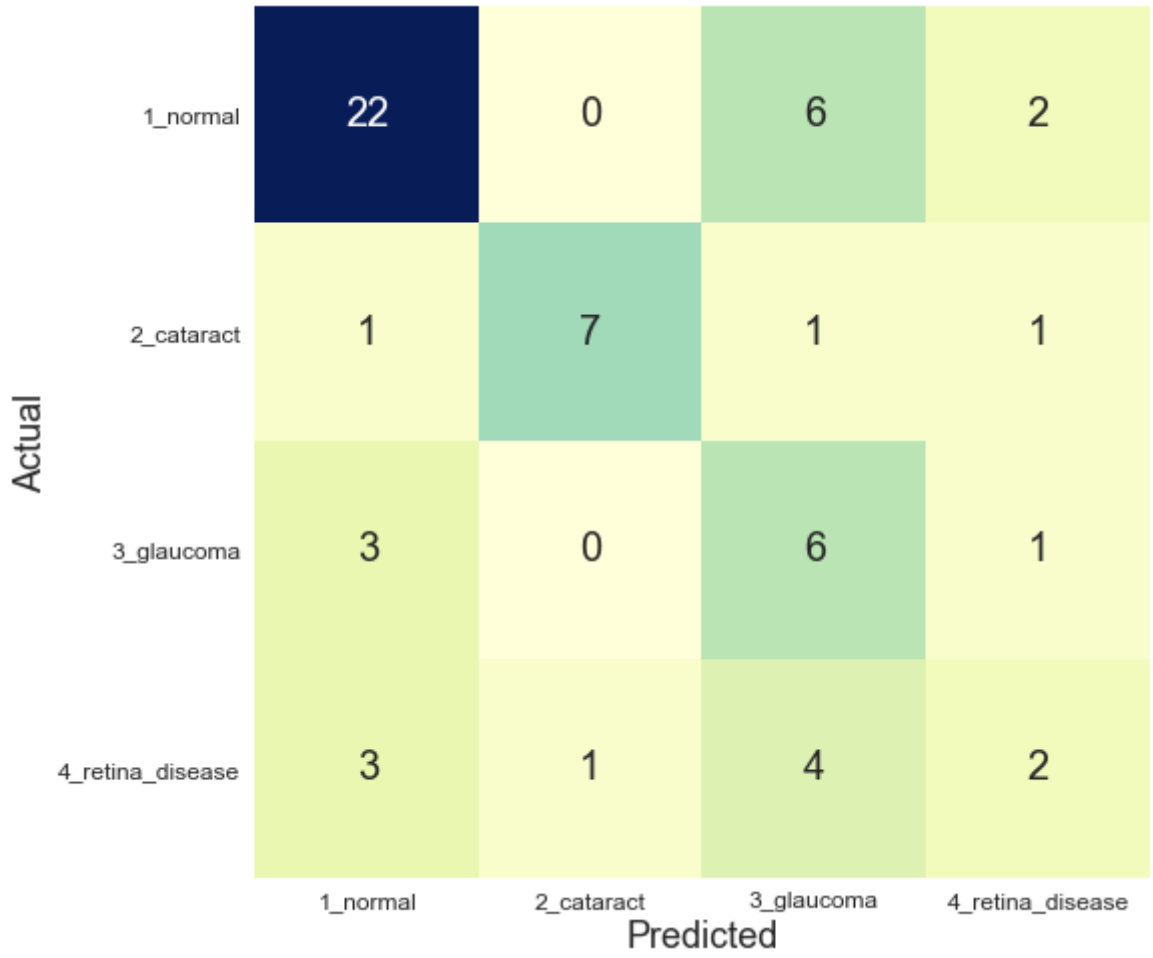
Visualizing Confusion Matrix for Each Fold

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

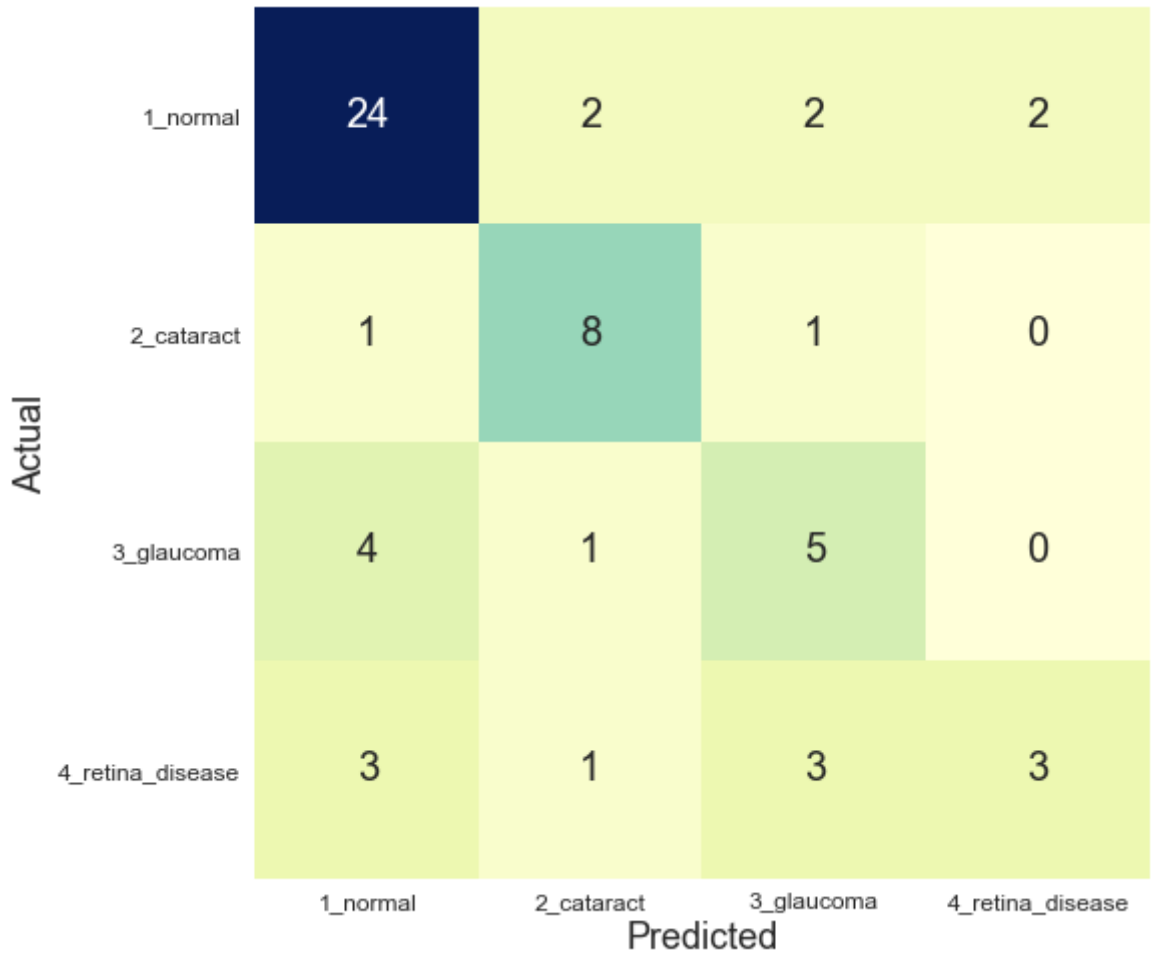
In [37]:
```
def confusionmatrix_vis(i):

        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)
```

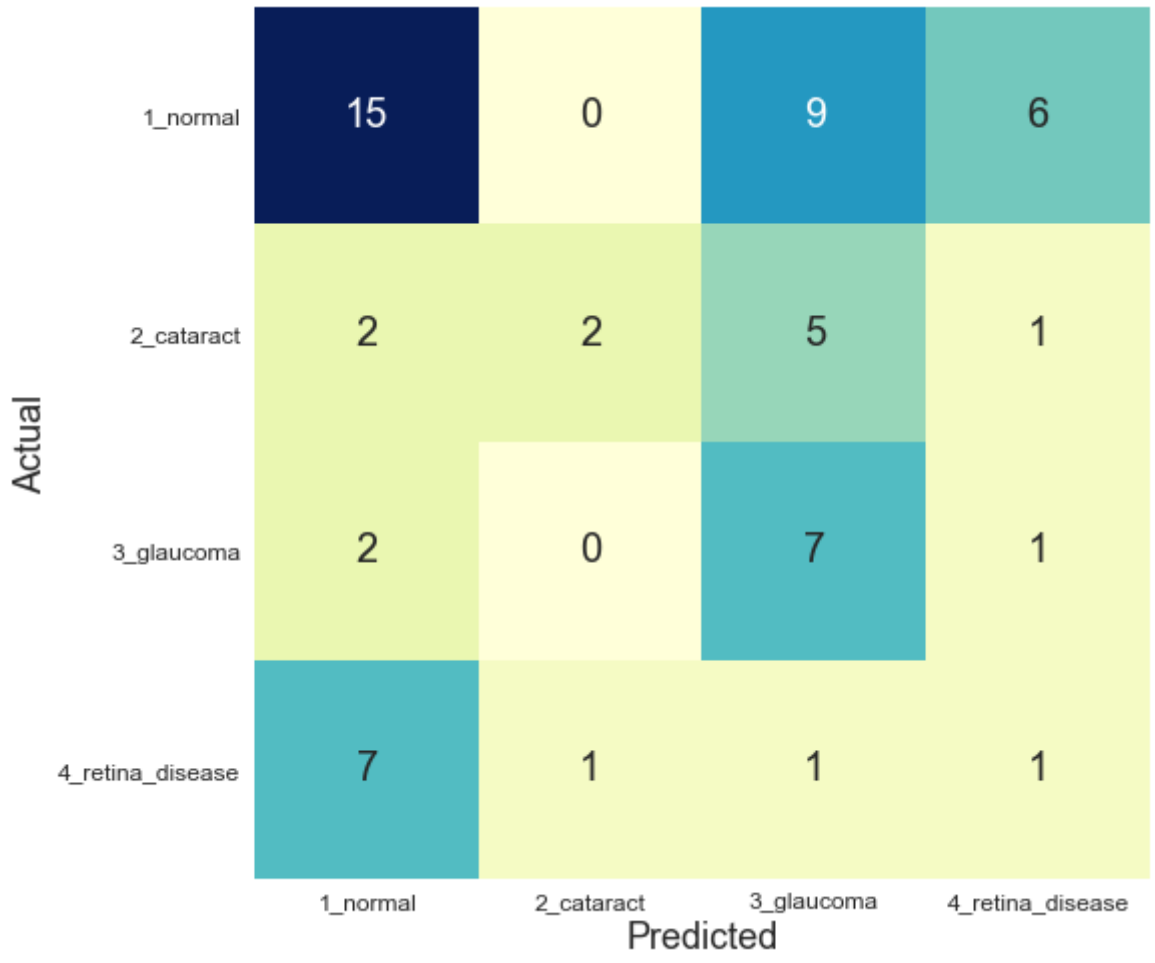
Confusion Matrix for 1 Cross Validation Test phase



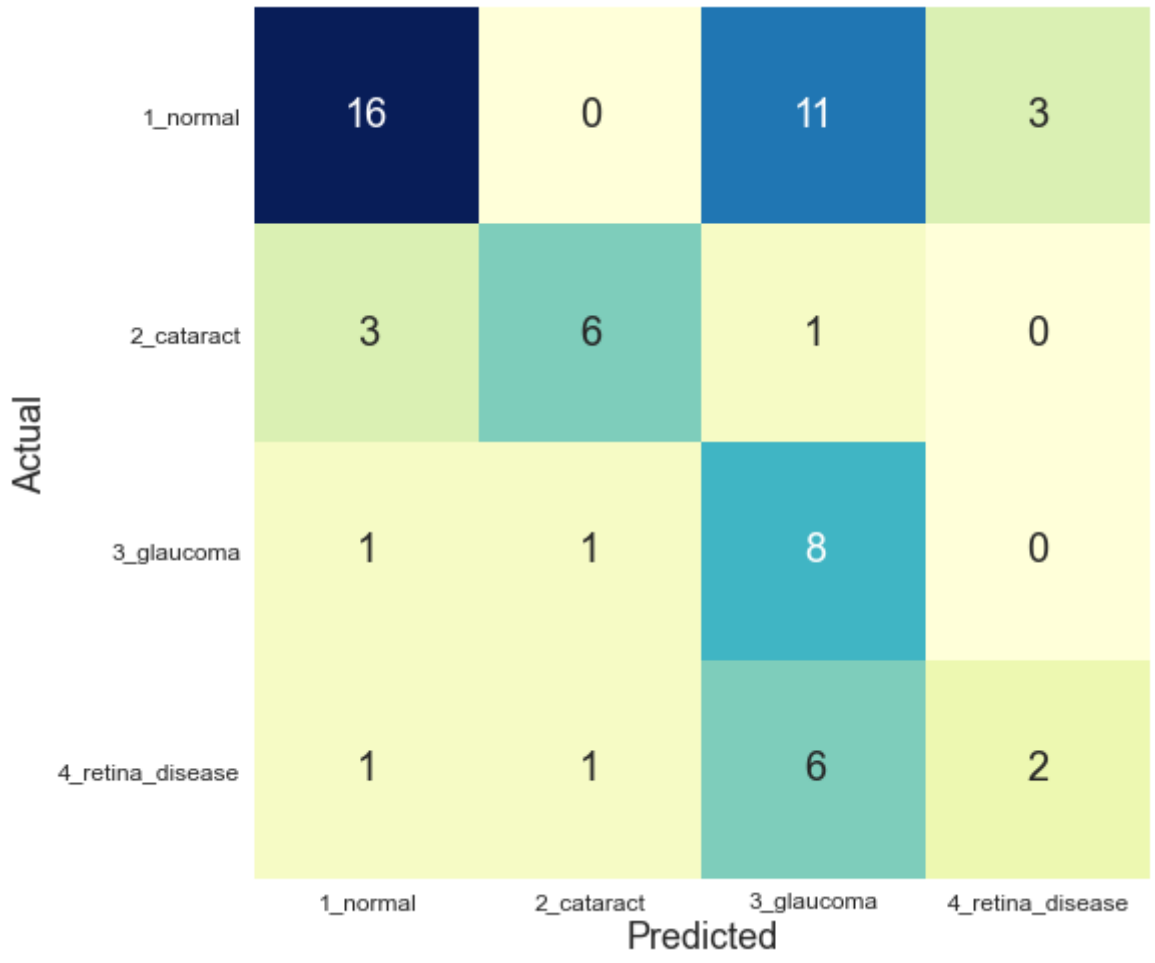
Confusion Matrix for 2 Cross Validation Test phase



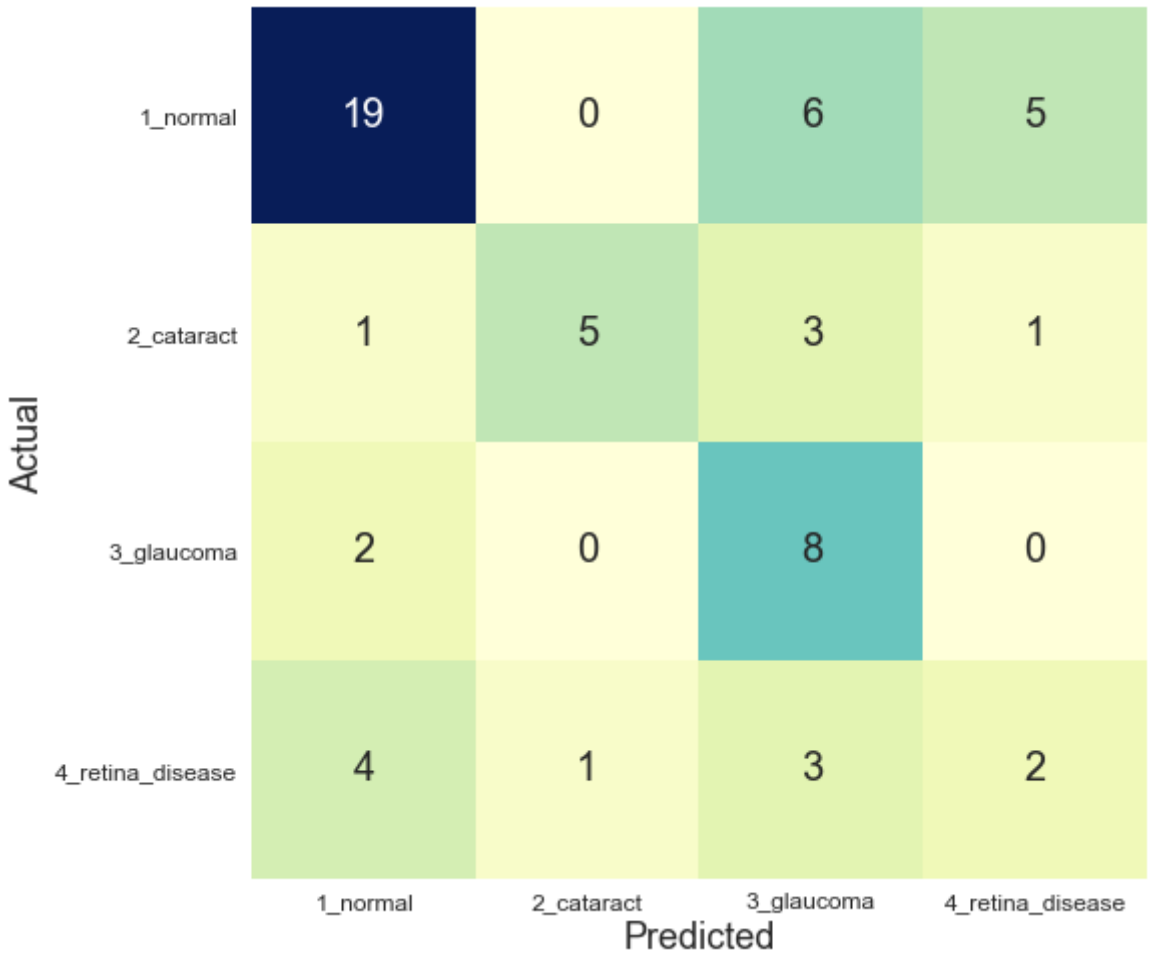
Confusion Matrix for 3 Cross Validation Test phase



Confusion Matrix for 4 Cross Validation Test phase



Confusion Matrix for 5 Cross Validation Test phase

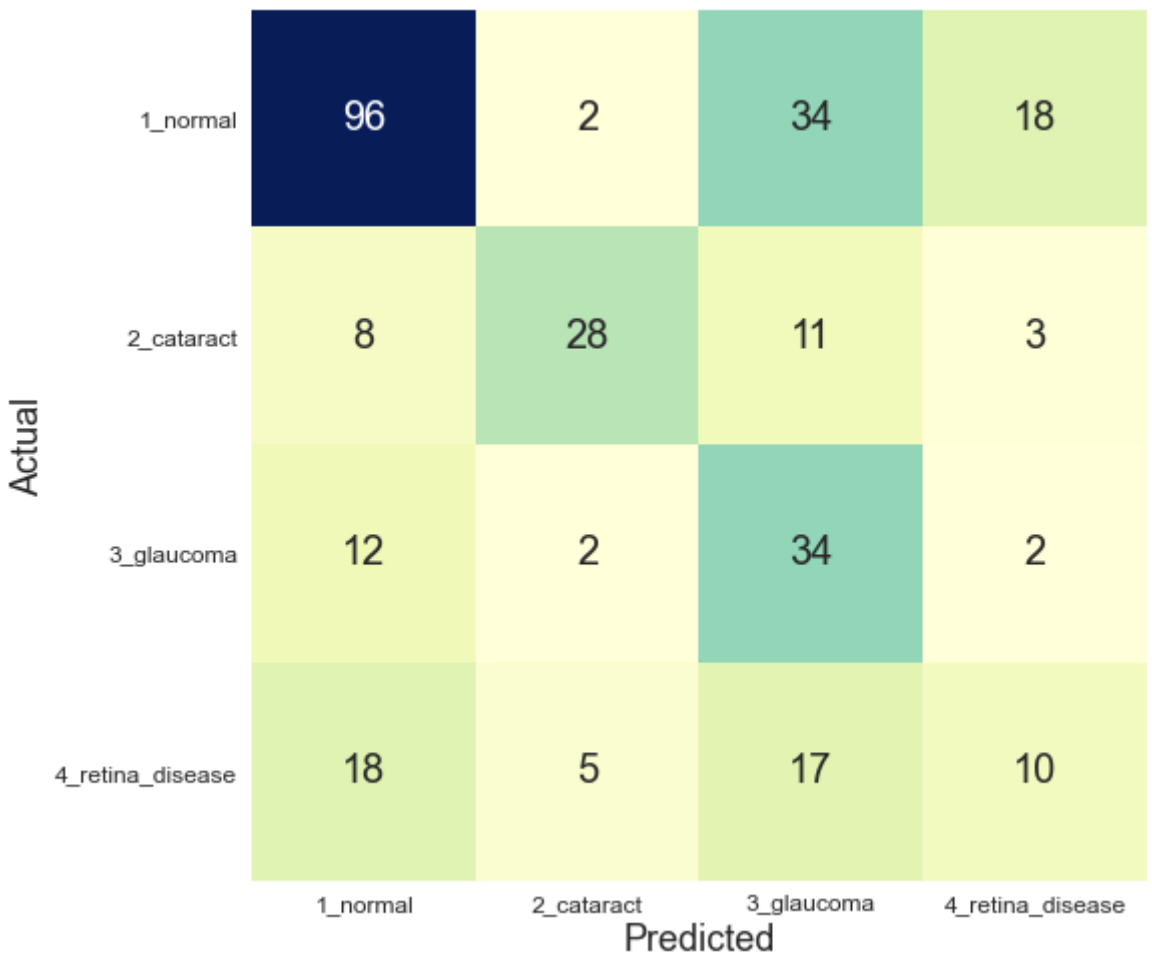


Visualizing Summarized Confusion Matrix of all 5 folds

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

Out[39]:
```
array([[96,  2, 34, 18],
       [ 8, 28, 11,  3],
       [12,  2, 34,  2],
       [18,  5, 17, 10]], 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    :  0.56
Specificity :  0.7876861040412577
Sensitivity :  0.5200000000000001
```

Model Summary

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

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

Out[42]:
```
<tensorflow.python.keras.engine.functional.Functional at 0x15d324a3520>
```

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

Model: "model_4"

Layer (type)	Output Shape	Param #	Connected to
input_5 (InputLayer)	[(None, 224, 224, 3)	0	
zero_padding2d_8 (ZeroPadding2D	(None, 230, 230, 3)	0	input_5[0][0]
conv1/conv (Conv2D)	(None, 112, 112, 64)	9408	zero_padding2d_8[0][0]
conv1/bn (BatchNormalization)	(None, 112, 112, 64)	256	conv1/conv[0][0]
conv1/relu (Activation)	(None, 112, 112, 64)	0	conv1/bn[0][0]
zero_padding2d_9 (ZeroPadding2D	(None, 114, 114, 64)	0	conv1/relu[0][0]
pool1 (MaxPooling2D)	(None, 56, 56, 64)	0	zero_padding2d_9[0][0]
conv2_block1_0_bn (BatchNormali	(None, 56, 56, 64)	256	pool1[0][0]
conv2_block1_0_relu (Activation	(None, 56, 56, 64)	0	conv2_block1_0_bn[0][0]
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 128)	8192	conv2_block1_0_relu[0][0]
conv2_block1_1_bn (BatchNormali	(None, 56, 56, 128)	512	conv2_block1_1_conv[0][0]
conv2_block1_1_relu (Activation	(None, 56, 56, 128)	0	conv2_block1_1_bn[0][0]
conv2_block1_2_conv (Conv2D)	(None, 56, 56, 32)	36864	conv2_block1_1_relu[0][0]
conv2_block1_concat (Concatenat	(None, 56, 56, 96)	0	pool1[0][0] conv2_block1_2_conv[0][0]
conv2_block2_0_bn (BatchNormali	(None, 56, 56, 96)	384	conv2_block1_concat[0][0]
conv2_block2_0_relu (Activation	(None, 56, 56, 96)	0	conv2_block2_0_bn[0][0]
conv2_block2_1_conv (Conv2D)	(None, 56, 56, 128)	12288	conv2_block2_0_relu[0][0]
conv2_block2_1_bn (BatchNormali	(None, 56, 56, 128)	512	conv2_block2_1_conv[0][0]
conv2_block2_1_relu (Activation	(None, 56, 56, 128)	0	conv2_block2_1_bn[0][0]
conv2_block2_2_conv (Conv2D)	(None, 56, 56, 32)	36864	conv2_block2_1_relu[0][0]
conv2_block2_concat (Concatenat	(None, 56, 56, 128)	0	conv2_block1_concat[0][0] conv2_block2_2_conv[0][0]
conv2_block3_0_bn (BatchNormali	(None, 56, 56, 128)	512	conv2_block2_concat[0][0]
conv2_block3_0_relu (Activation	(None, 56, 56, 128)	0	conv2_block3_0_bn[0][0]
conv2_block3_1_conv (Conv2D)	(None, 56, 56, 128)	16384	conv2_block3_0_relu[0][0]
conv2_block3_1_bn (BatchNormali	(None, 56, 56, 128)	512	conv2_block3_1_conv[0][0]
conv2_block3_1_relu (Activation	(None, 56, 56, 128)	0	conv2_block3_1_bn[0][0]
conv2_block3_2_conv (Conv2D)	(None, 56, 56, 32)	36864	conv2_block3_1_relu[0][0]
conv2_block3_concat (Concatenat	(None, 56, 56, 160)	0	conv2_block2_concat[0][0] conv2_block3_2_conv[0][0]
conv2_block4_0_bn (BatchNormali	(None, 56, 56, 160)	640	conv2_block3_concat[0][0]
conv2_block4_0_relu (Activation	(None, 56, 56, 160)	0	conv2_block4_0_bn[0][0]
conv2_block4_1_conv (Conv2D)	(None, 56, 56, 128)	20480	conv2_block4_0_relu[0][0]
conv2_block4_1_bn (BatchNormali	(None, 56, 56, 128)	512	conv2_block4_1_conv[0][0]
conv2_block4_1_relu (Activation	(None, 56, 56, 128)	0	conv2_block4_1_bn[0][0]
conv2_block4_2_conv (Conv2D)	(None, 56, 56, 32)	36864	conv2_block4_1_relu[0][0]
conv2_block4_concat (Concatenat	(None, 56, 56, 192)	0	conv2_block3_concat[0][0] conv2_block4_2_conv[0][0]
conv2_block5_0_bn (BatchNormali	(None, 56, 56, 192)	768	conv2_block4_concat[0][0]
conv2_block5_0_relu (Activation	(None, 56, 56, 192)	0	conv2_block5_0_bn[0][0]
conv2_block5_1_conv (Conv2D)	(None, 56, 56, 128)	24576	conv2_block5_0_relu[0][0]
conv2_block5_1_bn (BatchNormali	(None, 56, 56, 128)	512	conv2_block5_1_conv[0][0]
conv2_block5_1_relu (Activation	(None, 56, 56, 128)	0	conv2_block5_1_bn[0][0]

conv2_block5_2_conv (Conv2D)	(None, 56, 56, 32)	36864	conv2_block5_1_relu[0][0]
conv2_block5_concat (Concatenat	(None, 56, 56, 224)	0	conv2_block4_concat[0][0] conv2_block5_2_conv[0][0]
conv2_block6_0_bn (BatchNormali	(None, 56, 56, 224)	896	conv2_block5_concat[0][0]
conv2_block6_0_relu (Activation	(None, 56, 56, 224)	0	conv2_block6_0_bn[0][0]
conv2_block6_1_conv (Conv2D)	(None, 56, 56, 128)	28672	conv2_block6_0_relu[0][0]
conv2_block6_1_bn (BatchNormali	(None, 56, 56, 128)	512	conv2_block6_1_conv[0][0]
conv2_block6_1_relu (Activation	(None, 56, 56, 128)	0	conv2_block6_1_bn[0][0]
conv2_block6_2_conv (Conv2D)	(None, 56, 56, 32)	36864	conv2_block6_1_relu[0][0]
conv2_block6_concat (Concatenat	(None, 56, 56, 256)	0	conv2_block5_concat[0][0] conv2_block6_2_conv[0][0]
pool2_bn (BatchNormalization)	(None, 56, 56, 256)	1024	conv2_block6_concat[0][0]
pool2_relu (Activation)	(None, 56, 56, 256)	0	pool2_bn[0][0]
pool2_conv (Conv2D)	(None, 56, 56, 128)	32768	pool2_relu[0][0]
pool2_pool (AveragePooling2D)	(None, 28, 28, 128)	0	pool2_conv[0][0]
conv3_block1_0_bn (BatchNormali	(None, 28, 28, 128)	512	pool2_pool[0][0]
conv3_block1_0_relu (Activation	(None, 28, 28, 128)	0	conv3_block1_0_bn[0][0]
conv3_block1_1_conv (Conv2D)	(None, 28, 28, 128)	16384	conv3_block1_0_relu[0][0]
conv3_block1_1_bn (BatchNormali	(None, 28, 28, 128)	512	conv3_block1_1_conv[0][0]
conv3_block1_1_relu (Activation	(None, 28, 28, 128)	0	conv3_block1_1_bn[0][0]
conv3_block1_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block1_1_relu[0][0]
conv3_block1_concat (Concatenat	(None, 28, 28, 160)	0	pool2_pool[0][0] conv3_block1_2_conv[0][0]
conv3_block2_0_bn (BatchNormali	(None, 28, 28, 160)	640	conv3_block1_concat[0][0]
conv3_block2_0_relu (Activation	(None, 28, 28, 160)	0	conv3_block2_0_bn[0][0]
conv3_block2_1_conv (Conv2D)	(None, 28, 28, 128)	20480	conv3_block2_0_relu[0][0]
conv3_block2_1_bn (BatchNormali	(None, 28, 28, 128)	512	conv3_block2_1_conv[0][0]
conv3_block2_1_relu (Activation	(None, 28, 28, 128)	0	conv3_block2_1_bn[0][0]
conv3_block2_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block2_1_relu[0][0]
conv3_block2_concat (Concatenat	(None, 28, 28, 192)	0	conv3_block1_concat[0][0] conv3_block2_2_conv[0][0]
conv3_block3_0_bn (BatchNormali	(None, 28, 28, 192)	768	conv3_block2_concat[0][0]
conv3_block3_0_relu (Activation	(None, 28, 28, 192)	0	conv3_block3_0_bn[0][0]
conv3_block3_1_conv (Conv2D)	(None, 28, 28, 128)	24576	conv3_block3_0_relu[0][0]
conv3_block3_1_bn (BatchNormali	(None, 28, 28, 128)	512	conv3_block3_1_conv[0][0]
conv3_block3_1_relu (Activation	(None, 28, 28, 128)	0	conv3_block3_1_bn[0][0]
conv3_block3_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block3_1_relu[0][0]
conv3_block3_concat (Concatenat	(None, 28, 28, 224)	0	conv3_block2_concat[0][0] conv3_block3_2_conv[0][0]
conv3_block4_0_bn (BatchNormali	(None, 28, 28, 224)	896	conv3_block3_concat[0][0]
conv3_block4_0_relu (Activation	(None, 28, 28, 224)	0	conv3_block4_0_bn[0][0]
conv3_block4_1_conv (Conv2D)	(None, 28, 28, 128)	28672	conv3_block4_0_relu[0][0]
conv3_block4_1_bn (BatchNormali	(None, 28, 28, 128)	512	conv3_block4_1_conv[0][0]
conv3_block4_1_relu (Activation	(None, 28, 28, 128)	0	conv3_block4_1_bn[0][0]
conv3_block4_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block4_1_relu[0][0]
conv3_block4_concat (Concatenat	(None, 28, 28, 256)	0	conv3_block3_concat[0][0] conv3_block4_2_conv[0][0]

conv3_block5_0_bn (BatchNormali	(None, 28, 28, 256)	1024	conv3_block4_concat[0][0]
conv3_block5_0_relu (Activation	(None, 28, 28, 256)	0	conv3_block5_0_bn[0][0]
conv3_block5_1_conv (Conv2D)	(None, 28, 28, 128)	32768	conv3_block5_0_relu[0][0]
conv3_block5_1_bn (BatchNormali	(None, 28, 28, 128)	512	conv3_block5_1_conv[0][0]
conv3_block5_1_relu (Activation	(None, 28, 28, 128)	0	conv3_block5_1_bn[0][0]
conv3_block5_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block5_1_relu[0][0]
conv3_block5_concat (Concatenat	(None, 28, 28, 288)	0	conv3_block4_concat[0][0] conv3_block5_2_conv[0][0]
conv3_block6_0_bn (BatchNormali	(None, 28, 28, 288)	1152	conv3_block5_concat[0][0]
conv3_block6_0_relu (Activation	(None, 28, 28, 288)	0	conv3_block6_0_bn[0][0]
conv3_block6_1_conv (Conv2D)	(None, 28, 28, 128)	36864	conv3_block6_0_relu[0][0]
conv3_block6_1_bn (BatchNormali	(None, 28, 28, 128)	512	conv3_block6_1_conv[0][0]
conv3_block6_1_relu (Activation	(None, 28, 28, 128)	0	conv3_block6_1_bn[0][0]
conv3_block6_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block6_1_relu[0][0]
conv3_block6_concat (Concatenat	(None, 28, 28, 320)	0	conv3_block5_concat[0][0] conv3_block6_2_conv[0][0]
conv3_block7_0_bn (BatchNormali	(None, 28, 28, 320)	1280	conv3_block6_concat[0][0]
conv3_block7_0_relu (Activation	(None, 28, 28, 320)	0	conv3_block7_0_bn[0][0]
conv3_block7_1_conv (Conv2D)	(None, 28, 28, 128)	40960	conv3_block7_0_relu[0][0]
conv3_block7_1_bn (BatchNormali	(None, 28, 28, 128)	512	conv3_block7_1_conv[0][0]
conv3_block7_1_relu (Activation	(None, 28, 28, 128)	0	conv3_block7_1_bn[0][0]
conv3_block7_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block7_1_relu[0][0]
conv3_block7_concat (Concatenat	(None, 28, 28, 352)	0	conv3_block6_concat[0][0] conv3_block7_2_conv[0][0]
conv3_block8_0_bn (BatchNormali	(None, 28, 28, 352)	1408	conv3_block7_concat[0][0]
conv3_block8_0_relu (Activation	(None, 28, 28, 352)	0	conv3_block8_0_bn[0][0]
conv3_block8_1_conv (Conv2D)	(None, 28, 28, 128)	45056	conv3_block8_0_relu[0][0]
conv3_block8_1_bn (BatchNormali	(None, 28, 28, 128)	512	conv3_block8_1_conv[0][0]
conv3_block8_1_relu (Activation	(None, 28, 28, 128)	0	conv3_block8_1_bn[0][0]
conv3_block8_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block8_1_relu[0][0]
conv3_block8_concat (Concatenat	(None, 28, 28, 384)	0	conv3_block7_concat[0][0] conv3_block8_2_conv[0][0]
conv3_block9_0_bn (BatchNormali	(None, 28, 28, 384)	1536	conv3_block8_concat[0][0]
conv3_block9_0_relu (Activation	(None, 28, 28, 384)	0	conv3_block9_0_bn[0][0]
conv3_block9_1_conv (Conv2D)	(None, 28, 28, 128)	49152	conv3_block9_0_relu[0][0]
conv3_block9_1_bn (BatchNormali	(None, 28, 28, 128)	512	conv3_block9_1_conv[0][0]
conv3_block9_1_relu (Activation	(None, 28, 28, 128)	0	conv3_block9_1_bn[0][0]
conv3_block9_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block9_1_relu[0][0]
conv3_block9_concat (Concatenat	(None, 28, 28, 416)	0	conv3_block8_concat[0][0] conv3_block9_2_conv[0][0]
conv3_block10_0_bn (BatchNormal	(None, 28, 28, 416)	1664	conv3_block9_concat[0][0]
conv3_block10_0_relu (Activatio	(None, 28, 28, 416)	0	conv3_block10_0_bn[0][0]
conv3_block10_1_conv (Conv2D)	(None, 28, 28, 128)	53248	conv3_block10_0_relu[0][0]
conv3_block10_1_bn (BatchNormal	(None, 28, 28, 128)	512	conv3_block10_1_conv[0][0]
conv3_block10_1_relu (Activatio	(None, 28, 28, 128)	0	conv3_block10_1_bn[0][0]
conv3_block10_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block10_1_relu[0][0]
conv3_block10_concat (Concatena	(None, 28, 28, 448)	0	conv3_block9_concat[0][0]

			conv3_block10_2_conv[0][0]
conv3_block11_0_bn (BatchNormal	(None, 28, 28, 448)	1792	conv3_block10_concat[0][0]
conv3_block11_0_relu (Activatio	(None, 28, 28, 448)	0	conv3_block11_0_bn[0][0]
conv3_block11_1_conv (Conv2D)	(None, 28, 28, 128)	57344	conv3_block11_0_relu[0][0]
conv3_block11_1_bn (BatchNormal	(None, 28, 28, 128)	512	conv3_block11_1_conv[0][0]
conv3_block11_1_relu (Activatio	(None, 28, 28, 128)	0	conv3_block11_1_bn[0][0]
conv3_block11_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block11_1_relu[0][0]
conv3_block11_concat (Concatena	(None, 28, 28, 480)	0	conv3_block10_concat[0][0] conv3_block11_2_conv[0][0]
conv3_block12_0_bn (BatchNormal	(None, 28, 28, 480)	1920	conv3_block11_concat[0][0]
conv3_block12_0_relu (Activatio	(None, 28, 28, 480)	0	conv3_block12_0_bn[0][0]
conv3_block12_1_conv (Conv2D)	(None, 28, 28, 128)	61440	conv3_block12_0_relu[0][0]
conv3_block12_1_bn (BatchNormal	(None, 28, 28, 128)	512	conv3_block12_1_conv[0][0]
conv3_block12_1_relu (Activatio	(None, 28, 28, 128)	0	conv3_block12_1_bn[0][0]
conv3_block12_2_conv (Conv2D)	(None, 28, 28, 32)	36864	conv3_block12_1_relu[0][0]
conv3_block12_concat (Concatena	(None, 28, 28, 512)	0	conv3_block11_concat[0][0] conv3_block12_2_conv[0][0]
pool3_bn (BatchNormalization)	(None, 28, 28, 512)	2048	conv3_block12_concat[0][0]
pool3_relu (Activation)	(None, 28, 28, 512)	0	pool3_bn[0][0]
pool3_conv (Conv2D)	(None, 28, 28, 256)	131072	pool3_relu[0][0]
pool3_pool (AveragePooling2D)	(None, 14, 14, 256)	0	pool3_conv[0][0]
conv4_block1_0_bn (BatchNormali	(None, 14, 14, 256)	1024	pool3_pool[0][0]
conv4_block1_0_relu (Activation	(None, 14, 14, 256)	0	conv4_block1_0_bn[0][0]
conv4_block1_1_conv (Conv2D)	(None, 14, 14, 128)	32768	conv4_block1_0_relu[0][0]
conv4_block1_1_bn (BatchNormali	(None, 14, 14, 128)	512	conv4_block1_1_conv[0][0]
conv4_block1_1_relu (Activation	(None, 14, 14, 128)	0	conv4_block1_1_bn[0][0]
conv4_block1_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block1_1_relu[0][0]
conv4_block1_concat (Concatenat	(None, 14, 14, 288)	0	pool3_pool[0][0] conv4_block1_2_conv[0][0]
conv4_block2_0_bn (BatchNormali	(None, 14, 14, 288)	1152	conv4_block1_concat[0][0]
conv4_block2_0_relu (Activation	(None, 14, 14, 288)	0	conv4_block2_0_bn[0][0]
conv4_block2_1_conv (Conv2D)	(None, 14, 14, 128)	36864	conv4_block2_0_relu[0][0]
conv4_block2_1_bn (BatchNormali	(None, 14, 14, 128)	512	conv4_block2_1_conv[0][0]
conv4_block2_1_relu (Activation	(None, 14, 14, 128)	0	conv4_block2_1_bn[0][0]
conv4_block2_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block2_1_relu[0][0]
conv4_block2_concat (Concatenat	(None, 14, 14, 320)	0	conv4_block1_concat[0][0] conv4_block2_2_conv[0][0]
conv4_block3_0_bn (BatchNormali	(None, 14, 14, 320)	1280	conv4_block2_concat[0][0]
conv4_block3_0_relu (Activation	(None, 14, 14, 320)	0	conv4_block3_0_bn[0][0]
conv4_block3_1_conv (Conv2D)	(None, 14, 14, 128)	40960	conv4_block3_0_relu[0][0]
conv4_block3_1_bn (BatchNormali	(None, 14, 14, 128)	512	conv4_block3_1_conv[0][0]
conv4_block3_1_relu (Activation	(None, 14, 14, 128)	0	conv4_block3_1_bn[0][0]
conv4_block3_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block3_1_relu[0][0]
conv4_block3_concat (Concatenat	(None, 14, 14, 352)	0	conv4_block2_concat[0][0] conv4_block3_2_conv[0][0]
conv4_block4_0_bn (BatchNormali	(None, 14, 14, 352)	1408	conv4_block3_concat[0][0]
conv4_block4_0_relu (Activation	(None, 14, 14, 352)	0	conv4_block4_0_bn[0][0]

conv4_block4_1_conv (Conv2D)	(None, 14, 14, 128)	45056	conv4_block4_0_relu[0][0]
conv4_block4_1_bn (BatchNormali	(None, 14, 14, 128)	512	conv4_block4_1_conv[0][0]
conv4_block4_1_relu (Activation	(None, 14, 14, 128)	0	conv4_block4_1_bn[0][0]
conv4_block4_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block4_1_relu[0][0]
conv4_block4_concat (Concatenat	(None, 14, 14, 384)	0	conv4_block3_concat[0][0] conv4_block4_2_conv[0][0]
conv4_block5_0_bn (BatchNormali	(None, 14, 14, 384)	1536	conv4_block4_concat[0][0]
conv4_block5_0_relu (Activation	(None, 14, 14, 384)	0	conv4_block5_0_bn[0][0]
conv4_block5_1_conv (Conv2D)	(None, 14, 14, 128)	49152	conv4_block5_0_relu[0][0]
conv4_block5_1_bn (BatchNormali	(None, 14, 14, 128)	512	conv4_block5_1_conv[0][0]
conv4_block5_1_relu (Activation	(None, 14, 14, 128)	0	conv4_block5_1_bn[0][0]
conv4_block5_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block5_1_relu[0][0]
conv4_block5_concat (Concatenat	(None, 14, 14, 416)	0	conv4_block4_concat[0][0] conv4_block5_2_conv[0][0]
conv4_block6_0_bn (BatchNormali	(None, 14, 14, 416)	1664	conv4_block5_concat[0][0]
conv4_block6_0_relu (Activation	(None, 14, 14, 416)	0	conv4_block6_0_bn[0][0]
conv4_block6_1_conv (Conv2D)	(None, 14, 14, 128)	53248	conv4_block6_0_relu[0][0]
conv4_block6_1_bn (BatchNormali	(None, 14, 14, 128)	512	conv4_block6_1_conv[0][0]
conv4_block6_1_relu (Activation	(None, 14, 14, 128)	0	conv4_block6_1_bn[0][0]
conv4_block6_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block6_1_relu[0][0]
conv4_block6_concat (Concatenat	(None, 14, 14, 448)	0	conv4_block5_concat[0][0] conv4_block6_2_conv[0][0]
conv4_block7_0_bn (BatchNormali	(None, 14, 14, 448)	1792	conv4_block6_concat[0][0]
conv4_block7_0_relu (Activation	(None, 14, 14, 448)	0	conv4_block7_0_bn[0][0]
conv4_block7_1_conv (Conv2D)	(None, 14, 14, 128)	57344	conv4_block7_0_relu[0][0]
conv4_block7_1_bn (BatchNormali	(None, 14, 14, 128)	512	conv4_block7_1_conv[0][0]
conv4_block7_1_relu (Activation	(None, 14, 14, 128)	0	conv4_block7_1_bn[0][0]
conv4_block7_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block7_1_relu[0][0]
conv4_block7_concat (Concatenat	(None, 14, 14, 480)	0	conv4_block6_concat[0][0] conv4_block7_2_conv[0][0]
conv4_block8_0_bn (BatchNormali	(None, 14, 14, 480)	1920	conv4_block7_concat[0][0]
conv4_block8_0_relu (Activation	(None, 14, 14, 480)	0	conv4_block8_0_bn[0][0]
conv4_block8_1_conv (Conv2D)	(None, 14, 14, 128)	61440	conv4_block8_0_relu[0][0]
conv4_block8_1_bn (BatchNormali	(None, 14, 14, 128)	512	conv4_block8_1_conv[0][0]
conv4_block8_1_relu (Activation	(None, 14, 14, 128)	0	conv4_block8_1_bn[0][0]
conv4_block8_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block8_1_relu[0][0]
conv4_block8_concat (Concatenat	(None, 14, 14, 512)	0	conv4_block7_concat[0][0] conv4_block8_2_conv[0][0]
conv4_block9_0_bn (BatchNormali	(None, 14, 14, 512)	2048	conv4_block8_concat[0][0]
conv4_block9_0_relu (Activation	(None, 14, 14, 512)	0	conv4_block9_0_bn[0][0]
conv4_block9_1_conv (Conv2D)	(None, 14, 14, 128)	65536	conv4_block9_0_relu[0][0]
conv4_block9_1_bn (BatchNormali	(None, 14, 14, 128)	512	conv4_block9_1_conv[0][0]
conv4_block9_1_relu (Activation	(None, 14, 14, 128)	0	conv4_block9_1_bn[0][0]
conv4_block9_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block9_1_relu[0][0]
conv4_block9_concat (Concatenat	(None, 14, 14, 544)	0	conv4_block8_concat[0][0] conv4_block9_2_conv[0][0]
conv4_block10_0_bn (BatchNormal	(None, 14, 14, 544)	2176	conv4_block9_concat[0][0]

conv4_block10_0_relu (Activatio	(None, 14, 14, 544)	0	conv4_block10_0_bn[0][0]
conv4_block10_1_conv (Conv2D)	(None, 14, 14, 128)	69632	conv4_block10_0_relu[0][0]
conv4_block10_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block10_1_conv[0][0]
conv4_block10_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block10_1_bn[0][0]
conv4_block10_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block10_1_relu[0][0]
conv4_block10_concat (Concatena	(None, 14, 14, 576)	0	conv4_block9_concat[0][0] conv4_block10_2_conv[0][0]
conv4_block11_0_bn (BatchNormal	(None, 14, 14, 576)	2304	conv4_block10_concat[0][0]
conv4_block11_0_relu (Activatio	(None, 14, 14, 576)	0	conv4_block11_0_bn[0][0]
conv4_block11_1_conv (Conv2D)	(None, 14, 14, 128)	73728	conv4_block11_0_relu[0][0]
conv4_block11_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block11_1_conv[0][0]
conv4_block11_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block11_1_bn[0][0]
conv4_block11_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block11_1_relu[0][0]
conv4_block11_concat (Concatena	(None, 14, 14, 608)	0	conv4_block10_concat[0][0] conv4_block11_2_conv[0][0]
conv4_block12_0_bn (BatchNormal	(None, 14, 14, 608)	2432	conv4_block11_concat[0][0]
conv4_block12_0_relu (Activatio	(None, 14, 14, 608)	0	conv4_block12_0_bn[0][0]
conv4_block12_1_conv (Conv2D)	(None, 14, 14, 128)	77824	conv4_block12_0_relu[0][0]
conv4_block12_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block12_1_conv[0][0]
conv4_block12_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block12_1_bn[0][0]
conv4_block12_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block12_1_relu[0][0]
conv4_block12_concat (Concatena	(None, 14, 14, 640)	0	conv4_block11_concat[0][0] conv4_block12_2_conv[0][0]
conv4_block13_0_bn (BatchNormal	(None, 14, 14, 640)	2560	conv4_block12_concat[0][0]
conv4_block13_0_relu (Activatio	(None, 14, 14, 640)	0	conv4_block13_0_bn[0][0]
conv4_block13_1_conv (Conv2D)	(None, 14, 14, 128)	81920	conv4_block13_0_relu[0][0]
conv4_block13_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block13_1_conv[0][0]
conv4_block13_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block13_1_bn[0][0]
conv4_block13_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block13_1_relu[0][0]
conv4_block13_concat (Concatena	(None, 14, 14, 672)	0	conv4_block12_concat[0][0] conv4_block13_2_conv[0][0]
conv4_block14_0_bn (BatchNormal	(None, 14, 14, 672)	2688	conv4_block13_concat[0][0]
conv4_block14_0_relu (Activatio	(None, 14, 14, 672)	0	conv4_block14_0_bn[0][0]
conv4_block14_1_conv (Conv2D)	(None, 14, 14, 128)	86016	conv4_block14_0_relu[0][0]
conv4_block14_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block14_1_conv[0][0]
conv4_block14_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block14_1_bn[0][0]
conv4_block14_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block14_1_relu[0][0]
conv4_block14_concat (Concatena	(None, 14, 14, 704)	0	conv4_block13_concat[0][0] conv4_block14_2_conv[0][0]
conv4_block15_0_bn (BatchNormal	(None, 14, 14, 704)	2816	conv4_block14_concat[0][0]
conv4_block15_0_relu (Activatio	(None, 14, 14, 704)	0	conv4_block15_0_bn[0][0]
conv4_block15_1_conv (Conv2D)	(None, 14, 14, 128)	90112	conv4_block15_0_relu[0][0]
conv4_block15_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block15_1_conv[0][0]
conv4_block15_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block15_1_bn[0][0]
conv4_block15_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block15_1_relu[0][0]
conv4_block15_concat (Concatena	(None, 14, 14, 736)	0	conv4_block14_concat[0][0] conv4_block15_2_conv[0][0]

conv4_block16_0_bn (BatchNormal	(None, 14, 14, 736)	2944	conv4_block15_concat[0][0]
conv4_block16_0_relu (Activatio	(None, 14, 14, 736)	0	conv4_block16_0_bn[0][0]
conv4_block16_1_conv (Conv2D)	(None, 14, 14, 128)	94208	conv4_block16_0_relu[0][0]
conv4_block16_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block16_1_conv[0][0]
conv4_block16_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block16_1_bn[0][0]
conv4_block16_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block16_1_relu[0][0]
conv4_block16_concat (Concatena	(None, 14, 14, 768)	0	conv4_block15_concat[0][0] conv4_block16_2_conv[0][0]
conv4_block17_0_bn (BatchNormal	(None, 14, 14, 768)	3072	conv4_block16_concat[0][0]
conv4_block17_0_relu (Activatio	(None, 14, 14, 768)	0	conv4_block17_0_bn[0][0]
conv4_block17_1_conv (Conv2D)	(None, 14, 14, 128)	98304	conv4_block17_0_relu[0][0]
conv4_block17_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block17_1_conv[0][0]
conv4_block17_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block17_1_bn[0][0]
conv4_block17_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block17_1_relu[0][0]
conv4_block17_concat (Concatena	(None, 14, 14, 800)	0	conv4_block16_concat[0][0] conv4_block17_2_conv[0][0]
conv4_block18_0_bn (BatchNormal	(None, 14, 14, 800)	3200	conv4_block17_concat[0][0]
conv4_block18_0_relu (Activatio	(None, 14, 14, 800)	0	conv4_block18_0_bn[0][0]
conv4_block18_1_conv (Conv2D)	(None, 14, 14, 128)	102400	conv4_block18_0_relu[0][0]
conv4_block18_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block18_1_conv[0][0]
conv4_block18_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block18_1_bn[0][0]
conv4_block18_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block18_1_relu[0][0]
conv4_block18_concat (Concatena	(None, 14, 14, 832)	0	conv4_block17_concat[0][0] conv4_block18_2_conv[0][0]
conv4_block19_0_bn (BatchNormal	(None, 14, 14, 832)	3328	conv4_block18_concat[0][0]
conv4_block19_0_relu (Activatio	(None, 14, 14, 832)	0	conv4_block19_0_bn[0][0]
conv4_block19_1_conv (Conv2D)	(None, 14, 14, 128)	106496	conv4_block19_0_relu[0][0]
conv4_block19_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block19_1_conv[0][0]
conv4_block19_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block19_1_bn[0][0]
conv4_block19_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block19_1_relu[0][0]
conv4_block19_concat (Concatena	(None, 14, 14, 864)	0	conv4_block18_concat[0][0] conv4_block19_2_conv[0][0]
conv4_block20_0_bn (BatchNormal	(None, 14, 14, 864)	3456	conv4_block19_concat[0][0]
conv4_block20_0_relu (Activatio	(None, 14, 14, 864)	0	conv4_block20_0_bn[0][0]
conv4_block20_1_conv (Conv2D)	(None, 14, 14, 128)	110592	conv4_block20_0_relu[0][0]
conv4_block20_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block20_1_conv[0][0]
conv4_block20_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block20_1_bn[0][0]
conv4_block20_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block20_1_relu[0][0]
conv4_block20_concat (Concatena	(None, 14, 14, 896)	0	conv4_block19_concat[0][0] conv4_block20_2_conv[0][0]
conv4_block21_0_bn (BatchNormal	(None, 14, 14, 896)	3584	conv4_block20_concat[0][0]
conv4_block21_0_relu (Activatio	(None, 14, 14, 896)	0	conv4_block21_0_bn[0][0]
conv4_block21_1_conv (Conv2D)	(None, 14, 14, 128)	114688	conv4_block21_0_relu[0][0]
conv4_block21_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block21_1_conv[0][0]
conv4_block21_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block21_1_bn[0][0]
conv4_block21_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block21_1_relu[0][0]

conv4_block21_concat (Concatena	(None, 14, 14, 928)	0	conv4_block20_concat[0][0] conv4_block21_2_conv[0][0]
conv4_block22_0_bn (BatchNormal	(None, 14, 14, 928)	3712	conv4_block21_concat[0][0]
conv4_block22_0_relu (Activatio	(None, 14, 14, 928)	0	conv4_block22_0_bn[0][0]
conv4_block22_1_conv (Conv2D)	(None, 14, 14, 128)	118784	conv4_block22_0_relu[0][0]
conv4_block22_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block22_1_conv[0][0]
conv4_block22_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block22_1_bn[0][0]
conv4_block22_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block22_1_relu[0][0]
conv4_block22_concat (Concatena	(None, 14, 14, 960)	0	conv4_block21_concat[0][0] conv4_block22_2_conv[0][0]
conv4_block23_0_bn (BatchNormal	(None, 14, 14, 960)	3840	conv4_block22_concat[0][0]
conv4_block23_0_relu (Activatio	(None, 14, 14, 960)	0	conv4_block23_0_bn[0][0]
conv4_block23_1_conv (Conv2D)	(None, 14, 14, 128)	122880	conv4_block23_0_relu[0][0]
conv4_block23_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block23_1_conv[0][0]
conv4_block23_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block23_1_bn[0][0]
conv4_block23_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block23_1_relu[0][0]
conv4_block23_concat (Concatena	(None, 14, 14, 992)	0	conv4_block22_concat[0][0] conv4_block23_2_conv[0][0]
conv4_block24_0_bn (BatchNormal	(None, 14, 14, 992)	3968	conv4_block23_concat[0][0]
conv4_block24_0_relu (Activatio	(None, 14, 14, 992)	0	conv4_block24_0_bn[0][0]
conv4_block24_1_conv (Conv2D)	(None, 14, 14, 128)	126976	conv4_block24_0_relu[0][0]
conv4_block24_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block24_1_conv[0][0]
conv4_block24_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block24_1_bn[0][0]
conv4_block24_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block24_1_relu[0][0]
conv4_block24_concat (Concatena	(None, 14, 14, 1024)	0	conv4_block23_concat[0][0] conv4_block24_2_conv[0][0]
conv4_block25_0_bn (BatchNormal	(None, 14, 14, 1024)	4096	conv4_block24_concat[0][0]
conv4_block25_0_relu (Activatio	(None, 14, 14, 1024)	0	conv4_block25_0_bn[0][0]
conv4_block25_1_conv (Conv2D)	(None, 14, 14, 128)	131072	conv4_block25_0_relu[0][0]
conv4_block25_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block25_1_conv[0][0]
conv4_block25_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block25_1_bn[0][0]
conv4_block25_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block25_1_relu[0][0]
conv4_block25_concat (Concatena	(None, 14, 14, 1056)	0	conv4_block24_concat[0][0] conv4_block25_2_conv[0][0]
conv4_block26_0_bn (BatchNormal	(None, 14, 14, 1056)	4224	conv4_block25_concat[0][0]
conv4_block26_0_relu (Activatio	(None, 14, 14, 1056)	0	conv4_block26_0_bn[0][0]
conv4_block26_1_conv (Conv2D)	(None, 14, 14, 128)	135168	conv4_block26_0_relu[0][0]
conv4_block26_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block26_1_conv[0][0]
conv4_block26_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block26_1_bn[0][0]
conv4_block26_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block26_1_relu[0][0]
conv4_block26_concat (Concatena	(None, 14, 14, 1088)	0	conv4_block25_concat[0][0] conv4_block26_2_conv[0][0]
conv4_block27_0_bn (BatchNormal	(None, 14, 14, 1088)	4352	conv4_block26_concat[0][0]
conv4_block27_0_relu (Activatio	(None, 14, 14, 1088)	0	conv4_block27_0_bn[0][0]
conv4_block27_1_conv (Conv2D)	(None, 14, 14, 128)	139264	conv4_block27_0_relu[0][0]
conv4_block27_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block27_1_conv[0][0]
conv4_block27_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block27_1_bn[0][0]

conv4_block27_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block27_1_relu[0][0]
conv4_block27_concat (Concatena	(None, 14, 14, 1120)	0	conv4_block26_concat[0][0] conv4_block27_2_conv[0][0]
conv4_block28_0_bn (BatchNormal	(None, 14, 14, 1120)	4480	conv4_block27_concat[0][0]
conv4_block28_0_relu (Activatio	(None, 14, 14, 1120)	0	conv4_block28_0_bn[0][0]
conv4_block28_1_conv (Conv2D)	(None, 14, 14, 128)	143360	conv4_block28_0_relu[0][0]
conv4_block28_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block28_1_conv[0][0]
conv4_block28_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block28_1_bn[0][0]
conv4_block28_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block28_1_relu[0][0]
conv4_block28_concat (Concatena	(None, 14, 14, 1152)	0	conv4_block27_concat[0][0] conv4_block28_2_conv[0][0]
conv4_block29_0_bn (BatchNormal	(None, 14, 14, 1152)	4608	conv4_block28_concat[0][0]
conv4_block29_0_relu (Activatio	(None, 14, 14, 1152)	0	conv4_block29_0_bn[0][0]
conv4_block29_1_conv (Conv2D)	(None, 14, 14, 128)	147456	conv4_block29_0_relu[0][0]
conv4_block29_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block29_1_conv[0][0]
conv4_block29_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block29_1_bn[0][0]
conv4_block29_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block29_1_relu[0][0]
conv4_block29_concat (Concatena	(None, 14, 14, 1184)	0	conv4_block28_concat[0][0] conv4_block29_2_conv[0][0]
conv4_block30_0_bn (BatchNormal	(None, 14, 14, 1184)	4736	conv4_block29_concat[0][0]
conv4_block30_0_relu (Activatio	(None, 14, 14, 1184)	0	conv4_block30_0_bn[0][0]
conv4_block30_1_conv (Conv2D)	(None, 14, 14, 128)	151552	conv4_block30_0_relu[0][0]
conv4_block30_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block30_1_conv[0][0]
conv4_block30_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block30_1_bn[0][0]
conv4_block30_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block30_1_relu[0][0]
conv4_block30_concat (Concatena	(None, 14, 14, 1216)	0	conv4_block29_concat[0][0] conv4_block30_2_conv[0][0]
conv4_block31_0_bn (BatchNormal	(None, 14, 14, 1216)	4864	conv4_block30_concat[0][0]
conv4_block31_0_relu (Activatio	(None, 14, 14, 1216)	0	conv4_block31_0_bn[0][0]
conv4_block31_1_conv (Conv2D)	(None, 14, 14, 128)	155648	conv4_block31_0_relu[0][0]
conv4_block31_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block31_1_conv[0][0]
conv4_block31_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block31_1_bn[0][0]
conv4_block31_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block31_1_relu[0][0]
conv4_block31_concat (Concatena	(None, 14, 14, 1248)	0	conv4_block30_concat[0][0] conv4_block31_2_conv[0][0]
conv4_block32_0_bn (BatchNormal	(None, 14, 14, 1248)	4992	conv4_block31_concat[0][0]
conv4_block32_0_relu (Activatio	(None, 14, 14, 1248)	0	conv4_block32_0_bn[0][0]
conv4_block32_1_conv (Conv2D)	(None, 14, 14, 128)	159744	conv4_block32_0_relu[0][0]
conv4_block32_1_bn (BatchNormal	(None, 14, 14, 128)	512	conv4_block32_1_conv[0][0]
conv4_block32_1_relu (Activatio	(None, 14, 14, 128)	0	conv4_block32_1_bn[0][0]
conv4_block32_2_conv (Conv2D)	(None, 14, 14, 32)	36864	conv4_block32_1_relu[0][0]
conv4_block32_concat (Concatena	(None, 14, 14, 1280)	0	conv4_block31_concat[0][0] conv4_block32_2_conv[0][0]
pool4_bn (BatchNormalization)	(None, 14, 14, 1280)	5120	conv4_block32_concat[0][0]
pool4_relu (Activation)	(None, 14, 14, 1280)	0	pool4_bn[0][0]
pool4_conv (Conv2D)	(None, 14, 14, 640)	819200	pool4_relu[0][0]
pool4_pool (AveragePooling2D)	(None, 7, 7, 640)	0	pool4_conv[0][0]

conv5_block1_0_bn (BatchNormali	(None, 7, 7, 640)	2560	pool4_pool[0][0]
conv5_block1_0_relu (Activation	(None, 7, 7, 640)	0	conv5_block1_0_bn[0][0]
conv5_block1_1_conv (Conv2D)	(None, 7, 7, 128)	81920	conv5_block1_0_relu[0][0]
conv5_block1_1_bn (BatchNormali	(None, 7, 7, 128)	512	conv5_block1_1_conv[0][0]
conv5_block1_1_relu (Activation	(None, 7, 7, 128)	0	conv5_block1_1_bn[0][0]
conv5_block1_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block1_1_relu[0][0]
conv5_block1_concat (Concatenat	(None, 7, 7, 672)	0	pool4_pool[0][0] conv5_block1_2_conv[0][0]
conv5_block2_0_bn (BatchNormali	(None, 7, 7, 672)	2688	conv5_block1_concat[0][0]
conv5_block2_0_relu (Activation	(None, 7, 7, 672)	0	conv5_block2_0_bn[0][0]
conv5_block2_1_conv (Conv2D)	(None, 7, 7, 128)	86016	conv5_block2_0_relu[0][0]
conv5_block2_1_bn (BatchNormali	(None, 7, 7, 128)	512	conv5_block2_1_conv[0][0]
conv5_block2_1_relu (Activation	(None, 7, 7, 128)	0	conv5_block2_1_bn[0][0]
conv5_block2_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block2_1_relu[0][0]
conv5_block2_concat (Concatenat	(None, 7, 7, 704)	0	conv5_block1_concat[0][0] conv5_block2_2_conv[0][0]
conv5_block3_0_bn (BatchNormali	(None, 7, 7, 704)	2816	conv5_block2_concat[0][0]
conv5_block3_0_relu (Activation	(None, 7, 7, 704)	0	conv5_block3_0_bn[0][0]
conv5_block3_1_conv (Conv2D)	(None, 7, 7, 128)	90112	conv5_block3_0_relu[0][0]
conv5_block3_1_bn (BatchNormali	(None, 7, 7, 128)	512	conv5_block3_1_conv[0][0]
conv5_block3_1_relu (Activation	(None, 7, 7, 128)	0	conv5_block3_1_bn[0][0]
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block3_1_relu[0][0]
conv5_block3_concat (Concatenat	(None, 7, 7, 736)	0	conv5_block2_concat[0][0] conv5_block3_2_conv[0][0]
conv5_block4_0_bn (BatchNormali	(None, 7, 7, 736)	2944	conv5_block3_concat[0][0]
conv5_block4_0_relu (Activation	(None, 7, 7, 736)	0	conv5_block4_0_bn[0][0]
conv5_block4_1_conv (Conv2D)	(None, 7, 7, 128)	94208	conv5_block4_0_relu[0][0]
conv5_block4_1_bn (BatchNormali	(None, 7, 7, 128)	512	conv5_block4_1_conv[0][0]
conv5_block4_1_relu (Activation	(None, 7, 7, 128)	0	conv5_block4_1_bn[0][0]
conv5_block4_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block4_1_relu[0][0]
conv5_block4_concat (Concatenat	(None, 7, 7, 768)	0	conv5_block3_concat[0][0] conv5_block4_2_conv[0][0]
conv5_block5_0_bn (BatchNormali	(None, 7, 7, 768)	3072	conv5_block4_concat[0][0]
conv5_block5_0_relu (Activation	(None, 7, 7, 768)	0	conv5_block5_0_bn[0][0]
conv5_block5_1_conv (Conv2D)	(None, 7, 7, 128)	98304	conv5_block5_0_relu[0][0]
conv5_block5_1_bn (BatchNormali	(None, 7, 7, 128)	512	conv5_block5_1_conv[0][0]
conv5_block5_1_relu (Activation	(None, 7, 7, 128)	0	conv5_block5_1_bn[0][0]
conv5_block5_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block5_1_relu[0][0]
conv5_block5_concat (Concatenat	(None, 7, 7, 800)	0	conv5_block4_concat[0][0] conv5_block5_2_conv[0][0]
conv5_block6_0_bn (BatchNormali	(None, 7, 7, 800)	3200	conv5_block5_concat[0][0]
conv5_block6_0_relu (Activation	(None, 7, 7, 800)	0	conv5_block6_0_bn[0][0]
conv5_block6_1_conv (Conv2D)	(None, 7, 7, 128)	102400	conv5_block6_0_relu[0][0]
conv5_block6_1_bn (BatchNormali	(None, 7, 7, 128)	512	conv5_block6_1_conv[0][0]
conv5_block6_1_relu (Activation	(None, 7, 7, 128)	0	conv5_block6_1_bn[0][0]
conv5_block6_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block6_1_relu[0][0]
conv5_block6_concat (Concatenat	(None, 7, 7, 832)	0	conv5_block5_concat[0][0]

				conv5_block6_2_conv[0][0]
conv5_block7_0_bn (BatchNormali	(None, 7, 7, 832)	3328	conv5_block6_concat[0][0]	
conv5_block7_0_relu (Activation	(None, 7, 7, 832)	0	conv5_block7_0_bn[0][0]	
conv5_block7_1_conv (Conv2D)	(None, 7, 7, 128)	106496	conv5_block7_0_relu[0][0]	
conv5_block7_1_bn (BatchNormali	(None, 7, 7, 128)	512	conv5_block7_1_conv[0][0]	
conv5_block7_1_relu (Activation	(None, 7, 7, 128)	0	conv5_block7_1_bn[0][0]	
conv5_block7_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block7_1_relu[0][0]	
conv5_block7_concat (Concatenat	(None, 7, 7, 864)	0	conv5_block6_concat[0][0] conv5_block7_2_conv[0][0]	
conv5_block8_0_bn (BatchNormali	(None, 7, 7, 864)	3456	conv5_block7_concat[0][0]	
conv5_block8_0_relu (Activation	(None, 7, 7, 864)	0	conv5_block8_0_bn[0][0]	
conv5_block8_1_conv (Conv2D)	(None, 7, 7, 128)	110592	conv5_block8_0_relu[0][0]	
conv5_block8_1_bn (BatchNormali	(None, 7, 7, 128)	512	conv5_block8_1_conv[0][0]	
conv5_block8_1_relu (Activation	(None, 7, 7, 128)	0	conv5_block8_1_bn[0][0]	
conv5_block8_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block8_1_relu[0][0]	
conv5_block8_concat (Concatenat	(None, 7, 7, 896)	0	conv5_block7_concat[0][0] conv5_block8_2_conv[0][0]	
conv5_block9_0_bn (BatchNormali	(None, 7, 7, 896)	3584	conv5_block8_concat[0][0]	
conv5_block9_0_relu (Activation	(None, 7, 7, 896)	0	conv5_block9_0_bn[0][0]	
conv5_block9_1_conv (Conv2D)	(None, 7, 7, 128)	114688	conv5_block9_0_relu[0][0]	
conv5_block9_1_bn (BatchNormali	(None, 7, 7, 128)	512	conv5_block9_1_conv[0][0]	
conv5_block9_1_relu (Activation	(None, 7, 7, 128)	0	conv5_block9_1_bn[0][0]	
conv5_block9_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block9_1_relu[0][0]	
conv5_block9_concat (Concatenat	(None, 7, 7, 928)	0	conv5_block8_concat[0][0] conv5_block9_2_conv[0][0]	
conv5_block10_0_bn (BatchNormal	(None, 7, 7, 928)	3712	conv5_block9_concat[0][0]	
conv5_block10_0_relu (Activatio	(None, 7, 7, 928)	0	conv5_block10_0_bn[0][0]	
conv5_block10_1_conv (Conv2D)	(None, 7, 7, 128)	118784	conv5_block10_0_relu[0][0]	
conv5_block10_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block10_1_conv[0][0]	
conv5_block10_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block10_1_bn[0][0]	
conv5_block10_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block10_1_relu[0][0]	
conv5_block10_concat (Concatena	(None, 7, 7, 960)	0	conv5_block9_concat[0][0] conv5_block10_2_conv[0][0]	
conv5_block11_0_bn (BatchNormal	(None, 7, 7, 960)	3840	conv5_block10_concat[0][0]	
conv5_block11_0_relu (Activatio	(None, 7, 7, 960)	0	conv5_block11_0_bn[0][0]	
conv5_block11_1_conv (Conv2D)	(None, 7, 7, 128)	122880	conv5_block11_0_relu[0][0]	
conv5_block11_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block11_1_conv[0][0]	
conv5_block11_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block11_1_bn[0][0]	
conv5_block11_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block11_1_relu[0][0]	
conv5_block11_concat (Concatena	(None, 7, 7, 992)	0	conv5_block10_concat[0][0] conv5_block11_2_conv[0][0]	
conv5_block12_0_bn (BatchNormal	(None, 7, 7, 992)	3968	conv5_block11_concat[0][0]	
conv5_block12_0_relu (Activatio	(None, 7, 7, 992)	0	conv5_block12_0_bn[0][0]	
conv5_block12_1_conv (Conv2D)	(None, 7, 7, 128)	126976	conv5_block12_0_relu[0][0]	
conv5_block12_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block12_1_conv[0][0]	
conv5_block12_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block12_1_bn[0][0]	
conv5_block12_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block12_1_relu[0][0]	

conv5_block12_concat (Concatena	(None, 7, 7, 1024)	0	conv5_block11_concat[0][0] conv5_block12_2_conv[0][0]
conv5_block13_0_bn (BatchNormal	(None, 7, 7, 1024)	4096	conv5_block12_concat[0][0]
conv5_block13_0_relu (Activatio	(None, 7, 7, 1024)	0	conv5_block13_0_bn[0][0]
conv5_block13_1_conv (Conv2D)	(None, 7, 7, 128)	131072	conv5_block13_0_relu[0][0]
conv5_block13_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block13_1_conv[0][0]
conv5_block13_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block13_1_bn[0][0]
conv5_block13_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block13_1_relu[0][0]
conv5_block13_concat (Concatena	(None, 7, 7, 1056)	0	conv5_block12_concat[0][0] conv5_block13_2_conv[0][0]
conv5_block14_0_bn (BatchNormal	(None, 7, 7, 1056)	4224	conv5_block13_concat[0][0]
conv5_block14_0_relu (Activatio	(None, 7, 7, 1056)	0	conv5_block14_0_bn[0][0]
conv5_block14_1_conv (Conv2D)	(None, 7, 7, 128)	135168	conv5_block14_0_relu[0][0]
conv5_block14_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block14_1_conv[0][0]
conv5_block14_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block14_1_bn[0][0]
conv5_block14_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block14_1_relu[0][0]
conv5_block14_concat (Concatena	(None, 7, 7, 1088)	0	conv5_block13_concat[0][0] conv5_block14_2_conv[0][0]
conv5_block15_0_bn (BatchNormal	(None, 7, 7, 1088)	4352	conv5_block14_concat[0][0]
conv5_block15_0_relu (Activatio	(None, 7, 7, 1088)	0	conv5_block15_0_bn[0][0]
conv5_block15_1_conv (Conv2D)	(None, 7, 7, 128)	139264	conv5_block15_0_relu[0][0]
conv5_block15_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block15_1_conv[0][0]
conv5_block15_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block15_1_bn[0][0]
conv5_block15_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block15_1_relu[0][0]
conv5_block15_concat (Concatena	(None, 7, 7, 1120)	0	conv5_block14_concat[0][0] conv5_block15_2_conv[0][0]
conv5_block16_0_bn (BatchNormal	(None, 7, 7, 1120)	4480	conv5_block15_concat[0][0]
conv5_block16_0_relu (Activatio	(None, 7, 7, 1120)	0	conv5_block16_0_bn[0][0]
conv5_block16_1_conv (Conv2D)	(None, 7, 7, 128)	143360	conv5_block16_0_relu[0][0]
conv5_block16_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block16_1_conv[0][0]
conv5_block16_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block16_1_bn[0][0]
conv5_block16_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block16_1_relu[0][0]
conv5_block16_concat (Concatena	(None, 7, 7, 1152)	0	conv5_block15_concat[0][0] conv5_block16_2_conv[0][0]
conv5_block17_0_bn (BatchNormal	(None, 7, 7, 1152)	4608	conv5_block16_concat[0][0]
conv5_block17_0_relu (Activatio	(None, 7, 7, 1152)	0	conv5_block17_0_bn[0][0]
conv5_block17_1_conv (Conv2D)	(None, 7, 7, 128)	147456	conv5_block17_0_relu[0][0]
conv5_block17_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block17_1_conv[0][0]
conv5_block17_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block17_1_bn[0][0]
conv5_block17_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block17_1_relu[0][0]
conv5_block17_concat (Concatena	(None, 7, 7, 1184)	0	conv5_block16_concat[0][0] conv5_block17_2_conv[0][0]
conv5_block18_0_bn (BatchNormal	(None, 7, 7, 1184)	4736	conv5_block17_concat[0][0]
conv5_block18_0_relu (Activatio	(None, 7, 7, 1184)	0	conv5_block18_0_bn[0][0]
conv5_block18_1_conv (Conv2D)	(None, 7, 7, 128)	151552	conv5_block18_0_relu[0][0]
conv5_block18_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block18_1_conv[0][0]
conv5_block18_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block18_1_bn[0][0]

conv5_block18_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block18_1_relu[0][0]
conv5_block18_concat (Concatena	(None, 7, 7, 1216)	0	conv5_block17_concat[0][0] conv5_block18_2_conv[0][0]
conv5_block19_0_bn (BatchNormal	(None, 7, 7, 1216)	4864	conv5_block18_concat[0][0]
conv5_block19_0_relu (Activatio	(None, 7, 7, 1216)	0	conv5_block19_0_bn[0][0]
conv5_block19_1_conv (Conv2D)	(None, 7, 7, 128)	155648	conv5_block19_0_relu[0][0]
conv5_block19_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block19_1_conv[0][0]
conv5_block19_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block19_1_bn[0][0]
conv5_block19_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block19_1_relu[0][0]
conv5_block19_concat (Concatena	(None, 7, 7, 1248)	0	conv5_block18_concat[0][0] conv5_block19_2_conv[0][0]
conv5_block20_0_bn (BatchNormal	(None, 7, 7, 1248)	4992	conv5_block19_concat[0][0]
conv5_block20_0_relu (Activatio	(None, 7, 7, 1248)	0	conv5_block20_0_bn[0][0]
conv5_block20_1_conv (Conv2D)	(None, 7, 7, 128)	159744	conv5_block20_0_relu[0][0]
conv5_block20_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block20_1_conv[0][0]
conv5_block20_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block20_1_bn[0][0]
conv5_block20_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block20_1_relu[0][0]
conv5_block20_concat (Concatena	(None, 7, 7, 1280)	0	conv5_block19_concat[0][0] conv5_block20_2_conv[0][0]
conv5_block21_0_bn (BatchNormal	(None, 7, 7, 1280)	5120	conv5_block20_concat[0][0]
conv5_block21_0_relu (Activatio	(None, 7, 7, 1280)	0	conv5_block21_0_bn[0][0]
conv5_block21_1_conv (Conv2D)	(None, 7, 7, 128)	163840	conv5_block21_0_relu[0][0]
conv5_block21_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block21_1_conv[0][0]
conv5_block21_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block21_1_bn[0][0]
conv5_block21_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block21_1_relu[0][0]
conv5_block21_concat (Concatena	(None, 7, 7, 1312)	0	conv5_block20_concat[0][0] conv5_block21_2_conv[0][0]
conv5_block22_0_bn (BatchNormal	(None, 7, 7, 1312)	5248	conv5_block21_concat[0][0]
conv5_block22_0_relu (Activatio	(None, 7, 7, 1312)	0	conv5_block22_0_bn[0][0]
conv5_block22_1_conv (Conv2D)	(None, 7, 7, 128)	167936	conv5_block22_0_relu[0][0]
conv5_block22_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block22_1_conv[0][0]
conv5_block22_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block22_1_bn[0][0]
conv5_block22_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block22_1_relu[0][0]
conv5_block22_concat (Concatena	(None, 7, 7, 1344)	0	conv5_block21_concat[0][0] conv5_block22_2_conv[0][0]
conv5_block23_0_bn (BatchNormal	(None, 7, 7, 1344)	5376	conv5_block22_concat[0][0]
conv5_block23_0_relu (Activatio	(None, 7, 7, 1344)	0	conv5_block23_0_bn[0][0]
conv5_block23_1_conv (Conv2D)	(None, 7, 7, 128)	172032	conv5_block23_0_relu[0][0]
conv5_block23_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block23_1_conv[0][0]
conv5_block23_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block23_1_bn[0][0]
conv5_block23_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block23_1_relu[0][0]
conv5_block23_concat (Concatena	(None, 7, 7, 1376)	0	conv5_block22_concat[0][0] conv5_block23_2_conv[0][0]
conv5_block24_0_bn (BatchNormal	(None, 7, 7, 1376)	5504	conv5_block23_concat[0][0]
conv5_block24_0_relu (Activatio	(None, 7, 7, 1376)	0	conv5_block24_0_bn[0][0]
conv5_block24_1_conv (Conv2D)	(None, 7, 7, 128)	176128	conv5_block24_0_relu[0][0]
conv5_block24_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block24_1_conv[0][0]

conv5_block24_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block24_1_bn[0][0]
conv5_block24_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block24_1_relu[0][0]
conv5_block24_concat (Concatena	(None, 7, 7, 1408)	0	conv5_block23_concat[0][0] conv5_block24_2_conv[0][0]
conv5_block25_0_bn (BatchNormal	(None, 7, 7, 1408)	5632	conv5_block24_concat[0][0]
conv5_block25_0_relu (Activatio	(None, 7, 7, 1408)	0	conv5_block25_0_bn[0][0]
conv5_block25_1_conv (Conv2D)	(None, 7, 7, 128)	180224	conv5_block25_0_relu[0][0]
conv5_block25_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block25_1_conv[0][0]
conv5_block25_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block25_1_bn[0][0]
conv5_block25_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block25_1_relu[0][0]
conv5_block25_concat (Concatena	(None, 7, 7, 1440)	0	conv5_block24_concat[0][0] conv5_block25_2_conv[0][0]
conv5_block26_0_bn (BatchNormal	(None, 7, 7, 1440)	5760	conv5_block25_concat[0][0]
conv5_block26_0_relu (Activatio	(None, 7, 7, 1440)	0	conv5_block26_0_bn[0][0]
conv5_block26_1_conv (Conv2D)	(None, 7, 7, 128)	184320	conv5_block26_0_relu[0][0]
conv5_block26_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block26_1_conv[0][0]
conv5_block26_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block26_1_bn[0][0]
conv5_block26_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block26_1_relu[0][0]
conv5_block26_concat (Concatena	(None, 7, 7, 1472)	0	conv5_block25_concat[0][0] conv5_block26_2_conv[0][0]
conv5_block27_0_bn (BatchNormal	(None, 7, 7, 1472)	5888	conv5_block26_concat[0][0]
conv5_block27_0_relu (Activatio	(None, 7, 7, 1472)	0	conv5_block27_0_bn[0][0]
conv5_block27_1_conv (Conv2D)	(None, 7, 7, 128)	188416	conv5_block27_0_relu[0][0]
conv5_block27_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block27_1_conv[0][0]
conv5_block27_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block27_1_bn[0][0]
conv5_block27_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block27_1_relu[0][0]
conv5_block27_concat (Concatena	(None, 7, 7, 1504)	0	conv5_block26_concat[0][0] conv5_block27_2_conv[0][0]
conv5_block28_0_bn (BatchNormal	(None, 7, 7, 1504)	6016	conv5_block27_concat[0][0]
conv5_block28_0_relu (Activatio	(None, 7, 7, 1504)	0	conv5_block28_0_bn[0][0]
conv5_block28_1_conv (Conv2D)	(None, 7, 7, 128)	192512	conv5_block28_0_relu[0][0]
conv5_block28_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block28_1_conv[0][0]
conv5_block28_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block28_1_bn[0][0]
conv5_block28_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block28_1_relu[0][0]
conv5_block28_concat (Concatena	(None, 7, 7, 1536)	0	conv5_block27_concat[0][0] conv5_block28_2_conv[0][0]
conv5_block29_0_bn (BatchNormal	(None, 7, 7, 1536)	6144	conv5_block28_concat[0][0]
conv5_block29_0_relu (Activatio	(None, 7, 7, 1536)	0	conv5_block29_0_bn[0][0]
conv5_block29_1_conv (Conv2D)	(None, 7, 7, 128)	196608	conv5_block29_0_relu[0][0]
conv5_block29_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block29_1_conv[0][0]
conv5_block29_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block29_1_bn[0][0]
conv5_block29_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block29_1_relu[0][0]
conv5_block29_concat (Concatena	(None, 7, 7, 1568)	0	conv5_block28_concat[0][0] conv5_block29_2_conv[0][0]
conv5_block30_0_bn (BatchNormal	(None, 7, 7, 1568)	6272	conv5_block29_concat[0][0]
conv5_block30_0_relu (Activatio	(None, 7, 7, 1568)	0	conv5_block30_0_bn[0][0]
conv5_block30_1_conv (Conv2D)	(None, 7, 7, 128)	200704	conv5_block30_0_relu[0][0]

conv5_block30_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block30_1_conv[0][0]
conv5_block30_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block30_1_bn[0][0]
conv5_block30_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block30_1_relu[0][0]
conv5_block30_concat (Concatena	(None, 7, 7, 1600)	0	conv5_block29_concat[0][0] conv5_block30_2_conv[0][0]
conv5_block31_0_bn (BatchNormal	(None, 7, 7, 1600)	6400	conv5_block30_concat[0][0]
conv5_block31_0_relu (Activatio	(None, 7, 7, 1600)	0	conv5_block31_0_bn[0][0]
conv5_block31_1_conv (Conv2D)	(None, 7, 7, 128)	204800	conv5_block31_0_relu[0][0]
conv5_block31_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block31_1_conv[0][0]
conv5_block31_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block31_1_bn[0][0]
conv5_block31_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block31_1_relu[0][0]
conv5_block31_concat (Concatena	(None, 7, 7, 1632)	0	conv5_block30_concat[0][0] conv5_block31_2_conv[0][0]
conv5_block32_0_bn (BatchNormal	(None, 7, 7, 1632)	6528	conv5_block31_concat[0][0]
conv5_block32_0_relu (Activatio	(None, 7, 7, 1632)	0	conv5_block32_0_bn[0][0]
conv5_block32_1_conv (Conv2D)	(None, 7, 7, 128)	208896	conv5_block32_0_relu[0][0]
conv5_block32_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block32_1_conv[0][0]
conv5_block32_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block32_1_bn[0][0]
conv5_block32_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block32_1_relu[0][0]
conv5_block32_concat (Concatena	(None, 7, 7, 1664)	0	conv5_block31_concat[0][0] conv5_block32_2_conv[0][0]
bn (BatchNormalization)	(None, 7, 7, 1664)	6656	conv5_block32_concat[0][0]
relu (Activation)	(None, 7, 7, 1664)	0	bn[0][0]
reshape_4 (Reshape)	(None, 49, 1664)	0	relu[0][0]
lstm_4 (LSTM)	(None, 49, 1664)	22157824	reshape_4[0][0]
batch_normalization_12 (BatchNo	(None, 49, 1664)	6656	lstm_4[0][0]
flatten (Flatten)	(None, 81536)	0	batch_normalization_12[0][0]
dense_12 (Dense)	(None, 4096)	333975552	flatten[0][0]
batch_normalization_13 (BatchNo	(None, 4096)	16384	dense_12[0][0]
dense_13 (Dense)	(None, 4096)	16781312	batch_normalization_13[0][0]
batch_normalization_14 (BatchNo	(None, 4096)	16384	dense_13[0][0]
dense_14 (Dense)	(None, 4)	16388	batch_normalization_14[0][0]

Total params: 385,613,380
Trainable params: 350,792,964
Non-trainable params: 34,820,416