Introduction

Retinal optical coherence tomography (OCT) is an imaging technique used to captu
re high-resolution cross sections of the retinas of living patients. Approximately 30
million OCT scans are performed each year, and the analysis and interpretation of these ima
ges takes up a significant amount of time .Images are labeled as (disease)-(randomized
patient ID)-(image number by this patient) and split into 4 directories: CNV, DME, DRUSEN,
and NORMAL.Our task is to develop an automated system which would classify the images into
the different categories.

Importing os package

In [1]:

```
import os
print(os.listdir())
```

```
['Retina_OCT_Image_analysis.ipynb', '.ipynb_checkpoints', 'OCT2017 ']
```

In [2]:

```
from os import listdir
```

In [3]:

```
listdir('OCT2017 ')
```

Out[3]:

```
['NORMAL', 'CNV', '.ipynb_checkpoints', 'DRUSEN', 'DME']
```

Summary

*. Displaying all the list of files in the directory.

In [4]:

```
data = 'OCT2017 '
```

In [5]:

```
listdir(data)
```

Out[5]:

```
['NORMAL', 'CNV', '.ipynb_checkpoints', 'DRUSEN', 'DME']
```

Summary

* We are taking only 1000 images but of my ram constraint.

Importing all the necessary packages

In [6]:

```
import pandas as pd
import numpy as np
```

```
import os
from os import listdir
from glob import glob
import itertools
import fnmatch
import random
from PIL import Image
import zlib
import itertools
import fnmatch
import random
from PIL import Image
import zlib
import itertools
import csv
from tqdm import tqdm
import matplotlib.pylab as plt
import seaborn as sns
import cv2
import skimage
from skimage import transform
from skimage.transform import resize
import scipy
from scipy.misc import imresize, imread
from scipy import misc
import keras
from keras import backend as K
from keras import models, layers, optimizers
from keras.applications.inception_v3 import InceptionV3
from keras.applications import VGG16
from keras.applications.vgg16 import preprocess_input
from keras.models import Model, Sequential, model_from_json
from keras.layers import Dense, Dropout, Input, Activation, Flatten, BatchNormalization, Conv2D, Ma
xPool2D, MaxPooling2D, Lambda, AveragePooling2D
from keras.utils import np_utils
from keras.utils.np_utils import to_categorical
from keras.preprocessing.image import array_to_img, img_to_array, load_img, ImageDataGenerator
from keras.callbacks import Callback, EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
from keras.optimizers import SGD, RMSprop, Adam, Adagrad, Adadelta, RMSprop
import sklearn
from sklearn import model_selection
from sklearn.model_selection import train_test_split, KFold, cross_val_score, StratifiedKFold,
learning_curve, GridSearchCV
from sklearn.metrics import confusion_matrix, make_scorer, accuracy_score
from sklearn.utils import class_weight
%matplotlib inline
```

```
Using TensorFlow backend.
```

In [7]:

```
listdir(data+'/'+'NORMAL')[0:10]
```

Out[7]:

```
['NORMAL-330905-1.jpeg',
 'NORMAL-628048-1.jpeg',
 'NORMAL-3465750-1.jpeg',
 'NORMAL-101880-1.jpeg',
 'NORMAL-1998191-1.jpeg',
 'NORMAL-2182640-1.jpeg',
 'NORMAL-3077276-1.jpeg',
 'NORMAL-460711-1.jpeg',
 'NORMAL-12494-2.jpeg',
 'NORMAL-3256489-1.jpeg']
```

In [8]:

```
listdir(data+'/'+'CNV')[0:10]
```

Out[8]:

```
['CNV-1188386-1.jpeg',
 'CNV-103044-12.jpeg',
```

```
 'CNV-2158821-2.jpeg',
 'CNV-3163547-1.jpeg',
 'CNV-1997439-7.jpeg',
 'CNV-4674526-1.jpeg',
 'CNV-457907-1.jpeg',
 'CNV-163081-5.jpeg',
 'CNV-1699976-2.jpeg',
 'CNV-1699976-4.jpeg']
```

In [9]:

```
listdir(data+'/'+'DME')[0:10]
```

Out[9]:

```
['DME-4634094-1.jpeg',
 'DME-5864085-1.jpeg',
 'DME-3921035-1.jpeg',
 'DME-6314020-1.jpeg',
 'DME-57603-1.jpeg',
 'DME-7614088-1.jpeg',
 'DME-8177380-1.jpeg',
 'DME-7591008-1.jpeg',
 'DME-1274315-2.jpeg',
 'DME-7469235-1.jpeg']
```

In [10]:

```
listdir(data+'/'+'DRUSEN')[0:10]
```

Out[10]:

```
['DRUSEN-95633-1.jpeg',
 'DRUSEN-1912508-1.jpeg',
 'DRUSEN-4240777-1.jpeg',
 'DRUSEN-7393104-1.jpeg',
 'DRUSEN-9624303-1.jpeg',
 'DRUSEN-7915033-1.jpeg',
 'DRUSEN-4878077-5.jpeg',
 'DRUSEN-2541184-1.jpeg',
 'DRUSEN-3241692-1.jpeg',
 'DRUSEN-6193297-1.jpeg']
```

In [11]:

```
### source : kaggle
imageSize=256 # choosing imagesize=256
from tqdm import tqdm
def get_data(folder):
    """
    Load the data and labels from the given folder.
    """
    X = []
    y = []
    for folderName in os.listdir(folder):
        if not folderName.startswith('.'):
            #labelling each folders
            if folderName in ['NORMAL']:
                label = 0
            elif folderName in ['CNV']:
                label = 1
            elif folderName in ['DME']:
                label = 2
            elif folderName in ['DRUSEN']:
                label = 3
            else:
                label = 4
            for image_filename in tqdm(os.listdir(folder+'/'+folderName)):
                img_file = cv2.imread(folder+'/' + folderName + '/' + image_filename)
                if img_file is not None:
                    img_file = skimage.transform.resize(img_file, (imageSize, imageSize, 3))
                    img_arr = np.asarray(img_file)
                    X.append(img_arr)
                    y.append(label)
```

```
                    y.append(label)
    X = np.asarray(X)
    y = np.asarray(y)
    return X,y

X_test, y_test= get_data(data)

from sklearn.model_selection import train_test_split #train,test split
X_train, X_test, y_train, y_test = train_test_split(X_test, y_test, test_size=0.2)
```

```
  0%|          | 0/242 [00:00<?, ?it/s]/usr/local/lib/python3.5/dist-
packages/skimage/transform/_warps.py:105: UserWarning: The default mode, 'constant', will be
changed to 'reflect' in skimage 0.15.
  warn("The default mode, 'constant', will be changed to 'reflect' in "
/usr/local/lib/python3.5/dist-packages/skimage/transform/_warps.py:110: UserWarning: Anti-aliasing
will be enabled by default in skimage 0.15 to avoid aliasing artifacts when down-sampling images.
  warn("Anti-aliasing will be enabled by default in skimage 0.15 to "
100%|██████████| 242/242 [00:04<00:00, 53.79it/s]
100%|██████████| 243/243 [00:04<00:00, 49.56it/s]
100%|██████████| 242/242 [00:04<00:00, 50.20it/s]
100%|██████████| 242/242 [00:05<00:00, 47.50it/s]
```

Summary

*. Loading the data from all the folders.
*. Labelling each folder to different values 0,1,2,3 .
*. Taking this labelled values as targeted values.

In [12]:

```
X_train.shape
```

Out[12]:

```
(774, 256, 256, 3)
```

In [13]:

```
y_test[:10]
```

Out[13]:

```
array([0, 3, 2, 2, 0, 1, 1, 1, 3, 3])
```

In [14]:

```
X_train[0]
```

Out[14]:

```
array([[[0.12414216, 0.12414216, 0.12414216],
        [0.11685049, 0.11685049, 0.11685049],
        [0.11384804, 0.11384804, 0.11384804],
        ...,
        [0.98400735, 0.98400735, 0.98400735],
        [0.99375   , 0.99375   , 0.99375   ],
        [0.99344363, 0.99344363, 0.99344363]],

       [[0.09136029, 0.09136029, 0.09136029],
        [0.11691176, 0.11691176, 0.11691176],
        [0.10784314, 0.10784314, 0.10784314],
        ...,
        [0.61819853, 0.61819853, 0.61819853],
        [0.99601716, 0.99601716, 0.99601716],
        [0.98480392, 0.98480392, 0.98480392]],

       [[0.10202206, 0.10202206, 0.10202206],
        [0.13333333, 0.13333333, 0.13333333],
        [0.11378676, 0.11378676, 0.11378676],
        ...,
        [0.06452206, 0.06452206, 0.06452206],
```

```
         [0.03572304, 0.03572304, 0.03572304],
         [0.50514706, 0.50514706, 0.50514706]],

        ...,

        [[1.        , 1.        , 1.        ],
         [1.        , 1.        , 1.        ],
         [1.        , 1.        , 1.        ],
         ...,
         [0.00526961, 0.00526961, 0.00526961],
         [0.00459559, 0.00459559, 0.00459559],
         [0.50134804, 0.50134804, 0.50134804]],

        [[1.        , 1.        , 1.        ],
         [1.        , 1.        , 1.        ],
         [1.        , 1.        , 1.        ],
         ...,
         [0.003125  , 0.003125  , 0.003125  ],
         [0.00392157, 0.00392157, 0.00392157],
         [0.49920343, 0.49920343, 0.49920343]],

        [[1.        , 1.        , 1.        ],
         [1.        , 1.        , 1.        ],
         [1.        , 1.        , 1.        ],
         ...,
         [0.0060049 , 0.0060049 , 0.0060049 ],
         [0.00588235, 0.00588235, 0.00588235],
         [0.50300245, 0.50300245, 0.50300245]]])
```

Converting all the y values to categorical values

In [15]:

```
from keras.utils.np_utils import to_categorical
y_train = to_categorical(y_train, num_classes = 4)
y_test = to_categorical(y_test, num_classes = 4)
```

In [16]:

```
y_test[0]
```

Out[16]:

```
array([1., 0., 0., 0.], dtype=float32)
```

Summary

*. Converting all numeric y values to categorical values

Data Augmentation

In [21]:

```
#source :Kaggle
datagenerated = ImageDataGenerator(
        featurewise_center=False,  # set input mean to 0 over the dataset
        samplewise_center=False,  # set each sample mean to 0
        featurewise_std_normalization=False,  # divide inputs by std of the dataset
        samplewise_std_normalization=False,  # divide each input by its std
        zca_whitening=False,  # apply ZCA whitening
        rotation_range=10,  # randomly rotate images in the range (degrees, 0 to 180)
        width_shift_range=0.1,  # randomly shift images horizontally (fraction of total width)
        height_shift_range=0.1,  # randomly shift images vertically (fraction of total height)
        horizontal_flip=True,  # randomly flip images
        vertical_flip=False)  # randomly flip images
```

In [27]:

```
datagenerated
```

Out[27]:

```
<keras.preprocessing.image.ImageDataGenerator at 0x7fb77c33fef0>
```

Summary

We are using data augmentation, to generate different images with different rotation angles and flips.

Baseline model with 2 layer architecture

In [28]:

```
input_shape=(imageSize, imageSize, 3)
```

In [35]:

```
batch_size = 16
num_classes = 4
epochs = 12
```

In [36]:

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=input_shape,strides=1))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
```

In [37]:

```
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adam(),
              metrics=['accuracy'])

history = model.fit_generator(datagenerated.flow(X_train,y_train, batch_size=batch_size),
                        steps_per_epoch=len(X_train) / batch_size, epochs=epochs, validation_data =
[X_test,y_test])
score = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

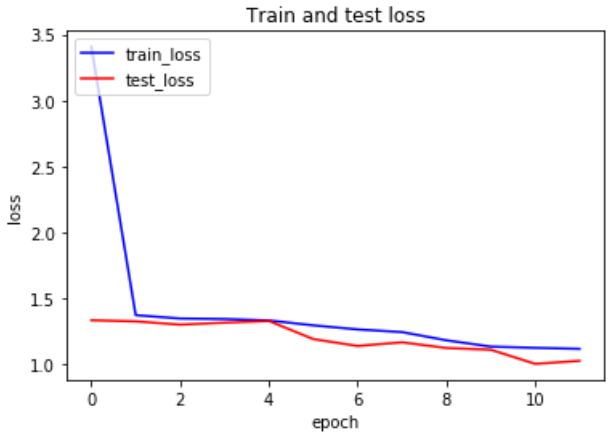
```
Epoch 1/12
49/48 [==============================] - 93s 2s/step - loss: 3.3830 - acc: 0.2772 - val_loss: 1.33
51 - val_acc: 0.2887
Epoch 2/12
49/48 [==============================] - 86s 2s/step - loss: 1.3751 - acc: 0.3474 - val_loss: 1.32
63 - val_acc: 0.3557
Epoch 3/12
49/48 [==============================] - 85s 2s/step - loss: 1.3483 - acc: 0.3295 - val_loss: 1.30
16 - val_acc: 0.3557
Epoch 4/12
49/48 [==============================] - 86s 2s/step - loss: 1.3458 - acc: 0.3580 - val_loss: 1.31
65 - val_acc: 0.3660
Epoch 5/12
49/48 [==============================] - 88s 2s/step - loss: 1.3339 - acc: 0.3555 - val_loss: 1.32
95 - val_acc: 0.3608
Epoch 6/12
49/48 [==============================] - 87s 2s/step - loss: 1.2968 - acc: 0.3691 - val_loss: 1.19
```

```
25 - val_acc: 0.4072
Epoch 7/12
49/48 [==============================] - 87s 2s/step - loss: 1.2610 - acc: 0.4026 - val_loss: 1.14
00 - val_acc: 0.4794
Epoch 8/12
49/48 [==============================] - 86s 2s/step - loss: 1.2460 - acc: 0.3976 - val_loss: 1.16
78 - val_acc: 0.4691
Epoch 9/12
49/48 [==============================] - 86s 2s/step - loss: 1.1817 - acc: 0.4472 - val_loss: 1.12
47 - val_acc: 0.4639
Epoch 10/12
49/48 [==============================] - 87s 2s/step - loss: 1.1372 - acc: 0.4911 - val_loss: 1.11
17 - val_acc: 0.4948
Epoch 11/12
49/48 [==============================] - 86s 2s/step - loss: 1.1277 - acc: 0.5000 - val_loss: 1.00
45 - val_acc: 0.5206
Epoch 12/12
49/48 [==============================] - 87s 2s/step - loss: 1.1200 - acc: 0.4737 - val_loss: 1.02
73 - val_acc: 0.5412
Test loss: 1.027250802394041
Test accuracy: 0.5412371134020618
```

Plot between train loss and test loss

In [38]:

```
plt.plot(history.history['loss'],color="blue")
plt.plot(history.history['val_loss'],color="red")
plt.title('Train and test loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train_loss', 'test_loss'], loc='upper left')
plt.show()
```



Summary

*. Training the model with only 2 layer architecture.
*. Test accuracy is only 54 %
*. From the graph, we can conclude, model is overfitting .

Baseline model with 7 layer architecture

In [39]:

```
model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape=input_shape,padding='same'))
model.add(BatchNormalization(axis=-1))
model.add(Activation('relu'))

model.add(Conv2D(32, (3, 3),padding='same'))
model.add(BatchNormalization(axis=-1))
model.add(Activation('relu'))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(64,(5, 5),padding='same'))
model.add(BatchNormalization(axis=-1))
model.add(Activation('relu'))

model.add(Conv2D(64, (3, 3),padding='same'))
model.add(BatchNormalization(axis=-1))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(64, (3, 3),padding='same'))
model.add(BatchNormalization(axis=-1))
model.add(Activation('relu'))

model.add(Conv2D(64, (3, 3),padding='same'))
model.add(BatchNormalization(axis=-1))
model.add(Activation('relu'))

model.add(Conv2D(64, (2,2),padding='same'))
model.add(BatchNormalization(axis=-1))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [40]:

```
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adam(),
              metrics=['accuracy'])

history = model.fit_generator(datagenerated.flow(X_train,y_train, batch_size=batch_size),
                        steps_per_epoch=len(X_train) / batch_size, epochs=epochs, validation_data =
[X_test,y_test])
score = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

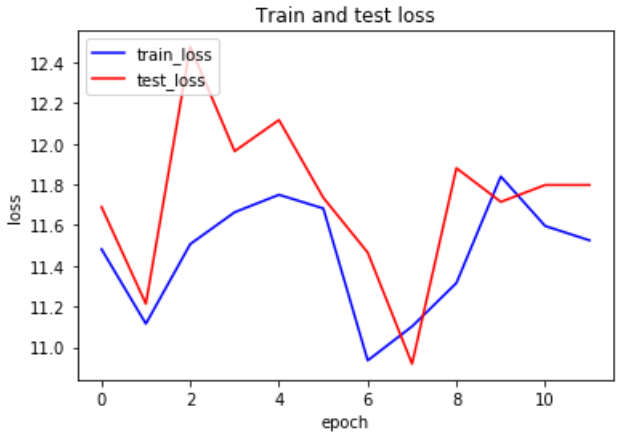
```
Epoch 1/12
49/48 [==============================] - 209s 4s/step - loss: 11.5066 - acc: 0.2458 - val_loss: 11
.6899 - val_acc: 0.2680
Epoch 2/12
49/48 [==============================] - 199s 4s/step - loss: 11.1105 - acc: 0.2861 - val_loss: 11
.2133 - val_acc: 0.2990
Epoch 3/12
49/48 [==============================] - 204s 4s/step - loss: 11.5646 - acc: 0.2615 - val_loss: 12
.4778 - val_acc: 0.2216
Epoch 4/12
49/48 [==============================] - 199s 4s/step - loss: 11.6186 - acc: 0.2640 - val_loss: 11
.9639 - val_acc: 0.2577
Epoch 5/12
49/48 [==============================] - 201s 4s/step - loss: 11.7090 - acc: 0.2593 - val_loss: 12
.1175 - val_acc: 0.2474
Epoch 6/12
49/48 [==============================] - 201s 4s/step - loss: 11.6707 - acc: 0.2593 - val_loss: 11
.7350 - val_acc: 0.2629
Epoch 7/12
49/48 [==============================] - 198s 4s/step - loss: 10.8986 - acc: 0.3061 - val_loss: 11
.4654 - val_acc: 0.2887
Epoch 8/12
49/48 [==============================] - 197s 4s/step - loss: 11.1649 - acc: 0.3024 - val_loss: 10
.9181 - val_acc: 0.3144
Epoch 9/12
49/48 [==============================] - 192s 4s/step - loss: 11.2752 - acc: 0.2920 - val_loss: 11
.8809 - val_acc: 0.2629
Epoch 10/12
49/48 [==============================] - 195s 4s/step - loss: 11.8261 - acc: 0.2619 - val_loss: 11
.7147 - val_acc: 0.2732
Epoch 11/12
49/48 [==============================] - 204s 4s/step - loss: 11.6191 - acc: 0.2777 - val_loss: 11
.7978 - val_acc: 0.2680
```

```
Epoch 12/12
49/48 [==============================] - 195s 4s/step - loss: 11.5156 - acc: 0.2823 - val_loss: 11
.7978 - val_acc: 0.2680
Test loss: 11.797781090146488
Test accuracy: 0.26804123711340205
```

Plot between train loss and test loss

In [41]:

```
plt.plot(history.history['loss'],color="blue")
plt.plot(history.history['val_loss'],color="red")
plt.title('Train and test loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train_loss', 'test_loss'], loc='upper left')
plt.show()
```



Summary

*. Training the model with only 7 layer architecture.
*. Test accuracy is only 26 %
*. The model is performing worse with 7 layer architecture .

Using transfer learning methods

Using Inception v3

In [42]:

```
from keras.applications import inception_v3
pretrained_model_incp = InceptionV3(include_top=False, input_shape=(imageSize, imageSize, 3))
```

In [43]:

```
base_model = pretrained_model_incp
opt = keras.optimizers.Adam()
x = base_model.output
x = Conv2D(256, kernel_size = (3,3), padding = 'valid')(x)
x = Flatten()(x)
x = Dropout(0.75)(x)
predictions = Dense(4, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)
# Train top layer
for layer in base_model.layers:
    layer.trainable = False
model.compile(loss='categorical_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
model.summary()
```

```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
input_1 (InputLayer)             (None, 256, 256, 3)   0
____________________________________________________________________________________________________
conv2d_16 (Conv2D)               (None, 127, 127, 32)  864         input_1[0][0]
____________________________________________________________________________________________________
batch_normalization_8 (BatchNor (None, 127, 127, 32)  96          conv2d_16[0][0]
____________________________________________________________________________________________________
activation_8 (Activation)        (None, 127, 127, 32)  0           batch_normalization_8[0][0]
____________________________________________________________________________________________________
conv2d_17 (Conv2D)               (None, 125, 125, 32)  9216        activation_8[0][0]
____________________________________________________________________________________________________
batch_normalization_9 (BatchNor (None, 125, 125, 32)  96          conv2d_17[0][0]
____________________________________________________________________________________________________
activation_9 (Activation)        (None, 125, 125, 32)  0           batch_normalization_9[0][0]
____________________________________________________________________________________________________
conv2d_18 (Conv2D)               (None, 125, 125, 64)  18432       activation_9[0][0]
____________________________________________________________________________________________________
batch_normalization_10 (BatchNo (None, 125, 125, 64)  192         conv2d_18[0][0]
____________________________________________________________________________________________________
activation_10 (Activation)       (None, 125, 125, 64)  0           batch_normalization_10[0][0]
____________________________________________________________________________________________________
max_pooling2d_8 (MaxPooling2D)   (None, 62, 62, 64)    0           activation_10[0][0]
____________________________________________________________________________________________________
conv2d_19 (Conv2D)               (None, 62, 62, 80)    5120        max_pooling2d_8[0][0]
____________________________________________________________________________________________________
batch_normalization_11 (BatchNo (None, 62, 62, 80)    240         conv2d_19[0][0]
____________________________________________________________________________________________________
activation_11 (Activation)       (None, 62, 62, 80)    0           batch_normalization_11[0][0]
____________________________________________________________________________________________________
conv2d_20 (Conv2D)               (None, 60, 60, 192)   138240      activation_11[0][0]
____________________________________________________________________________________________________
batch_normalization_12 (BatchNo (None, 60, 60, 192)   576         conv2d_20[0][0]
____________________________________________________________________________________________________
activation_12 (Activation)       (None, 60, 60, 192)   0           batch_normalization_12[0][0]
____________________________________________________________________________________________________
max_pooling2d_9 (MaxPooling2D)   (None, 29, 29, 192)   0           activation_12[0][0]
____________________________________________________________________________________________________
conv2d_24 (Conv2D)               (None, 29, 29, 64)    12288       max_pooling2d_9[0][0]
____________________________________________________________________________________________________
batch_normalization_16 (BatchNo (None, 29, 29, 64)    192         conv2d_24[0][0]
____________________________________________________________________________________________________
activation_16 (Activation)       (None, 29, 29, 64)    0           batch_normalization_16[0][0]
____________________________________________________________________________________________________
conv2d_22 (Conv2D)               (None, 29, 29, 48)    9216        max_pooling2d_9[0][0]
____________________________________________________________________________________________________
conv2d_25 (Conv2D)               (None, 29, 29, 96)    55296       activation_16[0][0]
____________________________________________________________________________________________________
batch_normalization_14 (BatchNo (None, 29, 29, 48)    144         conv2d_22[0][0]
____________________________________________________________________________________________________
batch_normalization_17 (BatchNo (None, 29, 29, 96)    288         conv2d_25[0][0]
____________________________________________________________________________________________________
activation_14 (Activation)       (None, 29, 29, 48)    0           batch_normalization_14[0][0]
____________________________________________________________________________________________________
activation_17 (Activation)       (None, 29, 29, 96)    0           batch_normalization_17[0][0]
____________________________________________________________________________________________________
average_pooling2d_1 (AveragePoo (None, 29, 29, 192)   0           max_pooling2d_9[0][0]
____________________________________________________________________________________________________
conv2d_21 (Conv2D)               (None, 29, 29, 64)    12288       max_pooling2d_9[0][0]
____________________________________________________________________________________________________
conv2d_23 (Conv2D)               (None, 29, 29, 64)    76800       activation_14[0][0]
____________________________________________________________________________________________________
conv2d_26 (Conv2D)               (None, 29, 29, 96)    82944       activation_17[0][0]
____________________________________________________________________________________________________
conv2d_27 (Conv2D)               (None, 29, 29, 32)    6144        average_pooling2d_1[0][0]
____________________________________________________________________________________________________
batch_normalization_13 (BatchNo (None, 29, 29, 64)    192         conv2d_21[0][0]
____________________________________________________________________________________________________
batch_normalization_15 (BatchNo (None, 29, 29, 64)    192         conv2d_23[0][0]
____________________________________________________________________________________________________
batch_normalization_18 (BatchNo (None, 29, 29, 96)    288         conv2d_26[0][0]
____________________________________________________________________________________________________
batch_normalization_19 (BatchNo (None, 29, 29, 32)    96          conv2d_27[0][0]
____________________________________________________________________________________________________
activation_13 (Activation)       (None, 29, 29, 64)    0           batch_normalization_13[0][0]
```

activation_15 (Activation)	(None, 29, 29, 64)	0	batch_normalization_15[0][0]
activation_18 (Activation)	(None, 29, 29, 96)	0	batch_normalization_18[0][0]
activation_19 (Activation)	(None, 29, 29, 32)	0	batch_normalization_19[0][0]
mixed0 (Concatenate)	(None, 29, 29, 256)	0	activation_13[0][0] activation_15[0][0] activation_18[0][0] activation_19[0][0]
conv2d_31 (Conv2D)	(None, 29, 29, 64)	16384	mixed0[0][0]
batch_normalization_23 (BatchNo	(None, 29, 29, 64)	192	conv2d_31[0][0]
activation_23 (Activation)	(None, 29, 29, 64)	0	batch_normalization_23[0][0]
conv2d_29 (Conv2D)	(None, 29, 29, 48)	12288	mixed0[0][0]
conv2d_32 (Conv2D)	(None, 29, 29, 96)	55296	activation_23[0][0]
batch_normalization_21 (BatchNo	(None, 29, 29, 48)	144	conv2d_29[0][0]
batch_normalization_24 (BatchNo	(None, 29, 29, 96)	288	conv2d_32[0][0]
activation_21 (Activation)	(None, 29, 29, 48)	0	batch_normalization_21[0][0]
activation_24 (Activation)	(None, 29, 29, 96)	0	batch_normalization_24[0][0]
average_pooling2d_2 (AveragePoo	(None, 29, 29, 256)	0	mixed0[0][0]
conv2d_28 (Conv2D)	(None, 29, 29, 64)	16384	mixed0[0][0]
conv2d_30 (Conv2D)	(None, 29, 29, 64)	76800	activation_21[0][0]
conv2d_33 (Conv2D)	(None, 29, 29, 96)	82944	activation_24[0][0]
conv2d_34 (Conv2D)	(None, 29, 29, 64)	16384	average_pooling2d_2[0][0]
batch_normalization_20 (BatchNo	(None, 29, 29, 64)	192	conv2d_28[0][0]
batch_normalization_22 (BatchNo	(None, 29, 29, 64)	192	conv2d_30[0][0]
batch_normalization_25 (BatchNo	(None, 29, 29, 96)	288	conv2d_33[0][0]
batch_normalization_26 (BatchNo	(None, 29, 29, 64)	192	conv2d_34[0][0]
activation_20 (Activation)	(None, 29, 29, 64)	0	batch_normalization_20[0][0]
activation_22 (Activation)	(None, 29, 29, 64)	0	batch_normalization_22[0][0]
activation_25 (Activation)	(None, 29, 29, 96)	0	batch_normalization_25[0][0]
activation_26 (Activation)	(None, 29, 29, 64)	0	batch_normalization_26[0][0]
mixed1 (Concatenate)	(None, 29, 29, 288)	0	activation_20[0][0] activation_22[0][0] activation_25[0][0] activation_26[0][0]
conv2d_38 (Conv2D)	(None, 29, 29, 64)	18432	mixed1[0][0]
batch_normalization_30 (BatchNo	(None, 29, 29, 64)	192	conv2d_38[0][0]
activation_30 (Activation)	(None, 29, 29, 64)	0	batch_normalization_30[0][0]
conv2d_36 (Conv2D)	(None, 29, 29, 48)	13824	mixed1[0][0]
conv2d_39 (Conv2D)	(None, 29, 29, 96)	55296	activation_30[0][0]
batch_normalization_28 (BatchNo	(None, 29, 29, 48)	144	conv2d_36[0][0]
batch_normalization_31 (BatchNo	(None, 29, 29, 96)	288	conv2d_39[0][0]
activation_28 (Activation)	(None, 29, 29, 48)	0	batch_normalization_28[0][0]

activation_31 (Activation)	(None, 29, 29, 96)	0	batch_normalization_31[0][0]
average_pooling2d_3 (AveragePoo	(None, 29, 29, 288)	0	mixed1[0][0]
conv2d_35 (Conv2D)	(None, 29, 29, 64)	18432	mixed1[0][0]
conv2d_37 (Conv2D)	(None, 29, 29, 64)	76800	activation_28[0][0]
conv2d_40 (Conv2D)	(None, 29, 29, 96)	82944	activation_31[0][0]
conv2d_41 (Conv2D)	(None, 29, 29, 64)	18432	average_pooling2d_3[0][0]
batch_normalization_27 (BatchNo	(None, 29, 29, 64)	192	conv2d_35[0][0]
batch_normalization_29 (BatchNo	(None, 29, 29, 64)	192	conv2d_37[0][0]
batch_normalization_32 (BatchNo	(None, 29, 29, 96)	288	conv2d_40[0][0]
batch_normalization_33 (BatchNo	(None, 29, 29, 64)	192	conv2d_41[0][0]
activation_27 (Activation)	(None, 29, 29, 64)	0	batch_normalization_27[0][0]
activation_29 (Activation)	(None, 29, 29, 64)	0	batch_normalization_29[0][0]
activation_32 (Activation)	(None, 29, 29, 96)	0	batch_normalization_32[0][0]
activation_33 (Activation)	(None, 29, 29, 64)	0	batch_normalization_33[0][0]
mixed2 (Concatenate)	(None, 29, 29, 288)	0	activation_27[0][0] activation_29[0][0] activation_32[0][0] activation_33[0][0]
conv2d_43 (Conv2D)	(None, 29, 29, 64)	18432	mixed2[0][0]
batch_normalization_35 (BatchNo	(None, 29, 29, 64)	192	conv2d_43[0][0]
activation_35 (Activation)	(None, 29, 29, 64)	0	batch_normalization_35[0][0]
conv2d_44 (Conv2D)	(None, 29, 29, 96)	55296	activation_35[0][0]
batch_normalization_36 (BatchNo	(None, 29, 29, 96)	288	conv2d_44[0][0]
activation_36 (Activation)	(None, 29, 29, 96)	0	batch_normalization_36[0][0]
conv2d_42 (Conv2D)	(None, 14, 14, 384)	995328	mixed2[0][0]
conv2d_45 (Conv2D)	(None, 14, 14, 96)	82944	activation_36[0][0]
batch_normalization_34 (BatchNo	(None, 14, 14, 384)	1152	conv2d_42[0][0]
batch_normalization_37 (BatchNo	(None, 14, 14, 96)	288	conv2d_45[0][0]
activation_34 (Activation)	(None, 14, 14, 384)	0	batch_normalization_34[0][0]
activation_37 (Activation)	(None, 14, 14, 96)	0	batch_normalization_37[0][0]
max_pooling2d_10 (MaxPooling2D)	(None, 14, 14, 288)	0	mixed2[0][0]
mixed3 (Concatenate)	(None, 14, 14, 768)	0	activation_34[0][0] activation_37[0][0] max_pooling2d_10[0][0]
conv2d_50 (Conv2D)	(None, 14, 14, 128)	98304	mixed3[0][0]
batch_normalization_42 (BatchNo	(None, 14, 14, 128)	384	conv2d_50[0][0]
activation_42 (Activation)	(None, 14, 14, 128)	0	batch_normalization_42[0][0]
conv2d_51 (Conv2D)	(None, 14, 14, 128)	114688	activation_42[0][0]
batch_normalization_43 (BatchNo	(None, 14, 14, 128)	384	conv2d_51[0][0]
activation_43 (Activation)	(None, 14, 14, 128)	0	batch_normalization_43[0][0]
conv2d_47 (Conv2D)	(None, 14, 14, 128)	98304	mixed3[0][0]

conv2d_52 (Conv2D)	(None, 14, 14, 128)	114688	activation_43[0][0]
batch_normalization_39 (BatchNo	(None, 14, 14, 128)	384	conv2d_47[0][0]
batch_normalization_44 (BatchNo	(None, 14, 14, 128)	384	conv2d_52[0][0]
activation_39 (Activation)	(None, 14, 14, 128)	0	batch_normalization_39[0][0]
activation_44 (Activation)	(None, 14, 14, 128)	0	batch_normalization_44[0][0]
conv2d_48 (Conv2D)	(None, 14, 14, 128)	114688	activation_39[0][0]
conv2d_53 (Conv2D)	(None, 14, 14, 128)	114688	activation_44[0][0]
batch_normalization_40 (BatchNo	(None, 14, 14, 128)	384	conv2d_48[0][0]
batch_normalization_45 (BatchNo	(None, 14, 14, 128)	384	conv2d_53[0][0]
activation_40 (Activation)	(None, 14, 14, 128)	0	batch_normalization_40[0][0]
activation_45 (Activation)	(None, 14, 14, 128)	0	batch_normalization_45[0][0]
average_pooling2d_4 (AveragePoo	(None, 14, 14, 768)	0	mixed3[0][0]
conv2d_46 (Conv2D)	(None, 14, 14, 192)	147456	mixed3[0][0]
conv2d_49 (Conv2D)	(None, 14, 14, 192)	172032	activation_40[0][0]
conv2d_54 (Conv2D)	(None, 14, 14, 192)	172032	activation_45[0][0]
conv2d_55 (Conv2D)	(None, 14, 14, 192)	147456	average_pooling2d_4[0][0]
batch_normalization_38 (BatchNo	(None, 14, 14, 192)	576	conv2d_46[0][0]
batch_normalization_41 (BatchNo	(None, 14, 14, 192)	576	conv2d_49[0][0]
batch_normalization_46 (BatchNo	(None, 14, 14, 192)	576	conv2d_54[0][0]
batch_normalization_47 (BatchNo	(None, 14, 14, 192)	576	conv2d_55[0][0]
activation_38 (Activation)	(None, 14, 14, 192)	0	batch_normalization_38[0][0]
activation_41 (Activation)	(None, 14, 14, 192)	0	batch_normalization_41[0][0]
activation_46 (Activation)	(None, 14, 14, 192)	0	batch_normalization_46[0][0]
activation_47 (Activation)	(None, 14, 14, 192)	0	batch_normalization_47[0][0]
mixed4 (Concatenate)	(None, 14, 14, 768)	0	activation_38[0][0] activation_41[0][0] activation_46[0][0] activation_47[0][0]
conv2d_60 (Conv2D)	(None, 14, 14, 160)	122880	mixed4[0][0]
batch_normalization_52 (BatchNo	(None, 14, 14, 160)	480	conv2d_60[0][0]
activation_52 (Activation)	(None, 14, 14, 160)	0	batch_normalization_52[0][0]
conv2d_61 (Conv2D)	(None, 14, 14, 160)	179200	activation_52[0][0]
batch_normalization_53 (BatchNo	(None, 14, 14, 160)	480	conv2d_61[0][0]
activation_53 (Activation)	(None, 14, 14, 160)	0	batch_normalization_53[0][0]
conv2d_57 (Conv2D)	(None, 14, 14, 160)	122880	mixed4[0][0]
conv2d_62 (Conv2D)	(None, 14, 14, 160)	179200	activation_53[0][0]
batch_normalization_49 (BatchNo	(None, 14, 14, 160)	480	conv2d_57[0][0]
batch_normalization_54 (BatchNo	(None, 14, 14, 160)	480	conv2d_62[0][0]
activation_49 (Activation)	(None, 14, 14, 160)	0	batch_normalization_49[0][0]
activation_54 (Activation)	(None, 14, 14, 160)	0	batch_normalization_54[0][0]

conv2d_58 (Conv2D)	(None, 14, 14, 160)	179200	activation_49[0][0]
conv2d_63 (Conv2D)	(None, 14, 14, 160)	179200	activation_54[0][0]
batch_normalization_50 (BatchNo	(None, 14, 14, 160)	480	conv2d_58[0][0]
batch_normalization_55 (BatchNo	(None, 14, 14, 160)	480	conv2d_63[0][0]
activation_50 (Activation)	(None, 14, 14, 160)	0	batch_normalization_50[0][0]
activation_55 (Activation)	(None, 14, 14, 160)	0	batch_normalization_55[0][0]
average_pooling2d_5 (AveragePoo	(None, 14, 14, 768)	0	mixed4[0][0]
conv2d_56 (Conv2D)	(None, 14, 14, 192)	147456	mixed4[0][0]
conv2d_59 (Conv2D)	(None, 14, 14, 192)	215040	activation_50[0][0]
conv2d_64 (Conv2D)	(None, 14, 14, 192)	215040	activation_55[0][0]
conv2d_65 (Conv2D)	(None, 14, 14, 192)	147456	average_pooling2d_5[0][0]
batch_normalization_48 (BatchNo	(None, 14, 14, 192)	576	conv2d_56[0][0]
batch_normalization_51 (BatchNo	(None, 14, 14, 192)	576	conv2d_59[0][0]
batch_normalization_56 (BatchNo	(None, 14, 14, 192)	576	conv2d_64[0][0]
batch_normalization_57 (BatchNo	(None, 14, 14, 192)	576	conv2d_65[0][0]
activation_48 (Activation)	(None, 14, 14, 192)	0	batch_normalization_48[0][0]
activation_51 (Activation)	(None, 14, 14, 192)	0	batch_normalization_51[0][0]
activation_56 (Activation)	(None, 14, 14, 192)	0	batch_normalization_56[0][0]
activation_57 (Activation)	(None, 14, 14, 192)	0	batch_normalization_57[0][0]
mixed5 (Concatenate)	(None, 14, 14, 768)	0	activation_48[0][0] activation_51[0][0] activation_56[0][0] activation_57[0][0]
conv2d_70 (Conv2D)	(None, 14, 14, 160)	122880	mixed5[0][0]
batch_normalization_62 (BatchNo	(None, 14, 14, 160)	480	conv2d_70[0][0]
activation_62 (Activation)	(None, 14, 14, 160)	0	batch_normalization_62[0][0]
conv2d_71 (Conv2D)	(None, 14, 14, 160)	179200	activation_62[0][0]
batch_normalization_63 (BatchNo	(None, 14, 14, 160)	480	conv2d_71[0][0]
activation_63 (Activation)	(None, 14, 14, 160)	0	batch_normalization_63[0][0]
conv2d_67 (Conv2D)	(None, 14, 14, 160)	122880	mixed5[0][0]
conv2d_72 (Conv2D)	(None, 14, 14, 160)	179200	activation_63[0][0]
batch_normalization_59 (BatchNo	(None, 14, 14, 160)	480	conv2d_67[0][0]
batch_normalization_64 (BatchNo	(None, 14, 14, 160)	480	conv2d_72[0][0]
activation_59 (Activation)	(None, 14, 14, 160)	0	batch_normalization_59[0][0]
activation_64 (Activation)	(None, 14, 14, 160)	0	batch_normalization_64[0][0]
conv2d_68 (Conv2D)	(None, 14, 14, 160)	179200	activation_59[0][0]
conv2d_73 (Conv2D)	(None, 14, 14, 160)	179200	activation_64[0][0]
batch_normalization_60 (BatchNo	(None, 14, 14, 160)	480	conv2d_68[0][0]
batch_normalization_65 (BatchNo	(None, 14, 14, 160)	480	conv2d_73[0][0]
activation_60 (Activation)	(None, 14, 14, 160)	0	batch_normalization_60[0][0]

activation_65 (Activation)	(None, 14, 14, 160)	0	batch_normalization_65[0][0]
average_pooling2d_6 (AveragePoo	(None, 14, 14, 768)	0	mixed5[0][0]
conv2d_66 (Conv2D)	(None, 14, 14, 192)	147456	mixed5[0][0]
conv2d_69 (Conv2D)	(None, 14, 14, 192)	215040	activation_60[0][0]
conv2d_74 (Conv2D)	(None, 14, 14, 192)	215040	activation_65[0][0]
conv2d_75 (Conv2D)	(None, 14, 14, 192)	147456	average_pooling2d_6[0][0]
batch_normalization_58 (BatchNo	(None, 14, 14, 192)	576	conv2d_66[0][0]
batch_normalization_61 (BatchNo	(None, 14, 14, 192)	576	conv2d_69[0][0]
batch_normalization_66 (BatchNo	(None, 14, 14, 192)	576	conv2d_74[0][0]
batch_normalization_67 (BatchNo	(None, 14, 14, 192)	576	conv2d_75[0][0]
activation_58 (Activation)	(None, 14, 14, 192)	0	batch_normalization_58[0][0]
activation_61 (Activation)	(None, 14, 14, 192)	0	batch_normalization_61[0][0]
activation_66 (Activation)	(None, 14, 14, 192)	0	batch_normalization_66[0][0]
activation_67 (Activation)	(None, 14, 14, 192)	0	batch_normalization_67[0][0]
mixed6 (Concatenate)	(None, 14, 14, 768)	0	activation_58[0][0] activation_61[0][0] activation_66[0][0] activation_67[0][0]
conv2d_80 (Conv2D)	(None, 14, 14, 192)	147456	mixed6[0][0]
batch_normalization_72 (BatchNo	(None, 14, 14, 192)	576	conv2d_80[0][0]
activation_72 (Activation)	(None, 14, 14, 192)	0	batch_normalization_72[0][0]
conv2d_81 (Conv2D)	(None, 14, 14, 192)	258048	activation_72[0][0]
batch_normalization_73 (BatchNo	(None, 14, 14, 192)	576	conv2d_81[0][0]
activation_73 (Activation)	(None, 14, 14, 192)	0	batch_normalization_73[0][0]
conv2d_77 (Conv2D)	(None, 14, 14, 192)	147456	mixed6[0][0]
conv2d_82 (Conv2D)	(None, 14, 14, 192)	258048	activation_73[0][0]
batch_normalization_69 (BatchNo	(None, 14, 14, 192)	576	conv2d_77[0][0]
batch_normalization_74 (BatchNo	(None, 14, 14, 192)	576	conv2d_82[0][0]
activation_69 (Activation)	(None, 14, 14, 192)	0	batch_normalization_69[0][0]
activation_74 (Activation)	(None, 14, 14, 192)	0	batch_normalization_74[0][0]
conv2d_78 (Conv2D)	(None, 14, 14, 192)	258048	activation_69[0][0]
conv2d_83 (Conv2D)	(None, 14, 14, 192)	258048	activation_74[0][0]
batch_normalization_70 (BatchNo	(None, 14, 14, 192)	576	conv2d_78[0][0]
batch_normalization_75 (BatchNo	(None, 14, 14, 192)	576	conv2d_83[0][0]
activation_70 (Activation)	(None, 14, 14, 192)	0	batch_normalization_70[0][0]
activation_75 (Activation)	(None, 14, 14, 192)	0	batch_normalization_75[0][0]
average_pooling2d_7 (AveragePoo	(None, 14, 14, 768)	0	mixed6[0][0]
conv2d_76 (Conv2D)	(None, 14, 14, 192)	147456	mixed6[0][0]
conv2d_79 (Conv2D)	(None, 14, 14, 192)	258048	activation_70[0][0]
conv2d_84 (Conv2D)	(None, 14, 14, 192)	258048	activation_75[0][0]

conv2d_85 (Conv2D)	(None, 14, 14, 192)	147456	average_pooling2d_7[0][0]
batch_normalization_68 (BatchNo	(None, 14, 14, 192)	576	conv2d_76[0][0]
batch_normalization_71 (BatchNo	(None, 14, 14, 192)	576	conv2d_79[0][0]
batch_normalization_76 (BatchNo	(None, 14, 14, 192)	576	conv2d_84[0][0]
batch_normalization_77 (BatchNo	(None, 14, 14, 192)	576	conv2d_85[0][0]
activation_68 (Activation)	(None, 14, 14, 192)	0	batch_normalization_68[0][0]
activation_71 (Activation)	(None, 14, 14, 192)	0	batch_normalization_71[0][0]
activation_76 (Activation)	(None, 14, 14, 192)	0	batch_normalization_76[0][0]
activation_77 (Activation)	(None, 14, 14, 192)	0	batch_normalization_77[0][0]
mixed7 (Concatenate)	(None, 14, 14, 768)	0	activation_68[0][0] activation_71[0][0] activation_76[0][0] activation_77[0][0]
conv2d_88 (Conv2D)	(None, 14, 14, 192)	147456	mixed7[0][0]
batch_normalization_80 (BatchNo	(None, 14, 14, 192)	576	conv2d_88[0][0]
activation_80 (Activation)	(None, 14, 14, 192)	0	batch_normalization_80[0][0]
conv2d_89 (Conv2D)	(None, 14, 14, 192)	258048	activation_80[0][0]
batch_normalization_81 (BatchNo	(None, 14, 14, 192)	576	conv2d_89[0][0]
activation_81 (Activation)	(None, 14, 14, 192)	0	batch_normalization_81[0][0]
conv2d_86 (Conv2D)	(None, 14, 14, 192)	147456	mixed7[0][0]
conv2d_90 (Conv2D)	(None, 14, 14, 192)	258048	activation_81[0][0]
batch_normalization_78 (BatchNo	(None, 14, 14, 192)	576	conv2d_86[0][0]
batch_normalization_82 (BatchNo	(None, 14, 14, 192)	576	conv2d_90[0][0]
activation_78 (Activation)	(None, 14, 14, 192)	0	batch_normalization_78[0][0]
activation_82 (Activation)	(None, 14, 14, 192)	0	batch_normalization_82[0][0]
conv2d_87 (Conv2D)	(None, 6, 6, 320)	552960	activation_78[0][0]
conv2d_91 (Conv2D)	(None, 6, 6, 192)	331776	activation_82[0][0]
batch_normalization_79 (BatchNo	(None, 6, 6, 320)	960	conv2d_87[0][0]
batch_normalization_83 (BatchNo	(None, 6, 6, 192)	576	conv2d_91[0][0]
activation_79 (Activation)	(None, 6, 6, 320)	0	batch_normalization_79[0][0]
activation_83 (Activation)	(None, 6, 6, 192)	0	batch_normalization_83[0][0]
max_pooling2d_11 (MaxPooling2D)	(None, 6, 6, 768)	0	mixed7[0][0]
mixed8 (Concatenate)	(None, 6, 6, 1280)	0	activation_79[0][0] activation_83[0][0] max_pooling2d_11[0][0]
conv2d_96 (Conv2D)	(None, 6, 6, 448)	573440	mixed8[0][0]
batch_normalization_88 (BatchNo	(None, 6, 6, 448)	1344	conv2d_96[0][0]
activation_88 (Activation)	(None, 6, 6, 448)	0	batch_normalization_88[0][0]
conv2d_93 (Conv2D)	(None, 6, 6, 384)	491520	mixed8[0][0]
conv2d_97 (Conv2D)	(None, 6, 6, 384)	1548288	activation_88[0][0]
batch_normalization_85 (BatchNo	(None, 6, 6, 384)	1152	conv2d_93[0][0]

batch_normalization_89 (BatchNo	(None, 6, 6, 384)	1152	conv2d_97[0][0]
activation_85 (Activation)	(None, 6, 6, 384)	0	batch_normalization_85[0][0]
activation_89 (Activation)	(None, 6, 6, 384)	0	batch_normalization_89[0][0]
conv2d_94 (Conv2D)	(None, 6, 6, 384)	442368	activation_85[0][0]
conv2d_95 (Conv2D)	(None, 6, 6, 384)	442368	activation_85[0][0]
conv2d_98 (Conv2D)	(None, 6, 6, 384)	442368	activation_89[0][0]
conv2d_99 (Conv2D)	(None, 6, 6, 384)	442368	activation_89[0][0]
average_pooling2d_8 (AveragePoo	(None, 6, 6, 1280)	0	mixed8[0][0]
conv2d_92 (Conv2D)	(None, 6, 6, 320)	409600	mixed8[0][0]
batch_normalization_86 (BatchNo	(None, 6, 6, 384)	1152	conv2d_94[0][0]
batch_normalization_87 (BatchNo	(None, 6, 6, 384)	1152	conv2d_95[0][0]
batch_normalization_90 (BatchNo	(None, 6, 6, 384)	1152	conv2d_98[0][0]
batch_normalization_91 (BatchNo	(None, 6, 6, 384)	1152	conv2d_99[0][0]
conv2d_100 (Conv2D)	(None, 6, 6, 192)	245760	average_pooling2d_8[0][0]
batch_normalization_84 (BatchNo	(None, 6, 6, 320)	960	conv2d_92[0][0]
activation_86 (Activation)	(None, 6, 6, 384)	0	batch_normalization_86[0][0]
activation_87 (Activation)	(None, 6, 6, 384)	0	batch_normalization_87[0][0]
activation_90 (Activation)	(None, 6, 6, 384)	0	batch_normalization_90[0][0]
activation_91 (Activation)	(None, 6, 6, 384)	0	batch_normalization_91[0][0]
batch_normalization_92 (BatchNo	(None, 6, 6, 192)	576	conv2d_100[0][0]
activation_84 (Activation)	(None, 6, 6, 320)	0	batch_normalization_84[0][0]
mixed9_0 (Concatenate)	(None, 6, 6, 768)	0	activation_86[0][0] activation_87[0][0]
concatenate_1 (Concatenate)	(None, 6, 6, 768)	0	activation_90[0][0] activation_91[0][0]
activation_92 (Activation)	(None, 6, 6, 192)	0	batch_normalization_92[0][0]
mixed9 (Concatenate)	(None, 6, 6, 2048)	0	activation_84[0][0] mixed9_0[0][0] concatenate_1[0][0] activation_92[0][0]
conv2d_105 (Conv2D)	(None, 6, 6, 448)	917504	mixed9[0][0]
batch_normalization_97 (BatchNo	(None, 6, 6, 448)	1344	conv2d_105[0][0]
activation_97 (Activation)	(None, 6, 6, 448)	0	batch_normalization_97[0][0]
conv2d_102 (Conv2D)	(None, 6, 6, 384)	786432	mixed9[0][0]
conv2d_106 (Conv2D)	(None, 6, 6, 384)	1548288	activation_97[0][0]
batch_normalization_94 (BatchNo	(None, 6, 6, 384)	1152	conv2d_102[0][0]
batch_normalization_98 (BatchNo	(None, 6, 6, 384)	1152	conv2d_106[0][0]
activation_94 (Activation)	(None, 6, 6, 384)	0	batch_normalization_94[0][0]
activation_98 (Activation)	(None, 6, 6, 384)	0	batch_normalization_98[0][0]
conv2d_103 (Conv2D)	(None, 6, 6, 384)	442368	activation_94[0][0]
conv2d_104 (Conv2D)	(None, 6, 6, 384)	442368	activation_94[0][0]

conv2d_107 (Conv2D)	(None, 6, 6, 384)	442368	activation_98[0][0]
conv2d_108 (Conv2D)	(None, 6, 6, 384)	442368	activation_98[0][0]
average_pooling2d_9 (AveragePoo	(None, 6, 6, 2048)	0	mixed9[0][0]
conv2d_101 (Conv2D)	(None, 6, 6, 320)	655360	mixed9[0][0]
batch_normalization_95 (BatchNo	(None, 6, 6, 384)	1152	conv2d_103[0][0]
batch_normalization_96 (BatchNo	(None, 6, 6, 384)	1152	conv2d_104[0][0]
batch_normalization_99 (BatchNo	(None, 6, 6, 384)	1152	conv2d_107[0][0]
batch_normalization_100 (BatchN	(None, 6, 6, 384)	1152	conv2d_108[0][0]
conv2d_109 (Conv2D)	(None, 6, 6, 192)	393216	average_pooling2d_9[0][0]
batch_normalization_93 (BatchNo	(None, 6, 6, 320)	960	conv2d_101[0][0]
activation_95 (Activation)	(None, 6, 6, 384)	0	batch_normalization_95[0][0]
activation_96 (Activation)	(None, 6, 6, 384)	0	batch_normalization_96[0][0]
activation_99 (Activation)	(None, 6, 6, 384)	0	batch_normalization_99[0][0]
activation_100 (Activation)	(None, 6, 6, 384)	0	batch_normalization_100[0][0]
batch_normalization_101 (BatchN	(None, 6, 6, 192)	576	conv2d_109[0][0]
activation_93 (Activation)	(None, 6, 6, 320)	0	batch_normalization_93[0][0]
mixed9_1 (Concatenate)	(None, 6, 6, 768)	0	activation_95[0][0] activation_96[0][0]
concatenate_2 (Concatenate)	(None, 6, 6, 768)	0	activation_99[0][0] activation_100[0][0]
activation_101 (Activation)	(None, 6, 6, 192)	0	batch_normalization_101[0][0]
mixed10 (Concatenate)	(None, 6, 6, 2048)	0	activation_93[0][0] mixed9_1[0][0] concatenate_2[0][0] activation_101[0][0]
conv2d_110 (Conv2D)	(None, 4, 4, 256)	4718848	mixed10[0][0]
flatten_6 (Flatten)	(None, 4096)	0	conv2d_110[0][0]
dropout_10 (Dropout)	(None, 4096)	0	flatten_6[0][0]
dense_11 (Dense)	(None, 4)	16388	dropout_10[0][0]

Total params: 26,538,020
Trainable params: 4,735,236
Non-trainable params: 21,802,784

In [45]:

```
history = model.fit_generator(datagenerated.flow(X_train,y_train, batch_size=batch_size),
                        steps_per_epoch=len(X_train) / batch_size, epochs=epochs, validation_data =
[X_test,y_test])
score = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

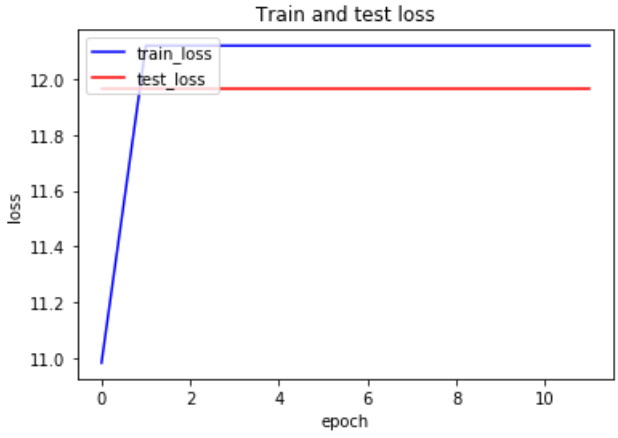
```
Epoch 1/12
49/48 [==============================] - 104s 2s/step - loss: 10.9795 - acc: 0.3014 - val_loss: 11
.9639 - val_acc: 0.2577
Epoch 2/12
49/48 [==============================] - 103s 2s/step - loss: 12.1362 - acc: 0.2470 - val_loss: 11
.9639 - val_acc: 0.2577
Epoch 3/12
49/48 [==============================] - 97s 2s/step - loss: 12.0690 - acc: 0.2512 - val_loss: 11
```

```
9639 - val_acc: 0.2577
Epoch 4/12
49/48 [==============================] - 97s 2s/step - loss: 12.1698 - acc: 0.2450 - val_loss: 11.
9639 - val_acc: 0.2577
Epoch 5/12
49/48 [==============================] - 97s 2s/step - loss: 12.0690 - acc: 0.2512 - val_loss: 11.
9639 - val_acc: 0.2577
Epoch 6/12
49/48 [==============================] - 101s 2s/step - loss: 12.1362 - acc: 0.2470 - val_loss: 11
.9639 - val_acc: 0.2577
Epoch 7/12
49/48 [==============================] - 98s 2s/step - loss: 12.1026 - acc: 0.2491 - val_loss: 11.
9639 - val_acc: 0.2577
Epoch 8/12
49/48 [==============================] - 88s 2s/step - loss: 12.1362 - acc: 0.2470 - val_loss: 11.
9639 - val_acc: 0.2577
Epoch 9/12
49/48 [==============================] - 93s 2s/step - loss: 12.1026 - acc: 0.2491 - val_loss: 11.
9639 - val_acc: 0.2577
Epoch 10/12
49/48 [==============================] - 97s 2s/step - loss: 12.1698 - acc: 0.2450 - val_loss: 11.
9639 - val_acc: 0.2577
Epoch 11/12
49/48 [==============================] - 96s 2s/step - loss: 12.1026 - acc: 0.2491 - val_loss: 11.
9639 - val_acc: 0.2577
Epoch 12/12
49/48 [==============================] - 90s 2s/step - loss: 12.1362 - acc: 0.2470 - val_loss: 11.
9639 - val_acc: 0.2577
Test loss: 11.963947100737661
Test accuracy: 0.25773195876288657
```

Plot between train loss and test loss

In [46]:

```
plt.plot(history.history['loss'],color="blue")
plt.plot(history.history['val_loss'],color="red")
plt.title('Train and test loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train_loss', 'test_loss'], loc='upper left')
plt.show()
```



Summary

*. Training the model with pretrained inception v3 model.
*. Test accuracy is only 45 %
*. The model is overfitting.

Using Resnet

In [47]:

```
from keras.applications import vgg16, inception_v3, ResNet50
pretrained_model_res = ResNet50(include_top=False, input_shape=(imageSize, imageSize, 3))
```

```
/usr/local/lib/python3.5/dist-packages/keras_applications/resnet50.py:265: UserWarning: The output shape of `ResNet50(include_top=False)` has been changed since Keras 2.2.0.
  warnings.warn('The output shape of `ResNet50(include_top=False)` '
```

In [48]:

```
base_model = pretrained_model_res
opt = keras.optimizers.Adam()
x = base_model.output
x = Conv2D(256, kernel_size = (3,3), padding = 'valid')(x)
x = Flatten()(x)
x = Dropout(0.75)(x)
predictions = Dense(4, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)
# Train top layer
for layer in base_model.layers:
    layer.trainable = False
model.compile(loss='categorical_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_2 (InputLayer)         (None, 256, 256, 3)       0         
_________________________________________________________________
conv1_pad (ZeroPadding2D)    (None, 262, 262, 3)       0         
_________________________________________________________________
conv1 (Conv2D)               (None, 128, 128, 64)      9472      
_________________________________________________________________
bn_conv1 (BatchNormalization (None, 128, 128, 64)      256       
_________________________________________________________________
activation_102 (Activation)  (None, 128, 128, 64)      0         
_________________________________________________________________
pool1_pad (ZeroPadding2D)    (None, 130, 130, 64)      0         
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 64, 64, 64)        0         
_________________________________________________________________
res2a_branch2a (Conv2D)      (None, 64, 64, 64)        4160      
_________________________________________________________________
bn2a_branch2a (BatchNormaliz (None, 64, 64, 64)        256       
_________________________________________________________________
activation_103 (Activation)  (None, 64, 64, 64)        0         
_________________________________________________________________
res2a_branch2b (Conv2D)      (None, 64, 64, 64)        36928     
_________________________________________________________________
bn2a_branch2b (BatchNormaliz (None, 64, 64, 64)        256       
_________________________________________________________________
activation_104 (Activation)  (None, 64, 64, 64)        0         
_________________________________________________________________
res2a_branch2c (Conv2D)      (None, 64, 64, 256)       16640     
_________________________________________________________________
res2a_branch1 (Conv2D)       (None, 64, 64, 256)       16640     
_________________________________________________________________
bn2a_branch2c (BatchNormaliz (None, 64, 64, 256)       1024      
_________________________________________________________________
bn2a_branch1 (BatchNormaliza (None, 64, 64, 256)       1024      
_________________________________________________________________
add_1 (Add)                  (None, 64, 64, 256)       0         
                                                                 
_________________________________________________________________
activation_105 (Activation)  (None, 64, 64, 256)       0         
_________________________________________________________________
res2b_branch2a (Conv2D)      (None, 64, 64, 64)        16448     
_________________________________________________________________
bn2b_branch2a (BatchNormaliz (None, 64, 64, 64)        256       
_________________________________________________________________
activation_106 (Activation)  (None, 64, 64, 64)        0         
_________________________________________________________________
res2b_branch2b (Conv2D)      (None, 64, 64, 64)        36928     
```

bn2b_branch2b (BatchNormalizati	(None, 64, 64, 64)	256	res2b_branch2b[0][0]
activation_107 (Activation)	(None, 64, 64, 64)	0	bn2b_branch2b[0][0]
res2b_branch2c (Conv2D)	(None, 64, 64, 256)	16640	activation_107[0][0]
bn2b_branch2c (BatchNormalizati	(None, 64, 64, 256)	1024	res2b_branch2c[0][0]
add_2 (Add)	(None, 64, 64, 256)	0	bn2b_branch2c[0][0] activation_105[0][0]
activation_108 (Activation)	(None, 64, 64, 256)	0	add_2[0][0]
res2c_branch2a (Conv2D)	(None, 64, 64, 64)	16448	activation_108[0][0]
bn2c_branch2a (BatchNormalizati	(None, 64, 64, 64)	256	res2c_branch2a[0][0]
activation_109 (Activation)	(None, 64, 64, 64)	0	bn2c_branch2a[0][0]
res2c_branch2b (Conv2D)	(None, 64, 64, 64)	36928	activation_109[0][0]
bn2c_branch2b (BatchNormalizati	(None, 64, 64, 64)	256	res2c_branch2b[0][0]
activation_110 (Activation)	(None, 64, 64, 64)	0	bn2c_branch2b[0][0]
res2c_branch2c (Conv2D)	(None, 64, 64, 256)	16640	activation_110[0][0]
bn2c_branch2c (BatchNormalizati	(None, 64, 64, 256)	1024	res2c_branch2c[0][0]
add_3 (Add)	(None, 64, 64, 256)	0	bn2c_branch2c[0][0] activation_108[0][0]
activation_111 (Activation)	(None, 64, 64, 256)	0	add_3[0][0]
res3a_branch2a (Conv2D)	(None, 32, 32, 128)	32896	activation_111[0][0]
bn3a_branch2a (BatchNormalizati	(None, 32, 32, 128)	512	res3a_branch2a[0][0]
activation_112 (Activation)	(None, 32, 32, 128)	0	bn3a_branch2a[0][0]
res3a_branch2b (Conv2D)	(None, 32, 32, 128)	147584	activation_112[0][0]
bn3a_branch2b (BatchNormalizati	(None, 32, 32, 128)	512	res3a_branch2b[0][0]
activation_113 (Activation)	(None, 32, 32, 128)	0	bn3a_branch2b[0][0]
res3a_branch2c (Conv2D)	(None, 32, 32, 512)	66048	activation_113[0][0]
res3a_branch1 (Conv2D)	(None, 32, 32, 512)	131584	activation_111[0][0]
bn3a_branch2c (BatchNormalizati	(None, 32, 32, 512)	2048	res3a_branch2c[0][0]
bn3a_branch1 (BatchNormalizatio	(None, 32, 32, 512)	2048	res3a_branch1[0][0]
add_4 (Add)	(None, 32, 32, 512)	0	bn3a_branch2c[0][0] bn3a_branch1[0][0]
activation_114 (Activation)	(None, 32, 32, 512)	0	add_4[0][0]
res3b_branch2a (Conv2D)	(None, 32, 32, 128)	65664	activation_114[0][0]
bn3b_branch2a (BatchNormalizati	(None, 32, 32, 128)	512	res3b_branch2a[0][0]
activation_115 (Activation)	(None, 32, 32, 128)	0	bn3b_branch2a[0][0]
res3b_branch2b (Conv2D)	(None, 32, 32, 128)	147584	activation_115[0][0]
bn3b_branch2b (BatchNormalizati	(None, 32, 32, 128)	512	res3b_branch2b[0][0]
activation_116 (Activation)	(None, 32, 32, 128)	0	bn3b_branch2b[0][0]
res3b_branch2c (Conv2D)	(None, 32, 32, 512)	66048	activation_116[0][0]
bn3b_branch2c (BatchNormalizati	(None, 32, 32, 512)	2048	res3b_branch2c[0][0]
add_5 (Add)	(None, 32, 32, 512)	0	bn3b_branch2c[0][0]

			activation_114[0][0]
activation_117 (Activation)	(None, 32, 32, 512)	0	add_5[0][0]
res3c_branch2a (Conv2D)	(None, 32, 32, 128)	65664	activation_117[0][0]
bn3c_branch2a (BatchNormalizati	(None, 32, 32, 128)	512	res3c_branch2a[0][0]
activation_118 (Activation)	(None, 32, 32, 128)	0	bn3c_branch2a[0][0]
res3c_branch2b (Conv2D)	(None, 32, 32, 128)	147584	activation_118[0][0]
bn3c_branch2b (BatchNormalizati	(None, 32, 32, 128)	512	res3c_branch2b[0][0]
activation_119 (Activation)	(None, 32, 32, 128)	0	bn3c_branch2b[0][0]
res3c_branch2c (Conv2D)	(None, 32, 32, 512)	66048	activation_119[0][0]
bn3c_branch2c (BatchNormalizati	(None, 32, 32, 512)	2048	res3c_branch2c[0][0]
add_6 (Add)	(None, 32, 32, 512)	0	bn3c_branch2c[0][0] activation_117[0][0]
activation_120 (Activation)	(None, 32, 32, 512)	0	add_6[0][0]
res3d_branch2a (Conv2D)	(None, 32, 32, 128)	65664	activation_120[0][0]
bn3d_branch2a (BatchNormalizati	(None, 32, 32, 128)	512	res3d_branch2a[0][0]
activation_121 (Activation)	(None, 32, 32, 128)	0	bn3d_branch2a[0][0]
res3d_branch2b (Conv2D)	(None, 32, 32, 128)	147584	activation_121[0][0]
bn3d_branch2b (BatchNormalizati	(None, 32, 32, 128)	512	res3d_branch2b[0][0]
activation_122 (Activation)	(None, 32, 32, 128)	0	bn3d_branch2b[0][0]
res3d_branch2c (Conv2D)	(None, 32, 32, 512)	66048	activation_122[0][0]
bn3d_branch2c (BatchNormalizati	(None, 32, 32, 512)	2048	res3d_branch2c[0][0]
add_7 (Add)	(None, 32, 32, 512)	0	bn3d_branch2c[0][0] activation_120[0][0]
activation_123 (Activation)	(None, 32, 32, 512)	0	add_7[0][0]
res4a_branch2a (Conv2D)	(None, 16, 16, 256)	131328	activation_123[0][0]
bn4a_branch2a (BatchNormalizati	(None, 16, 16, 256)	1024	res4a_branch2a[0][0]
activation_124 (Activation)	(None, 16, 16, 256)	0	bn4a_branch2a[0][0]
res4a_branch2b (Conv2D)	(None, 16, 16, 256)	590080	activation_124[0][0]
bn4a_branch2b (BatchNormalizati	(None, 16, 16, 256)	1024	res4a_branch2b[0][0]
activation_125 (Activation)	(None, 16, 16, 256)	0	bn4a_branch2b[0][0]
res4a_branch2c (Conv2D)	(None, 16, 16, 1024)	263168	activation_125[0][0]
res4a_branch1 (Conv2D)	(None, 16, 16, 1024)	525312	activation_123[0][0]
bn4a_branch2c (BatchNormalizati	(None, 16, 16, 1024)	4096	res4a_branch2c[0][0]
bn4a_branch1 (BatchNormalizatio	(None, 16, 16, 1024)	4096	res4a_branch1[0][0]
add_8 (Add)	(None, 16, 16, 1024)	0	bn4a_branch2c[0][0] bn4a_branch1[0][0]
activation_126 (Activation)	(None, 16, 16, 1024)	0	add_8[0][0]
res4b_branch2a (Conv2D)	(None, 16, 16, 256)	262400	activation_126[0][0]
bn4b_branch2a (BatchNormalizati	(None, 16, 16, 256)	1024	res4b_branch2a[0][0]
activation_127 (Activation)	(None, 16, 16, 256)	0	bn4b_branch2a[0][0]

```
res4b_branch2b (Conv2D)          (None, 16, 16, 256)  590080      activation_127[0][0]
__________________________________________________________________________________________________
bn4b_branch2b (BatchNormalizati (None, 16, 16, 256)  1024        res4b_branch2b[0][0]
__________________________________________________________________________________________________
activation_128 (Activation)      (None, 16, 16, 256)  0           bn4b_branch2b[0][0]
__________________________________________________________________________________________________
res4b_branch2c (Conv2D)          (None, 16, 16, 1024) 263168      activation_128[0][0]
__________________________________________________________________________________________________
bn4b_branch2c (BatchNormalizati (None, 16, 16, 1024) 4096        res4b_branch2c[0][0]
__________________________________________________________________________________________________
add_9 (Add)                      (None, 16, 16, 1024) 0           bn4b_branch2c[0][0]
                                                                  activation_126[0][0]
__________________________________________________________________________________________________
activation_129 (Activation)      (None, 16, 16, 1024) 0           add_9[0][0]
__________________________________________________________________________________________________
res4c_branch2a (Conv2D)          (None, 16, 16, 256)  262400      activation_129[0][0]
__________________________________________________________________________________________________
bn4c_branch2a (BatchNormalizati (None, 16, 16, 256)  1024        res4c_branch2a[0][0]
__________________________________________________________________________________________________
activation_130 (Activation)      (None, 16, 16, 256)  0           bn4c_branch2a[0][0]
__________________________________________________________________________________________________
res4c_branch2b (Conv2D)          (None, 16, 16, 256)  590080      activation_130[0][0]
__________________________________________________________________________________________________
bn4c_branch2b (BatchNormalizati (None, 16, 16, 256)  1024        res4c_branch2b[0][0]
__________________________________________________________________________________________________
activation_131 (Activation)      (None, 16, 16, 256)  0           bn4c_branch2b[0][0]
__________________________________________________________________________________________________
res4c_branch2c (Conv2D)          (None, 16, 16, 1024) 263168      activation_131[0][0]
__________________________________________________________________________________________________
bn4c_branch2c (BatchNormalizati (None, 16, 16, 1024) 4096        res4c_branch2c[0][0]
__________________________________________________________________________________________________
add_10 (Add)                     (None, 16, 16, 1024) 0           bn4c_branch2c[0][0]
                                                                  activation_129[0][0]
__________________________________________________________________________________________________
activation_132 (Activation)      (None, 16, 16, 1024) 0           add_10[0][0]
__________________________________________________________________________________________________
res4d_branch2a (Conv2D)          (None, 16, 16, 256)  262400      activation_132[0][0]
__________________________________________________________________________________________________
bn4d_branch2a (BatchNormalizati (None, 16, 16, 256)  1024        res4d_branch2a[0][0]
__________________________________________________________________________________________________
activation_133 (Activation)      (None, 16, 16, 256)  0           bn4d_branch2a[0][0]
__________________________________________________________________________________________________
res4d_branch2b (Conv2D)          (None, 16, 16, 256)  590080      activation_133[0][0]
__________________________________________________________________________________________________
bn4d_branch2b (BatchNormalizati (None, 16, 16, 256)  1024        res4d_branch2b[0][0]
__________________________________________________________________________________________________
activation_134 (Activation)      (None, 16, 16, 256)  0           bn4d_branch2b[0][0]
__________________________________________________________________________________________________
res4d_branch2c (Conv2D)          (None, 16, 16, 1024) 263168      activation_134[0][0]
__________________________________________________________________________________________________
bn4d_branch2c (BatchNormalizati (None, 16, 16, 1024) 4096        res4d_branch2c[0][0]
__________________________________________________________________________________________________
add_11 (Add)                     (None, 16, 16, 1024) 0           bn4d_branch2c[0][0]
                                                                  activation_132[0][0]
__________________________________________________________________________________________________
activation_135 (Activation)      (None, 16, 16, 1024) 0           add_11[0][0]
__________________________________________________________________________________________________
res4e_branch2a (Conv2D)          (None, 16, 16, 256)  262400      activation_135[0][0]
__________________________________________________________________________________________________
bn4e_branch2a (BatchNormalizati (None, 16, 16, 256)  1024        res4e_branch2a[0][0]
__________________________________________________________________________________________________
activation_136 (Activation)      (None, 16, 16, 256)  0           bn4e_branch2a[0][0]
__________________________________________________________________________________________________
res4e_branch2b (Conv2D)          (None, 16, 16, 256)  590080      activation_136[0][0]
__________________________________________________________________________________________________
bn4e_branch2b (BatchNormalizati (None, 16, 16, 256)  1024        res4e_branch2b[0][0]
__________________________________________________________________________________________________
activation_137 (Activation)      (None, 16, 16, 256)  0           bn4e_branch2b[0][0]
__________________________________________________________________________________________________
res4e_branch2c (Conv2D)          (None, 16, 16, 1024) 263168      activation_137[0][0]
__________________________________________________________________________________________________
bn4e_branch2c (BatchNormalizati (None, 16, 16, 1024) 4096        res4e_branch2c[0][0]
__________________________________________________________________________________________________
add_12 (Add)                     (None, 16, 16, 1024) 0           bn4e_branch2c[0][0]
                                                                  activation_135[0][0]
__________________________________________________________________________________________________
activation_138 (Activation)      (None, 16, 16, 1024) 0           add_12[0][0]
```

res4f_branch2a (Conv2D)	(None, 16, 16, 256)	262400	activation_138[0][0]
bn4f_branch2a (BatchNormalizati	(None, 16, 16, 256)	1024	res4f_branch2a[0][0]
activation_139 (Activation)	(None, 16, 16, 256)	0	bn4f_branch2a[0][0]
res4f_branch2b (Conv2D)	(None, 16, 16, 256)	590080	activation_139[0][0]
bn4f_branch2b (BatchNormalizati	(None, 16, 16, 256)	1024	res4f_branch2b[0][0]
activation_140 (Activation)	(None, 16, 16, 256)	0	bn4f_branch2b[0][0]
res4f_branch2c (Conv2D)	(None, 16, 16, 1024)	263168	activation_140[0][0]
bn4f_branch2c (BatchNormalizati	(None, 16, 16, 1024)	4096	res4f_branch2c[0][0]
add_13 (Add)	(None, 16, 16, 1024)	0	bn4f_branch2c[0][0] activation_138[0][0]
activation_141 (Activation)	(None, 16, 16, 1024)	0	add_13[0][0]
res5a_branch2a (Conv2D)	(None, 8, 8, 512)	524800	activation_141[0][0]
bn5a_branch2a (BatchNormalizati	(None, 8, 8, 512)	2048	res5a_branch2a[0][0]
activation_142 (Activation)	(None, 8, 8, 512)	0	bn5a_branch2a[0][0]
res5a_branch2b (Conv2D)	(None, 8, 8, 512)	2359808	activation_142[0][0]
bn5a_branch2b (BatchNormalizati	(None, 8, 8, 512)	2048	res5a_branch2b[0][0]
activation_143 (Activation)	(None, 8, 8, 512)	0	bn5a_branch2b[0][0]
res5a_branch2c (Conv2D)	(None, 8, 8, 2048)	1050624	activation_143[0][0]
res5a_branch1 (Conv2D)	(None, 8, 8, 2048)	2099200	activation_141[0][0]
bn5a_branch2c (BatchNormalizati	(None, 8, 8, 2048)	8192	res5a_branch2c[0][0]
bn5a_branch1 (BatchNormalizatio	(None, 8, 8, 2048)	8192	res5a_branch1[0][0]
add_14 (Add)	(None, 8, 8, 2048)	0	bn5a_branch2c[0][0] bn5a_branch1[0][0]
activation_144 (Activation)	(None, 8, 8, 2048)	0	add_14[0][0]
res5b_branch2a (Conv2D)	(None, 8, 8, 512)	1049088	activation_144[0][0]
bn5b_branch2a (BatchNormalizati	(None, 8, 8, 512)	2048	res5b_branch2a[0][0]
activation_145 (Activation)	(None, 8, 8, 512)	0	bn5b_branch2a[0][0]
res5b_branch2b (Conv2D)	(None, 8, 8, 512)	2359808	activation_145[0][0]
bn5b_branch2b (BatchNormalizati	(None, 8, 8, 512)	2048	res5b_branch2b[0][0]
activation_146 (Activation)	(None, 8, 8, 512)	0	bn5b_branch2b[0][0]
res5b_branch2c (Conv2D)	(None, 8, 8, 2048)	1050624	activation_146[0][0]
bn5b_branch2c (BatchNormalizati	(None, 8, 8, 2048)	8192	res5b_branch2c[0][0]
add_15 (Add)	(None, 8, 8, 2048)	0	bn5b_branch2c[0][0] activation_144[0][0]
activation_147 (Activation)	(None, 8, 8, 2048)	0	add_15[0][0]
res5c_branch2a (Conv2D)	(None, 8, 8, 512)	1049088	activation_147[0][0]
bn5c_branch2a (BatchNormalizati	(None, 8, 8, 512)	2048	res5c_branch2a[0][0]
activation_148 (Activation)	(None, 8, 8, 512)	0	bn5c_branch2a[0][0]
res5c_branch2b (Conv2D)	(None, 8, 8, 512)	2359808	activation_148[0][0]
bn5c_branch2b (BatchNormalizati	(None, 8, 8, 512)	2048	res5c_branch2b[0][0]

```
activation_149 (Activation)     (None, 8, 8, 512)    0           bn5c_branch2b[0][0]
res5c_branch2c (Conv2D)         (None, 8, 8, 2048)   1050624     activation_149[0][0]
bn5c_branch2c (BatchNormalizati (None, 8, 8, 2048)   8192        res5c_branch2c[0][0]
add_16 (Add)                    (None, 8, 8, 2048)   0           bn5c_branch2c[0][0]
                                                                 activation_147[0][0]
activation_150 (Activation)     (None, 8, 8, 2048)   0           add_16[0][0]
conv2d_111 (Conv2D)             (None, 6, 6, 256)    4718848     activation_150[0][0]
flatten_7 (Flatten)             (None, 9216)         0           conv2d_111[0][0]
dropout_11 (Dropout)            (None, 9216)         0           flatten_7[0][0]
dense_12 (Dense)                (None, 4)            36868       dropout_11[0][0]
==================================================================================================
Total params: 28,343,428
Trainable params: 4,755,716
Non-trainable params: 23,587,712
__________________________________________________________________________________________________
```

In [49]:

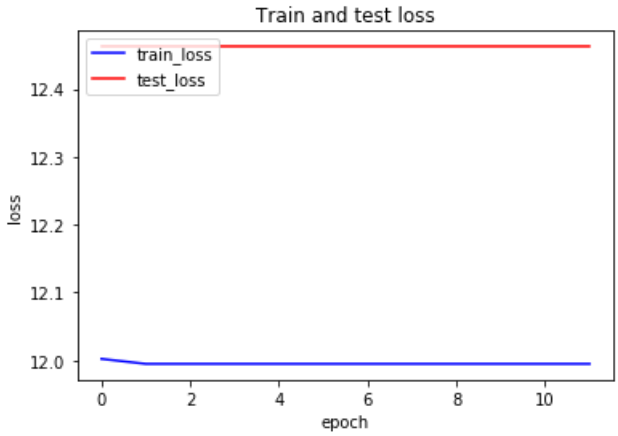
```
history = model.fit_generator(datagenerated.flow(X_train,y_train, batch_size=batch_size),
                        steps_per_epoch=len(X_train) / batch_size, epochs=epochs, validation_data =
[X_test,y_test])
score = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Epoch 1/12
49/48 [==============================] - 168s 3s/step - loss: 12.0201 - acc: 0.2317 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 2/12
49/48 [==============================] - 160s 3s/step - loss: 12.0128 - acc: 0.2547 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 3/12
49/48 [==============================] - 163s 3s/step - loss: 11.9792 - acc: 0.2568 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 4/12
49/48 [==============================] - 161s 3s/step - loss: 12.0464 - acc: 0.2526 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 5/12
49/48 [==============================] - 162s 3s/step - loss: 11.9792 - acc: 0.2568 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 6/12
49/48 [==============================] - 160s 3s/step - loss: 11.9121 - acc: 0.2609 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 7/12
49/48 [==============================] - 160s 3s/step - loss: 11.9457 - acc: 0.2589 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 8/12
49/48 [==============================] - 160s 3s/step - loss: 12.0464 - acc: 0.2526 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 9/12
49/48 [==============================] - 161s 3s/step - loss: 12.0464 - acc: 0.2526 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 10/12
49/48 [==============================] - 163s 3s/step - loss: 11.8785 - acc: 0.2630 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 11/12
49/48 [==============================] - 163s 3s/step - loss: 11.9457 - acc: 0.2589 - val_loss: 12
.4624 - val_acc: 0.2268
Epoch 12/12
49/48 [==============================] - 161s 3s/step - loss: 11.9792 - acc: 0.2568 - val_loss: 12
.4624 - val_acc: 0.2268
Test loss: 12.46244489532156
Test accuracy: 0.2268041237113402
```

Plot between train loss and test loss

In [50]:

```
plt.plot(history.history['loss'],color="blue")
plt.plot(history.history['val_loss'],color="red")
plt.title('Train and test loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train_loss', 'test_loss'], loc='upper left')
plt.show()
```



Summary

*. Training the model with pretrained resnet model.
*. Test accuracy is only 22 %
*. The model is performing worse than any other models .

Using VGG16

In [51]:

```
pretrained_model_vgg = VGG16(include_top=False, input_shape=(imageSize, imageSize, 3))
```

In [52]:

```
base_model = pretrained_model_vgg
opt = keras.optimizers.Adam()
x = base_model.output
x = Conv2D(256, kernel_size = (3,3), padding = 'valid')(x)
x = Flatten()(x)
x = Dropout(0.75)(x)
predictions = Dense(4, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)
# Train top layer
for layer in base_model.layers:
    layer.trainable = False
model.compile(loss='categorical_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 256, 256, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 256, 256, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 256, 256, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 128, 128, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 128, 128, 128)     73856
```

```
block2_conv2 (Conv2D)        (None, 128, 128, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 64, 64, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 64, 64, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 64, 64, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 64, 64, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 32, 32, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 32, 32, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 32, 32, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 32, 32, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 16, 16, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 8, 8, 512)         0
_________________________________________________________________
conv2d_112 (Conv2D)          (None, 6, 6, 256)         1179904
_________________________________________________________________
flatten_8 (Flatten)          (None, 9216)              0
_________________________________________________________________
dropout_12 (Dropout)         (None, 9216)              0
_________________________________________________________________
dense_13 (Dense)             (None, 4)                 36868
=================================================================
Total params: 15,931,460
Trainable params: 1,216,772
Non-trainable params: 14,714,688
_________________________________________________________________
```

In [53]:

```
history = model.fit_generator(datagenerated.flow(X_train,y_train, batch_size=batch_size),
                        steps_per_epoch=len(X_train) / batch_size, epochs=epochs, validation_data =
[X_test,y_test])
score = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

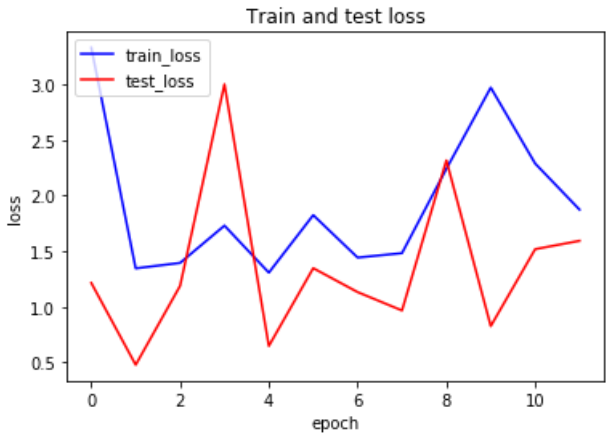
```
Epoch 1/12
49/48 [==============================] - 158s 3s/step - loss: 3.3151 - acc: 0.4396 - val_loss: 1.2
139 - val_acc: 0.7113
Epoch 2/12
49/48 [==============================] - 152s 3s/step - loss: 1.3570 - acc: 0.7088 - val_loss: 0.4
765 - val_acc: 0.8814
Epoch 3/12
49/48 [==============================] - 154s 3s/step - loss: 1.3842 - acc: 0.7530 - val_loss: 1.1
874 - val_acc: 0.7938
Epoch 4/12
49/48 [==============================] - 153s 3s/step - loss: 1.7091 - acc: 0.7359 - val_loss: 3.0
028 - val_acc: 0.7113
Epoch 5/12
49/48 [==============================] - 157s 3s/step - loss: 1.2917 - acc: 0.8201 - val_loss: 0.6
463 - val_acc: 0.8660
Epoch 6/12
49/48 [==============================] - 156s 3s/step - loss: 1.8357 - acc: 0.7840 - val_loss: 1.3
471 - val_acc: 0.8454
Epoch 7/12
49/48 [==============================] - 154s 3s/step - loss: 1.4244 - acc: 0.8252 - val_loss: 1.1
315 - val_acc: 0.8660
Epoch 8/12
49/48 [==============================] - 157s 3s/step - loss: 1.4948 - acc: 0.8274 - val_loss: 0.9
660 - val_acc: 0.8866
```

```
Epoch 9/12
49/48 [==============================] - 157s 3s/step - loss: 2.2136 - acc: 0.7895 - val_loss: 2.3
177 - val_acc: 0.7629
Epoch 10/12
49/48 [==============================] - 156s 3s/step - loss: 2.9358 - acc: 0.7601 - val_loss: 0.8
253 - val_acc: 0.9227
Epoch 11/12
49/48 [==============================] - 154s 3s/step - loss: 2.3277 - acc: 0.8006 - val_loss: 1.5
180 - val_acc: 0.8866
Epoch 12/12
49/48 [==============================] - 152s 3s/step - loss: 1.8843 - acc: 0.8333 - val_loss: 1.5
933 - val_acc: 0.8711
Test loss: 1.5933270614171766
Test accuracy: 0.8711340206185567
```

Plot between train loss and test loss

In [54]:

```
plt.plot(history.history['loss'],color="blue")
plt.plot(history.history['val_loss'],color="red")
plt.title('Train and test loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train_loss', 'test_loss'], loc='upper left')
plt.show()
```



Summary

```
        *. Training the model with pretrained VGG16 model.
        *. Test accuracy is only 87 %
        *. The model is performing better than other baseline and other transfer learning mo
dels .
```

Table

Model	Test loss	Test accuracy
2 layer architecture	1.02	0.54
3 layer architecture	11.79	0.26
Inception v3	11.96	0.25
ResNet	12.46	0.22
VGG	1.59	0.87

Summary

*. VGG16 model is performing better than all the other model
*. With VGG16, we have an accuracy of 87 %

Conclusion

*. Loading the data
*. Labelling the data, and taking the labelled value as target value
*. Converting all the y values to categorical values.
*. We are augmenting the images
*. Training the model with basline, trying with different architecture.
*. Applying transfer learning methodology to train the model.
*. With VGG16, we are getting better accuracy than other models.