In [12]:
```
!pip install keras
```

```
Requirement already satisfied: keras in c:\users\shadm\anaconda3\lib\site-packages
(2.8.0)
```

In [13]:
```
!pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\shadm\anaconda3\lib\site-pac
kages (2.8.0)
Requirement already satisfied: setuptools in c:\users\shadm\anaconda3\lib\site-pac
kages (from tensorflow) (61.2.0)
Requirement already satisfied: keras<2.9,>=2.8.0rc0 in c:\users\shadm\anaconda3\li
b\site-packages (from tensorflow) (2.8.0)
Requirement already satisfied: h5py>=2.9.0 in c:\users\shadm\anaconda3\lib\site-pa
ckages (from tensorflow) (3.6.0)
Requirement already satisfied: tf-estimator-nightly==2.8.0.dev2021122109 in c:\use
rs\shadm\anaconda3\lib\site-packages (from tensorflow) (2.8.0.dev2021122109)
Requirement already satisfied: gast>=0.2.1 in c:\users\shadm\anaconda3\lib\site-pa
ckages (from tensorflow) (0.5.3)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\shadm\anaconda3\lib
\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\shadm\anaconda3\lib
\site-packages (from tensorflow) (1.42.0)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\shadm\anaconda3\lib\site-
packages (from tensorflow) (1.12.1)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\shadm\anaconda3\lib\s
ite-packages (from tensorflow) (1.6.3)
Requirement already satisfied: libclang>=9.0.1 in c:\users\shadm\anaconda3\lib\sit
e-packages (from tensorflow) (14.0.1)
Requirement already satisfied: keras-preprocessing>=1.1.1 in c:\users\shadm\anacon
da3\lib\site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\shadm\anaconda3\lib\si
te-packages (from tensorflow) (1.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\shadm\anaconda
3\lib\site-packages (from tensorflow) (4.1.1)
Requirement already satisfied: protobuf>=3.9.2 in c:\users\shadm\anaconda3\lib\sit
e-packages (from tensorflow) (3.19.1)
Requirement already satisfied: absl-py>=0.4.0 in c:\users\shadm\anaconda3\lib\site
-packages (from tensorflow) (1.0.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in c:\users\sh
adm\anaconda3\lib\site-packages (from tensorflow) (0.25.0)
Requirement already satisfied: six>=1.12.0 in c:\users\shadm\anaconda3\lib\site-pa
ckages (from tensorflow) (1.16.0)
Requirement already satisfied: tensorboard<2.9,>=2.8 in c:\users\shadm\anaconda3\l
ib\site-packages (from tensorflow) (2.8.0)
Requirement already satisfied: numpy>=1.20 in c:\users\shadm\anaconda3\lib\site-pa
ckages (from tensorflow) (1.21.5)
Requirement already satisfied: flatbuffers>=1.12 in c:\users\shadm\anaconda3\lib\s
ite-packages (from tensorflow) (2.0)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\shadm\anaconda3\lib\s
ite-packages (from tensorflow) (3.3.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\shadm\anaconda3\lib
\site-packages (from astunparse>=1.6.0->tensorflow) (0.37.1)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\shadm\anaconda3\lib
\site-packages (from tensorboard<2.9,>=2.8->tensorflow) (2.27.1)
Requirement already satisfied: werkzeug>=0.11.15 in c:\users\shadm\anaconda3\lib\s
ite-packages (from tensorboard<2.9,>=2.8->tensorflow) (2.0.3)
Requirement already satisfied: google-auth<3,>=1.6.3 in c:\users\shadm\anaconda3\l
ib\site-packages (from tensorboard<2.9,>=2.8->tensorflow) (1.33.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in c:\users\shadm
\anaconda3\lib\site-packages (from tensorboard<2.9,>=2.8->tensorflow) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in c:\users\shadm\anaconda3\lib\sit
e-packages (from tensorboard<2.9,>=2.8->tensorflow) (3.3.4)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in c:\users\shadm\ana
conda3\lib\site-packages (from tensorboard<2.9,>=2.8->tensorflow) (1.8.1)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in c:\users\s
hadm\anaconda3\lib\site-packages (from tensorboard<2.9,>=2.8->tensorflow) (0.6.1)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in c:\users\shadm\anaconda3
\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.9,>=2.8->tensorflow)
(4.2.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\shadm\anaconda3\l
```

```
ib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.9,>=2.8->tensorflow)
(0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in c:\users\shadm\anaconda3\lib\site-
packages (from google-auth<3,>=1.6.3->tensorboard<2.9,>=2.8->tensorflow) (4.7.2)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\shadm\anaconda
3\lib\site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.9,>=2.8-
>tensorflow) (1.3.1)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\shadm\anaconda3\li
b\site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.
9,>=2.8->tensorflow) (0.4.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\shadm\anaconda3\l
ib\site-packages (from requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow) (1.
26.9)
Requirement already satisfied: idna<4,>=2.5 in c:\users\shadm\anaconda3\lib\site-p
ackages (from requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\shadm\anaconda3\lib
\site-packages (from requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow) (202
1.10.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\shadm\anacond
a3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow)
(2.0.4)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\shadm\anaconda3\lib\sit
e-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tenso
rboard<2.9,>=2.8->tensorflow) (3.2.0)
```

In [14]:

```
import os

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os
from glob import glob
import seaborn as sns
from PIL import Image
np.random.seed(11)
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, KFold, cross_val_score, GridS
from sklearn.metrics import accuracy_score
import itertools

import keras
from keras.utils.np_utils import to_categorical
from keras.models import Sequential, Model
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D
from keras import backend as K
from tensorflow.keras.layers import BatchNormalization
from keras.utils.np_utils import to_categorical
from tensorflow.keras.optimizers import Adam, RMSprop
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ReduceLROnPlateau
from keras.wrappers.scikit_learn import KerasClassifier
from tensorflow.keras.applications.resnet50 import ResNet50
from keras import backend as K
```

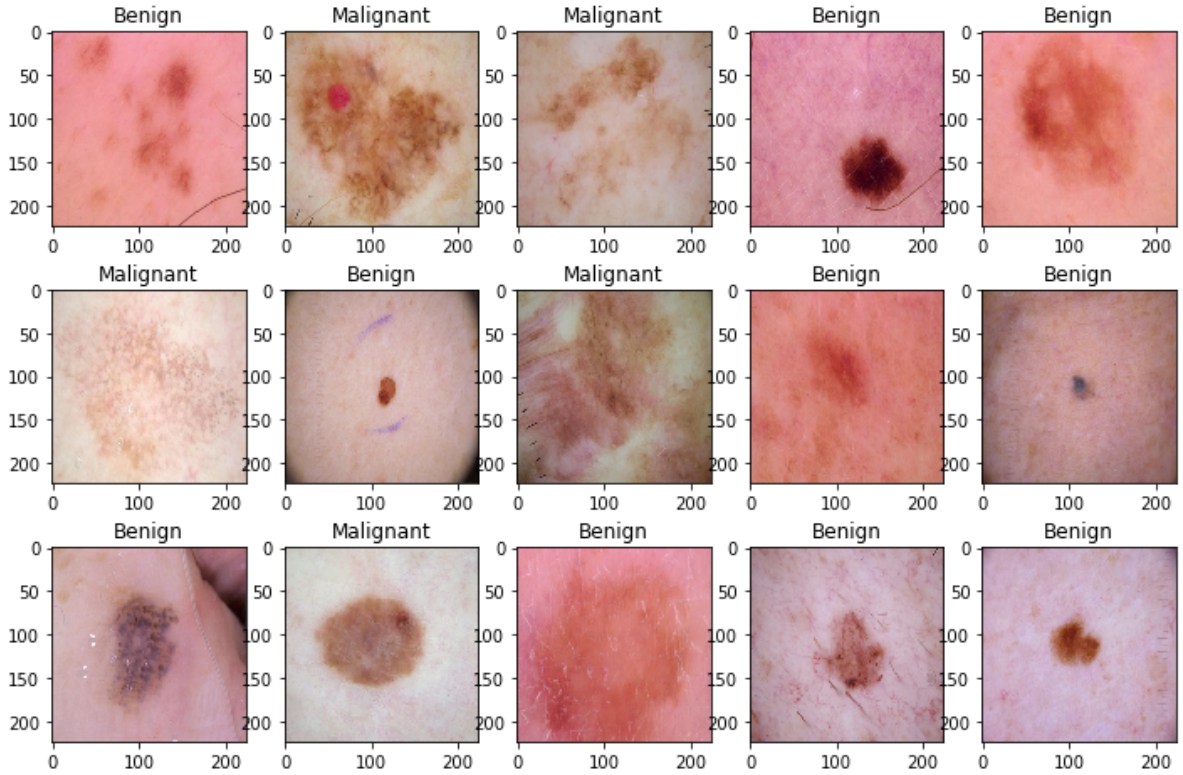
In [15]:

```
folder_benign_train = r'C:\Users\shadm\Desktop\Project\Biomedical_Image_Processing_
folder_malignant_train = r'C:\Users\shadm\Desktop\Project\Biomedical_Image_Process

folder_benign_test = r'C:\Users\shadm\Desktop\Project\Biomedical_Image_Processing_U
folder_malignant_test = r'C:\Users\shadm\Desktop\Project\Biomedical_Image_Processi

read = lambda imname: np.asarray(Image.open(imname).convert("RGB"))
```

```
ims_benign = [read(os.path.join(folder_benign_train, filename)) for filename in os
X_benign = np.array(ims_benign, dtype='uint8')
ims_malignant = [read(os.path.join(folder_malignant_train, filename)) for filename
X_malignant = np.array(ims_malignant, dtype='uint8')

ims_benign = [read(os.path.join(folder_benign_test, filename)) for filename in os.l
X_benign_test = np.array(ims_benign, dtype='uint8')
ims_malignant = [read(os.path.join(folder_malignant_test, filename)) for filename :
X_malignant_test = np.array(ims_malignant, dtype='uint8')

y_benign = np.zeros(X_benign.shape[0])
y_malignant = np.ones(X_malignant.shape[0])

y_benign_test = np.zeros(X_benign_test.shape[0])
y_malignant_test = np.ones(X_malignant_test.shape[0])

X_train = np.concatenate((X_benign, X_malignant), axis = 0)
y_train = np.concatenate((y_benign, y_malignant), axis = 0)

X_test = np.concatenate((X_benign_test, X_malignant_test), axis = 0)
y_test = np.concatenate((y_benign_test, y_malignant_test), axis = 0)

s = np.arange(X_train.shape[0])
np.random.shuffle(s)
X_train = X_train[s]
y_train = y_train[s]

s = np.arange(X_test.shape[0])
np.random.shuffle(s)
X_test = X_test[s]
y_test = y_test[s]
```

In [16]:

```
w=40
h=30
fig=plt.figure(figsize=(12, 8))
columns = 5
rows = 3

for i in range(1, columns*rows +1):
    ax = fig.add_subplot(rows, columns, i)
    if y_train[i] == 0:
        ax.title.set_text('Benign')
    else:
        ax.title.set_text('Malignant')
    plt.imshow(X_train[i], interpolation='nearest')
plt.show()
```



In [17]:
```
y_train = to_categorical(y_train, num_classes= 2)
y_test = to_categorical(y_test, num_classes= 2)
```

In [18]:
```
X_train = X_train/255.
X_test = X_test/255.
```

In [19]:
```
def build(input_shape= (224,224,3), lr = 1e-3, num_classes= 2,
          init= 'normal', activ= 'relu', optim= 'adam'):
    model = Sequential()
    model.add(Conv2D(64, kernel_size=(3, 3),padding = 'Same',input_shape=input_sha
                     activation= activ, kernel_initializer='glorot_uniform'))
    model.add(MaxPool2D(pool_size = (2, 2)))
    model.add(Dropout(0.25))

    model.add(Conv2D(64, kernel_size=(3, 3),padding = 'Same',
                     activation =activ, kernel_initializer = 'glorot_uniform'))
    model.add(MaxPool2D(pool_size = (2, 2)))
    model.add(Dropout(0.25))

    model.add(Flatten())
    model.add(Dense(128, activation='relu', kernel_initializer=init))
    model.add(Dense(num_classes, activation='softmax'))
    model.summary()

    if optim == 'rmsprop':
        optimizer = RMSprop(lr=lr)

    else:
        optimizer = Adam(lr=lr)

    model.compile(optimizer = optimizer ,loss = "binary_crossentropy", metrics=["a
    return model

learning_rate_reduction = ReduceLROnPlateau(monitor='val_accuracy',
                                            patience=5,
                                            verbose=1,
                                            factor=0.5,
                                            min_lr=1e-7)
```

In [20]:
```
input_shape = (224,224,3)
lr = 1e-5
init = 'normal'
activ = 'relu'
optim = 'adam'
epochs = 40
batch_size = 64

model = build(lr=lr, init= init, activ= activ, optim=optim, input_shape= input_shap
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 224, 224, 64)      1792

 max_pooling2d_2 (MaxPooling  (None, 112, 112, 64)     0
 2D)

 dropout_2 (Dropout)         (None, 112, 112, 64)      0

 conv2d_3 (Conv2D)           (None, 112, 112, 64)      36928

 max_pooling2d_3 (MaxPooling  (None, 56, 56, 64)       0
 2D)

 dropout_3 (Dropout)         (None, 56, 56, 64)        0

 flatten_1 (Flatten)         (None, 200704)            0

 dense_2 (Dense)             (None, 128)               25690240

 dense_3 (Dense)             (None, 2)                 258

=================================================================
Total params: 25,729,218
Trainable params: 25,729,218
Non-trainable params: 0
_________________________________________________________________
```

In [21]:
```
history = model.fit(X_train, y_train, validation_split=0.2,
                    epochs= epochs, batch_size= batch_size, verbose=1,
                    callbacks=[learning_rate_reduction]
                   )
```

```
Epoch 1/40
3/3 [==============================] - 19s 6s/step - loss: 1.2502 - accuracy: 0.53
12 - val_loss: 0.7285 - val_accuracy: 0.5250 - lr: 1.0000e-05
Epoch 2/40
3/3 [==============================] - 16s 5s/step - loss: 1.0799 - accuracy: 0.50
63 - val_loss: 0.7692 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 3/40
3/3 [==============================] - 16s 5s/step - loss: 1.0633 - accuracy: 0.51
25 - val_loss: 0.7041 - val_accuracy: 0.5250 - lr: 1.0000e-05
Epoch 4/40
3/3 [==============================] - 16s 5s/step - loss: 1.0985 - accuracy: 0.48
12 - val_loss: 0.7002 - val_accuracy: 0.5500 - lr: 1.0000e-05
Epoch 5/40
3/3 [==============================] - 16s 5s/step - loss: 0.9788 - accuracy: 0.51
88 - val_loss: 0.7283 - val_accuracy: 0.5750 - lr: 1.0000e-05
Epoch 6/40
3/3 [==============================] - 16s 5s/step - loss: 0.9588 - accuracy: 0.55
00 - val_loss: 0.6782 - val_accuracy: 0.5750 - lr: 1.0000e-05
Epoch 7/40
3/3 [==============================] - 16s 5s/step - loss: 0.9073 - accuracy: 0.57
50 - val_loss: 0.6522 - val_accuracy: 0.5500 - lr: 1.0000e-05
Epoch 8/40
3/3 [==============================] - 16s 5s/step - loss: 0.9650 - accuracy: 0.50
63 - val_loss: 0.6367 - val_accuracy: 0.5500 - lr: 1.0000e-05
Epoch 9/40
3/3 [==============================] - 16s 5s/step - loss: 0.9343 - accuracy: 0.56
88 - val_loss: 0.6334 - val_accuracy: 0.5500 - lr: 1.0000e-05
Epoch 10/40
3/3 [==============================] - ETA: 0s - loss: 0.8591 - accuracy: 0.5875
Epoch 10: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-06.
3/3 [==============================] - 16s 5s/step - loss: 0.8591 - accuracy: 0.58
75 - val_loss: 0.6557 - val_accuracy: 0.5500 - lr: 1.0000e-05
Epoch 11/40
3/3 [==============================] - 16s 5s/step - loss: 0.8849 - accuracy: 0.58
13 - val_loss: 0.6741 - val_accuracy: 0.5250 - lr: 5.0000e-06
Epoch 12/40
3/3 [==============================] - 16s 5s/step - loss: 0.9483 - accuracy: 0.55
00 - val_loss: 0.6736 - val_accuracy: 0.5250 - lr: 5.0000e-06
Epoch 13/40
3/3 [==============================] - 16s 5s/step - loss: 0.8818 - accuracy: 0.56
25 - val_loss: 0.6479 - val_accuracy: 0.5500 - lr: 5.0000e-06
Epoch 14/40
3/3 [==============================] - 16s 5s/step - loss: 0.8292 - accuracy: 0.64
38 - val_loss: 0.6229 - val_accuracy: 0.5500 - lr: 5.0000e-06
Epoch 15/40
3/3 [==============================] - ETA: 0s - loss: 0.8251 - accuracy: 0.6062
Epoch 15: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-06.
3/3 [==============================] - 16s 5s/step - loss: 0.8251 - accuracy: 0.60
62 - val_loss: 0.6011 - val_accuracy: 0.5500 - lr: 5.0000e-06
Epoch 16/40
3/3 [==============================] - 16s 5s/step - loss: 0.7220 - accuracy: 0.68
75 - val_loss: 0.5962 - val_accuracy: 0.5500 - lr: 2.5000e-06
Epoch 17/40
3/3 [==============================] - 16s 5s/step - loss: 0.7687 - accuracy: 0.65
00 - val_loss: 0.5967 - val_accuracy: 0.5500 - lr: 2.5000e-06
Epoch 18/40
3/3 [==============================] - 15s 5s/step - loss: 0.8818 - accuracy: 0.56
25 - val_loss: 0.6014 - val_accuracy: 0.5500 - lr: 2.5000e-06
Epoch 19/40
3/3 [==============================] - 16s 5s/step - loss: 0.8205 - accuracy: 0.64
38 - val_loss: 0.6047 - val_accuracy: 0.5500 - lr: 2.5000e-06
Epoch 20/40
3/3 [==============================] - ETA: 0s - loss: 0.8235 - accuracy: 0.6812
Epoch 20: ReduceLROnPlateau reducing learning rate to 1.249999968422344e-06.
```

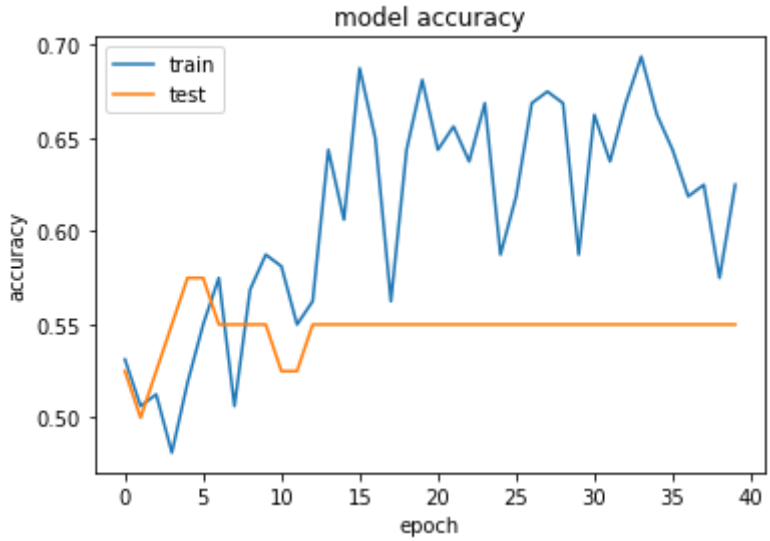
```
3/3 [==============================] - 16s 5s/step - loss: 0.8235 - accuracy: 0.68
12 - val_loss: 0.6130 - val_accuracy: 0.5500 - lr: 2.5000e-06
Epoch 21/40
3/3 [==============================] - 16s 5s/step - loss: 0.7343 - accuracy: 0.64
38 - val_loss: 0.6184 - val_accuracy: 0.5500 - lr: 1.2500e-06
Epoch 22/40
3/3 [==============================] - 16s 5s/step - loss: 0.7762 - accuracy: 0.65
62 - val_loss: 0.6229 - val_accuracy: 0.5500 - lr: 1.2500e-06
Epoch 23/40
3/3 [==============================] - 16s 5s/step - loss: 0.7806 - accuracy: 0.63
75 - val_loss: 0.6264 - val_accuracy: 0.5500 - lr: 1.2500e-06
Epoch 24/40
3/3 [==============================] - 16s 5s/step - loss: 0.7173 - accuracy: 0.66
87 - val_loss: 0.6276 - val_accuracy: 0.5500 - lr: 1.2500e-06
Epoch 25/40
3/3 [==============================] - ETA: 0s - loss: 0.8931 - accuracy: 0.5875
Epoch 25: ReduceLROnPlateau reducing learning rate to 6.24999984211172e-07.
3/3 [==============================] - 16s 5s/step - loss: 0.8931 - accuracy: 0.58
75 - val_loss: 0.6226 - val_accuracy: 0.5500 - lr: 1.2500e-06
Epoch 26/40
3/3 [==============================] - 16s 5s/step - loss: 0.7701 - accuracy: 0.61
87 - val_loss: 0.6187 - val_accuracy: 0.5500 - lr: 6.2500e-07
Epoch 27/40
3/3 [==============================] - 16s 5s/step - loss: 0.7159 - accuracy: 0.66
87 - val_loss: 0.6142 - val_accuracy: 0.5500 - lr: 6.2500e-07
Epoch 28/40
3/3 [==============================] - 16s 5s/step - loss: 0.7296 - accuracy: 0.67
50 - val_loss: 0.6095 - val_accuracy: 0.5500 - lr: 6.2500e-07
Epoch 29/40
3/3 [==============================] - 16s 5s/step - loss: 0.7329 - accuracy: 0.66
87 - val_loss: 0.6055 - val_accuracy: 0.5500 - lr: 6.2500e-07
Epoch 30/40
3/3 [==============================] - ETA: 0s - loss: 0.7907 - accuracy: 0.5875
Epoch 30: ReduceLROnPlateau reducing learning rate to 3.12499992105586e-07.
3/3 [==============================] - 17s 5s/step - loss: 0.7907 - accuracy: 0.58
75 - val_loss: 0.6016 - val_accuracy: 0.5500 - lr: 6.2500e-07
Epoch 31/40
3/3 [==============================] - 16s 5s/step - loss: 0.7209 - accuracy: 0.66
25 - val_loss: 0.6004 - val_accuracy: 0.5500 - lr: 3.1250e-07
Epoch 32/40
3/3 [==============================] - 16s 5s/step - loss: 0.7827 - accuracy: 0.63
75 - val_loss: 0.5998 - val_accuracy: 0.5500 - lr: 3.1250e-07
Epoch 33/40
3/3 [==============================] - 16s 5s/step - loss: 0.8126 - accuracy: 0.66
87 - val_loss: 0.5995 - val_accuracy: 0.5500 - lr: 3.1250e-07
Epoch 34/40
3/3 [==============================] - 16s 5s/step - loss: 0.7055 - accuracy: 0.69
38 - val_loss: 0.5994 - val_accuracy: 0.5500 - lr: 3.1250e-07
Epoch 35/40
3/3 [==============================] - ETA: 0s - loss: 0.7627 - accuracy: 0.6625
Epoch 35: ReduceLROnPlateau reducing learning rate to 1.56249996052793e-07.
3/3 [==============================] - 16s 5s/step - loss: 0.7627 - accuracy: 0.66
25 - val_loss: 0.5984 - val_accuracy: 0.5500 - lr: 3.1250e-07
Epoch 36/40
3/3 [==============================] - 16s 5s/step - loss: 0.8029 - accuracy: 0.64
38 - val_loss: 0.5978 - val_accuracy: 0.5500 - lr: 1.5625e-07
Epoch 37/40
3/3 [==============================] - 16s 5s/step - loss: 0.7627 - accuracy: 0.61
87 - val_loss: 0.5973 - val_accuracy: 0.5500 - lr: 1.5625e-07
Epoch 38/40
3/3 [==============================] - 16s 5s/step - loss: 0.8174 - accuracy: 0.62
50 - val_loss: 0.5969 - val_accuracy: 0.5500 - lr: 1.5625e-07
Epoch 39/40
3/3 [==============================] - 16s 5s/step - loss: 0.8455 - accuracy: 0.57
```

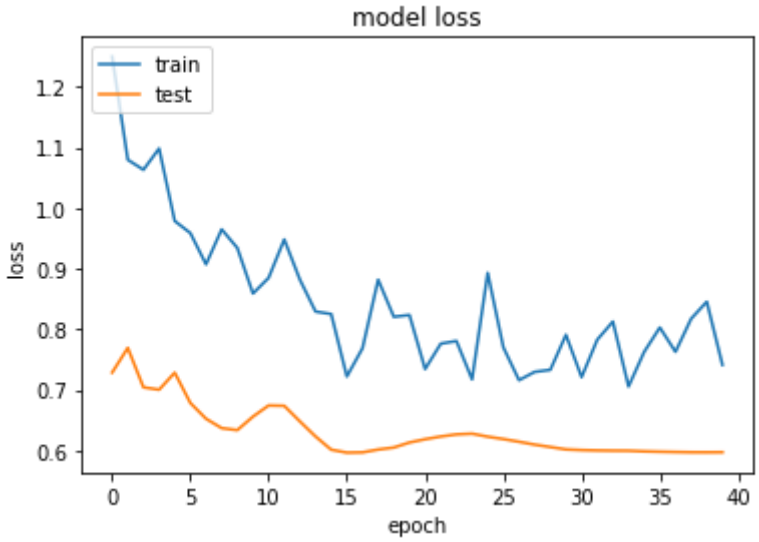
```
50 - val_loss: 0.5968 - val_accuracy: 0.5500 - lr: 1.5625e-07
Epoch 40/40
3/3 [==============================] - ETA: 0s - loss: 0.7412 - accuracy: 0.6250
Epoch 40: ReduceLROnPlateau reducing learning rate to 1e-07.
3/3 [==============================] - 16s 5s/step - loss: 0.7412 - accuracy: 0.62
50 - val_loss: 0.5969 - val_accuracy: 0.5500 - lr: 1.5625e-07
```

print(history.history.keys()) plt.plot(history.history['accuracy']) plt.plot(history.history['val_accuracy']) plt.title('model accuracy') plt.ylabel('accuracy') plt.xlabel('epoch') plt.legend(['train', 'test'], loc='upper left') plt.show() plt.plot(history.history['loss']) plt.plot(history.history['val_loss']) plt.title('model loss') plt.ylabel('loss') plt.xlabel('epoch') plt.legend(['train', 'test'], loc='upper left') plt.show()

In [22]:
```
print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy', 'lr'])
```





In [23]:
```
K.clear_session()
del model
del history
```

In [13]:
```
kfold = KFold(n_splits=3, shuffle=True, random_state=11)

cvscores = []
for train, test in kfold.split(X_train, y_train):
    model = build(lr=lr,
                  init= init,
                  activ= activ,
                  optim=optim,
                  input_shape= input_shape)

    model.fit(X_train[train], y_train[train], epochs=epochs, batch_size=batch_size,
    scores = model.evaluate(X_train[test], y_train[test], verbose=0)
    print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
    cvscores.append(scores[1] * 100)
    K.clear_session()
    del model

print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 224, 224, 64)      1792

 max_pooling2d (MaxPooling2D  (None, 112, 112, 64)     0
 )

 dropout (Dropout)           (None, 112, 112, 64)      0

 conv2d_1 (Conv2D)           (None, 112, 112, 64)      36928

 max_pooling2d_1 (MaxPooling  (None, 56, 56, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 56, 56, 64)        0

 flatten (Flatten)           (None, 200704)            0

 dense (Dense)               (None, 128)               25690240

 dense_1 (Dense)             (None, 2)                 258

=================================================================
Total params: 25,729,218
Trainable params: 25,729,218
Non-trainable params: 0
_________________________________________________________________
Epoch 1/5
3/3 [==============================] - 10s 2s/step - loss: 1.3326 - accuracy: 0.57
14
Epoch 2/5
3/3 [==============================] - 8s 2s/step - loss: 1.2247 - accuracy: 0.518
8
Epoch 3/5
3/3 [==============================] - 8s 2s/step - loss: 1.2222 - accuracy: 0.488
7
Epoch 4/5
3/3 [==============================] - 9s 2s/step - loss: 1.2725 - accuracy: 0.518
8
Epoch 5/5
3/3 [==============================] - 8s 2s/step - loss: 1.0865 - accuracy: 0.511
3
accuracy: 52.24%
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 224, 224, 64)      1792

 max_pooling2d (MaxPooling2D  (None, 112, 112, 64)     0
 )

 dropout (Dropout)           (None, 112, 112, 64)      0

 conv2d_1 (Conv2D)           (None, 112, 112, 64)      36928

 max_pooling2d_1 (MaxPooling  (None, 56, 56, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 56, 56, 64)        0

 flatten (Flatten)           (None, 200704)            0
```

```
 dense (Dense)               (None, 128)               25690240

 dense_1 (Dense)             (None, 2)                 258

=================================================================
Total params: 25,729,218
Trainable params: 25,729,218
Non-trainable params: 0
_________________________________________________________________
Epoch 1/5
3/3 [==============================] - 10s 2s/step - loss: 0.9418 - accuracy: 0.48
87
Epoch 2/5
3/3 [==============================] - 8s 2s/step - loss: 0.9604 - accuracy: 0.594
0
Epoch 3/5
3/3 [==============================] - 8s 2s/step - loss: 0.9396 - accuracy: 0.533
8
Epoch 4/5
3/3 [==============================] - 8s 2s/step - loss: 0.9063 - accuracy: 0.518
8
Epoch 5/5
3/3 [==============================] - 8s 2s/step - loss: 0.9199 - accuracy: 0.533
8
accuracy: 52.24%
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 224, 224, 64)      1792

 max_pooling2d (MaxPooling2D  (None, 112, 112, 64)     0
 )

 dropout (Dropout)           (None, 112, 112, 64)      0

 conv2d_1 (Conv2D)           (None, 112, 112, 64)      36928

 max_pooling2d_1 (MaxPooling  (None, 56, 56, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 56, 56, 64)        0

 flatten (Flatten)           (None, 200704)            0

 dense (Dense)               (None, 128)               25690240

 dense_1 (Dense)             (None, 2)                 258

=================================================================
Total params: 25,729,218
Trainable params: 25,729,218
Non-trainable params: 0
_________________________________________________________________
Epoch 1/5
3/3 [==============================] - 10s 2s/step - loss: 2.1285 - accuracy: 0.55
97
Epoch 2/5
3/3 [==============================] - 9s 2s/step - loss: 1.3042 - accuracy: 0.514
9
Epoch 3/5
3/3 [==============================] - 8s 2s/step - loss: 1.4246 - accuracy: 0.582
1
```

```
Epoch 4/5
3/3 [==============================] - 9s 2s/step - loss: 1.5070 - accuracy: 0.477
6
Epoch 5/5
3/3 [==============================] - 8s 2s/step - loss: 1.1514 - accuracy: 0.619
4
accuracy: 48.48%
50.99% (+/- 1.77%)
```

In [14]:

```
model = build(lr=lr,
              init= init,
              activ= activ,
              optim=optim,
              input_shape= input_shape)

model.fit(X_train, y_train,
          epochs=epochs, batch_size= batch_size, verbose=1,
          callbacks=[learning_rate_reduction]
         )

y_predict = model.predict_classes(X_test)

print(accuracy_score(np.argmax(y_test, axis=1),y_predict))
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 224, 224, 64)      1792

 max_pooling2d (MaxPooling2D  (None, 112, 112, 64)     0
 )

 dropout (Dropout)           (None, 112, 112, 64)      0

 conv2d_1 (Conv2D)           (None, 112, 112, 64)      36928

 max_pooling2d_1 (MaxPooling  (None, 56, 56, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 56, 56, 64)        0

 flatten (Flatten)           (None, 200704)            0

 dense (Dense)               (None, 128)               25690240

 dense_1 (Dense)             (None, 2)                 258

=================================================================
Total params: 25,729,218
Trainable params: 25,729,218
Non-trainable params: 0
_________________________________________________________________
Epoch 1/5
4/4 [==============================] - ETA: 0s - loss: 1.1890 - accuracy: 0.4650WA
RNING:tensorflow:Learning rate reduction is conditioned on metric `val_accuracy` w
hich is not available. Available metrics are: loss,accuracy,lr
4/4 [==============================] - 13s 3s/step - loss: 1.1890 - accuracy: 0.46
50 - lr: 1.0000e-05
Epoch 2/5
4/4 [==============================] - ETA: 0s - loss: 1.1305 - accuracy: 0.4850WA
RNING:tensorflow:Learning rate reduction is conditioned on metric `val_accuracy` w
hich is not available. Available metrics are: loss,accuracy,lr
4/4 [==============================] - 12s 3s/step - loss: 1.1305 - accuracy: 0.48
50 - lr: 1.0000e-05
Epoch 3/5
4/4 [==============================] - ETA: 0s - loss: 1.0075 - accuracy: 0.5100WA
RNING:tensorflow:Learning rate reduction is conditioned on metric `val_accuracy` w
hich is not available. Available metrics are: loss,accuracy,lr
4/4 [==============================] - 12s 3s/step - loss: 1.0075 - accuracy: 0.51
00 - lr: 1.0000e-05
Epoch 4/5
4/4 [==============================] - ETA: 0s - loss: 0.9106 - accuracy: 0.6250WA
RNING:tensorflow:Learning rate reduction is conditioned on metric `val_accuracy` w
hich is not available. Available metrics are: loss,accuracy,lr
4/4 [==============================] - 11s 3s/step - loss: 0.9106 - accuracy: 0.62
50 - lr: 1.0000e-05
Epoch 5/5
4/4 [==============================] - ETA: 0s - loss: 0.8984 - accuracy: 0.6400WA
RNING:tensorflow:Learning rate reduction is conditioned on metric `val_accuracy` w
hich is not available. Available metrics are: loss,accuracy,lr
4/4 [==============================] - 11s 3s/step - loss: 0.8984 - accuracy: 0.64
00 - lr: 1.0000e-05
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [14], in <cell line: 12>()
      1 model = build(lr=lr,
      2               init= init,
      3               activ= activ,
      4               optim=optim,
      5               input_shape= input_shape)
      7 model.fit(X_train, y_train,
      8           epochs=epochs, batch_size= batch_size, verbose=1,
      9           callbacks=[learning_rate_reduction]
     10          )
---> 12 y_predict = model.predict_classes(X_test)
     14 print(accuracy_score(np.argmax(y_test, axis=1),y_predict))

AttributeError: 'Sequential' object has no attribute 'predict_classes'
```

In [15]:
```
model_json = model.to_json()

with open("model.json", "w") as json_file:
    json_file.write(model_json)

model.save_weights("model.h5")
print("Saved model to disk")

del model
K.clear_session()
```

```
Saved model to disk
```

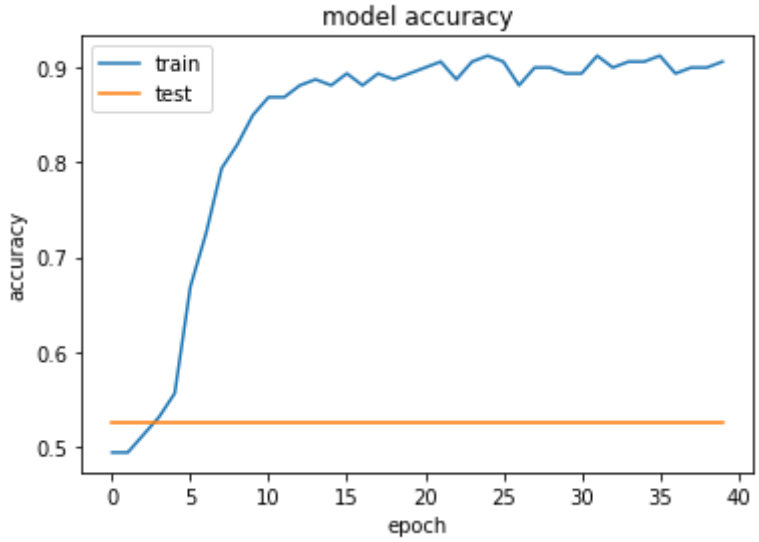
In [24]:
```
input_shape = (224,224,3)
lr = 1e-5
epochs = 40
batch_size = 64

model = ResNet50(include_top=True,
                 weights= None,
                 input_tensor=None,
                 input_shape=input_shape,
                 pooling='avg',
                 classes=2)

model.compile(optimizer = Adam(lr) ,
              loss = "binary_crossentropy",
              metrics=["accuracy"])

history = model.fit(X_train, y_train, validation_split=0.2,
                    epochs= epochs, batch_size= batch_size, verbose=1,
                    callbacks=[learning_rate_reduction]
                   )

print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
```

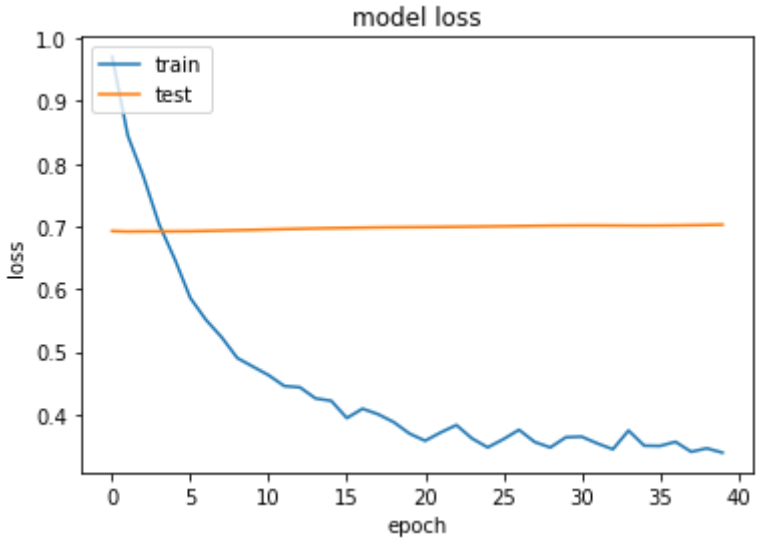
```
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

```
Epoch 1/40
3/3 [==============================] - 94s 24s/step - loss: 0.9710 - accuracy: 0.4
938 - val_loss: 0.6926 - val_accuracy: 0.5250 - lr: 1.0000e-05
Epoch 2/40
3/3 [==============================] - 72s 22s/step - loss: 0.8450 - accuracy: 0.4
938 - val_loss: 0.6919 - val_accuracy: 0.5250 - lr: 1.0000e-05
Epoch 3/40
3/3 [==============================] - 71s 22s/step - loss: 0.7796 - accuracy: 0.5
125 - val_loss: 0.6921 - val_accuracy: 0.5250 - lr: 1.0000e-05
Epoch 4/40
3/3 [==============================] - 72s 23s/step - loss: 0.7034 - accuracy: 0.5
312 - val_loss: 0.6921 - val_accuracy: 0.5250 - lr: 1.0000e-05
Epoch 5/40
3/3 [==============================] - 72s 23s/step - loss: 0.6479 - accuracy: 0.5
562 - val_loss: 0.6922 - val_accuracy: 0.5250 - lr: 1.0000e-05
Epoch 6/40
3/3 [==============================] - ETA: 0s - loss: 0.5857 - accuracy: 0.6687
Epoch 6: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-06.
3/3 [==============================] - 71s 22s/step - loss: 0.5857 - accuracy: 0.6
687 - val_loss: 0.6924 - val_accuracy: 0.5250 - lr: 1.0000e-05
Epoch 7/40
3/3 [==============================] - 71s 22s/step - loss: 0.5508 - accuracy: 0.7
250 - val_loss: 0.6929 - val_accuracy: 0.5250 - lr: 5.0000e-06
Epoch 8/40
3/3 [==============================] - 73s 23s/step - loss: 0.5232 - accuracy: 0.7
937 - val_loss: 0.6934 - val_accuracy: 0.5250 - lr: 5.0000e-06
Epoch 9/40
3/3 [==============================] - 75s 23s/step - loss: 0.4900 - accuracy: 0.8
188 - val_loss: 0.6939 - val_accuracy: 0.5250 - lr: 5.0000e-06
Epoch 10/40
3/3 [==============================] - 72s 22s/step - loss: 0.4763 - accuracy: 0.8
500 - val_loss: 0.6945 - val_accuracy: 0.5250 - lr: 5.0000e-06
Epoch 11/40
3/3 [==============================] - ETA: 0s - loss: 0.4626 - accuracy: 0.8687
Epoch 11: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-06.
3/3 [==============================] - 71s 23s/step - loss: 0.4626 - accuracy: 0.8
687 - val_loss: 0.6951 - val_accuracy: 0.5250 - lr: 5.0000e-06
Epoch 12/40
3/3 [==============================] - 71s 22s/step - loss: 0.4454 - accuracy: 0.8
687 - val_loss: 0.6957 - val_accuracy: 0.5250 - lr: 2.5000e-06
Epoch 13/40
3/3 [==============================] - 72s 23s/step - loss: 0.4433 - accuracy: 0.8
813 - val_loss: 0.6963 - val_accuracy: 0.5250 - lr: 2.5000e-06
Epoch 14/40
3/3 [==============================] - 71s 22s/step - loss: 0.4256 - accuracy: 0.8
875 - val_loss: 0.6968 - val_accuracy: 0.5250 - lr: 2.5000e-06
Epoch 15/40
3/3 [==============================] - 70s 22s/step - loss: 0.4218 - accuracy: 0.8
813 - val_loss: 0.6973 - val_accuracy: 0.5250 - lr: 2.5000e-06
Epoch 16/40
3/3 [==============================] - ETA: 0s - loss: 0.3942 - accuracy: 0.8938
Epoch 16: ReduceLROnPlateau reducing learning rate to 1.249999968422344e-06.
3/3 [==============================] - 72s 22s/step - loss: 0.3942 - accuracy: 0.8
938 - val_loss: 0.6976 - val_accuracy: 0.5250 - lr: 2.5000e-06
Epoch 17/40
3/3 [==============================] - 72s 23s/step - loss: 0.4091 - accuracy: 0.8
813 - val_loss: 0.6980 - val_accuracy: 0.5250 - lr: 1.2500e-06
Epoch 18/40
3/3 [==============================] - 71s 22s/step - loss: 0.4003 - accuracy: 0.8
938 - val_loss: 0.6984 - val_accuracy: 0.5250 - lr: 1.2500e-06
Epoch 19/40
3/3 [==============================] - 70s 22s/step - loss: 0.3877 - accuracy: 0.8
875 - val_loss: 0.6986 - val_accuracy: 0.5250 - lr: 1.2500e-06
Epoch 20/40
```

```
3/3 [==============================] - 70s 22s/step - loss: 0.3695 - accuracy: 0.8
938 - val_loss: 0.6988 - val_accuracy: 0.5250 - lr: 1.2500e-06
Epoch 21/40
3/3 [==============================] - ETA: 0s - loss: 0.3580 - accuracy: 0.9000
Epoch 21: ReduceLROnPlateau reducing learning rate to 6.24999984211172e-07.
3/3 [==============================] - 71s 22s/step - loss: 0.3580 - accuracy: 0.9
000 - val_loss: 0.6990 - val_accuracy: 0.5250 - lr: 1.2500e-06
Epoch 22/40
3/3 [==============================] - 70s 22s/step - loss: 0.3713 - accuracy: 0.9
062 - val_loss: 0.6992 - val_accuracy: 0.5250 - lr: 6.2500e-07
Epoch 23/40
3/3 [==============================] - 71s 22s/step - loss: 0.3831 - accuracy: 0.8
875 - val_loss: 0.6995 - val_accuracy: 0.5250 - lr: 6.2500e-07
Epoch 24/40
3/3 [==============================] - 71s 22s/step - loss: 0.3617 - accuracy: 0.9
062 - val_loss: 0.6997 - val_accuracy: 0.5250 - lr: 6.2500e-07
Epoch 25/40
3/3 [==============================] - 72s 22s/step - loss: 0.3474 - accuracy: 0.9
125 - val_loss: 0.7000 - val_accuracy: 0.5250 - lr: 6.2500e-07
Epoch 26/40
3/3 [==============================] - ETA: 0s - loss: 0.3605 - accuracy: 0.9062
Epoch 26: ReduceLROnPlateau reducing learning rate to 3.12499992105586e-07.
3/3 [==============================] - 71s 22s/step - loss: 0.3605 - accuracy: 0.9
062 - val_loss: 0.7003 - val_accuracy: 0.5250 - lr: 6.2500e-07
Epoch 27/40
3/3 [==============================] - 70s 22s/step - loss: 0.3755 - accuracy: 0.8
813 - val_loss: 0.7006 - val_accuracy: 0.5250 - lr: 3.1250e-07
Epoch 28/40
3/3 [==============================] - 71s 22s/step - loss: 0.3561 - accuracy: 0.9
000 - val_loss: 0.7009 - val_accuracy: 0.5250 - lr: 3.1250e-07
Epoch 29/40
3/3 [==============================] - 70s 22s/step - loss: 0.3473 - accuracy: 0.9
000 - val_loss: 0.7012 - val_accuracy: 0.5250 - lr: 3.1250e-07
Epoch 30/40
3/3 [==============================] - 74s 22s/step - loss: 0.3636 - accuracy: 0.8
938 - val_loss: 0.7014 - val_accuracy: 0.5250 - lr: 3.1250e-07
Epoch 31/40
3/3 [==============================] - ETA: 0s - loss: 0.3646 - accuracy: 0.8938
Epoch 31: ReduceLROnPlateau reducing learning rate to 1.56249996052793e-07.
3/3 [==============================] - 73s 23s/step - loss: 0.3646 - accuracy: 0.8
938 - val_loss: 0.7016 - val_accuracy: 0.5250 - lr: 3.1250e-07
Epoch 32/40
3/3 [==============================] - 72s 22s/step - loss: 0.3538 - accuracy: 0.9
125 - val_loss: 0.7016 - val_accuracy: 0.5250 - lr: 1.5625e-07
Epoch 33/40
3/3 [==============================] - 71s 22s/step - loss: 0.3445 - accuracy: 0.9
000 - val_loss: 0.7015 - val_accuracy: 0.5250 - lr: 1.5625e-07
Epoch 34/40
3/3 [==============================] - 72s 22s/step - loss: 0.3743 - accuracy: 0.9
062 - val_loss: 0.7014 - val_accuracy: 0.5250 - lr: 1.5625e-07
Epoch 35/40
3/3 [==============================] - 72s 22s/step - loss: 0.3502 - accuracy: 0.9
062 - val_loss: 0.7013 - val_accuracy: 0.5250 - lr: 1.5625e-07
Epoch 36/40
3/3 [==============================] - ETA: 0s - loss: 0.3497 - accuracy: 0.9125
Epoch 36: ReduceLROnPlateau reducing learning rate to 1e-07.
3/3 [==============================] - 71s 22s/step - loss: 0.3497 - accuracy: 0.9
125 - val_loss: 0.7014 - val_accuracy: 0.5250 - lr: 1.5625e-07
Epoch 37/40
3/3 [==============================] - 70s 22s/step - loss: 0.3564 - accuracy: 0.8
938 - val_loss: 0.7017 - val_accuracy: 0.5250 - lr: 1.0000e-07
Epoch 38/40
3/3 [==============================] - 70s 22s/step - loss: 0.3405 - accuracy: 0.9
000 - val_loss: 0.7020 - val_accuracy: 0.5250 - lr: 1.0000e-07
```

```
Epoch 39/40
3/3 [==============================] - 70s 22s/step - loss: 0.3460 - accuracy: 0.9
000 - val_loss: 0.7024 - val_accuracy: 0.5250 - lr: 1.0000e-07
Epoch 40/40
3/3 [==============================] - 69s 22s/step - loss: 0.3391 - accuracy: 0.9
062 - val_loss: 0.7029 - val_accuracy: 0.5250 - lr: 1.0000e-07
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy', 'lr'])
```





In [25]:
```
y_pred = model.predict(X_test)
print(accuracy_score(np.argmax(y_test, axis=1), np.argmax(y_pred, axis=1)))

resnet50_json = model.to_json()

with open("resnet50.json", "w") as json_file:
    json_file.write(resnet50_json)

model.save_weights("resnet50.h5")
print("Saved model to disk")
```

```
0.5
Saved model to disk
```

In [26]:
```
del model
K.clear_session()
```

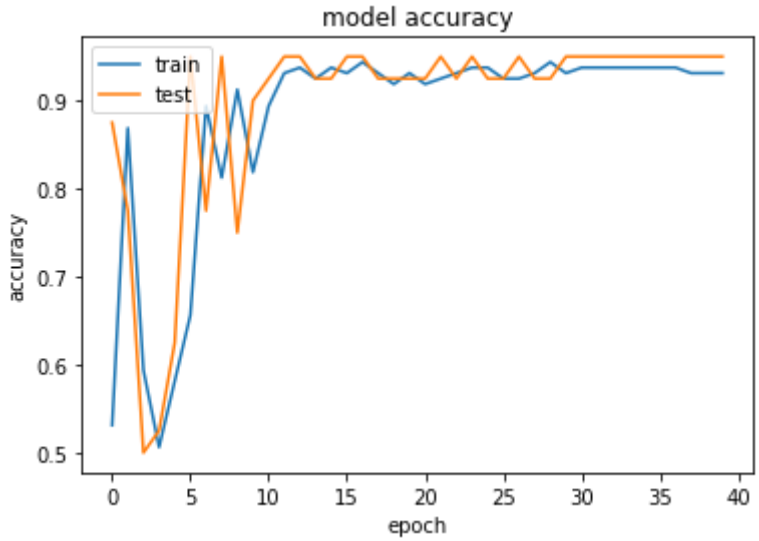
In [27]:
```
from keras.applications.vgg16 import VGG16
from keras.preprocessing import image
from keras.applications.vgg16 import preprocess_input
import numpy as np
```

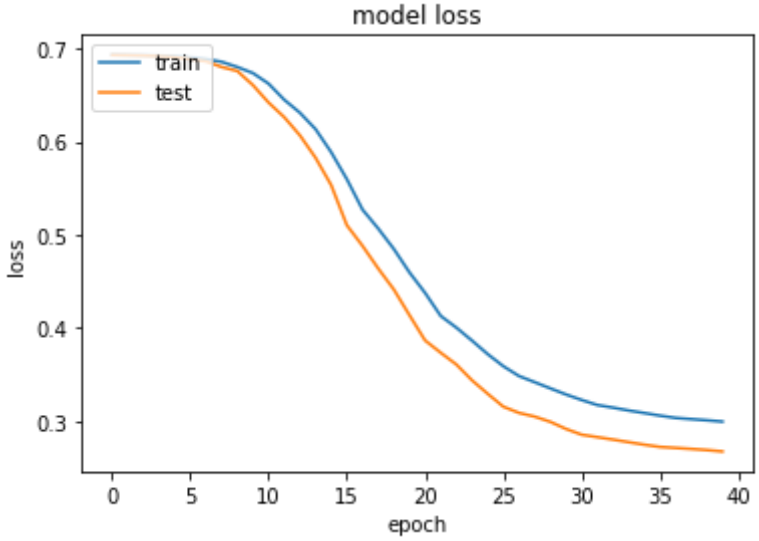
```
input_shape = (224,224,3)
lr = 1e-5
epochs = 40
batch_size = 64

model = VGG16(include_top=True,
                 weights= None,
                 input_tensor=None,
                 input_shape=input_shape,
                 pooling='avg',
                 classes=2)

model.compile(optimizer = Adam(lr) ,
              loss = "binary_crossentropy",
              metrics=["accuracy"])

history = model.fit(X_train, y_train, validation_split=0.2,
                    epochs= epochs, batch_size= batch_size, verbose=1,
                    callbacks=[learning_rate_reduction]
                   )

print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

```
Epoch 1/40
3/3 [==============================] - 174s 54s/step - loss: 0.6932 - accuracy: 0.
5312 - val_loss: 0.6929 - val_accuracy: 0.8750 - lr: 1.0000e-05
Epoch 2/40
3/3 [==============================] - 165s 51s/step - loss: 0.6929 - accuracy: 0.
8687 - val_loss: 0.6926 - val_accuracy: 0.7750 - lr: 1.0000e-05
Epoch 3/40
3/3 [==============================] - 168s 53s/step - loss: 0.6927 - accuracy: 0.
5938 - val_loss: 0.6922 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 4/40
3/3 [==============================] - 166s 52s/step - loss: 0.6922 - accuracy: 0.
5063 - val_loss: 0.6916 - val_accuracy: 0.5250 - lr: 1.0000e-05
Epoch 5/40
3/3 [==============================] - 169s 54s/step - loss: 0.6915 - accuracy: 0.
5813 - val_loss: 0.6905 - val_accuracy: 0.6250 - lr: 1.0000e-05
Epoch 6/40
3/3 [==============================] - 140s 39s/step - loss: 0.6903 - accuracy: 0.
6562 - val_loss: 0.6884 - val_accuracy: 0.9500 - lr: 1.0000e-05
Epoch 7/40
3/3 [==============================] - 105s 32s/step - loss: 0.6885 - accuracy: 0.
8938 - val_loss: 0.6861 - val_accuracy: 0.7750 - lr: 1.0000e-05
Epoch 8/40
3/3 [==============================] - 111s 34s/step - loss: 0.6856 - accuracy: 0.
8125 - val_loss: 0.6800 - val_accuracy: 0.9500 - lr: 1.0000e-05
Epoch 9/40
3/3 [==============================] - 129s 38s/step - loss: 0.6800 - accuracy: 0.
9125 - val_loss: 0.6757 - val_accuracy: 0.7500 - lr: 1.0000e-05
Epoch 10/40
3/3 [==============================] - 126s 40s/step - loss: 0.6734 - accuracy: 0.
8188 - val_loss: 0.6605 - val_accuracy: 0.9000 - lr: 1.0000e-05
Epoch 11/40
3/3 [==============================] - ETA: 0s - loss: 0.6621 - accuracy: 0.8938
Epoch 11: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-06.
3/3 [==============================] - 116s 35s/step - loss: 0.6621 - accuracy: 0.
8938 - val_loss: 0.6418 - val_accuracy: 0.9250 - lr: 1.0000e-05
Epoch 12/40
3/3 [==============================] - 112s 35s/step - loss: 0.6449 - accuracy: 0.
9312 - val_loss: 0.6262 - val_accuracy: 0.9500 - lr: 5.0000e-06
Epoch 13/40
3/3 [==============================] - 112s 35s/step - loss: 0.6310 - accuracy: 0.
9375 - val_loss: 0.6067 - val_accuracy: 0.9500 - lr: 5.0000e-06
Epoch 14/40
3/3 [==============================] - 113s 35s/step - loss: 0.6134 - accuracy: 0.
9250 - val_loss: 0.5826 - val_accuracy: 0.9250 - lr: 5.0000e-06
Epoch 15/40
3/3 [==============================] - 109s 34s/step - loss: 0.5886 - accuracy: 0.
9375 - val_loss: 0.5533 - val_accuracy: 0.9250 - lr: 5.0000e-06
Epoch 16/40
3/3 [==============================] - ETA: 0s - loss: 0.5597 - accuracy: 0.9312
Epoch 16: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-06.
3/3 [==============================] - 122s 39s/step - loss: 0.5597 - accuracy: 0.
9312 - val_loss: 0.5103 - val_accuracy: 0.9500 - lr: 5.0000e-06
Epoch 17/40
3/3 [==============================] - 127s 38s/step - loss: 0.5271 - accuracy: 0.
9438 - val_loss: 0.4882 - val_accuracy: 0.9500 - lr: 2.5000e-06
Epoch 18/40
3/3 [==============================] - 165s 53s/step - loss: 0.5068 - accuracy: 0.
9312 - val_loss: 0.4640 - val_accuracy: 0.9250 - lr: 2.5000e-06
Epoch 19/40
3/3 [==============================] - 126s 38s/step - loss: 0.4849 - accuracy: 0.
9187 - val_loss: 0.4414 - val_accuracy: 0.9250 - lr: 2.5000e-06
Epoch 20/40
3/3 [==============================] - 124s 38s/step - loss: 0.4597 - accuracy: 0.
9312 - val_loss: 0.4139 - val_accuracy: 0.9250 - lr: 2.5000e-06
```

```
Epoch 21/40
3/3 [==============================] - ETA: 0s - loss: 0.4377 - accuracy: 0.9187
Epoch 21: ReduceLROnPlateau reducing learning rate to 1.249999968422344e-06.
3/3 [==============================] - 129s 40s/step - loss: 0.4377 - accuracy: 0.
9187 - val_loss: 0.3867 - val_accuracy: 0.9250 - lr: 2.5000e-06
Epoch 22/40
3/3 [==============================] - 122s 37s/step - loss: 0.4130 - accuracy: 0.
9250 - val_loss: 0.3736 - val_accuracy: 0.9500 - lr: 1.2500e-06
Epoch 23/40
3/3 [==============================] - 118s 36s/step - loss: 0.4005 - accuracy: 0.
9312 - val_loss: 0.3609 - val_accuracy: 0.9250 - lr: 1.2500e-06
Epoch 24/40
3/3 [==============================] - 122s 37s/step - loss: 0.3865 - accuracy: 0.
9375 - val_loss: 0.3440 - val_accuracy: 0.9500 - lr: 1.2500e-06
Epoch 25/40
3/3 [==============================] - 117s 36s/step - loss: 0.3721 - accuracy: 0.
9375 - val_loss: 0.3296 - val_accuracy: 0.9250 - lr: 1.2500e-06
Epoch 26/40
3/3 [==============================] - ETA: 0s - loss: 0.3591 - accuracy: 0.9250
Epoch 26: ReduceLROnPlateau reducing learning rate to 6.24999984211172e-07.
3/3 [==============================] - 120s 37s/step - loss: 0.3591 - accuracy: 0.
9250 - val_loss: 0.3157 - val_accuracy: 0.9250 - lr: 1.2500e-06
Epoch 27/40
3/3 [==============================] - 180s 66s/step - loss: 0.3486 - accuracy: 0.
9250 - val_loss: 0.3091 - val_accuracy: 0.9500 - lr: 6.2500e-07
Epoch 28/40
3/3 [==============================] - 125s 37s/step - loss: 0.3420 - accuracy: 0.
9312 - val_loss: 0.3051 - val_accuracy: 0.9250 - lr: 6.2500e-07
Epoch 29/40
3/3 [==============================] - 195s 60s/step - loss: 0.3354 - accuracy: 0.
9438 - val_loss: 0.2995 - val_accuracy: 0.9250 - lr: 6.2500e-07
Epoch 30/40
3/3 [==============================] - 160s 55s/step - loss: 0.3289 - accuracy: 0.
9312 - val_loss: 0.2919 - val_accuracy: 0.9500 - lr: 6.2500e-07
Epoch 31/40
3/3 [==============================] - ETA: 0s - loss: 0.3229 - accuracy: 0.9375
Epoch 31: ReduceLROnPlateau reducing learning rate to 3.12499992105586e-07.
3/3 [==============================] - 182s 65s/step - loss: 0.3229 - accuracy: 0.
9375 - val_loss: 0.2857 - val_accuracy: 0.9500 - lr: 6.2500e-07
Epoch 32/40
3/3 [==============================] - 223s 65s/step - loss: 0.3175 - accuracy: 0.
9375 - val_loss: 0.2831 - val_accuracy: 0.9500 - lr: 3.1250e-07
Epoch 33/40
3/3 [==============================] - 232s 84s/step - loss: 0.3147 - accuracy: 0.
9375 - val_loss: 0.2805 - val_accuracy: 0.9500 - lr: 3.1250e-07
Epoch 34/40
3/3 [==============================] - 263s 70s/step - loss: 0.3117 - accuracy: 0.
9375 - val_loss: 0.2777 - val_accuracy: 0.9500 - lr: 3.1250e-07
Epoch 35/40
3/3 [==============================] - 189s 56s/step - loss: 0.3090 - accuracy: 0.
9375 - val_loss: 0.2750 - val_accuracy: 0.9500 - lr: 3.1250e-07
Epoch 36/40
3/3 [==============================] - ETA: 0s - loss: 0.3061 - accuracy: 0.9375
Epoch 36: ReduceLROnPlateau reducing learning rate to 1.56249996052793e-07.
3/3 [==============================] - 124s 38s/step - loss: 0.3061 - accuracy: 0.
9375 - val_loss: 0.2726 - val_accuracy: 0.9500 - lr: 3.1250e-07
Epoch 37/40
3/3 [==============================] - 200s 65s/step - loss: 0.3037 - accuracy: 0.
9375 - val_loss: 0.2715 - val_accuracy: 0.9500 - lr: 1.5625e-07
Epoch 38/40
3/3 [==============================] - 166s 40s/step - loss: 0.3024 - accuracy: 0.
9312 - val_loss: 0.2704 - val_accuracy: 0.9500 - lr: 1.5625e-07
Epoch 39/40
3/3 [==============================] - 118s 37s/step - loss: 0.3011 - accuracy: 0.
```

```
9312 - val_loss: 0.2692 - val_accuracy: 0.9500 - lr: 1.5625e-07
Epoch 40/40
3/3 [==============================] - 210s 67s/step - loss: 0.2997 - accuracy: 0.
9312 - val_loss: 0.2677 - val_accuracy: 0.9500 - lr: 1.5625e-07
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy', 'lr'])
```





In [28]:
```
y_pred = model.predict(X_test)
print(accuracy_score(np.argmax(y_test, axis=1), np.argmax(y_pred, axis=1)))

vgg16_json = model.to_json()

with open("vgg16.json", "w") as json_file:
    json_file.write(vgg16_json)

model.save_weights("vgg16.h5")
print("Saved model to disk")

del model
K.clear_session()
```

```
0.805
Saved model to disk
```

In [ ]:

In [ ]:

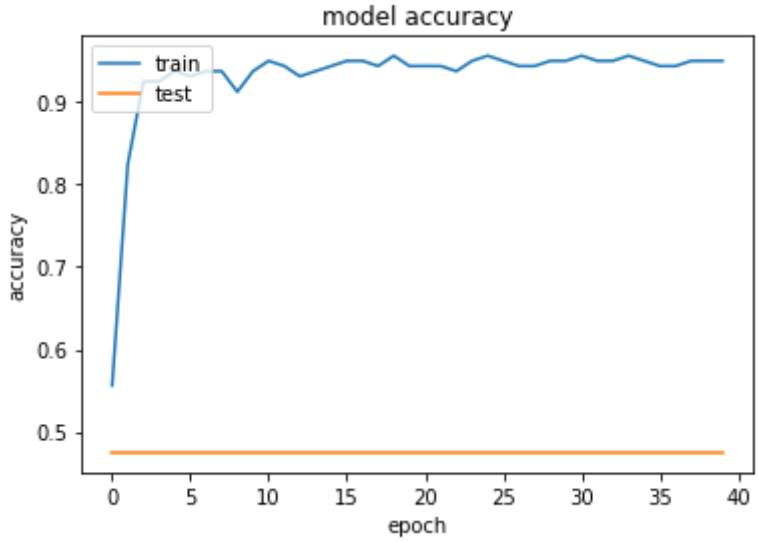
In [29]:
```
from keras.applications.xception import Xception
```

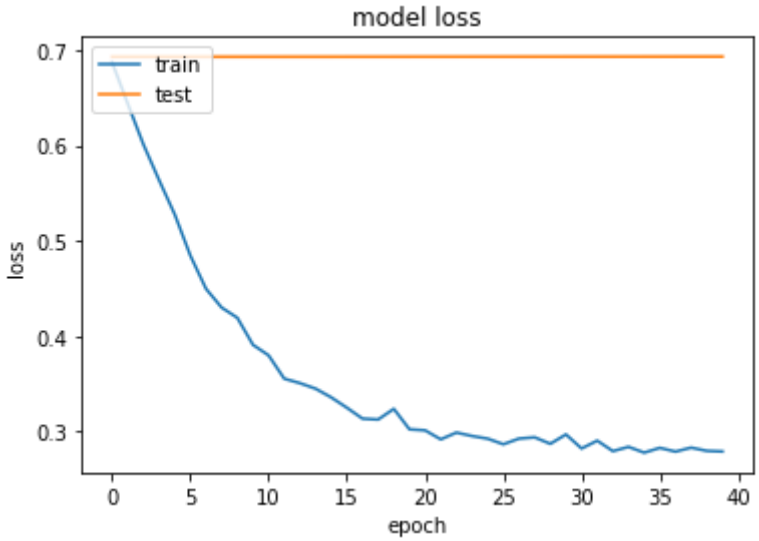
```
input_shape = (224,224,3)
lr = 1e-5
epochs = 40
batch_size = 64

model = Xception(include_top=True,
                 weights= None,
                 input_tensor=None,
                 input_shape=input_shape,
                 pooling='avg',
                 classes=2)

model.compile(optimizer = Adam(lr) ,
              loss = "binary_crossentropy",
              metrics=["accuracy"])

history = model.fit(X_train, y_train, validation_split=0.2,
                    epochs= epochs, batch_size= batch_size, verbose=1,
                    callbacks=[learning_rate_reduction]
                   )

print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

```
Epoch 1/40
3/3 [==============================] - 524s 128s/step - loss: 0.6872 - accuracy:
0.5562 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.0000e-05
Epoch 2/40
3/3 [==============================] - 354s 97s/step - loss: 0.6438 - accuracy: 0.
8250 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.0000e-05
Epoch 3/40
3/3 [==============================] - 397s 103s/step - loss: 0.6011 - accuracy:
0.9250 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.0000e-05
Epoch 4/40
3/3 [==============================] - 339s 84s/step - loss: 0.5638 - accuracy: 0.
9250 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.0000e-05
Epoch 5/40
3/3 [==============================] - 354s 90s/step - loss: 0.5278 - accuracy: 0.
9375 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.0000e-05
Epoch 6/40
3/3 [==============================] - ETA: 0s - loss: 0.4845 - accuracy: 0.9312
Epoch 6: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-06.
3/3 [==============================] - 366s 93s/step - loss: 0.4845 - accuracy: 0.
9312 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.0000e-05
Epoch 7/40
3/3 [==============================] - 361s 96s/step - loss: 0.4497 - accuracy: 0.
9375 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 5.0000e-06
Epoch 8/40
3/3 [==============================] - 315s 85s/step - loss: 0.4300 - accuracy: 0.
9375 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 5.0000e-06
Epoch 9/40
3/3 [==============================] - 352s 89s/step - loss: 0.4193 - accuracy: 0.
9125 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 5.0000e-06
Epoch 10/40
3/3 [==============================] - 355s 89s/step - loss: 0.3909 - accuracy: 0.
9375 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 5.0000e-06
Epoch 11/40
3/3 [==============================] - ETA: 0s - loss: 0.3798 - accuracy: 0.9500
Epoch 11: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-06.
3/3 [==============================] - 382s 109s/step - loss: 0.3798 - accuracy:
0.9500 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 5.0000e-06
Epoch 12/40
3/3 [==============================] - 360s 85s/step - loss: 0.3554 - accuracy: 0.
9438 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 2.5000e-06
Epoch 13/40
3/3 [==============================] - 351s 90s/step - loss: 0.3507 - accuracy: 0.
9312 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 2.5000e-06
Epoch 14/40
3/3 [==============================] - 345s 92s/step - loss: 0.3449 - accuracy: 0.
9375 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 2.5000e-06
Epoch 15/40
3/3 [==============================] - 297s 79s/step - loss: 0.3358 - accuracy: 0.
9438 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 2.5000e-06
Epoch 16/40
3/3 [==============================] - ETA: 0s - loss: 0.3249 - accuracy: 0.9500
Epoch 16: ReduceLROnPlateau reducing learning rate to 1.249999968422344e-06.
3/3 [==============================] - 368s 99s/step - loss: 0.3249 - accuracy: 0.
9500 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 2.5000e-06
Epoch 17/40
3/3 [==============================] - 350s 96s/step - loss: 0.3134 - accuracy: 0.
9500 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.2500e-06
Epoch 18/40
3/3 [==============================] - 350s 91s/step - loss: 0.3125 - accuracy: 0.
9438 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.2500e-06
Epoch 19/40
3/3 [==============================] - 306s 89s/step - loss: 0.3237 - accuracy: 0.
9563 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.2500e-06
Epoch 20/40
```

```
3/3 [==============================] - 394s 102s/step - loss: 0.3023 - accuracy:
0.9438 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.2500e-06
Epoch 21/40
3/3 [==============================] - ETA: 0s - loss: 0.3012 - accuracy: 0.9438
Epoch 21: ReduceLROnPlateau reducing learning rate to 6.24999984211172e-07.
3/3 [==============================] - 407s 116s/step - loss: 0.3012 - accuracy:
0.9438 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 1.2500e-06
Epoch 22/40
3/3 [==============================] - 391s 107s/step - loss: 0.2917 - accuracy:
0.9438 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 6.2500e-07
Epoch 23/40
3/3 [==============================] - 376s 103s/step - loss: 0.2988 - accuracy:
0.9375 - val_loss: 0.6932 - val_accuracy: 0.4750 - lr: 6.2500e-07
Epoch 24/40
3/3 [==============================] - 414s 111s/step - loss: 0.2952 - accuracy:
0.9500 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 6.2500e-07
Epoch 25/40
3/3 [==============================] - 335s 97s/step - loss: 0.2924 - accuracy: 0.
9563 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 6.2500e-07
Epoch 26/40
3/3 [==============================] - ETA: 0s - loss: 0.2866 - accuracy: 0.9500
Epoch 26: ReduceLROnPlateau reducing learning rate to 3.12499992105586e-07.
3/3 [==============================] - 343s 101s/step - loss: 0.2866 - accuracy:
0.9500 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 6.2500e-07
Epoch 27/40
3/3 [==============================] - 407s 103s/step - loss: 0.2925 - accuracy:
0.9438 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 3.1250e-07
Epoch 28/40
3/3 [==============================] - 330s 89s/step - loss: 0.2938 - accuracy: 0.
9438 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 3.1250e-07
Epoch 29/40
3/3 [==============================] - 313s 78s/step - loss: 0.2870 - accuracy: 0.
9500 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 3.1250e-07
Epoch 30/40
3/3 [==============================] - 333s 76s/step - loss: 0.2969 - accuracy: 0.
9500 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 3.1250e-07
Epoch 31/40
3/3 [==============================] - ETA: 0s - loss: 0.2821 - accuracy: 0.9563
Epoch 31: ReduceLROnPlateau reducing learning rate to 1.56249996052793e-07.
3/3 [==============================] - 374s 99s/step - loss: 0.2821 - accuracy: 0.
9563 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 3.1250e-07
Epoch 32/40
3/3 [==============================] - 375s 83s/step - loss: 0.2903 - accuracy: 0.
9500 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 1.5625e-07
Epoch 33/40
3/3 [==============================] - 365s 99s/step - loss: 0.2794 - accuracy: 0.
9500 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 1.5625e-07
Epoch 34/40
3/3 [==============================] - 307s 82s/step - loss: 0.2838 - accuracy: 0.
9563 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 1.5625e-07
Epoch 35/40
3/3 [==============================] - 347s 91s/step - loss: 0.2778 - accuracy: 0.
9500 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 1.5625e-07
Epoch 36/40
3/3 [==============================] - ETA: 0s - loss: 0.2828 - accuracy: 0.9438
Epoch 36: ReduceLROnPlateau reducing learning rate to 1e-07.
3/3 [==============================] - 332s 84s/step - loss: 0.2828 - accuracy: 0.
9438 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 1.5625e-07
Epoch 37/40
3/3 [==============================] - 292s 78s/step - loss: 0.2789 - accuracy: 0.
9438 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 1.0000e-07
Epoch 38/40
3/3 [==============================] - 283s 70s/step - loss: 0.2830 - accuracy: 0.
9500 - val_loss: 0.6933 - val_accuracy: 0.4750 - lr: 1.0000e-07
```

```
Epoch 39/40
3/3 [==============================] - 284s 73s/step - loss: 0.2796 - accuracy: 0.
9500 - val_loss: 0.6934 - val_accuracy: 0.4750 - lr: 1.0000e-07
Epoch 40/40
3/3 [==============================] - 310s 86s/step - loss: 0.2790 - accuracy: 0.
9500 - val_loss: 0.6934 - val_accuracy: 0.4750 - lr: 1.0000e-07
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy', 'lr'])
```





In [30]:
```
y_pred = model.predict(X_test)
print(accuracy_score(np.argmax(y_test, axis=1), np.argmax(y_pred, axis=1)))

xception_json = model.to_json()

with open("xception.json", "w") as json_file:
    json_file.write(xception_json)

model.save_weights("xception.h5")
print("Saved model to disk")

del model
K.clear_session()
```

```
0.5
Saved model to disk
```

In [ ]: