

In [1]: !pip install keras

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

In [2]: !pip install tensorflow

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

Requirement already satisfied: termcolor>=1.1.0 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (1.1.0)

Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (4.1.1)

Requirement already satisfied: wrapt>=1.11.0 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (1.12.1)

Requirement already satisfied: tf-estimator-nightly==2.8.0.dev2021122109 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (2.8.0.dev2021122109)

Requirement already satisfied: six>=1.12.0 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (1.16.0)

Requirement already satisfied: protobuf>=3.9.2 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (3.19.1)

Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (3.3.0)

Requirement already satisfied: numpy>=1.20 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (1.21.5)

Requirement already satisfied: setuptools in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (61.2.0)

Requirement already satisfied: keras-preprocessing>=1.1.1 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (1.1.2)

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: astunparse>=1.6.0 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (1.6.3)

Requirement already satisfied: keras<2.9,>=2.8.0rc0 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (2.8.0)

Requirement already satisfied: google-pasta>=0.1.1 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (0.2.0)

Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (0.25.0)

Requirement already satisfied: flatbuffers>=1.12 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (2.0)

Requirement already satisfied: tensorboard<2.9,>=2.8 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (2.8.0)

Requirement already satisfied: gast>=0.2.1 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (0.5.3)

Requirement already satisfied: h5py>=2.9.0 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (3.6.0)

Requirement already satisfied: absl-py>=0.4.0 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (1.0.0)

Requirement already satisfied: libclang>=9.0.1 in c:\users\shadm\anaconda3\lib\site-packages (from tensorflow) (14.0.1)

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: 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\site-packages (from tensorboard<2.9,>=2.8->tensorflow) (3.3.4)

Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in c:\users\shadm\anaconda3\lib\site-packages (from tensorboard<2.9,>=2.8->tensorflow) (1.8.1)

Requirement already satisfied: google-auth<3,>=1.6.3 in c:\users\shadm\anaconda3\lib\site-packages (from tensorboard<2.9,>=2.8->tensorflow) (1.33.0)

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\site-packages (from tensorboard<2.9,>=2.8->tensorflow) (2.0.3)

Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in c:\users\shadm\anaconda3\lib\site-packages (from tensorboard<2.9,>=2.8->tensorflow) (0.6.1)

Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\shadm\anaconda3\lib\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: 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: requests-oauthlib>=0.7.0 in c:\users\shadm\anaconda3\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\lib\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: idna<4,>=2.5 in c:\users\shadm\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow) (3.3)
 Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\shadm\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow) (1.26.9)
 Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\shadm\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.9,>=2.8->tensorflow) (2.0.4)
 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) (2021.10.8)
 Requirement already satisfied: oauthlib>=3.0.0 in c:\users\shadm\anaconda3\lib\site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.9,>=2.8->tensorflow) (3.2.0)

```
In [3]: 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, GridSearchCV
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 [4]: folder_benign_train = r'C:\Users\shadm\Desktop\Project\Biomedical_Image_Processing\train\benign'
folder_malignant_train = r'C:\Users\shadm\Desktop\Project\Biomedical_Image_Processing\train\malignant'

folder_benign_test = r'C:\Users\shadm\Desktop\Project\Biomedical_Image_Processing\test\benign'
folder_malignant_test = r'C:\Users\shadm\Desktop\Project\Biomedical_Image_Processing\test\malignant'

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

```

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

ims_benign_test = [read(os.path.join(folder_benign_test, filename)) for filename in os.listdir(folder_benign_test)]
X_benign_test = np.array(ims_benign_test, dtype='uint8')
ims_malignant_test = [read(os.path.join(folder_malignant_test, filename)) for filename in os.listdir(folder_malignant_test)]
X_malignant_test = np.array(ims_malignant_test, 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]

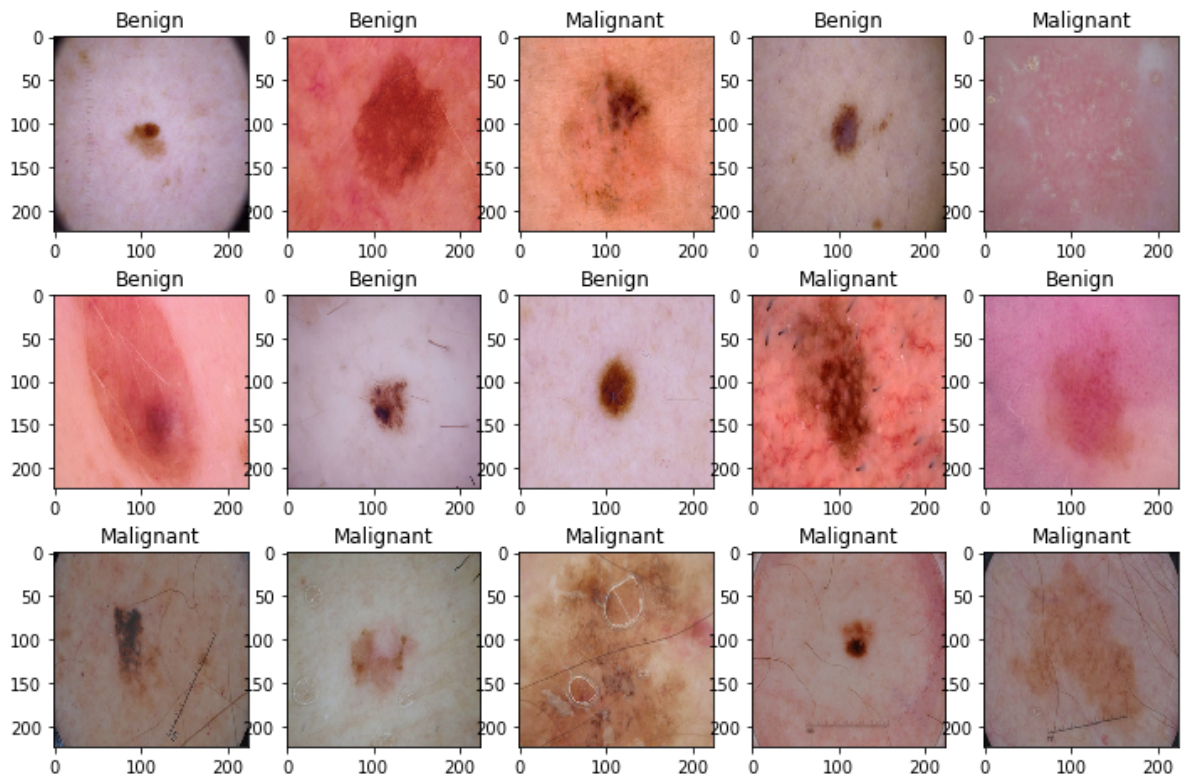
```

```

In [5]: 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 [6]: y_train = to_categorical(y_train, num_classes= 2)
        y_test = to_categorical(y_test, num_classes= 2)
```

```
In [7]: X_train = X_train/255.
        X_test = X_test/255.
```

```
In [8]: 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_shape,
                    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=["accuracy"])
    return model

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

```
In [10]: 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_shape)
```

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 (MaxPooling2D)	(None, 56, 56, 64)	0
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		

C:\Users\shadm\anaconda3\lib\site-packages\keras\optimizer_v2\adam.py:105: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
 super(Adam, self).__init__(name, **kwargs)

```
In [11]: 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
33/33 [=====] - 162s 5s/step - loss: 1.0335 - accuracy:
0.5462 - val_loss: 0.6564 - val_accuracy: 0.7121 - lr: 1.0000e-05
Epoch 2/40
33/33 [=====] - 134s 4s/step - loss: 0.9143 - accuracy:
0.5391 - val_loss: 0.6518 - val_accuracy: 0.5322 - lr: 1.0000e-05
Epoch 3/40
33/33 [=====] - 131s 4s/step - loss: 0.8371 - accuracy:
0.5799 - val_loss: 0.6402 - val_accuracy: 0.5341 - lr: 1.0000e-05
Epoch 4/40
33/33 [=====] - 129s 4s/step - loss: 0.7645 - accuracy:
0.6083 - val_loss: 0.6640 - val_accuracy: 0.5341 - lr: 1.0000e-05
Epoch 5/40
33/33 [=====] - 131s 4s/step - loss: 0.7490 - accuracy:
0.5951 - val_loss: 0.6505 - val_accuracy: 0.5341 - lr: 1.0000e-05
Epoch 6/40
33/33 [=====] - ETA: 0s - loss: 0.7083 - accuracy: 0.6126
Epoch 6: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-06.
33/33 [=====] - 133s 4s/step - loss: 0.7083 - accuracy:
0.6126 - val_loss: 0.6288 - val_accuracy: 0.5436 - lr: 1.0000e-05
Epoch 7/40
33/33 [=====] - 135s 4s/step - loss: 0.6984 - accuracy:
0.6259 - val_loss: 0.6555 - val_accuracy: 0.5341 - lr: 5.0000e-06
Epoch 8/40
33/33 [=====] - 136s 4s/step - loss: 0.6754 - accuracy:
0.6321 - val_loss: 0.6514 - val_accuracy: 0.5341 - lr: 5.0000e-06
Epoch 9/40
33/33 [=====] - 131s 4s/step - loss: 0.6714 - accuracy:
0.6358 - val_loss: 0.6841 - val_accuracy: 0.5341 - lr: 5.0000e-06
Epoch 10/40
33/33 [=====] - 137s 4s/step - loss: 0.6488 - accuracy:
0.6543 - val_loss: 0.6506 - val_accuracy: 0.5341 - lr: 5.0000e-06
Epoch 11/40
33/33 [=====] - ETA: 0s - loss: 0.6574 - accuracy: 0.6486
Epoch 11: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-06.
33/33 [=====] - 131s 4s/step - loss: 0.6574 - accuracy:
0.6486 - val_loss: 0.6644 - val_accuracy: 0.5341 - lr: 5.0000e-06
Epoch 12/40
33/33 [=====] - 130s 4s/step - loss: 0.6523 - accuracy:
0.6520 - val_loss: 0.6558 - val_accuracy: 0.5341 - lr: 2.5000e-06
Epoch 13/40
33/33 [=====] - 134s 4s/step - loss: 0.6253 - accuracy:
0.6847 - val_loss: 0.6636 - val_accuracy: 0.5341 - lr: 2.5000e-06
Epoch 14/40
33/33 [=====] - 129s 4s/step - loss: 0.6216 - accuracy:
0.6728 - val_loss: 0.6584 - val_accuracy: 0.5341 - lr: 2.5000e-06
Epoch 15/40
33/33 [=====] - 131s 4s/step - loss: 0.6362 - accuracy:
0.6724 - val_loss: 0.6620 - val_accuracy: 0.5341 - lr: 2.5000e-06
Epoch 16/40
33/33 [=====] - ETA: 0s - loss: 0.6295 - accuracy: 0.6648
Epoch 16: ReduceLROnPlateau reducing learning rate to 1.249999968422344e-06.
33/33 [=====] - 128s 4s/step - loss: 0.6295 - accuracy:
0.6648 - val_loss: 0.6646 - val_accuracy: 0.5341 - lr: 2.5000e-06
Epoch 17/40
33/33 [=====] - 136s 4s/step - loss: 0.6188 - accuracy:
0.6757 - val_loss: 0.6528 - val_accuracy: 0.5379 - lr: 1.2500e-06
Epoch 18/40
33/33 [=====] - 196s 6s/step - loss: 0.6230 - accuracy:
0.6671 - val_loss: 0.6614 - val_accuracy: 0.5341 - lr: 1.2500e-06
Epoch 19/40
33/33 [=====] - 164s 5s/step - loss: 0.6379 - accuracy:
0.6534 - val_loss: 0.6568 - val_accuracy: 0.5341 - lr: 1.2500e-06
Epoch 20/40
```

```
33/33 [=====] - 163s 5s/step - loss: 0.6094 - accuracy:
0.6700 - val_loss: 0.6652 - val_accuracy: 0.5341 - lr: 1.2500e-06
Epoch 21/40
33/33 [=====] - ETA: 0s - loss: 0.6212 - accuracy: 0.6752
Epoch 21: ReduceLROnPlateau reducing learning rate to 6.24999984211172e-07.
33/33 [=====] - 163s 5s/step - loss: 0.6212 - accuracy:
0.6752 - val_loss: 0.6637 - val_accuracy: 0.5341 - lr: 1.2500e-06
Epoch 22/40
33/33 [=====] - 162s 5s/step - loss: 0.6227 - accuracy:
0.6728 - val_loss: 0.6624 - val_accuracy: 0.5341 - lr: 6.2500e-07
Epoch 23/40
33/33 [=====] - 163s 5s/step - loss: 0.6178 - accuracy:
0.6686 - val_loss: 0.6633 - val_accuracy: 0.5341 - lr: 6.2500e-07
Epoch 24/40
33/33 [=====] - 187s 6s/step - loss: 0.6235 - accuracy:
0.6667 - val_loss: 0.6616 - val_accuracy: 0.5341 - lr: 6.2500e-07
Epoch 25/40
33/33 [=====] - 169s 5s/step - loss: 0.6221 - accuracy:
0.6671 - val_loss: 0.6590 - val_accuracy: 0.5341 - lr: 6.2500e-07
Epoch 26/40
33/33 [=====] - ETA: 0s - loss: 0.6050 - accuracy: 0.6761
Epoch 26: ReduceLROnPlateau reducing learning rate to 3.12499992105586e-07.
33/33 [=====] - 165s 5s/step - loss: 0.6050 - accuracy:
0.6761 - val_loss: 0.6620 - val_accuracy: 0.5341 - lr: 6.2500e-07
Epoch 27/40
33/33 [=====] - 164s 5s/step - loss: 0.6200 - accuracy:
0.6567 - val_loss: 0.6602 - val_accuracy: 0.5341 - lr: 3.1250e-07
Epoch 28/40
33/33 [=====] - 164s 5s/step - loss: 0.6053 - accuracy:
0.6818 - val_loss: 0.6588 - val_accuracy: 0.5341 - lr: 3.1250e-07
Epoch 29/40
33/33 [=====] - 167s 5s/step - loss: 0.6171 - accuracy:
0.6709 - val_loss: 0.6591 - val_accuracy: 0.5341 - lr: 3.1250e-07
Epoch 30/40
33/33 [=====] - 165s 5s/step - loss: 0.5994 - accuracy:
0.6818 - val_loss: 0.6578 - val_accuracy: 0.5341 - lr: 3.1250e-07
Epoch 31/40
33/33 [=====] - ETA: 0s - loss: 0.6232 - accuracy: 0.6690
Epoch 31: ReduceLROnPlateau reducing learning rate to 1.56249996052793e-07.
33/33 [=====] - 167s 5s/step - loss: 0.6232 - accuracy:
0.6690 - val_loss: 0.6578 - val_accuracy: 0.5341 - lr: 3.1250e-07
Epoch 32/40
33/33 [=====] - 165s 5s/step - loss: 0.6138 - accuracy:
0.6724 - val_loss: 0.6584 - val_accuracy: 0.5341 - lr: 1.5625e-07
Epoch 33/40
33/33 [=====] - 164s 5s/step - loss: 0.6156 - accuracy:
0.6780 - val_loss: 0.6582 - val_accuracy: 0.5341 - lr: 1.5625e-07
Epoch 34/40
33/33 [=====] - 162s 5s/step - loss: 0.6093 - accuracy:
0.6871 - val_loss: 0.6607 - val_accuracy: 0.5341 - lr: 1.5625e-07
Epoch 35/40
33/33 [=====] - 163s 5s/step - loss: 0.5976 - accuracy:
0.6965 - val_loss: 0.6604 - val_accuracy: 0.5341 - lr: 1.5625e-07
Epoch 36/40
33/33 [=====] - ETA: 0s - loss: 0.6138 - accuracy: 0.6719
Epoch 36: ReduceLROnPlateau reducing learning rate to 1e-07.
33/33 [=====] - 165s 5s/step - loss: 0.6138 - accuracy:
0.6719 - val_loss: 0.6597 - val_accuracy: 0.5341 - lr: 1.5625e-07
Epoch 37/40
33/33 [=====] - 136s 4s/step - loss: 0.6233 - accuracy:
0.6780 - val_loss: 0.6604 - val_accuracy: 0.5341 - lr: 1.0000e-07
Epoch 38/40
33/33 [=====] - 128s 4s/step - loss: 0.6064 - accuracy:
0.6866 - val_loss: 0.6609 - val_accuracy: 0.5341 - lr: 1.0000e-07
```


Epoch 39/40

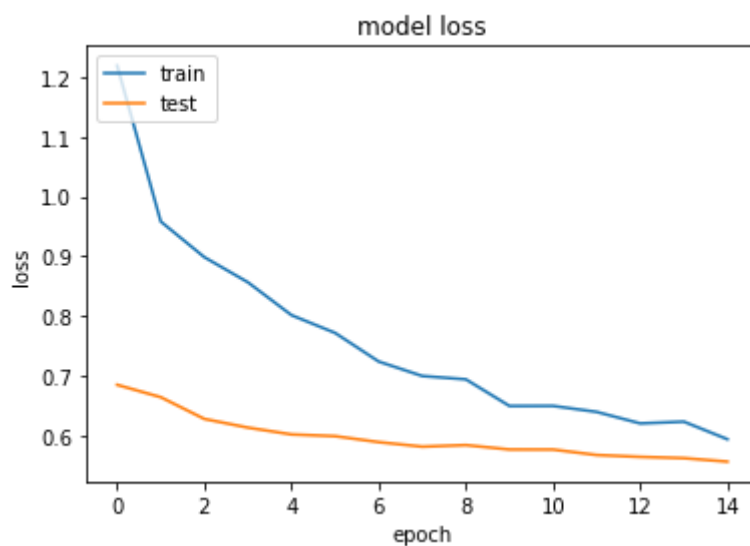
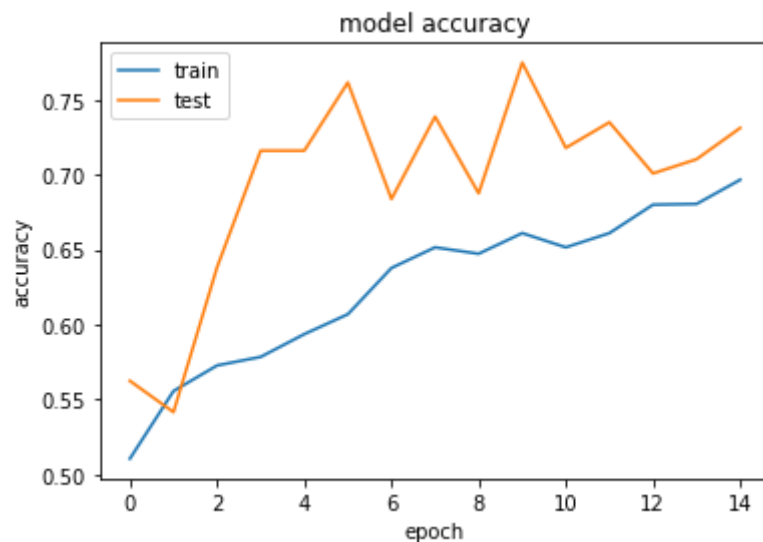
33/33 [=====] - 127s 4s/step - loss: 0.6050 - accuracy: 0.6795 - val_loss: 0.6614 - val_accuracy: 0.5341 - lr: 1.0000e-07

Epoch 40/40

33/33 [=====] - 138s 4s/step - loss: 0.6055 - accuracy: 0.6776 - val_loss: 0.6604 - val_accuracy: 0.5341 - lr: 1.0000e-07

```
In [57]: 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 []: