Facial Recognition with Deep Learning in Keras Using CNN

Project Description:
Facial recognition is a biometric alternative that measures unique characteristics of a human face. Applications available today include flight check in, tagging friends and family members in photos, and “tailored” advertising. You are a computer vision engineer who needs to develop a face recognition programme with deep convolutional neural networks. Objective: Use a deep convolutional neural network to perform facial recognition using Keras. Dataset Details: ORL face database composed of 400 images of size 112 x 92. There are 40 people, 10 images per person. The images were taken at different times, lighting and facial expressions. The faces are in an upright position in frontal view, with a slight left-right rotation.

Step1 - Input the required libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

import keras
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, Dropout
from keras.optimizers import Adam
```

Step2 - Load the dataset after loading the dataset and have to normalize every image.

In [2]:
```
dataset = np.load("ORL_faces.npz")
print("Files in unzipped dataset: ", dataset.files)
```

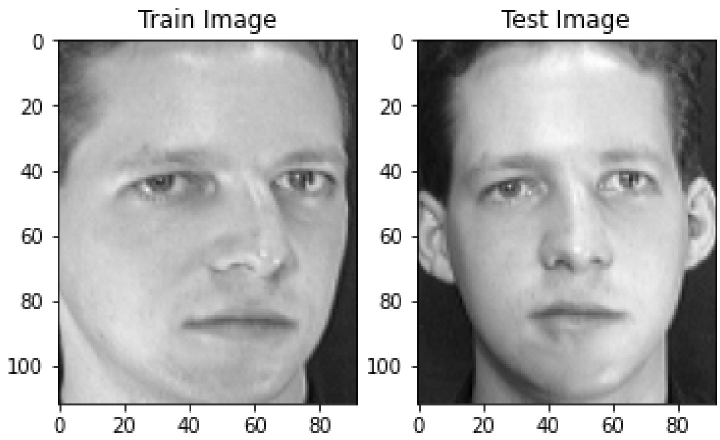
```
Files in unzipped dataset:  ['testY', 'testX', 'trainX', 'trainY']
```

In [3]:
```
X_train = dataset['trainX']
X_train = np.array(X_train,dtype='float32')/255

X_test = dataset['testX']
X_test = np.array(X_test,dtype='float32')/255

y_train= dataset['trainY']
y_test= dataset['testY']
```

Vizualizing image data

In [4]:
```
img_train = X_train[1].reshape(112,92)
plt.subplot(1,2,1)
plt.imshow(img_train, cmap='gray')
plt.title('Train Image')

img_test = X_test[1].reshape(112, 92)
plt.subplot(1,2,2)
```

```
plt.imshow(img_test, cmap='gray')
plt.title('Test Image')
plt.show()
```



Step3 - Split the dataset

In [5]:

```
X_train, X_valid, y_train, y_valid= train_test_split(
    X_train, y_train, test_size=.2, random_state=42,)
```

Step4 - Transform the images to equal sizes to feed in CNN

In [6]:

```
im_rows=112
im_cols=92
batch_size=512
im_shape=(im_rows, im_cols, 1)

#change the size of images
X_train = X_train.reshape(X_train.shape[0], *im_shape)
X_test = X_test.reshape(X_test.shape[0], *im_shape)
X_valid = X_valid.reshape(X_valid.shape[0], *im_shape)

print('x_train shape: {}'.format(y_train.shape[0]))
print('x_test shape: {}'.format(y_test.shape[0]))
```

```
x_train shape: 192
x_test shape: 160
```

Step5 - Build a CNN with 3 main layers

Build CNN model with 3 layers

1- Convolotional layer

2- Pooling layer

3- Fully connected layer

In [7]:

```
cnn_model= Sequential([
    Conv2D(filters=36, kernel_size=7, activation='relu', input_shape= im_shape),
    MaxPooling2D(pool_size=2),
    Conv2D(filters=54, kernel_size=5, activation='relu', input_shape= im_shape),
    MaxPooling2D(pool_size=2),
    Flatten(),
    Dense(2024, activation='relu'),
     Dropout(0.5),
```

```
    Dense(1024, activation='relu'),
    Dropout(0.5),
    Dense(512, activation='relu'),
    Dropout(0.5),
    #20 is the number of outputs
    Dense(20, activation='softmax')
])

cnn_model.compile(
    loss='sparse_categorical_crossentropy',
    optimizer=Adam(learning_rate=0.0001),
    metrics=['accuracy']
)
```

Show the model's parameters.

In [8]:
```
cnn_model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 106, 86, 36)       1800      
                                                                 
 max_pooling2d (MaxPooling2D  (None, 53, 43, 36)       0         
 )                                                               
                                                                 
 conv2d_1 (Conv2D)           (None, 49, 39, 54)        48654     
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 24, 19, 54)       0         
 2D)                                                             
                                                                 
 flatten (Flatten)           (None, 24624)             0         
                                                                 
 dense (Dense)               (None, 2024)              49841000  
                                                                 
 dropout (Dropout)           (None, 2024)              0         
                                                                 
 dense_1 (Dense)             (None, 1024)              2073600   
                                                                 
 dropout_1 (Dropout)         (None, 1024)              0         
                                                                 
 dense_2 (Dense)             (None, 512)               524800    
                                                                 
 dropout_2 (Dropout)         (None, 512)               0         
                                                                 
 dense_3 (Dense)             (None, 20)                10260     
                                                                 
=================================================================
Total params: 52,500,114
Trainable params: 52,500,114
Non-trainable params: 0
_________________________________________________________________
```

Step6 - Training the model

In [9]:
```
history=cnn_model.fit(
    np.array(X_train), np.array(y_train), batch_size=512,
    epochs=150,
    validation_data=(np.array(X_valid),np.array(y_valid)),
)
```

```
Epoch 1/150
1/1 [==============================] - 24s 24s/step - loss: 3.0032 - accuracy: 0.041
7 - val_loss: 3.0018 - val_accuracy: 0.0417
Epoch 2/150
1/1 [==============================] - 16s 16s/step - loss: 3.0010 - accuracy: 0.046
9 - val_loss: 3.0023 - val_accuracy: 0.0417
Epoch 3/150
1/1 [==============================] - 5s 5s/step - loss: 3.0010 - accuracy: 0.0781
- val_loss: 3.0006 - val_accuracy: 0.0417
Epoch 4/150
1/1 [==============================] - 6s 6s/step - loss: 3.0024 - accuracy: 0.0469
- val_loss: 2.9996 - val_accuracy: 0.0417
Epoch 5/150
1/1 [==============================] - 5s 5s/step - loss: 2.9786 - accuracy: 0.0781
- val_loss: 2.9981 - val_accuracy: 0.0417
Epoch 6/150
1/1 [==============================] - 5s 5s/step - loss: 3.0005 - accuracy: 0.0729
- val_loss: 2.9962 - val_accuracy: 0.0625
Epoch 7/150
1/1 [==============================] - 5s 5s/step - loss: 2.9553 - accuracy: 0.0677
- val_loss: 2.9947 - val_accuracy: 0.0833
Epoch 8/150
1/1 [==============================] - 5s 5s/step - loss: 2.9872 - accuracy: 0.0521
- val_loss: 2.9929 - val_accuracy: 0.0833
Epoch 9/150
1/1 [==============================] - 5s 5s/step - loss: 2.9643 - accuracy: 0.1094
- val_loss: 2.9902 - val_accuracy: 0.0833
Epoch 10/150
1/1 [==============================] - 5s 5s/step - loss: 2.9498 - accuracy: 0.0990
- val_loss: 2.9869 - val_accuracy: 0.1042
Epoch 11/150
1/1 [==============================] - 5s 5s/step - loss: 2.9382 - accuracy: 0.1042
- val_loss: 2.9829 - val_accuracy: 0.1250
Epoch 12/150
1/1 [==============================] - 5s 5s/step - loss: 2.9357 - accuracy: 0.1302
- val_loss: 2.9789 - val_accuracy: 0.1250
Epoch 13/150
1/1 [==============================] - 5s 5s/step - loss: 2.9218 - accuracy: 0.1198
- val_loss: 2.9736 - val_accuracy: 0.1458
Epoch 14/150
1/1 [==============================] - 5s 5s/step - loss: 2.9123 - accuracy: 0.1198
- val_loss: 2.9674 - val_accuracy: 0.1458
Epoch 15/150
1/1 [==============================] - 5s 5s/step - loss: 2.9285 - accuracy: 0.0833
- val_loss: 2.9612 - val_accuracy: 0.1667
Epoch 16/150
1/1 [==============================] - 5s 5s/step - loss: 2.9045 - accuracy: 0.0990
- val_loss: 2.9544 - val_accuracy: 0.1667
Epoch 17/150
1/1 [==============================] - 5s 5s/step - loss: 2.9065 - accuracy: 0.1302
- val_loss: 2.9476 - val_accuracy: 0.1667
Epoch 18/150
1/1 [==============================] - 5s 5s/step - loss: 2.8783 - accuracy: 0.1719
- val_loss: 2.9391 - val_accuracy: 0.1667
Epoch 19/150
1/1 [==============================] - 6s 6s/step - loss: 2.8669 - accuracy: 0.1562
- val_loss: 2.9295 - val_accuracy: 0.1458
Epoch 20/150
1/1 [==============================] - 5s 5s/step - loss: 2.9049 - accuracy: 0.1042
- val_loss: 2.9175 - val_accuracy: 0.1667
Epoch 21/150
1/1 [==============================] - 5s 5s/step - loss: 2.8623 - accuracy: 0.1510
- val_loss: 2.9033 - val_accuracy: 0.1667
Epoch 22/150
1/1 [==============================] - 5s 5s/step - loss: 2.8388 - accuracy: 0.1875
```

```
- val_loss: 2.8877 - val_accuracy: 0.1667
Epoch 23/150
1/1 [==============================] - 5s 5s/step - loss: 2.8567 - accuracy: 0.1354
- val_loss: 2.8705 - val_accuracy: 0.2083
Epoch 24/150
1/1 [==============================] - 5s 5s/step - loss: 2.8333 - accuracy: 0.1875
- val_loss: 2.8508 - val_accuracy: 0.2083
Epoch 25/150
1/1 [==============================] - 5s 5s/step - loss: 2.7695 - accuracy: 0.2448
- val_loss: 2.8297 - val_accuracy: 0.2292
Epoch 26/150
1/1 [==============================] - 5s 5s/step - loss: 2.7583 - accuracy: 0.2083
- val_loss: 2.8059 - val_accuracy: 0.2083
Epoch 27/150
1/1 [==============================] - 5s 5s/step - loss: 2.7667 - accuracy: 0.2396
- val_loss: 2.7797 - val_accuracy: 0.2917
Epoch 28/150
1/1 [==============================] - 5s 5s/step - loss: 2.7357 - accuracy: 0.2083
- val_loss: 2.7531 - val_accuracy: 0.3125
Epoch 29/150
1/1 [==============================] - 5s 5s/step - loss: 2.6730 - accuracy: 0.2760
- val_loss: 2.7210 - val_accuracy: 0.3125
Epoch 30/150
1/1 [==============================] - 5s 5s/step - loss: 2.6436 - accuracy: 0.2812
- val_loss: 2.6845 - val_accuracy: 0.3750
Epoch 31/150
1/1 [==============================] - 5s 5s/step - loss: 2.6615 - accuracy: 0.2500
- val_loss: 2.6485 - val_accuracy: 0.3750
Epoch 32/150
1/1 [==============================] - 5s 5s/step - loss: 2.6153 - accuracy: 0.2552
- val_loss: 2.6100 - val_accuracy: 0.3750
Epoch 33/150
1/1 [==============================] - 5s 5s/step - loss: 2.5773 - accuracy: 0.2917
- val_loss: 2.5708 - val_accuracy: 0.3750
Epoch 34/150
1/1 [==============================] - 5s 5s/step - loss: 2.4904 - accuracy: 0.3125
- val_loss: 2.5316 - val_accuracy: 0.3750
Epoch 35/150
1/1 [==============================] - 5s 5s/step - loss: 2.4634 - accuracy: 0.3177
- val_loss: 2.4909 - val_accuracy: 0.3750
Epoch 36/150
1/1 [==============================] - 5s 5s/step - loss: 2.4075 - accuracy: 0.3385
- val_loss: 2.4454 - val_accuracy: 0.4167
Epoch 37/150
1/1 [==============================] - 5s 5s/step - loss: 2.3428 - accuracy: 0.3125
- val_loss: 2.3965 - val_accuracy: 0.4375
Epoch 38/150
1/1 [==============================] - 5s 5s/step - loss: 2.3784 - accuracy: 0.3490
- val_loss: 2.3473 - val_accuracy: 0.4375
Epoch 39/150
1/1 [==============================] - 5s 5s/step - loss: 2.3575 - accuracy: 0.3229
- val_loss: 2.2956 - val_accuracy: 0.4375
Epoch 40/150
1/1 [==============================] - 5s 5s/step - loss: 2.2699 - accuracy: 0.3281
- val_loss: 2.2404 - val_accuracy: 0.4792
Epoch 41/150
1/1 [==============================] - 5s 5s/step - loss: 2.2076 - accuracy: 0.3750
- val_loss: 2.1848 - val_accuracy: 0.4792
Epoch 42/150
1/1 [==============================] - 5s 5s/step - loss: 2.1521 - accuracy: 0.3646
- val_loss: 2.1266 - val_accuracy: 0.5000
Epoch 43/150
1/1 [==============================] - 5s 5s/step - loss: 2.0982 - accuracy: 0.4375
- val_loss: 2.0655 - val_accuracy: 0.5000
Epoch 44/150
```

```
1/1 [==============================] - 5s 5s/step - loss: 2.0331 - accuracy: 0.4323
- val_loss: 2.0031 - val_accuracy: 0.5208
Epoch 45/150
1/1 [==============================] - 5s 5s/step - loss: 1.9943 - accuracy: 0.4531
- val_loss: 1.9467 - val_accuracy: 0.5208
Epoch 46/150
1/1 [==============================] - 5s 5s/step - loss: 1.9298 - accuracy: 0.4792
- val_loss: 1.8915 - val_accuracy: 0.5417
Epoch 47/150
1/1 [==============================] - 5s 5s/step - loss: 1.8585 - accuracy: 0.4792
- val_loss: 1.8297 - val_accuracy: 0.5417
Epoch 48/150
1/1 [==============================] - 5s 5s/step - loss: 1.8201 - accuracy: 0.5052
- val_loss: 1.7618 - val_accuracy: 0.5833
Epoch 49/150
1/1 [==============================] - 5s 5s/step - loss: 1.7673 - accuracy: 0.4844
- val_loss: 1.6862 - val_accuracy: 0.6042
Epoch 50/150
1/1 [==============================] - 5s 5s/step - loss: 1.7444 - accuracy: 0.4531
- val_loss: 1.6154 - val_accuracy: 0.6042
Epoch 51/150
1/1 [==============================] - 5s 5s/step - loss: 1.6922 - accuracy: 0.5104
- val_loss: 1.5469 - val_accuracy: 0.6250
Epoch 52/150
1/1 [==============================] - 5s 5s/step - loss: 1.5754 - accuracy: 0.5625
- val_loss: 1.4793 - val_accuracy: 0.6667
Epoch 53/150
1/1 [==============================] - 5s 5s/step - loss: 1.5168 - accuracy: 0.5677
- val_loss: 1.4078 - val_accuracy: 0.6667
Epoch 54/150
1/1 [==============================] - 6s 6s/step - loss: 1.5670 - accuracy: 0.5521
- val_loss: 1.3363 - val_accuracy: 0.6875
Epoch 55/150
1/1 [==============================] - 6s 6s/step - loss: 1.3788 - accuracy: 0.6354
- val_loss: 1.2711 - val_accuracy: 0.7500
Epoch 56/150
1/1 [==============================] - 5s 5s/step - loss: 1.3612 - accuracy: 0.6198
- val_loss: 1.2042 - val_accuracy: 0.7917
Epoch 57/150
1/1 [==============================] - 5s 5s/step - loss: 1.3848 - accuracy: 0.5729
- val_loss: 1.1460 - val_accuracy: 0.7917
Epoch 58/150
1/1 [==============================] - 6s 6s/step - loss: 1.3670 - accuracy: 0.5729
- val_loss: 1.0957 - val_accuracy: 0.7917
Epoch 59/150
1/1 [==============================] - 5s 5s/step - loss: 1.2445 - accuracy: 0.6667
- val_loss: 1.0437 - val_accuracy: 0.7708
Epoch 60/150
1/1 [==============================] - 5s 5s/step - loss: 1.1145 - accuracy: 0.6875
- val_loss: 0.9876 - val_accuracy: 0.7708
Epoch 61/150
1/1 [==============================] - 5s 5s/step - loss: 1.1921 - accuracy: 0.6719
- val_loss: 0.9310 - val_accuracy: 0.8125
Epoch 62/150
1/1 [==============================] - 5s 5s/step - loss: 1.0668 - accuracy: 0.7135
- val_loss: 0.8717 - val_accuracy: 0.8333
Epoch 63/150
1/1 [==============================] - 5s 5s/step - loss: 1.0399 - accuracy: 0.6979
- val_loss: 0.8163 - val_accuracy: 0.8333
Epoch 64/150
1/1 [==============================] - 5s 5s/step - loss: 0.8722 - accuracy: 0.7812
- val_loss: 0.7633 - val_accuracy: 0.8750
Epoch 65/150
1/1 [==============================] - 5s 5s/step - loss: 0.9491 - accuracy: 0.6979
- val_loss: 0.7183 - val_accuracy: 0.8958
```

```
Epoch 66/150
1/1 [==============================] - 5s 5s/step - loss: 0.8928 - accuracy: 0.7031
- val_loss: 0.6764 - val_accuracy: 0.8958
Epoch 67/150
1/1 [==============================] - 5s 5s/step - loss: 0.8815 - accuracy: 0.7812
- val_loss: 0.6379 - val_accuracy: 0.8958
Epoch 68/150
1/1 [==============================] - 5s 5s/step - loss: 0.7758 - accuracy: 0.8021
- val_loss: 0.5967 - val_accuracy: 0.8958
Epoch 69/150
1/1 [==============================] - 5s 5s/step - loss: 0.7625 - accuracy: 0.8125
- val_loss: 0.5600 - val_accuracy: 0.9167
Epoch 70/150
1/1 [==============================] - 5s 5s/step - loss: 0.7312 - accuracy: 0.7969
- val_loss: 0.5251 - val_accuracy: 0.8958
Epoch 71/150
1/1 [==============================] - 5s 5s/step - loss: 0.6888 - accuracy: 0.8177
- val_loss: 0.4895 - val_accuracy: 0.9167
Epoch 72/150
1/1 [==============================] - 5s 5s/step - loss: 0.6171 - accuracy: 0.8177
- val_loss: 0.4665 - val_accuracy: 0.9375
Epoch 73/150
1/1 [==============================] - 5s 5s/step - loss: 0.5564 - accuracy: 0.8698
- val_loss: 0.4432 - val_accuracy: 0.9375
Epoch 74/150
1/1 [==============================] - 5s 5s/step - loss: 0.6741 - accuracy: 0.7969
- val_loss: 0.4199 - val_accuracy: 0.8958
Epoch 75/150
1/1 [==============================] - 5s 5s/step - loss: 0.6109 - accuracy: 0.8438
- val_loss: 0.4014 - val_accuracy: 0.8958
Epoch 76/150
1/1 [==============================] - 5s 5s/step - loss: 0.5622 - accuracy: 0.8646
- val_loss: 0.3726 - val_accuracy: 0.9167
Epoch 77/150
1/1 [==============================] - 5s 5s/step - loss: 0.5012 - accuracy: 0.8750
- val_loss: 0.3543 - val_accuracy: 0.9375
Epoch 78/150
1/1 [==============================] - 5s 5s/step - loss: 0.5168 - accuracy: 0.8906
- val_loss: 0.3380 - val_accuracy: 0.9375
Epoch 79/150
1/1 [==============================] - 5s 5s/step - loss: 0.5231 - accuracy: 0.8229
- val_loss: 0.3213 - val_accuracy: 0.9167
Epoch 80/150
1/1 [==============================] - 5s 5s/step - loss: 0.4225 - accuracy: 0.8854
- val_loss: 0.3168 - val_accuracy: 0.9167
Epoch 81/150
1/1 [==============================] - 5s 5s/step - loss: 0.4320 - accuracy: 0.8646
- val_loss: 0.3110 - val_accuracy: 0.9167
Epoch 82/150
1/1 [==============================] - 5s 5s/step - loss: 0.4270 - accuracy: 0.8802
- val_loss: 0.2845 - val_accuracy: 0.9375
Epoch 83/150
1/1 [==============================] - 5s 5s/step - loss: 0.3771 - accuracy: 0.8958
- val_loss: 0.2764 - val_accuracy: 0.9375
Epoch 84/150
1/1 [==============================] - 5s 5s/step - loss: 0.3825 - accuracy: 0.9010
- val_loss: 0.2749 - val_accuracy: 0.9375
Epoch 85/150
1/1 [==============================] - 5s 5s/step - loss: 0.3659 - accuracy: 0.8906
- val_loss: 0.2671 - val_accuracy: 0.9167
Epoch 86/150
1/1 [==============================] - 5s 5s/step - loss: 0.3062 - accuracy: 0.9375
- val_loss: 0.2621 - val_accuracy: 0.9167
Epoch 87/150
1/1 [==============================] - 5s 5s/step - loss: 0.2405 - accuracy: 0.9531
```

```
- val_loss: 0.2631 - val_accuracy: 0.8958
Epoch 88/150
1/1 [==============================] - 5s 5s/step - loss: 0.3305 - accuracy: 0.9219
- val_loss: 0.2474 - val_accuracy: 0.8958
Epoch 89/150
1/1 [==============================] - 5s 5s/step - loss: 0.3589 - accuracy: 0.8854
- val_loss: 0.2204 - val_accuracy: 0.9167
Epoch 90/150
1/1 [==============================] - 5s 5s/step - loss: 0.2508 - accuracy: 0.9583
- val_loss: 0.1982 - val_accuracy: 0.9375
Epoch 91/150
1/1 [==============================] - 5s 5s/step - loss: 0.2766 - accuracy: 0.9271
- val_loss: 0.1857 - val_accuracy: 0.9583
Epoch 92/150
1/1 [==============================] - 5s 5s/step - loss: 0.2388 - accuracy: 0.9375
- val_loss: 0.1761 - val_accuracy: 0.9583
Epoch 93/150
1/1 [==============================] - 5s 5s/step - loss: 0.2211 - accuracy: 0.9427
- val_loss: 0.1698 - val_accuracy: 0.9583
Epoch 94/150
1/1 [==============================] - 5s 5s/step - loss: 0.2048 - accuracy: 0.9427
- val_loss: 0.1714 - val_accuracy: 0.9375
Epoch 95/150
1/1 [==============================] - 5s 5s/step - loss: 0.2169 - accuracy: 0.9375
- val_loss: 0.1794 - val_accuracy: 0.9375
Epoch 96/150
1/1 [==============================] - 5s 5s/step - loss: 0.2205 - accuracy: 0.9427
- val_loss: 0.1716 - val_accuracy: 0.9375
Epoch 97/150
1/1 [==============================] - 5s 5s/step - loss: 0.2619 - accuracy: 0.9427
- val_loss: 0.1567 - val_accuracy: 0.9583
Epoch 98/150
1/1 [==============================] - 5s 5s/step - loss: 0.1739 - accuracy: 0.9635
- val_loss: 0.1528 - val_accuracy: 0.9375
Epoch 99/150
1/1 [==============================] - 5s 5s/step - loss: 0.2050 - accuracy: 0.9531
- val_loss: 0.1519 - val_accuracy: 0.9375
Epoch 100/150
1/1 [==============================] - 5s 5s/step - loss: 0.2042 - accuracy: 0.9427
- val_loss: 0.1523 - val_accuracy: 0.9375
Epoch 101/150
1/1 [==============================] - 5s 5s/step - loss: 0.1991 - accuracy: 0.9427
- val_loss: 0.1469 - val_accuracy: 0.9167
Epoch 102/150
1/1 [==============================] - 5s 5s/step - loss: 0.1890 - accuracy: 0.9479
- val_loss: 0.1434 - val_accuracy: 0.9167
Epoch 103/150
1/1 [==============================] - 5s 5s/step - loss: 0.1286 - accuracy: 0.9740
- val_loss: 0.1360 - val_accuracy: 0.9375
Epoch 104/150
1/1 [==============================] - 5s 5s/step - loss: 0.1136 - accuracy: 0.9896
- val_loss: 0.1229 - val_accuracy: 0.9583
Epoch 105/150
1/1 [==============================] - 5s 5s/step - loss: 0.1535 - accuracy: 0.9531
- val_loss: 0.1103 - val_accuracy: 0.9583
Epoch 106/150
1/1 [==============================] - 5s 5s/step - loss: 0.1493 - accuracy: 0.9740
- val_loss: 0.0970 - val_accuracy: 0.9792
Epoch 107/150
1/1 [==============================] - 5s 5s/step - loss: 0.1541 - accuracy: 0.9531
- val_loss: 0.0895 - val_accuracy: 0.9792
Epoch 108/150
1/1 [==============================] - 5s 5s/step - loss: 0.1282 - accuracy: 0.9740
- val_loss: 0.0823 - val_accuracy: 0.9792
Epoch 109/150
```

```
1/1 [==============================] - 5s 5s/step - loss: 0.1184 - accuracy: 0.9740
- val_loss: 0.0786 - val_accuracy: 0.9792
Epoch 110/150
1/1 [==============================] - 5s 5s/step - loss: 0.1035 - accuracy: 0.9740
- val_loss: 0.0778 - val_accuracy: 0.9792
Epoch 111/150
1/1 [==============================] - 5s 5s/step - loss: 0.1439 - accuracy: 0.9583
- val_loss: 0.0766 - val_accuracy: 0.9792
Epoch 112/150
1/1 [==============================] - 5s 5s/step - loss: 0.1313 - accuracy: 0.9635
- val_loss: 0.0765 - val_accuracy: 0.9792
Epoch 113/150
1/1 [==============================] - 5s 5s/step - loss: 0.0766 - accuracy: 0.9896
- val_loss: 0.0746 - val_accuracy: 0.9792
Epoch 114/150
1/1 [==============================] - 5s 5s/step - loss: 0.1225 - accuracy: 0.9688
- val_loss: 0.0728 - val_accuracy: 0.9792
Epoch 115/150
1/1 [==============================] - 5s 5s/step - loss: 0.1006 - accuracy: 0.9844
- val_loss: 0.0691 - val_accuracy: 0.9792
Epoch 116/150
1/1 [==============================] - 5s 5s/step - loss: 0.1226 - accuracy: 0.9688
- val_loss: 0.0682 - val_accuracy: 0.9792
Epoch 117/150
1/1 [==============================] - 5s 5s/step - loss: 0.1101 - accuracy: 0.9792
- val_loss: 0.0652 - val_accuracy: 0.9792
Epoch 118/150
1/1 [==============================] - 5s 5s/step - loss: 0.0982 - accuracy: 0.9844
- val_loss: 0.0617 - val_accuracy: 0.9792
Epoch 119/150
1/1 [==============================] - 5s 5s/step - loss: 0.0994 - accuracy: 0.9792
- val_loss: 0.0596 - val_accuracy: 0.9792
Epoch 120/150
1/1 [==============================] - 6s 6s/step - loss: 0.1000 - accuracy: 0.9688
- val_loss: 0.0593 - val_accuracy: 0.9792
Epoch 121/150
1/1 [==============================] - 5s 5s/step - loss: 0.1025 - accuracy: 0.9688
- val_loss: 0.0567 - val_accuracy: 0.9792
Epoch 122/150
1/1 [==============================] - 5s 5s/step - loss: 0.0675 - accuracy: 0.9896
- val_loss: 0.0543 - val_accuracy: 0.9792
Epoch 123/150
1/1 [==============================] - 5s 5s/step - loss: 0.1126 - accuracy: 0.9792
- val_loss: 0.0536 - val_accuracy: 0.9792
Epoch 124/150
1/1 [==============================] - 5s 5s/step - loss: 0.0837 - accuracy: 0.9792
- val_loss: 0.0532 - val_accuracy: 0.9792
Epoch 125/150
1/1 [==============================] - 5s 5s/step - loss: 0.0821 - accuracy: 0.9844
- val_loss: 0.0519 - val_accuracy: 0.9792
Epoch 126/150
1/1 [==============================] - 5s 5s/step - loss: 0.0722 - accuracy: 0.9896
- val_loss: 0.0504 - val_accuracy: 0.9792
Epoch 127/150
1/1 [==============================] - 5s 5s/step - loss: 0.0715 - accuracy: 0.9948
- val_loss: 0.0491 - val_accuracy: 0.9792
Epoch 128/150
1/1 [==============================] - 5s 5s/step - loss: 0.0722 - accuracy: 0.9792
- val_loss: 0.0481 - val_accuracy: 0.9792
Epoch 129/150
1/1 [==============================] - 5s 5s/step - loss: 0.0678 - accuracy: 0.9948
- val_loss: 0.0473 - val_accuracy: 0.9792
Epoch 130/150
1/1 [==============================] - 5s 5s/step - loss: 0.0630 - accuracy: 0.9896
- val_loss: 0.0466 - val_accuracy: 0.9792
```

```
Epoch 131/150
1/1 [==============================] - 5s 5s/step - loss: 0.0522 - accuracy: 0.9948
- val_loss: 0.0443 - val_accuracy: 0.9792
Epoch 132/150
1/1 [==============================] - 5s 5s/step - loss: 0.0495 - accuracy: 1.0000
- val_loss: 0.0432 - val_accuracy: 0.9792
Epoch 133/150
1/1 [==============================] - 5s 5s/step - loss: 0.0519 - accuracy: 0.9844
- val_loss: 0.0428 - val_accuracy: 0.9792
Epoch 134/150
1/1 [==============================] - 5s 5s/step - loss: 0.0544 - accuracy: 0.9948
- val_loss: 0.0420 - val_accuracy: 0.9792
Epoch 135/150
1/1 [==============================] - 5s 5s/step - loss: 0.0586 - accuracy: 0.9896
- val_loss: 0.0394 - val_accuracy: 0.9792
Epoch 136/150
1/1 [==============================] - 5s 5s/step - loss: 0.0381 - accuracy: 0.9948
- val_loss: 0.0381 - val_accuracy: 1.0000
Epoch 137/150
1/1 [==============================] - 5s 5s/step - loss: 0.0620 - accuracy: 0.9844
- val_loss: 0.0351 - val_accuracy: 1.0000
Epoch 138/150
1/1 [==============================] - 5s 5s/step - loss: 0.0793 - accuracy: 0.9844
- val_loss: 0.0329 - val_accuracy: 1.0000
Epoch 139/150
1/1 [==============================] - 6s 6s/step - loss: 0.0636 - accuracy: 0.9740
- val_loss: 0.0330 - val_accuracy: 1.0000
Epoch 140/150
1/1 [==============================] - 5s 5s/step - loss: 0.0575 - accuracy: 0.9896
- val_loss: 0.0316 - val_accuracy: 1.0000
Epoch 141/150
1/1 [==============================] - 5s 5s/step - loss: 0.0375 - accuracy: 0.9896
- val_loss: 0.0324 - val_accuracy: 1.0000
Epoch 142/150
1/1 [==============================] - 5s 5s/step - loss: 0.0450 - accuracy: 0.9896
- val_loss: 0.0328 - val_accuracy: 1.0000
Epoch 143/150
1/1 [==============================] - 5s 5s/step - loss: 0.0410 - accuracy: 0.9896
- val_loss: 0.0339 - val_accuracy: 1.0000
Epoch 144/150
1/1 [==============================] - 6s 6s/step - loss: 0.0548 - accuracy: 0.9948
- val_loss: 0.0317 - val_accuracy: 1.0000
Epoch 145/150
1/1 [==============================] - 5s 5s/step - loss: 0.0603 - accuracy: 0.9844
- val_loss: 0.0303 - val_accuracy: 1.0000
Epoch 146/150
1/1 [==============================] - 5s 5s/step - loss: 0.0585 - accuracy: 0.9844
- val_loss: 0.0263 - val_accuracy: 1.0000
Epoch 147/150
1/1 [==============================] - 5s 5s/step - loss: 0.0438 - accuracy: 0.9844
- val_loss: 0.0237 - val_accuracy: 1.0000
Epoch 148/150
1/1 [==============================] - 5s 5s/step - loss: 0.0350 - accuracy: 0.9948
- val_loss: 0.0213 - val_accuracy: 1.0000
Epoch 149/150
1/1 [==============================] - 5s 5s/step - loss: 0.0445 - accuracy: 0.9896
- val_loss: 0.0204 - val_accuracy: 1.0000
Epoch 150/150
1/1 [==============================] - 5s 5s/step - loss: 0.0234 - accuracy: 1.0000
- val_loss: 0.0203 - val_accuracy: 1.0000
```

In [13]:

```
score = cnn_model.evaluate( np.array(X_test),  np.array(y_test), verbose=0)

print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

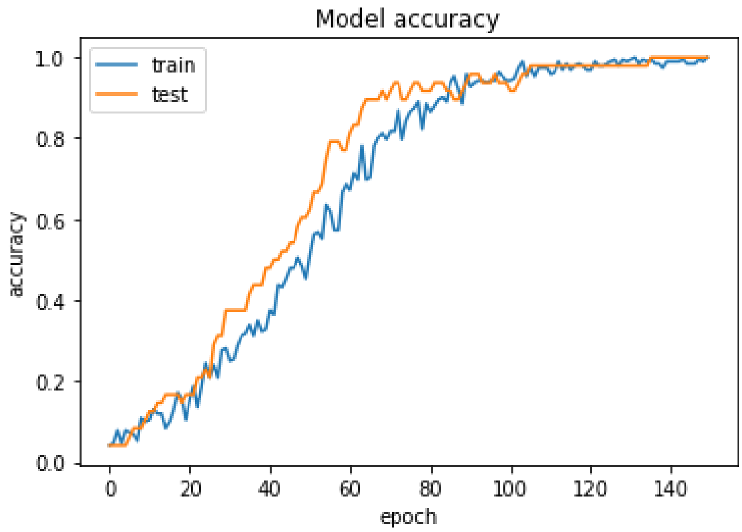
```
Test loss: 0.38005945086479187
Test accuracy: 0.918749988079071
```

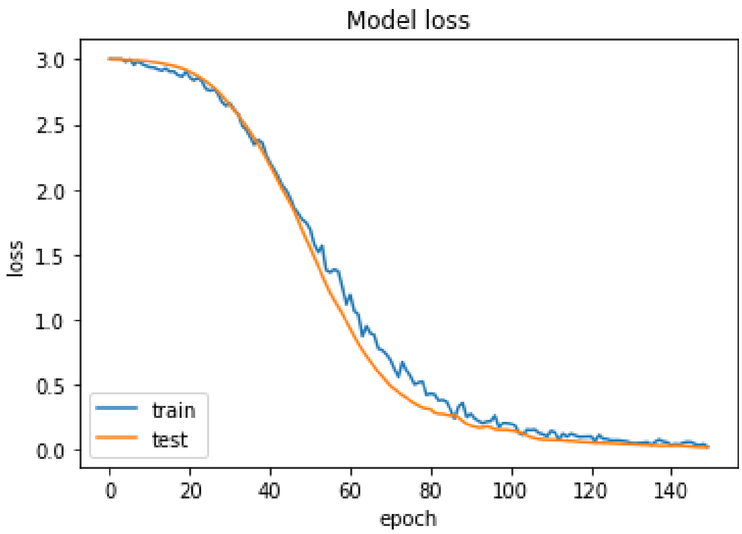
Step7 - Plotting the result

In [14]:
```python
history.history.keys()
```

Out[14]:
```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

In [15]:
```python
# Plotting history for accuracy
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()

# Plotting history for loss
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='lower left')
plt.show()
```





In [ ]:

In [ ]: