Facial Recognition with Deep Learning in Keras Using CNN Project

Aurore Prevot

Importation of the librairies

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2

from sklearn.model_selection import train_test_split
import tensorflow as tf
```

Loading the dataset

In [2]:
```
data = np.load("ORL_faces.npz")
list_files = data.files
print(f"There are {len(list_files)} files, named:\n")
for item in list_files:
    print(f"{item}:")
    print(f"{data[item]}\n")
```

```
There are 4 files, named:

testY:
[ 0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2
  3  3  3  3  3  3  3  3  4  4  4  4  4  4  4  4  5  5  5  5  5  5  5  5
  6  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8
  9  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 11 11 11 11 11 11 11 11
 12 12 12 12 12 12 12 12 13 13 13 13 13 13 13 13 14 14 14 14 14 14 14 14
 15 15 15 15 15 15 15 15 16 16 16 16 16 16 16 16 17 17 17 17 17 17 17 17
 18 18 18 18 18 18 18 18 19 19 19 19 19 19 19 19]

testX:
[[ 41.  47.  47. ...  35.  37.  38.]
 [ 44.  43.  32. ...  43.  43.  37.]
 [ 42.  41.  44. ...  42.  43.  41.]
 ...
 [101. 100. 103. ...  31.  40.  42.]
 [105. 108. 106. ...  44.  40.  47.]
 [113. 114. 111. ...  62.  81.  89.]]

trainX:
[[ 48.  49.  45. ...  47.  46.  46.]
 [ 60.  60.  62. ...  32.  34.  34.]
 [ 39.  44.  53. ...  29.  26.  29.]
 ...
 [114. 117. 114. ...  98.  96.  98.]
 [105. 105. 107. ...  54.  47.  41.]
 [116. 114. 117. ...  95. 100. 101.]]

trainY:
[ 0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1  1
  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  3  3  3  3  3  3
  4  4  4  4  4  4  4  4  4  4  4  4  5  5  5  5  5  5  5  5  5  5  5  5
  6  6  6  6  6  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  7  7  7  7
  8  8  8  8  8  8  8  8  8  8  8  8  9  9  9  9  9  9  9  9  9  9  9  9
 10 10 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11 11 11 11 11 11 11 11
 12 12 12 12 12 12 12 12 12 12 12 12 13 13 13 13 13 13 13 13 13 13 13 13
 14 14 14 14 14 14 14 14 14 14 14 14 15 15 15 15 15 15 15 15 15 15 15 15
 16 16 16 16 16 16 16 16 16 16 16 16 17 17 17 17 17 17 17 17 17 17 17 17
 18 18 18 18 18 18 18 18 18 18 18 18 19 19 19 19 19 19 19 19 19 19 19 19]
```

In [3]:
```python
print(f"The shape of the training dataset is : {data['trainX'].shape}")
print(f"The shape of the training target is : {data['trainY'].shape}")
print(f"The shape of the test dataset is : {data['testX'].shape}")
print(f"The shape of the test target is : {data['testY'].shape}")
```

```
The shape of the training dataset is : (240, 10304)
The shape of the training target is : (240,)
The shape of the test dataset is : (160, 10304)
The shape of the test target is : (160,)
```

Obervations:

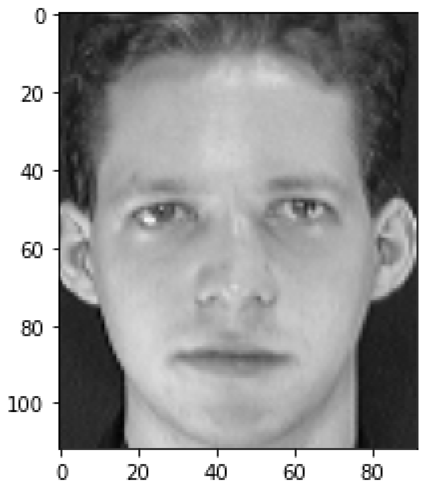
- The dataset is split in training and test sets.
- The training set has 240 images of 20 persons (12 images per person).
- The test set has 160 images of 20 persons (8 images per person).

Visualization of the data

- Visualization of the first image with matplotlib

In [4]:
```
plt.imshow(data["trainX"][0].reshape(112, 92), cmap="gray")
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x290b1bacc40>
```



- Visualization of the first image with opencv

In [5]:
```
img1 = data["trainX"][0].reshape(112, 92).astype(np.uint8)

cv2.imshow("Premiere image", img1)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- Visualization of the last image with opencv

In [6]:
```
img240 = data["trainX"][-1].reshape(112, 92).astype(np.uint8)

cv2.imshow("last image", img240)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

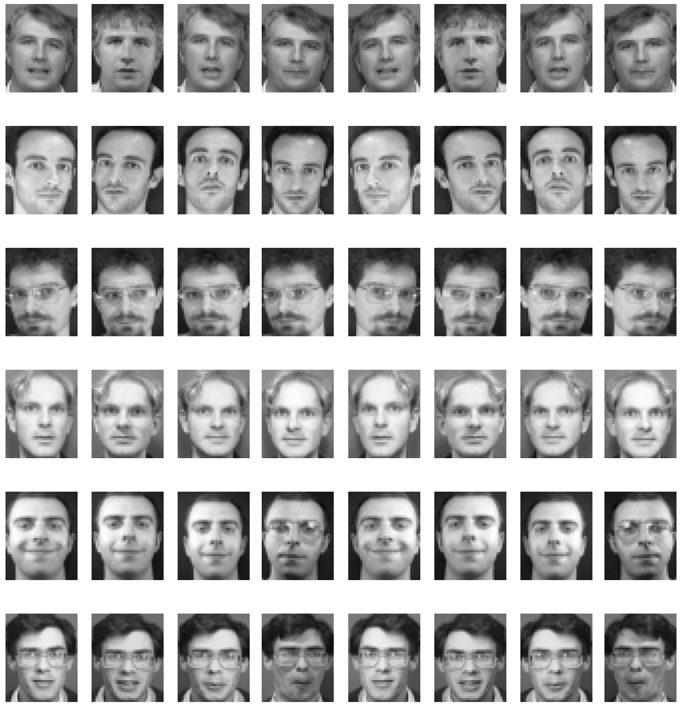
- Visualization of the images in the training dataset

In [7]:
```
for j in range(0, data["trainX"].shape[0], 12):
    for i in range(12):
        plt.subplot(1, 12, i+1)
        plt.imshow(data["trainX"][i+j].reshape(112, 92), cmap="gray")
        plt.axis("off")
    plt.show()
```

- Visualisation of the images in the test dataset

In [8]:
```
for j in range(0, data["testX"].shape[0], 8):
```

```
    for i in range(8):
        plt.subplot(1, 8, i+1)
        plt.imshow(data["testX"][i+j].reshape(112, 92), cmap="gray")
        plt.axis("off")
    plt.show()
```



Observations:

- The 20 persons in the training dataset are the same as the 20 persons in the test dataset.

Normalization of the images and splitting the dataset

In [9]:
```
X_train_norm = (data["trainX"])/255
print(X_train_norm)
print(f"\nThe shape of the normalized training dataset is {X_train_norm.shape}")
```

```
[[0.18823529 0.19215686 0.17647059 ... 0.18431373 0.18039216 0.18039216]
 [0.23529412 0.23529412 0.24313725 ... 0.1254902  0.13333333 0.13333333]
 [0.15294118 0.17254902 0.20784314 ... 0.11372549 0.10196078 0.11372549]
 ...
 [0.44705882 0.45882353 0.44705882 ... 0.38431373 0.37647059 0.38431373]
 [0.41176471 0.41176471 0.41960784 ... 0.21176471 0.18431373 0.16078431]
 [0.45490196 0.44705882 0.45882353 ... 0.37254902 0.39215686 0.39607843]]

The shape of the normalized training dataset is (240, 10304)
```

In [10]:
```
X_test_norm = (data["testX"])/255
print(X_test_norm)
print(f"\nThe shape of the normalized test dataset is {X_test_norm.shape}")
```

```
[[0.16078431 0.18431373 0.18431373 ... 0.1372549  0.14509804 0.14901961]
 [0.17254902 0.16862745 0.1254902  ... 0.16862745 0.16862745 0.14509804]
 [0.16470588 0.16078431 0.17254902 ... 0.16470588 0.16862745 0.16078431]
 ...
 [0.39607843 0.39215686 0.40392157 ... 0.12156863 0.15686275 0.16470588]
 [0.41176471 0.42352941 0.41568627 ... 0.17254902 0.15686275 0.18431373]
 [0.44313725 0.44705882 0.43529412 ... 0.24313725 0.31764706 0.34901961]]

The shape of the normalized test dataset is (160, 10304)
```

```
In [11]: Y_train = data['trainY']
Y_test = data['testY']
print(f"The shape of the training target is {Y_train.shape}")
print(f"The shape of the test target is {Y_test.shape}")
```

```
The shape of the training target is (240,)
The shape of the test target is (160,)
```

Resizing the images

```
In [12]: X_train = []
for i in range(X_train_norm.shape[0]):
    X_train.append(X_train_norm[i].reshape(112, 92, 1))
X_train = np.array(X_train)
print(f"\nThe shape of the training dataset is {X_train.shape}")
```

```
The shape of the training dataset is (240, 112, 92, 1)
```

```
In [13]: X_test = []
for i in range(X_test_norm.shape[0]):
    X_test.append(X_test_norm[i].reshape(112, 92, 1))
X_test = np.array(X_test)
print(f"\nThe shape of the test dataset is {X_test.shape}")
```

```
The shape of the test dataset is (160, 112, 92, 1)
```

CNN Model with 3 layers

Building the model

```
In [14]: model = tf.keras.models.Sequential()

model.add(tf.keras.layers.Conv2D(32 , (4,4) , input_shape= X_train.shape[1:] , activat
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2,2)))

model.add(tf.keras.layers.Conv2D(16 , (3,3), activation= 'relu' ))
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2,2)))

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(units= 4096 , activation="relu"))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(units= 1024, activation="relu"))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(units= 256, activation="relu"))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(units= 20 , activation="softmax"))

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 109, 89, 32)       544

 max_pooling2d (MaxPooling2D  (None, 54, 44, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 52, 42, 16)        4624

 max_pooling2d_1 (MaxPooling  (None, 26, 21, 16)       0
 2D)

 flatten (Flatten)           (None, 8736)              0

 dense (Dense)               (None, 4096)              35786752

 dropout (Dropout)           (None, 4096)              0

 dense_1 (Dense)             (None, 1024)              4195328

 dropout_1 (Dropout)         (None, 1024)              0

 dense_2 (Dense)             (None, 256)               262400

 dropout_2 (Dropout)         (None, 256)               0

 dense_3 (Dense)             (None, 20)                5140

=================================================================
Total params: 40,254,788
Trainable params: 40,254,788
Non-trainable params: 0
_________________________________________________________________
```

In [15]:

```
class MyThresholdCallback(tf.keras.callbacks.Callback):
  def __init__(self, cl):
    super(MyThresholdCallback).__init__()
    self.cl = cl

  def on_epoch_end(self, epoch, logs=None):
    testScore = logs['val_accuracy']
    trainScore = logs['accuracy']
    if testScore >= self.cl and testScore >= trainScore:
      self.model.stop_training = True
```

In [16]:

```
model.compile(optimizer="adam", loss="sparse_categorical_crossentropy", metrics=['accu
```

Training the model

In [17]:

```
history = model.fit(x=X_train, y=Y_train,
              validation_data=(X_test, Y_test),
              epochs=150,
              callbacks=[MyThresholdCallback(0.9)])
```

```
Epoch 1/150
8/8 [==============================] - 3s 276ms/step - loss: 3.2982 - accuracy: 0.045
8 - val_loss: 2.9915 - val_accuracy: 0.0688
Epoch 2/150
8/8 [==============================] - 2s 258ms/step - loss: 2.9921 - accuracy: 0.079
2 - val_loss: 2.9624 - val_accuracy: 0.1312
Epoch 3/150
8/8 [==============================] - 2s 257ms/step - loss: 2.8610 - accuracy: 0.183
3 - val_loss: 2.6793 - val_accuracy: 0.2062
Epoch 4/150
8/8 [==============================] - 2s 250ms/step - loss: 2.4429 - accuracy: 0.241
7 - val_loss: 2.2862 - val_accuracy: 0.3938
Epoch 5/150
8/8 [==============================] - 2s 254ms/step - loss: 1.7686 - accuracy: 0.470
8 - val_loss: 1.3772 - val_accuracy: 0.5437
Epoch 6/150
8/8 [==============================] - 2s 257ms/step - loss: 1.1751 - accuracy: 0.654
2 - val_loss: 0.8739 - val_accuracy: 0.7125
Epoch 7/150
8/8 [==============================] - 2s 256ms/step - loss: 0.6341 - accuracy: 0.825
0 - val_loss: 0.8425 - val_accuracy: 0.7625
Epoch 8/150
8/8 [==============================] - 2s 256ms/step - loss: 0.3383 - accuracy: 0.891
7 - val_loss: 0.5076 - val_accuracy: 0.8438
Epoch 9/150
8/8 [==============================] - 2s 253ms/step - loss: 0.1352 - accuracy: 0.970
8 - val_loss: 0.5429 - val_accuracy: 0.8313
Epoch 10/150
8/8 [==============================] - 2s 248ms/step - loss: 0.0939 - accuracy: 0.979
2 - val_loss: 0.3698 - val_accuracy: 0.8813
Epoch 11/150
8/8 [==============================] - 2s 253ms/step - loss: 0.0621 - accuracy: 0.979
2 - val_loss: 0.5893 - val_accuracy: 0.8875
Epoch 12/150
8/8 [==============================] - 2s 250ms/step - loss: 0.0540 - accuracy: 0.979
2 - val_loss: 0.5322 - val_accuracy: 0.8188
Epoch 13/150
8/8 [==============================] - 2s 252ms/step - loss: 0.0228 - accuracy: 1.000
0 - val_loss: 0.6258 - val_accuracy: 0.8562
Epoch 14/150
8/8 [==============================] - 2s 256ms/step - loss: 0.0291 - accuracy: 0.983
3 - val_loss: 0.4678 - val_accuracy: 0.9062
Epoch 15/150
8/8 [==============================] - 2s 256ms/step - loss: 0.0193 - accuracy: 0.991
7 - val_loss: 0.4890 - val_accuracy: 0.8813
Epoch 16/150
8/8 [==============================] - 2s 253ms/step - loss: 0.0253 - accuracy: 0.991
7 - val_loss: 0.3564 - val_accuracy: 0.8813
Epoch 17/150
8/8 [==============================] - 2s 261ms/step - loss: 0.0149 - accuracy: 0.995
8 - val_loss: 0.4437 - val_accuracy: 0.8750
Epoch 18/150
8/8 [==============================] - 2s 250ms/step - loss: 0.0115 - accuracy: 0.995
8 - val_loss: 0.4068 - val_accuracy: 0.9000
Epoch 19/150
8/8 [==============================] - 2s 258ms/step - loss: 0.0035 - accuracy: 1.000
0 - val_loss: 0.4054 - val_accuracy: 0.9062
Epoch 20/150
8/8 [==============================] - 2s 265ms/step - loss: 0.0101 - accuracy: 1.000
0 - val_loss: 0.3780 - val_accuracy: 0.9062
```

```
Epoch 21/150
8/8 [==============================] - 2s 263ms/step - loss: 0.0017 - accuracy: 1.000
0 - val_loss: 0.4157 - val_accuracy: 0.8813
Epoch 22/150
8/8 [==============================] - 2s 266ms/step - loss: 0.0014 - accuracy: 1.000
0 - val_loss: 0.4013 - val_accuracy: 0.9062
Epoch 23/150
8/8 [==============================] - 2s 263ms/step - loss: 2.5349e-04 - accuracy:
1.0000 - val_loss: 0.3988 - val_accuracy: 0.9000
Epoch 24/150
8/8 [==============================] - 2s 272ms/step - loss: 4.3395e-04 - accuracy:
1.0000 - val_loss: 0.4169 - val_accuracy: 0.9000
Epoch 25/150
8/8 [==============================] - 2s 255ms/step - loss: 1.9691e-04 - accuracy:
1.0000 - val_loss: 0.4332 - val_accuracy: 0.9000
Epoch 26/150
8/8 [==============================] - 2s 253ms/step - loss: 1.9333e-04 - accuracy:
1.0000 - val_loss: 0.4454 - val_accuracy: 0.8875
Epoch 27/150
8/8 [==============================] - 2s 266ms/step - loss: 3.2482e-04 - accuracy:
1.0000 - val_loss: 0.4497 - val_accuracy: 0.8875
Epoch 28/150
8/8 [==============================] - 2s 275ms/step - loss: 2.9441e-04 - accuracy:
1.0000 - val_loss: 0.4462 - val_accuracy: 0.8875
Epoch 29/150
8/8 [==============================] - 2s 263ms/step - loss: 6.4477e-04 - accuracy:
1.0000 - val_loss: 0.4521 - val_accuracy: 0.8938
Epoch 30/150
8/8 [==============================] - 2s 259ms/step - loss: 3.1644e-04 - accuracy:
1.0000 - val_loss: 0.5331 - val_accuracy: 0.9000
Epoch 31/150
8/8 [==============================] - 2s 263ms/step - loss: 3.6455e-04 - accuracy:
1.0000 - val_loss: 0.5634 - val_accuracy: 0.8938
Epoch 32/150
8/8 [==============================] - 2s 264ms/step - loss: 1.9034e-04 - accuracy:
1.0000 - val_loss: 0.5571 - val_accuracy: 0.8938
Epoch 33/150
8/8 [==============================] - 2s 255ms/step - loss: 2.4293e-04 - accuracy:
1.0000 - val_loss: 0.5446 - val_accuracy: 0.8938
Epoch 34/150
8/8 [==============================] - 2s 254ms/step - loss: 1.4141e-04 - accuracy:
1.0000 - val_loss: 0.5310 - val_accuracy: 0.9000
Epoch 35/150
8/8 [==============================] - 2s 253ms/step - loss: 1.9788e-04 - accuracy:
1.0000 - val_loss: 0.5161 - val_accuracy: 0.9000
Epoch 36/150
8/8 [==============================] - 2s 260ms/step - loss: 2.0248e-04 - accuracy:
1.0000 - val_loss: 0.4758 - val_accuracy: 0.9000
Epoch 37/150
8/8 [==============================] - 2s 254ms/step - loss: 7.4697e-04 - accuracy:
1.0000 - val_loss: 0.8290 - val_accuracy: 0.8687
Epoch 38/150
8/8 [==============================] - 2s 254ms/step - loss: 8.1020e-05 - accuracy:
1.0000 - val_loss: 0.6906 - val_accuracy: 0.8875
Epoch 39/150
8/8 [==============================] - 2s 258ms/step - loss: 3.0324e-04 - accuracy:
1.0000 - val_loss: 0.7106 - val_accuracy: 0.8750
Epoch 40/150
8/8 [==============================] - 2s 258ms/step - loss: 0.0066 - accuracy: 0.995
8 - val_loss: 0.5971 - val_accuracy: 0.8625
```

```
Epoch 41/150
8/8 [==============================] - 2s 255ms/step - loss: 0.0657 - accuracy: 0.983
3 - val_loss: 0.3993 - val_accuracy: 0.9062
Epoch 42/150
8/8 [==============================] - 2s 251ms/step - loss: 0.0499 - accuracy: 0.991
7 - val_loss: 0.4640 - val_accuracy: 0.8687
Epoch 43/150
8/8 [==============================] - 2s 253ms/step - loss: 0.0129 - accuracy: 0.995
8 - val_loss: 0.4926 - val_accuracy: 0.9062
Epoch 44/150
8/8 [==============================] - 2s 253ms/step - loss: 0.0158 - accuracy: 0.995
8 - val_loss: 0.6908 - val_accuracy: 0.8250
Epoch 45/150
8/8 [==============================] - 2s 251ms/step - loss: 0.0428 - accuracy: 0.995
8 - val_loss: 0.3678 - val_accuracy: 0.8813
Epoch 46/150
8/8 [==============================] - 2s 255ms/step - loss: 0.0100 - accuracy: 1.000
0 - val_loss: 0.4206 - val_accuracy: 0.8875
Epoch 47/150
8/8 [==============================] - 2s 263ms/step - loss: 0.0019 - accuracy: 1.000
0 - val_loss: 0.6728 - val_accuracy: 0.8750
Epoch 48/150
8/8 [==============================] - 2s 259ms/step - loss: 0.0170 - accuracy: 0.991
7 - val_loss: 0.4429 - val_accuracy: 0.9000
Epoch 49/150
8/8 [==============================] - 2s 253ms/step - loss: 0.0321 - accuracy: 0.979
2 - val_loss: 0.2175 - val_accuracy: 0.9375
Epoch 50/150
8/8 [==============================] - 2s 250ms/step - loss: 0.0296 - accuracy: 0.991
7 - val_loss: 0.5678 - val_accuracy: 0.8813
Epoch 51/150
8/8 [==============================] - 2s 253ms/step - loss: 0.0281 - accuracy: 0.987
5 - val_loss: 0.7219 - val_accuracy: 0.8562
Epoch 52/150
8/8 [==============================] - 2s 250ms/step - loss: 0.0325 - accuracy: 0.991
7 - val_loss: 0.6338 - val_accuracy: 0.8313
Epoch 53/150
8/8 [==============================] - 2s 253ms/step - loss: 0.0190 - accuracy: 1.000
0 - val_loss: 0.6377 - val_accuracy: 0.8562
Epoch 54/150
8/8 [==============================] - 2s 255ms/step - loss: 0.0239 - accuracy: 0.991
7 - val_loss: 0.4470 - val_accuracy: 0.8938
Epoch 55/150
8/8 [==============================] - 2s 261ms/step - loss: 0.0197 - accuracy: 0.991
7 - val_loss: 0.4379 - val_accuracy: 0.8625
Epoch 56/150
8/8 [==============================] - 2s 254ms/step - loss: 0.0062 - accuracy: 1.000
0 - val_loss: 0.4386 - val_accuracy: 0.8500
Epoch 57/150
8/8 [==============================] - 2s 256ms/step - loss: 0.0027 - accuracy: 1.000
0 - val_loss: 0.4881 - val_accuracy: 0.8562
Epoch 58/150
8/8 [==============================] - 2s 257ms/step - loss: 3.6457e-04 - accuracy:
1.0000 - val_loss: 0.4908 - val_accuracy: 0.8625
Epoch 59/150
8/8 [==============================] - 2s 251ms/step - loss: 0.0018 - accuracy: 1.000
0 - val_loss: 0.9059 - val_accuracy: 0.8500
Epoch 60/150
8/8 [==============================] - 2s 253ms/step - loss: 0.0228 - accuracy: 0.991
7 - val_loss: 0.7274 - val_accuracy: 0.8750
```

```
Epoch 61/150
8/8 [==============================] - 2s 255ms/step - loss: 0.0074 - accuracy: 0.995
8 - val_loss: 1.0714 - val_accuracy: 0.8562
Epoch 62/150
8/8 [==============================] - 2s 253ms/step - loss: 0.0356 - accuracy: 0.987
5 - val_loss: 0.6609 - val_accuracy: 0.8438
Epoch 63/150
8/8 [==============================] - 2s 259ms/step - loss: 0.0240 - accuracy: 0.991
7 - val_loss: 0.6515 - val_accuracy: 0.8625
Epoch 64/150
8/8 [==============================] - 2s 265ms/step - loss: 0.0089 - accuracy: 1.000
0 - val_loss: 1.0911 - val_accuracy: 0.8562
Epoch 65/150
8/8 [==============================] - 2s 254ms/step - loss: 8.6713e-04 - accuracy:
1.0000 - val_loss: 0.9725 - val_accuracy: 0.8438
Epoch 66/150
8/8 [==============================] - 2s 260ms/step - loss: 0.0014 - accuracy: 1.000
0 - val_loss: 0.9595 - val_accuracy: 0.8750
Epoch 67/150
8/8 [==============================] - 2s 257ms/step - loss: 1.1725e-04 - accuracy:
1.0000 - val_loss: 1.1134 - val_accuracy: 0.8938
Epoch 68/150
8/8 [==============================] - 2s 250ms/step - loss: 0.0049 - accuracy: 0.995
8 - val_loss: 1.3268 - val_accuracy: 0.8687
Epoch 69/150
8/8 [==============================] - 2s 251ms/step - loss: 0.0027 - accuracy: 1.000
0 - val_loss: 0.9978 - val_accuracy: 0.8687
Epoch 70/150
8/8 [==============================] - 2s 252ms/step - loss: 3.4118e-04 - accuracy:
1.0000 - val_loss: 0.7596 - val_accuracy: 0.8750
Epoch 71/150
8/8 [==============================] - 2s 254ms/step - loss: 0.0017 - accuracy: 1.000
0 - val_loss: 0.7262 - val_accuracy: 0.8750
Epoch 72/150
8/8 [==============================] - 2s 253ms/step - loss: 1.1876e-04 - accuracy:
1.0000 - val_loss: 0.7717 - val_accuracy: 0.8813
Epoch 73/150
8/8 [==============================] - 2s 252ms/step - loss: 1.1583e-04 - accuracy:
1.0000 - val_loss: 0.8145 - val_accuracy: 0.8750
Epoch 74/150
8/8 [==============================] - 2s 250ms/step - loss: 1.2424e-04 - accuracy:
1.0000 - val_loss: 0.8167 - val_accuracy: 0.8687
Epoch 75/150
8/8 [==============================] - 2s 250ms/step - loss: 4.8196e-04 - accuracy:
1.0000 - val_loss: 0.8579 - val_accuracy: 0.8813
Epoch 76/150
8/8 [==============================] - 2s 253ms/step - loss: 0.0012 - accuracy: 1.000
0 - val_loss: 0.9781 - val_accuracy: 0.8750
Epoch 77/150
8/8 [==============================] - 2s 252ms/step - loss: 3.5256e-05 - accuracy:
1.0000 - val_loss: 1.1204 - val_accuracy: 0.8687
Epoch 78/150
8/8 [==============================] - 2s 250ms/step - loss: 1.0449e-04 - accuracy:
1.0000 - val_loss: 1.2030 - val_accuracy: 0.8687
Epoch 79/150
8/8 [==============================] - 2s 256ms/step - loss: 8.2616e-06 - accuracy:
1.0000 - val_loss: 1.2513 - val_accuracy: 0.8687
Epoch 80/150
8/8 [==============================] - 2s 258ms/step - loss: 3.5939e-06 - accuracy:
1.0000 - val_loss: 1.2716 - val_accuracy: 0.8687
```

```
Epoch 81/150
8/8 [==============================] - 2s 253ms/step - loss: 7.0537e-05 - accuracy:
1.0000 - val_loss: 1.2655 - val_accuracy: 0.8687
Epoch 82/150
8/8 [==============================] - 2s 259ms/step - loss: 2.0630e-05 - accuracy:
1.0000 - val_loss: 1.2505 - val_accuracy: 0.8750
Epoch 83/150
8/8 [==============================] - 2s 255ms/step - loss: 1.5080e-04 - accuracy:
1.0000 - val_loss: 1.1907 - val_accuracy: 0.8750
Epoch 84/150
8/8 [==============================] - 2s 253ms/step - loss: 7.7403e-06 - accuracy:
1.0000 - val_loss: 1.1175 - val_accuracy: 0.8813
Epoch 85/150
8/8 [==============================] - 2s 259ms/step - loss: 3.5901e-06 - accuracy:
1.0000 - val_loss: 1.0853 - val_accuracy: 0.8813
Epoch 86/150
8/8 [==============================] - 2s 263ms/step - loss: 3.3713e-05 - accuracy:
1.0000 - val_loss: 1.0665 - val_accuracy: 0.8813
Epoch 87/150
8/8 [==============================] - 2s 261ms/step - loss: 4.2299e-06 - accuracy:
1.0000 - val_loss: 1.0568 - val_accuracy: 0.8813
Epoch 88/150
8/8 [==============================] - 2s 264ms/step - loss: 1.0217e-06 - accuracy:
1.0000 - val_loss: 1.0515 - val_accuracy: 0.8813
Epoch 89/150
8/8 [==============================] - 2s 255ms/step - loss: 5.3124e-06 - accuracy:
1.0000 - val_loss: 1.0491 - val_accuracy: 0.8813
Epoch 90/150
8/8 [==============================] - 2s 251ms/step - loss: 4.7932e-06 - accuracy:
1.0000 - val_loss: 1.0544 - val_accuracy: 0.8813
Epoch 91/150
8/8 [==============================] - 2s 265ms/step - loss: 4.8201e-06 - accuracy:
1.0000 - val_loss: 1.0558 - val_accuracy: 0.8813
Epoch 92/150
8/8 [==============================] - 2s 250ms/step - loss: 8.9419e-06 - accuracy:
1.0000 - val_loss: 1.0563 - val_accuracy: 0.8813
Epoch 93/150
8/8 [==============================] - 2s 253ms/step - loss: 3.1851e-05 - accuracy:
1.0000 - val_loss: 1.0657 - val_accuracy: 0.8813
Epoch 94/150
8/8 [==============================] - 2s 251ms/step - loss: 7.5330e-05 - accuracy:
1.0000 - val_loss: 1.0341 - val_accuracy: 0.8875
Epoch 95/150
8/8 [==============================] - 2s 256ms/step - loss: 1.8709e-06 - accuracy:
1.0000 - val_loss: 1.0173 - val_accuracy: 0.8875
Epoch 96/150
8/8 [==============================] - 2s 255ms/step - loss: 6.2176e-04 - accuracy:
1.0000 - val_loss: 0.9662 - val_accuracy: 0.8875
Epoch 97/150
8/8 [==============================] - 2s 251ms/step - loss: 7.1169e-05 - accuracy:
1.0000 - val_loss: 0.9579 - val_accuracy: 0.8813
Epoch 98/150
8/8 [==============================] - 2s 255ms/step - loss: 4.6179e-06 - accuracy:
1.0000 - val_loss: 0.9490 - val_accuracy: 0.8938
Epoch 99/150
8/8 [==============================] - 2s 250ms/step - loss: 2.7863e-06 - accuracy:
1.0000 - val_loss: 0.9443 - val_accuracy: 0.9000
Epoch 100/150
8/8 [==============================] - 2s 251ms/step - loss: 4.8381e-06 - accuracy:
1.0000 - val_loss: 0.9434 - val_accuracy: 0.9000
```

```
Epoch 101/150
8/8 [==============================] - 2s 249ms/step - loss: 0.0104 - accuracy: 0.995
8 - val_loss: 0.7161 - val_accuracy: 0.8687
Epoch 102/150
8/8 [==============================] - 2s 255ms/step - loss: 0.0125 - accuracy: 0.995
8 - val_loss: 0.5327 - val_accuracy: 0.8687
Epoch 103/150
8/8 [==============================] - 2s 258ms/step - loss: 0.0195 - accuracy: 0.995
8 - val_loss: 0.7726 - val_accuracy: 0.8500
Epoch 104/150
8/8 [==============================] - 2s 254ms/step - loss: 0.0089 - accuracy: 1.000
0 - val_loss: 0.7487 - val_accuracy: 0.8375
Epoch 105/150
8/8 [==============================] - 2s 250ms/step - loss: 0.0035 - accuracy: 1.000
0 - val_loss: 0.5716 - val_accuracy: 0.8750
Epoch 106/150
8/8 [==============================] - 2s 252ms/step - loss: 7.8148e-04 - accuracy:
1.0000 - val_loss: 0.5903 - val_accuracy: 0.8687
Epoch 107/150
8/8 [==============================] - 2s 252ms/step - loss: 0.0038 - accuracy: 1.000
0 - val_loss: 0.6302 - val_accuracy: 0.8438
Epoch 108/150
8/8 [==============================] - 2s 248ms/step - loss: 0.0130 - accuracy: 0.995
8 - val_loss: 0.8704 - val_accuracy: 0.8500
Epoch 109/150
8/8 [==============================] - 2s 250ms/step - loss: 0.0026 - accuracy: 1.000
0 - val_loss: 0.9171 - val_accuracy: 0.8500
Epoch 110/150
8/8 [==============================] - 2s 254ms/step - loss: 7.8475e-04 - accuracy:
1.0000 - val_loss: 0.9983 - val_accuracy: 0.8562
Epoch 111/150
8/8 [==============================] - 2s 254ms/step - loss: 5.0919e-04 - accuracy:
1.0000 - val_loss: 0.9900 - val_accuracy: 0.8687
Epoch 112/150
8/8 [==============================] - 2s 259ms/step - loss: 1.5103e-04 - accuracy:
1.0000 - val_loss: 0.8968 - val_accuracy: 0.8687
Epoch 113/150
8/8 [==============================] - 2s 257ms/step - loss: 4.3023e-05 - accuracy:
1.0000 - val_loss: 0.8496 - val_accuracy: 0.8813
Epoch 114/150
8/8 [==============================] - 2s 258ms/step - loss: 3.3187e-05 - accuracy:
1.0000 - val_loss: 0.8203 - val_accuracy: 0.8813
Epoch 115/150
8/8 [==============================] - 2s 265ms/step - loss: 5.1177e-05 - accuracy:
1.0000 - val_loss: 0.8121 - val_accuracy: 0.8813
Epoch 116/150
8/8 [==============================] - 2s 250ms/step - loss: 2.8609e-05 - accuracy:
1.0000 - val_loss: 0.8222 - val_accuracy: 0.8813
Epoch 117/150
8/8 [==============================] - 2s 250ms/step - loss: 2.1734e-05 - accuracy:
1.0000 - val_loss: 0.8366 - val_accuracy: 0.8813
Epoch 118/150
8/8 [==============================] - 2s 254ms/step - loss: 7.1892e-06 - accuracy:
1.0000 - val_loss: 0.8517 - val_accuracy: 0.8813
Epoch 119/150
8/8 [==============================] - 2s 258ms/step - loss: 2.5106e-05 - accuracy:
1.0000 - val_loss: 0.8626 - val_accuracy: 0.8813
Epoch 120/150
8/8 [==============================] - 2s 254ms/step - loss: 9.9270e-05 - accuracy:
1.0000 - val_loss: 0.9057 - val_accuracy: 0.8750
```

```
Epoch 121/150
8/8 [==============================] - 2s 249ms/step - loss: 6.0769e-05 - accuracy: 1.0000 - val_loss: 0.9086 - val_accuracy: 0.8750
Epoch 122/150
8/8 [==============================] - 2s 249ms/step - loss: 1.1960e-05 - accuracy: 1.0000 - val_loss: 0.9129 - val_accuracy: 0.8750
Epoch 123/150
8/8 [==============================] - 2s 249ms/step - loss: 1.1746e-05 - accuracy: 1.0000 - val_loss: 0.9194 - val_accuracy: 0.8750
Epoch 124/150
8/8 [==============================] - 2s 250ms/step - loss: 2.6077e-05 - accuracy: 1.0000 - val_loss: 0.9195 - val_accuracy: 0.8750
Epoch 125/150
8/8 [==============================] - 2s 250ms/step - loss: 5.0194e-06 - accuracy: 1.0000 - val_loss: 0.9200 - val_accuracy: 0.8750
Epoch 126/150
8/8 [==============================] - 2s 252ms/step - loss: 1.4384e-05 - accuracy: 1.0000 - val_loss: 0.9193 - val_accuracy: 0.8750
Epoch 127/150
8/8 [==============================] - 2s 255ms/step - loss: 5.2946e-05 - accuracy: 1.0000 - val_loss: 0.8893 - val_accuracy: 0.8813
Epoch 128/150
8/8 [==============================] - 2s 253ms/step - loss: 6.0325e-06 - accuracy: 1.0000 - val_loss: 0.8702 - val_accuracy: 0.8813
Epoch 129/150
8/8 [==============================] - 2s 248ms/step - loss: 2.0364e-05 - accuracy: 1.0000 - val_loss: 0.8624 - val_accuracy: 0.8875
Epoch 130/150
8/8 [==============================] - 2s 248ms/step - loss: 8.1027e-06 - accuracy: 1.0000 - val_loss: 0.8622 - val_accuracy: 0.8938
Epoch 131/150
8/8 [==============================] - 2s 252ms/step - loss: 1.1068e-05 - accuracy: 1.0000 - val_loss: 0.8630 - val_accuracy: 0.8938
Epoch 132/150
8/8 [==============================] - 2s 257ms/step - loss: 5.5568e-06 - accuracy: 1.0000 - val_loss: 0.8647 - val_accuracy: 0.8938
Epoch 133/150
8/8 [==============================] - 2s 257ms/step - loss: 3.5557e-06 - accuracy: 1.0000 - val_loss: 0.8677 - val_accuracy: 0.8938
Epoch 134/150
8/8 [==============================] - 2s 252ms/step - loss: 4.1632e-06 - accuracy: 1.0000 - val_loss: 0.8692 - val_accuracy: 0.8938
Epoch 135/150
8/8 [==============================] - 2s 256ms/step - loss: 6.8551e-06 - accuracy: 1.0000 - val_loss: 0.8706 - val_accuracy: 0.8938
Epoch 136/150
8/8 [==============================] - 2s 254ms/step - loss: 7.2840e-06 - accuracy: 1.0000 - val_loss: 0.8724 - val_accuracy: 0.8938
Epoch 137/150
8/8 [==============================] - 2s 251ms/step - loss: 3.1207e-06 - accuracy: 1.0000 - val_loss: 0.8764 - val_accuracy: 0.8938
Epoch 138/150
8/8 [==============================] - 2s 249ms/step - loss: 6.9221e-06 - accuracy: 1.0000 - val_loss: 0.8813 - val_accuracy: 0.8938
Epoch 139/150
8/8 [==============================] - 2s 248ms/step - loss: 4.9091e-06 - accuracy: 1.0000 - val_loss: 0.8857 - val_accuracy: 0.8938
Epoch 140/150
8/8 [==============================] - 2s 251ms/step - loss: 1.5368e-06 - accuracy: 1.0000 - val_loss: 0.8873 - val_accuracy: 0.8938
```

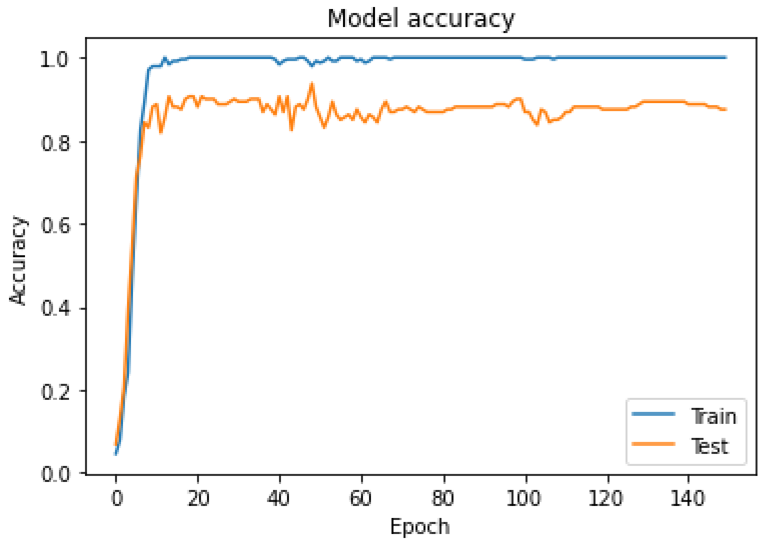
```
Epoch 141/150
8/8 [==============================] - 2s 252ms/step - loss: 4.5762e-06 - accuracy:
1.0000 - val_loss: 0.8894 - val_accuracy: 0.8875
Epoch 142/150
8/8 [==============================] - 2s 250ms/step - loss: 1.5933e-05 - accuracy:
1.0000 - val_loss: 0.9032 - val_accuracy: 0.8875
Epoch 143/150
8/8 [==============================] - 2s 254ms/step - loss: 7.4712e-06 - accuracy:
1.0000 - val_loss: 0.9136 - val_accuracy: 0.8875
Epoch 144/150
8/8 [==============================] - 2s 256ms/step - loss: 2.4576e-06 - accuracy:
1.0000 - val_loss: 0.9201 - val_accuracy: 0.8875
Epoch 145/150
8/8 [==============================] - 2s 249ms/step - loss: 1.3788e-05 - accuracy:
1.0000 - val_loss: 0.9391 - val_accuracy: 0.8875
Epoch 146/150
8/8 [==============================] - 2s 250ms/step - loss: 1.2877e-05 - accuracy:
1.0000 - val_loss: 0.9511 - val_accuracy: 0.8813
Epoch 147/150
8/8 [==============================] - 2s 251ms/step - loss: 6.8997e-06 - accuracy:
1.0000 - val_loss: 0.9515 - val_accuracy: 0.8813
Epoch 148/150
8/8 [==============================] - 2s 249ms/step - loss: 3.7371e-06 - accuracy:
1.0000 - val_loss: 0.9497 - val_accuracy: 0.8813
Epoch 149/150
8/8 [==============================] - 2s 253ms/step - loss: 2.1352e-05 - accuracy:
1.0000 - val_loss: 0.9546 - val_accuracy: 0.8750
Epoch 150/150
8/8 [==============================] - 2s 257ms/step - loss: 1.8332e-06 - accuracy:
1.0000 - val_loss: 0.9615 - val_accuracy: 0.8750
```

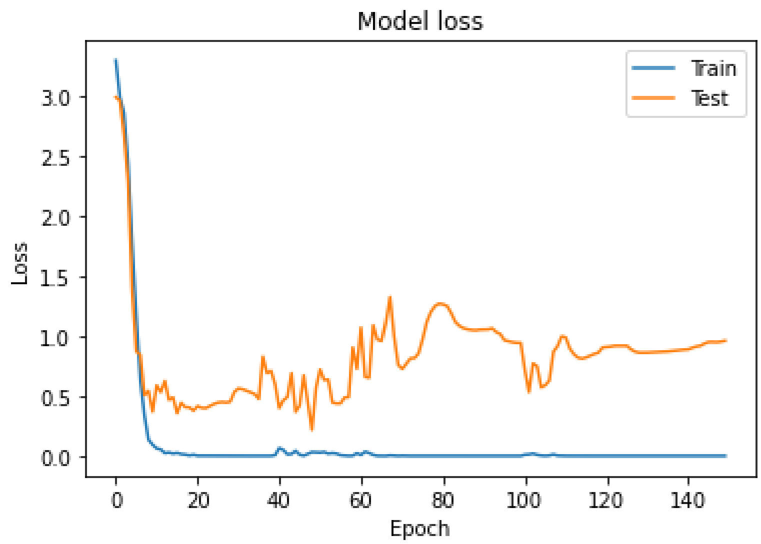
Showing the results

In [18]:

```
plt.plot(history.history["accuracy"], label="Train")
plt.plot(history.history["val_accuracy"], label="Test")
plt.title("Model accuracy")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()
plt.show()

plt.plot(history.history["loss"], label="Train")
plt.plot(history.history["val_loss"], label="Test")
plt.title("Model loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()
plt.show()
```


Model accuracy
Accuracy
Epoch
Train
Test


Model loss
Loss
Epoch
Train
Test

```
In [ ]:
```