In [2]:

```
# Face recognition - CNN model
# Step 1: Loading the required libraries
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow import keras
%matplotlib inline
```

In [3]:

```
# Step 2: Loading the dataset.
img_data = np.load('ORL_faces.npz')
```

In [4]:

```
tf.__version__
```

Out[4]:

```
'2.7.0'
```

In [5]:

```
# Reading the data into train and validation samples
#img_data.files
testX = img_data['testX']
testY = img_data['testY']
trainX = img_data['trainX']
trainY = img_data['trainY']
```

In [6]:

```
print(testX)
print(testX.shape)
```

```
[[ 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.]]
(160, 10304)
```

In [7]:

```
print(testY)
print(testY.shape)
```

```
[ 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]
(160,)
```

In [8]:

```
print(trainX)
print(trainX.shape)
```

```
[[ 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.]]
(240, 10304)
```

In [9]:

```python
print(trainY)
print(trainY.shape)
```

```
[ 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]
(240,)
```

In [10]:

```python
# Normalizing every image
trainX_norm = trainX * (1.0/255.)
testX_norm = testX * (1.0/255.)
```

In [11]:

```python
trainX_norm
```

Out[11]:

```
array([[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]])
```

In [12]:

```python
testX_norm
```

Out[12]:

```
array([[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]])
```

In [13]:

```python
# Step 3: Split the dataset
# The dataset is already split into training and test sets
```

In [14]:

```
# Step 4: Transform the images to equal sizes to feed in CNN
height = 112
width = 92
channels = 1

trainX_reshaped = np.reshape(trainX_norm, (trainX_norm.shape[0], height, width, channels
))
testX_reshaped = np.reshape(testX_norm, (testX_norm.shape[0], height, width, channels))
```

In [15]:

```
#trainX_reshaped
```

In [16]:

```
#testX_reshaped
```

In [17]:

```
# Step 5: Build a CNN model
#Architect our CNN
model = tf.keras.models.Sequential()
#Conv2D(noFeatureMaps , kernelShape, input_shape, activation)
```

In [18]:

```
#Step 5.i:
#Create First Convolutional Layer
#Convolve Layer
model.add(tf.keras.layers.Conv2D(16, (3,3), input_shape = (height, width, channels), act
ivation = 'tanh' , padding='same'))
#Pooling Layer
model.add(tf.keras.layers.MaxPooling2D(pool_size = (2,2)))
#model.add(tf.keras.layers.AveragePooling2D(pool_size = (2,2) ))

#Create Second Convolutional Layer
#Convolve Layer
model.add(tf.keras.layers.Conv2D(16 , (3,3) , activation = 'tanh' , padding='same'))
#Pooling Layer
model.add(tf.keras.layers.MaxPooling2D(pool_size = (2,2)))

#Flatten
model.add(tf.keras.layers.Flatten())

# Fully Connected Layer -- adding one dropout layer
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(units= 512 , activation = 'relu'))
model.add(tf.keras.layers.Dense(units= 256 , activation = 'relu'))
# number of labels
num_labels = len(np.unique(trainY))
model.add(tf.keras.layers.Dense(units=num_labels , activation='softmax'))
```

In [19]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 112, 92, 16)       160

 max_pooling2d (MaxPooling2D  (None, 56, 46, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 56, 46, 16)        2320

 max_pooling2d_1 (MaxPooling  (None, 28, 23, 16)       0
```

```
 2D)

 flatten (Flatten)           (None, 10304)           0

 dropout (Dropout)           (None, 10304)           0

 dense (Dense)               (None, 512)             5276160

 dense_1 (Dense)             (None, 256)             131328

 dense_2 (Dense)             (None, 20)              5140

=================================================================
Total params: 5,415,108
Trainable params: 5,415,108
Non-trainable params: 0
_________________________________________________________________
```

In [20]:

```
#Compile
model.compile(optimizer=tf.keras.optimizers.Nadam(),
#model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy']
             )
```

In [21]:

```
#Fit
history = model.fit(trainX_reshaped,
                    trainY,
                    validation_data=(testX_reshaped, testY),
                    epochs=100,
                    steps_per_epoch= int(240 / 20), #No of Images / batch size
                    validation_steps= int(160 / 20)
                    )
```

```
Epoch 1/100
12/12 [==============================] - 5s 207ms/step - loss: 3.1085 - accuracy: 0.0708
- val_loss: 2.7205 - val_accuracy: 0.1937
Epoch 2/100
12/12 [==============================] - 2s 131ms/step - loss: 2.3596 - accuracy: 0.3792
- val_loss: 1.8292 - val_accuracy: 0.5500
Epoch 3/100
12/12 [==============================] - 2s 148ms/step - loss: 1.3299 - accuracy: 0.6583
- val_loss: 0.8091 - val_accuracy: 0.8313
Epoch 4/100
12/12 [==============================] - 2s 135ms/step - loss: 0.4765 - accuracy: 0.8875
- val_loss: 0.4689 - val_accuracy: 0.9125
Epoch 5/100
12/12 [==============================] - 2s 129ms/step - loss: 0.0657 - accuracy: 1.0000
- val_loss: 0.3724 - val_accuracy: 0.9187
Epoch 6/100
12/12 [==============================] - 2s 134ms/step - loss: 0.0170 - accuracy: 1.0000
- val_loss: 0.3078 - val_accuracy: 0.9312
Epoch 7/100
12/12 [==============================] - 2s 137ms/step - loss: 0.0078 - accuracy: 1.0000
- val_loss: 0.3040 - val_accuracy: 0.9375
Epoch 8/100
12/12 [==============================] - 2s 130ms/step - loss: 0.0053 - accuracy: 1.0000
- val_loss: 0.3102 - val_accuracy: 0.9250
Epoch 9/100
12/12 [==============================] - 2s 158ms/step - loss: 0.0037 - accuracy: 1.0000
- val_loss: 0.3077 - val_accuracy: 0.9250
Epoch 10/100
12/12 [==============================] - 2s 135ms/step - loss: 0.0030 - accuracy: 1.0000
- val_loss: 0.3061 - val_accuracy: 0.9312
Epoch 11/100
12/12 [==============================] - 2s 144ms/step - loss: 0.0025 - accuracy: 1.0000
- val_loss: 0.3085 - val_accuracy: 0.9312
Epoch 12/100
```

```
12/12 [==============================] - 2s 139ms/step - loss: 0.0021 - accuracy: 1.0000
- val_loss: 0.3147 - val_accuracy: 0.9250
Epoch 13/100
12/12 [==============================] - 2s 147ms/step - loss: 0.0019 - accuracy: 1.0000
- val_loss: 0.3239 - val_accuracy: 0.9250
Epoch 14/100
12/12 [==============================] - 2s 134ms/step - loss: 0.0016 - accuracy: 1.0000
- val_loss: 0.3139 - val_accuracy: 0.9250
Epoch 15/100
12/12 [==============================] - 2s 130ms/step - loss: 0.0013 - accuracy: 1.0000
- val_loss: 0.3114 - val_accuracy: 0.9250
Epoch 16/100
12/12 [==============================] - 2s 146ms/step - loss: 0.0013 - accuracy: 1.0000
- val_loss: 0.3235 - val_accuracy: 0.9250
Epoch 17/100
12/12 [==============================] - 2s 146ms/step - loss: 0.0010 - accuracy: 1.0000
- val_loss: 0.3296 - val_accuracy: 0.9250
Epoch 18/100
12/12 [==============================] - 2s 135ms/step - loss: 9.1311e-04 - accuracy: 1.0
000 - val_loss: 0.3206 - val_accuracy: 0.9250
Epoch 19/100
12/12 [==============================] - 2s 161ms/step - loss: 8.4465e-04 - accuracy: 1.0
000 - val_loss: 0.3255 - val_accuracy: 0.9250
Epoch 20/100
12/12 [==============================] - 2s 172ms/step - loss: 7.7014e-04 - accuracy: 1.0
000 - val_loss: 0.3247 - val_accuracy: 0.9250
Epoch 21/100
12/12 [==============================] - 2s 197ms/step - loss: 7.1552e-04 - accuracy: 1.0
000 - val_loss: 0.3265 - val_accuracy: 0.9250
Epoch 22/100
12/12 [==============================] - 2s 210ms/step - loss: 6.6159e-04 - accuracy: 1.0
000 - val_loss: 0.3346 - val_accuracy: 0.9250
Epoch 23/100
12/12 [==============================] - 2s 165ms/step - loss: 6.1311e-04 - accuracy: 1.0
000 - val_loss: 0.3332 - val_accuracy: 0.9250
Epoch 24/100
12/12 [==============================] - 2s 160ms/step - loss: 5.4952e-04 - accuracy: 1.0
000 - val_loss: 0.3301 - val_accuracy: 0.9250
Epoch 25/100
12/12 [==============================] - 2s 164ms/step - loss: 4.9795e-04 - accuracy: 1.0
000 - val_loss: 0.3384 - val_accuracy: 0.9250
Epoch 26/100
12/12 [==============================] - 2s 164ms/step - loss: 4.5939e-04 - accuracy: 1.0
000 - val_loss: 0.3363 - val_accuracy: 0.9250
Epoch 27/100
12/12 [==============================] - 2s 159ms/step - loss: 4.1979e-04 - accuracy: 1.0
000 - val_loss: 0.3337 - val_accuracy: 0.9250
Epoch 28/100
12/12 [==============================] - 2s 185ms/step - loss: 3.7944e-04 - accuracy: 1.0
000 - val_loss: 0.3328 - val_accuracy: 0.9250
Epoch 29/100
12/12 [==============================] - 2s 174ms/step - loss: 3.5350e-04 - accuracy: 1.0
000 - val_loss: 0.3396 - val_accuracy: 0.9250
Epoch 30/100
12/12 [==============================] - 3s 226ms/step - loss: 3.4606e-04 - accuracy: 1.0
000 - val_loss: 0.3444 - val_accuracy: 0.9250
Epoch 31/100
12/12 [==============================] - 2s 167ms/step - loss: 3.3276e-04 - accuracy: 1.0
000 - val_loss: 0.3416 - val_accuracy: 0.9250
Epoch 32/100
12/12 [==============================] - 2s 161ms/step - loss: 3.0815e-04 - accuracy: 1.0
000 - val_loss: 0.3441 - val_accuracy: 0.9250
Epoch 33/100
12/12 [==============================] - 2s 166ms/step - loss: 2.9637e-04 - accuracy: 1.0
000 - val_loss: 0.3480 - val_accuracy: 0.9250
Epoch 34/100
12/12 [==============================] - 2s 168ms/step - loss: 2.6275e-04 - accuracy: 1.0
000 - val_loss: 0.3456 - val_accuracy: 0.9250
Epoch 35/100
12/12 [==============================] - 2s 188ms/step - loss: 2.5677e-04 - accuracy: 1.0
000 - val_loss: 0.3465 - val_accuracy: 0.9250
Epoch 36/100
```

```
12/12 [==============================] - 3s 240ms/step - loss: 2.5194e-04 - accuracy: 1.0
000 - val_loss: 0.3498 - val_accuracy: 0.9250
Epoch 37/100
12/12 [==============================] - 2s 172ms/step - loss: 2.3479e-04 - accuracy: 1.0
000 - val_loss: 0.3542 - val_accuracy: 0.9250
Epoch 38/100
12/12 [==============================] - 2s 156ms/step - loss: 2.1493e-04 - accuracy: 1.0
000 - val_loss: 0.3540 - val_accuracy: 0.9250
Epoch 39/100
12/12 [==============================] - 2s 157ms/step - loss: 2.1126e-04 - accuracy: 1.0
000 - val_loss: 0.3526 - val_accuracy: 0.9250
Epoch 40/100
12/12 [==============================] - 2s 166ms/step - loss: 1.8998e-04 - accuracy: 1.0
000 - val_loss: 0.3540 - val_accuracy: 0.9250
Epoch 41/100
12/12 [==============================] - 2s 161ms/step - loss: 1.7645e-04 - accuracy: 1.0
000 - val_loss: 0.3589 - val_accuracy: 0.9250
Epoch 42/100
12/12 [==============================] - 2s 160ms/step - loss: 1.8210e-04 - accuracy: 1.0
000 - val_loss: 0.3575 - val_accuracy: 0.9250
Epoch 43/100
12/12 [==============================] - 2s 158ms/step - loss: 1.6523e-04 - accuracy: 1.0
000 - val_loss: 0.3544 - val_accuracy: 0.9250
Epoch 44/100
12/12 [==============================] - 2s 159ms/step - loss: 1.6562e-04 - accuracy: 1.0
000 - val_loss: 0.3560 - val_accuracy: 0.9250
Epoch 45/100
12/12 [==============================] - 2s 151ms/step - loss: 1.5847e-04 - accuracy: 1.0
000 - val_loss: 0.3576 - val_accuracy: 0.9250
Epoch 46/100
12/12 [==============================] - 2s 157ms/step - loss: 1.4632e-04 - accuracy: 1.0
000 - val_loss: 0.3572 - val_accuracy: 0.9250
Epoch 47/100
12/12 [==============================] - 2s 148ms/step - loss: 1.3839e-04 - accuracy: 1.0
000 - val_loss: 0.3569 - val_accuracy: 0.9250
Epoch 48/100
12/12 [==============================] - 2s 150ms/step - loss: 1.3903e-04 - accuracy: 1.0
000 - val_loss: 0.3538 - val_accuracy: 0.9250
Epoch 49/100
12/12 [==============================] - 2s 147ms/step - loss: 1.3508e-04 - accuracy: 1.0
000 - val_loss: 0.3547 - val_accuracy: 0.9250
Epoch 50/100
12/12 [==============================] - 2s 146ms/step - loss: 1.3078e-04 - accuracy: 1.0
000 - val_loss: 0.3585 - val_accuracy: 0.9250
Epoch 51/100
12/12 [==============================] - 2s 164ms/step - loss: 1.2634e-04 - accuracy: 1.0
000 - val_loss: 0.3589 - val_accuracy: 0.9250
Epoch 52/100
12/12 [==============================] - 2s 176ms/step - loss: 1.1765e-04 - accuracy: 1.0
000 - val_loss: 0.3573 - val_accuracy: 0.9250
Epoch 53/100
12/12 [==============================] - 2s 155ms/step - loss: 1.1176e-04 - accuracy: 1.0
000 - val_loss: 0.3596 - val_accuracy: 0.9250
Epoch 54/100
12/12 [==============================] - 2s 190ms/step - loss: 1.0713e-04 - accuracy: 1.0
000 - val_loss: 0.3631 - val_accuracy: 0.9250
Epoch 55/100
12/12 [==============================] - 2s 185ms/step - loss: 1.0004e-04 - accuracy: 1.0
000 - val_loss: 0.3620 - val_accuracy: 0.9250
Epoch 56/100
12/12 [==============================] - 2s 183ms/step - loss: 9.6106e-05 - accuracy: 1.0
000 - val_loss: 0.3642 - val_accuracy: 0.9250
Epoch 57/100
12/12 [==============================] - 2s 192ms/step - loss: 9.7267e-05 - accuracy: 1.0
000 - val_loss: 0.3604 - val_accuracy: 0.9250
Epoch 58/100
12/12 [==============================] - 2s 196ms/step - loss: 9.3347e-05 - accuracy: 1.0
000 - val_loss: 0.3609 - val_accuracy: 0.9250
Epoch 59/100
12/12 [==============================] - 2s 175ms/step - loss: 8.8025e-05 - accuracy: 1.0
000 - val_loss: 0.3633 - val_accuracy: 0.9250
Epoch 60/100
```

```
12/12 [==============================] - 2s 166ms/step - loss: 9.2371e-05 - accuracy: 1.0
000 - val_loss: 0.3687 - val_accuracy: 0.9250
Epoch 61/100
12/12 [==============================] - 2s 182ms/step - loss: 8.3583e-05 - accuracy: 1.0
000 - val_loss: 0.3677 - val_accuracy: 0.9250
Epoch 62/100
12/12 [==============================] - 2s 188ms/step - loss: 8.3838e-05 - accuracy: 1.0
000 - val_loss: 0.3650 - val_accuracy: 0.9250
Epoch 63/100
12/12 [==============================] - 2s 160ms/step - loss: 7.6480e-05 - accuracy: 1.0
000 - val_loss: 0.3668 - val_accuracy: 0.9250
Epoch 64/100
12/12 [==============================] - 2s 152ms/step - loss: 7.1418e-05 - accuracy: 1.0
000 - val_loss: 0.3681 - val_accuracy: 0.9250
Epoch 65/100
12/12 [==============================] - 2s 184ms/step - loss: 7.1219e-05 - accuracy: 1.0
000 - val_loss: 0.3651 - val_accuracy: 0.9250
Epoch 66/100
12/12 [==============================] - 3s 224ms/step - loss: 7.2584e-05 - accuracy: 1.0
000 - val_loss: 0.3632 - val_accuracy: 0.9312
Epoch 67/100
12/12 [==============================] - 3s 215ms/step - loss: 6.4022e-05 - accuracy: 1.0
000 - val_loss: 0.3653 - val_accuracy: 0.9312
Epoch 68/100
12/12 [==============================] - 3s 229ms/step - loss: 6.3127e-05 - accuracy: 1.0
000 - val_loss: 0.3649 - val_accuracy: 0.9250
Epoch 69/100
12/12 [==============================] - 2s 181ms/step - loss: 6.6416e-05 - accuracy: 1.0
000 - val_loss: 0.3699 - val_accuracy: 0.9250
Epoch 70/100
12/12 [==============================] - 3s 216ms/step - loss: 6.0323e-05 - accuracy: 1.0
000 - val_loss: 0.3727 - val_accuracy: 0.9250
Epoch 71/100
12/12 [==============================] - 2s 181ms/step - loss: 6.0059e-05 - accuracy: 1.0
000 - val_loss: 0.3701 - val_accuracy: 0.9250
Epoch 72/100
12/12 [==============================] - 2s 167ms/step - loss: 5.6603e-05 - accuracy: 1.0
000 - val_loss: 0.3711 - val_accuracy: 0.9250
Epoch 73/100
12/12 [==============================] - 2s 183ms/step - loss: 5.6052e-05 - accuracy: 1.0
000 - val_loss: 0.3701 - val_accuracy: 0.9250
Epoch 74/100
12/12 [==============================] - 2s 177ms/step - loss: 5.4654e-05 - accuracy: 1.0
000 - val_loss: 0.3711 - val_accuracy: 0.9250
Epoch 75/100
12/12 [==============================] - 3s 212ms/step - loss: 5.4021e-05 - accuracy: 1.0
000 - val_loss: 0.3735 - val_accuracy: 0.9250
Epoch 76/100
12/12 [==============================] - 2s 199ms/step - loss: 5.2400e-05 - accuracy: 1.0
000 - val_loss: 0.3718 - val_accuracy: 0.9250
Epoch 77/100
12/12 [==============================] - 2s 167ms/step - loss: 5.0378e-05 - accuracy: 1.0
000 - val_loss: 0.3767 - val_accuracy: 0.9250
Epoch 78/100
12/12 [==============================] - 2s 157ms/step - loss: 4.7491e-05 - accuracy: 1.0
000 - val_loss: 0.3758 - val_accuracy: 0.9250
Epoch 79/100
12/12 [==============================] - 2s 161ms/step - loss: 4.8760e-05 - accuracy: 1.0
000 - val_loss: 0.3732 - val_accuracy: 0.9250
Epoch 80/100
12/12 [==============================] - 2s 166ms/step - loss: 4.3713e-05 - accuracy: 1.0
000 - val_loss: 0.3716 - val_accuracy: 0.9250
Epoch 81/100
12/12 [==============================] - 2s 164ms/step - loss: 4.7964e-05 - accuracy: 1.0
000 - val_loss: 0.3727 - val_accuracy: 0.9250
Epoch 82/100
12/12 [==============================] - 2s 164ms/step - loss: 4.3574e-05 - accuracy: 1.0
000 - val_loss: 0.3756 - val_accuracy: 0.9312
Epoch 83/100
12/12 [==============================] - 2s 181ms/step - loss: 4.2792e-05 - accuracy: 1.0
000 - val_loss: 0.3732 - val_accuracy: 0.9312
Epoch 84/100
```

```
12/12 [==============================] - 2s 205ms/step - loss: 4.2493e-05 - accuracy: 1.0
000 - val_loss: 0.3738 - val_accuracy: 0.9312
Epoch 85/100
12/12 [==============================] - 2s 156ms/step - loss: 4.0514e-05 - accuracy: 1.0
000 - val_loss: 0.3759 - val_accuracy: 0.9250
Epoch 86/100
12/12 [==============================] - 2s 173ms/step - loss: 3.7766e-05 - accuracy: 1.0
000 - val_loss: 0.3754 - val_accuracy: 0.9250
Epoch 87/100
12/12 [==============================] - 2s 177ms/step - loss: 3.6111e-05 - accuracy: 1.0
000 - val_loss: 0.3790 - val_accuracy: 0.9250
Epoch 88/100
12/12 [==============================] - 2s 166ms/step - loss: 3.4860e-05 - accuracy: 1.0
000 - val_loss: 0.3801 - val_accuracy: 0.9250
Epoch 89/100
12/12 [==============================] - 2s 151ms/step - loss: 3.4073e-05 - accuracy: 1.0
000 - val_loss: 0.3800 - val_accuracy: 0.9250
Epoch 90/100
12/12 [==============================] - 2s 154ms/step - loss: 3.5239e-05 - accuracy: 1.0
000 - val_loss: 0.3775 - val_accuracy: 0.9312
Epoch 91/100
12/12 [==============================] - 2s 154ms/step - loss: 3.5677e-05 - accuracy: 1.0
000 - val_loss: 0.3788 - val_accuracy: 0.9250
Epoch 92/100
12/12 [==============================] - 2s 155ms/step - loss: 3.7037e-05 - accuracy: 1.0
000 - val_loss: 0.3834 - val_accuracy: 0.9250
Epoch 93/100
12/12 [==============================] - 2s 153ms/step - loss: 3.3418e-05 - accuracy: 1.0
000 - val_loss: 0.3804 - val_accuracy: 0.9312
Epoch 94/100
12/12 [==============================] - 2s 150ms/step - loss: 3.3287e-05 - accuracy: 1.0
000 - val_loss: 0.3803 - val_accuracy: 0.9312
Epoch 95/100
12/12 [==============================] - 2s 153ms/step - loss: 2.9443e-05 - accuracy: 1.0
000 - val_loss: 0.3835 - val_accuracy: 0.9250
Epoch 96/100
12/12 [==============================] - 2s 156ms/step - loss: 3.3621e-05 - accuracy: 1.0
000 - val_loss: 0.3835 - val_accuracy: 0.9250
Epoch 97/100
12/12 [==============================] - 2s 161ms/step - loss: 2.9620e-05 - accuracy: 1.0
000 - val_loss: 0.3869 - val_accuracy: 0.9250
Epoch 98/100
12/12 [==============================] - 2s 169ms/step - loss: 2.9345e-05 - accuracy: 1.0
000 - val_loss: 0.3861 - val_accuracy: 0.9250
Epoch 99/100
12/12 [==============================] - 2s 202ms/step - loss: 2.9358e-05 - accuracy: 1.0
000 - val_loss: 0.3860 - val_accuracy: 0.9250
Epoch 100/100
12/12 [==============================] - 2s 185ms/step - loss: 2.7933e-05 - accuracy: 1.0
000 - val_loss: 0.3851 - val_accuracy: 0.9312
```

In [22]:

```
print('Training Score :', model.evaluate(trainX_reshaped, trainY)[1])
print('Testing Score :', model.evaluate(testX_reshaped, testY)[1])
# Model is overfitted
```

```
8/8 [==============================] - 1s 39ms/step - loss: 2.3681e-05 - accuracy: 1.0000
Training Score : 1.0
5/5 [==============================] - 0s 45ms/step - loss: 0.3851 - accuracy: 0.9312
Testing Score : 0.9312499761581421
```

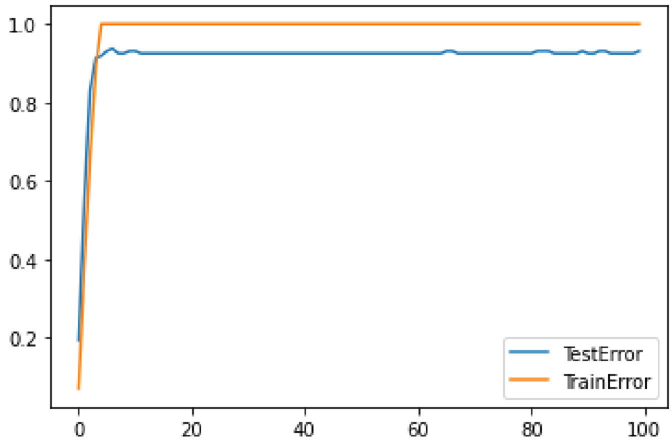
In [23]:

```
# plotting
#%matplotlib inline
plt.plot(history.history['val_accuracy'])
plt.plot(history.history['accuracy'])
plt.legend(['TestError','TrainError'])

#model.predict_classes(testX_reshaped)
```

Out[23]:

<matplotlib.legend.Legend at 0x7fa8441ae820>



In [ ]: