IMPORTING ALL THE REQUIRED LIBRARIES

[28]:
```
import numpy as np,pandas as pd
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Flatten,Dense,Dropout
from keras.optimizers import Adam

from sklearn.model_selection import train_test_split
import cv2
```

PRINT SOME IMAGES

[5]:
```
import matplotlib.pyplot as plt
from numpy import load
data=load('/content/ORL_faces.npz')
lst =data.files
for item in lst:
  print(item)
  print(data[item])
```

```
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]
```

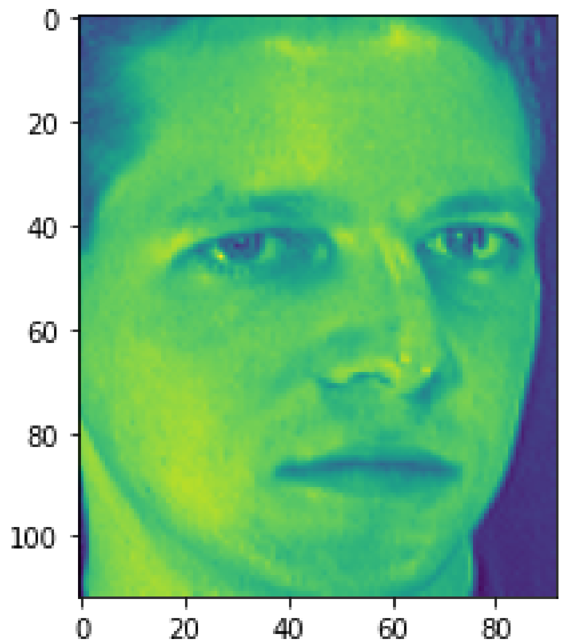
[11]:

```python
x_train= data['trainX']
x_train= np.array(x_train, dtype='float32')/255.
x_test = data['testX']
x_test= np.array(x_test, dtype='float32')/255.
y_train= data['trainY']
y_test= data['testY']
print('x_train:{}'.format(x_train.shape))
print('x_test:{}'.format(x_test.shape))
print('y_train:{}'.format(y_train.shape))
print('y_test:{}'.format(y_test.shape))
```

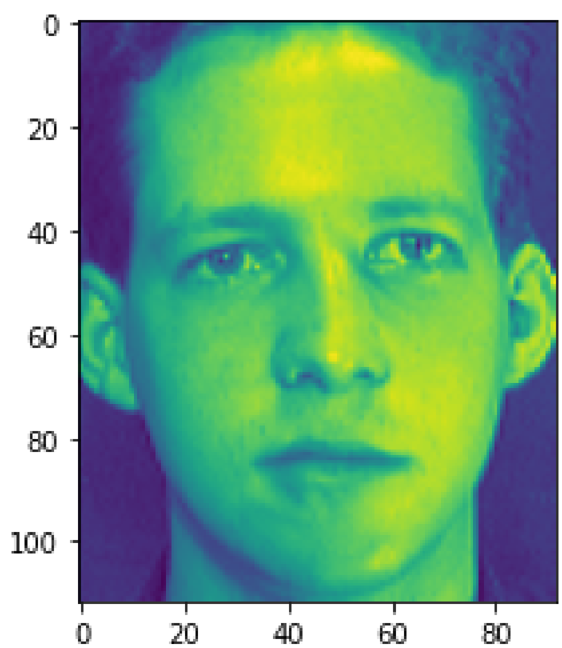
```
x_train:(240, 10304)
x_test:(160, 10304)
y_train:(240,)
y_test:(160,)
```

[12]:

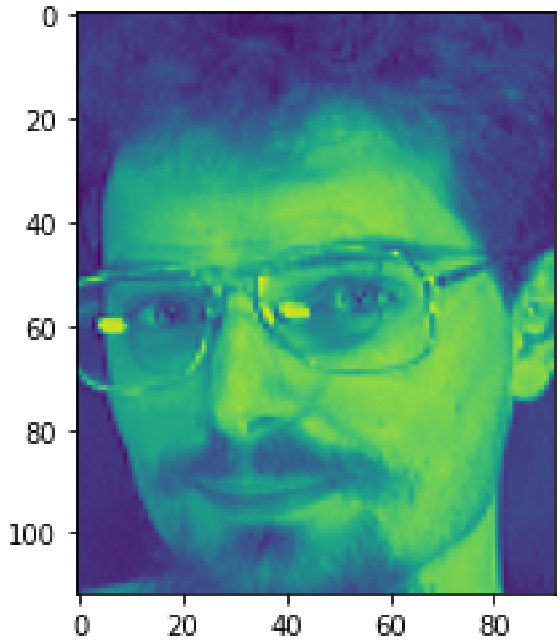
```python
img_train = x_train[1].reshape(112, 92)
plt.imshow(img_train)
plt.show()
```



[14]:
```
img_test = x_test[1].reshape(112,92)
plt.imshow(img_test)
plt.show()
```



[15]:
```python
img_test = x_test[128].reshape(112,92)
plt.imshow(img_test)
plt.show()
```



[20]:
```python
x_train, x_valid, y_train, y_valid = train_test_split(x_train, y_train,␣
↪test_size=0.2)
```

[21]:
```python
img_rows= 112
img_cols= 92
batch_size = 512
im_shape=(img_rows, img_cols, 1)
x_train = x_train.reshape(x_train.shape[0], *im_shape)
x_valid = x_valid.reshape(x_valid.shape[0], *im_shape)
```

[22]:
```python
x_train.shape
```

[22]: (192, 112, 92, 1)

[24]:
```python
data['trainX'].shape
```

[24]: (240, 10304)

[38]:
```python
model = Sequential()
model.add(Conv2D(36,(7,7),1, activation='relu', input_shape= im_shape))
```

```
model.add(MaxPooling2D(pool_size=2))
model.add(Conv2D(54,(5,5),1, activation='relu',input_shape=im_shape))

model.add(MaxPooling2D(pool_size=2))
model.add(Flatten())

model.add(Dense(2024, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))

#40 is the number of outputs
model.add(Dense(40, activation='softmax'))
model.
 ↪compile(loss='sparse_categorical_crossentropy',optimizer=Adam(learning_rate␣
 ↪=0.01),metrics=['accuracy'])
model.summary()
```

```
Model: "sequential_2"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 106, 86, 36)       1800

 max_pooling2d_4 (MaxPooling  (None, 53, 43, 36)       0
 2D)

 conv2d_5 (Conv2D)           (None, 49, 39, 54)        48654

 max_pooling2d_5 (MaxPooling  (None, 24, 19, 54)       0
 2D)

 flatten_2 (Flatten)         (None, 24624)             0

 dense_5 (Dense)             (None, 2024)              49841000

 dropout_3 (Dropout)         (None, 2024)              0

 dense_6 (Dense)             (None, 1024)              2073600

 dropout_4 (Dropout)         (None, 1024)              0

 dense_7 (Dense)             (None, 512)               524800
```

```
 dropout_5 (Dropout)         (None, 512)               0

 dense_8 (Dense)             (None, 40)                20520

=================================================================
Total params: 52,510,374
Trainable params: 52,510,374
Non-trainable params: 0
_________________________________________________________________
```

[39]:

```
history= model.fit(np.array(x_train), np.array(y_train), batch_size= 512,
 epochs=100, verbose=2,
 validation_data=(np.array(x_valid),np.array(y_valid)))
```

```
Epoch 1/100
1/1 - 6s - loss: 3.7223 - accuracy: 0.0052 - val_loss: 300.7651 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 2/100
1/1 - 5s - loss: 374.5501 - accuracy: 0.0417 - val_loss: 3.9468 - val_accuracy:
0.0625 - 5s/epoch - 5s/step
Epoch 3/100
1/1 - 5s - loss: 6.1254 - accuracy: 0.0469 - val_loss: 10.2724 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 4/100
1/1 - 5s - loss: 15.9092 - accuracy: 0.0312 - val_loss: 5.0859 - val_accuracy:
0.0833 - 5s/epoch - 5s/step
Epoch 5/100
1/1 - 5s - loss: 11.4774 - accuracy: 0.0781 - val_loss: 3.9432 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 6/100
1/1 - 5s - loss: 4.6153 - accuracy: 0.0573 - val_loss: 3.2436 - val_accuracy:
0.0625 - 5s/epoch - 5s/step
Epoch 7/100
1/1 - 5s - loss: 3.2816 - accuracy: 0.0521 - val_loss: 3.2469 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 8/100
1/1 - 5s - loss: 3.2651 - accuracy: 0.0208 - val_loss: 3.2158 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 9/100
1/1 - 5s - loss: 3.2164 - accuracy: 0.0156 - val_loss: 3.1220 - val_accuracy:
0.0833 - 5s/epoch - 5s/step
Epoch 10/100
1/1 - 5s - loss: 3.1612 - accuracy: 0.0573 - val_loss: 3.1366 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 11/100
1/1 - 6s - loss: 3.2434 - accuracy: 0.0417 - val_loss: 3.2181 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
```

```
Epoch 12/100
1/1 - 5s - loss: 3.2055 - accuracy: 0.0312 - val_loss: 3.2224 - val_accuracy:
0.0625 - 5s/epoch - 5s/step
Epoch 13/100
1/1 - 6s - loss: 3.2112 - accuracy: 0.0625 - val_loss: 3.2012 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 14/100
1/1 - 5s - loss: 3.1383 - accuracy: 0.0781 - val_loss: 3.1617 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 15/100
1/1 - 5s - loss: 3.1912 - accuracy: 0.0677 - val_loss: 3.1471 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 16/100
1/1 - 5s - loss: 3.1391 - accuracy: 0.0573 - val_loss: 3.2316 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 17/100
1/1 - 5s - loss: 3.2095 - accuracy: 0.0260 - val_loss: 3.2308 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 18/100
1/1 - 6s - loss: 3.1688 - accuracy: 0.0677 - val_loss: 3.2276 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 19/100
1/1 - 5s - loss: 3.1412 - accuracy: 0.0781 - val_loss: 3.1939 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 20/100
1/1 - 5s - loss: 3.1058 - accuracy: 0.0573 - val_loss: 3.1336 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 21/100
1/1 - 5s - loss: 3.1004 - accuracy: 0.0833 - val_loss: 3.1058 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 22/100
1/1 - 5s - loss: 3.1107 - accuracy: 0.0417 - val_loss: 3.0972 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 23/100
1/1 - 5s - loss: 3.1138 - accuracy: 0.0729 - val_loss: 3.1064 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 24/100
1/1 - 5s - loss: 3.0802 - accuracy: 0.1042 - val_loss: 3.0954 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 25/100
1/1 - 6s - loss: 3.0580 - accuracy: 0.0677 - val_loss: 3.0889 - val_accuracy:
0.0625 - 6s/epoch - 6s/step
Epoch 26/100
1/1 - 5s - loss: 3.0562 - accuracy: 0.0521 - val_loss: 3.0898 - val_accuracy:
0.0625 - 5s/epoch - 5s/step
Epoch 27/100
1/1 - 5s - loss: 3.1176 - accuracy: 0.0312 - val_loss: 3.0966 - val_accuracy:
0.0625 - 5s/epoch - 5s/step
```

```
Epoch 28/100
1/1 - 5s - loss: 3.0431 - accuracy: 0.0625 - val_loss: 3.1016 - val_accuracy:
0.0625 - 5s/epoch - 5s/step
Epoch 29/100
1/1 - 5s - loss: 3.0631 - accuracy: 0.0469 - val_loss: 3.0952 - val_accuracy:
0.0625 - 5s/epoch - 5s/step
Epoch 30/100
1/1 - 5s - loss: 3.0403 - accuracy: 0.0469 - val_loss: 3.0911 - val_accuracy:
0.0625 - 5s/epoch - 5s/step
Epoch 31/100
1/1 - 5s - loss: 3.0827 - accuracy: 0.0573 - val_loss: 3.0932 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 32/100
1/1 - 5s - loss: 3.0338 - accuracy: 0.0417 - val_loss: 3.0933 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 33/100
1/1 - 5s - loss: 3.0387 - accuracy: 0.0625 - val_loss: 3.0978 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 34/100
1/1 - 5s - loss: 3.0357 - accuracy: 0.0677 - val_loss: 3.1061 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 35/100
1/1 - 5s - loss: 3.0337 - accuracy: 0.0521 - val_loss: 3.1087 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 36/100
1/1 - 5s - loss: 3.0435 - accuracy: 0.0469 - val_loss: 3.1116 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 37/100
1/1 - 6s - loss: 3.0465 - accuracy: 0.0781 - val_loss: 3.1180 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 38/100
1/1 - 5s - loss: 3.0484 - accuracy: 0.0365 - val_loss: 3.1115 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 39/100
1/1 - 5s - loss: 3.0491 - accuracy: 0.0625 - val_loss: 3.1050 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 40/100
1/1 - 5s - loss: 3.0224 - accuracy: 0.0365 - val_loss: 3.0980 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 41/100
1/1 - 5s - loss: 3.0517 - accuracy: 0.0521 - val_loss: 3.0925 - val_accuracy:
0.0417 - 5s/epoch - 5s/step
Epoch 42/100
1/1 - 6s - loss: 3.0329 - accuracy: 0.0469 - val_loss: 3.0881 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 43/100
1/1 - 5s - loss: 3.0407 - accuracy: 0.0521 - val_loss: 3.0846 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
```

```
Epoch 44/100
1/1 - 6s - loss: 2.9928 - accuracy: 0.0677 - val_loss: 3.0818 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 45/100
1/1 - 5s - loss: 3.0124 - accuracy: 0.0573 - val_loss: 3.0810 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 46/100
1/1 - 5s - loss: 3.0357 - accuracy: 0.0521 - val_loss: 3.0811 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 47/100
1/1 - 5s - loss: 3.0097 - accuracy: 0.0781 - val_loss: 3.0782 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 48/100
1/1 - 5s - loss: 3.0572 - accuracy: 0.0365 - val_loss: 3.0750 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 49/100
1/1 - 6s - loss: 3.0241 - accuracy: 0.0469 - val_loss: 3.0721 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 50/100
1/1 - 5s - loss: 3.0473 - accuracy: 0.0469 - val_loss: 3.0708 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 51/100
1/1 - 5s - loss: 3.0119 - accuracy: 0.0573 - val_loss: 3.0715 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 52/100
1/1 - 5s - loss: 3.0128 - accuracy: 0.0521 - val_loss: 3.0715 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 53/100
1/1 - 5s - loss: 3.0239 - accuracy: 0.0417 - val_loss: 3.0719 - val_accuracy:
0.0417 - 5s/epoch - 5s/step
Epoch 54/100
1/1 - 5s - loss: 3.0517 - accuracy: 0.0729 - val_loss: 3.0729 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 55/100
1/1 - 5s - loss: 3.0039 - accuracy: 0.0417 - val_loss: 3.0720 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 56/100
1/1 - 5s - loss: 3.0285 - accuracy: 0.0521 - val_loss: 3.0725 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 57/100
1/1 - 5s - loss: 3.0147 - accuracy: 0.0521 - val_loss: 3.0741 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 58/100
1/1 - 6s - loss: 3.0185 - accuracy: 0.0469 - val_loss: 3.0737 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 59/100
1/1 - 5s - loss: 3.0391 - accuracy: 0.0469 - val_loss: 3.0740 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
```

```
Epoch 60/100
1/1 - 5s - loss: 3.0081 - accuracy: 0.0625 - val_loss: 3.0737 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 61/100
1/1 - 6s - loss: 3.0012 - accuracy: 0.0729 - val_loss: 3.0737 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 62/100
1/1 - 5s - loss: 3.0158 - accuracy: 0.0521 - val_loss: 3.0736 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 63/100
1/1 - 6s - loss: 2.9898 - accuracy: 0.0729 - val_loss: 3.0720 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 64/100
1/1 - 5s - loss: 3.0128 - accuracy: 0.0573 - val_loss: 3.0688 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 65/100
1/1 - 6s - loss: 3.0033 - accuracy: 0.0365 - val_loss: 3.0624 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 66/100
1/1 - 5s - loss: 3.0212 - accuracy: 0.0469 - val_loss: 3.0631 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 67/100
1/1 - 5s - loss: 3.0228 - accuracy: 0.0573 - val_loss: 3.0671 - val_accuracy:
0.0417 - 5s/epoch - 5s/step
Epoch 68/100
1/1 - 5s - loss: 3.0232 - accuracy: 0.0521 - val_loss: 3.0719 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 69/100
1/1 - 5s - loss: 3.0208 - accuracy: 0.0469 - val_loss: 3.0753 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 70/100
1/1 - 5s - loss: 3.0077 - accuracy: 0.0417 - val_loss: 3.0760 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 71/100
1/1 - 5s - loss: 3.0039 - accuracy: 0.0521 - val_loss: 3.0754 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 72/100
1/1 - 5s - loss: 2.9854 - accuracy: 0.0885 - val_loss: 3.0756 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 73/100
1/1 - 5s - loss: 3.0249 - accuracy: 0.0365 - val_loss: 3.0752 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 74/100
1/1 - 5s - loss: 3.0059 - accuracy: 0.0625 - val_loss: 3.0743 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 75/100
1/1 - 5s - loss: 2.9966 - accuracy: 0.0469 - val_loss: 3.0745 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
```

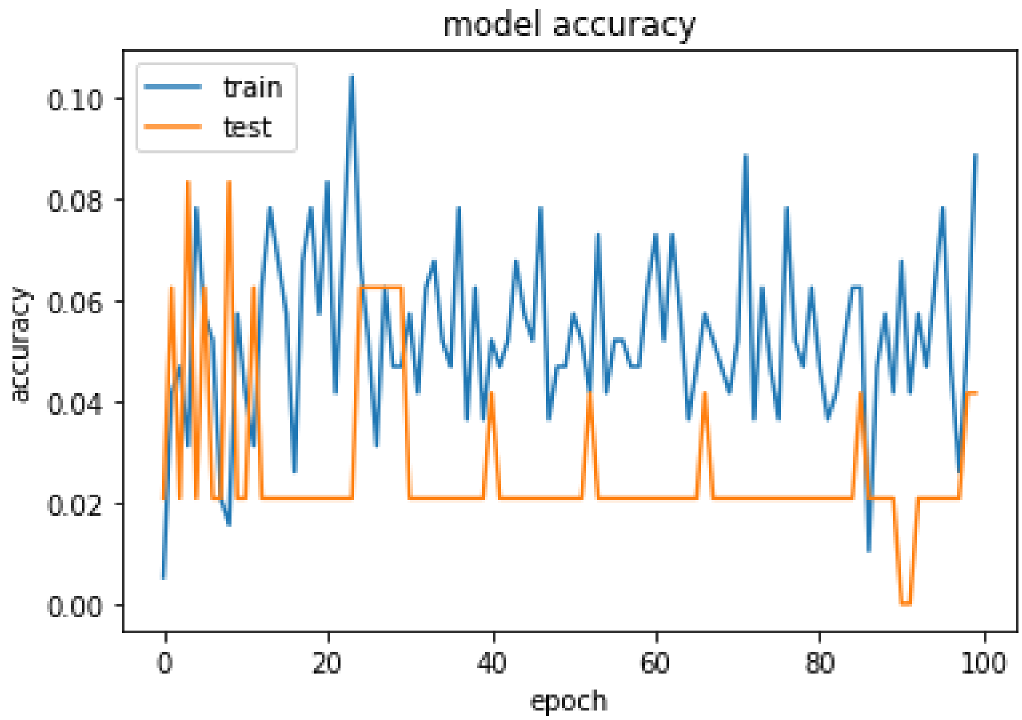
```
Epoch 76/100
1/1 - 5s - loss: 2.9963 - accuracy: 0.0365 - val_loss: 3.0756 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 77/100
1/1 - 5s - loss: 2.9942 - accuracy: 0.0781 - val_loss: 3.0762 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 78/100
1/1 - 5s - loss: 3.0164 - accuracy: 0.0521 - val_loss: 3.0722 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 79/100
1/1 - 5s - loss: 3.0073 - accuracy: 0.0469 - val_loss: 3.0686 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 80/100
1/1 - 5s - loss: 3.0179 - accuracy: 0.0625 - val_loss: 3.0672 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 81/100
1/1 - 5s - loss: 3.0285 - accuracy: 0.0469 - val_loss: 3.0650 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 82/100
1/1 - 5s - loss: 3.0008 - accuracy: 0.0365 - val_loss: 3.0628 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 83/100
1/1 - 5s - loss: 3.0098 - accuracy: 0.0417 - val_loss: 3.0604 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 84/100
1/1 - 5s - loss: 3.0028 - accuracy: 0.0521 - val_loss: 3.0595 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 85/100
1/1 - 6s - loss: 3.0183 - accuracy: 0.0625 - val_loss: 3.0592 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 86/100
1/1 - 5s - loss: 3.0031 - accuracy: 0.0625 - val_loss: 3.0596 - val_accuracy:
0.0417 - 5s/epoch - 5s/step
Epoch 87/100
1/1 - 5s - loss: 3.0203 - accuracy: 0.0104 - val_loss: 3.0594 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 88/100
1/1 - 5s - loss: 3.0075 - accuracy: 0.0469 - val_loss: 3.0592 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 89/100
1/1 - 6s - loss: 3.0131 - accuracy: 0.0573 - val_loss: 3.0578 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 90/100
1/1 - 5s - loss: 3.0154 - accuracy: 0.0417 - val_loss: 3.0574 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 91/100
1/1 - 6s - loss: 2.9818 - accuracy: 0.0677 - val_loss: 3.0565 - val_accuracy:
0.0000e+00 - 6s/epoch - 6s/step
```

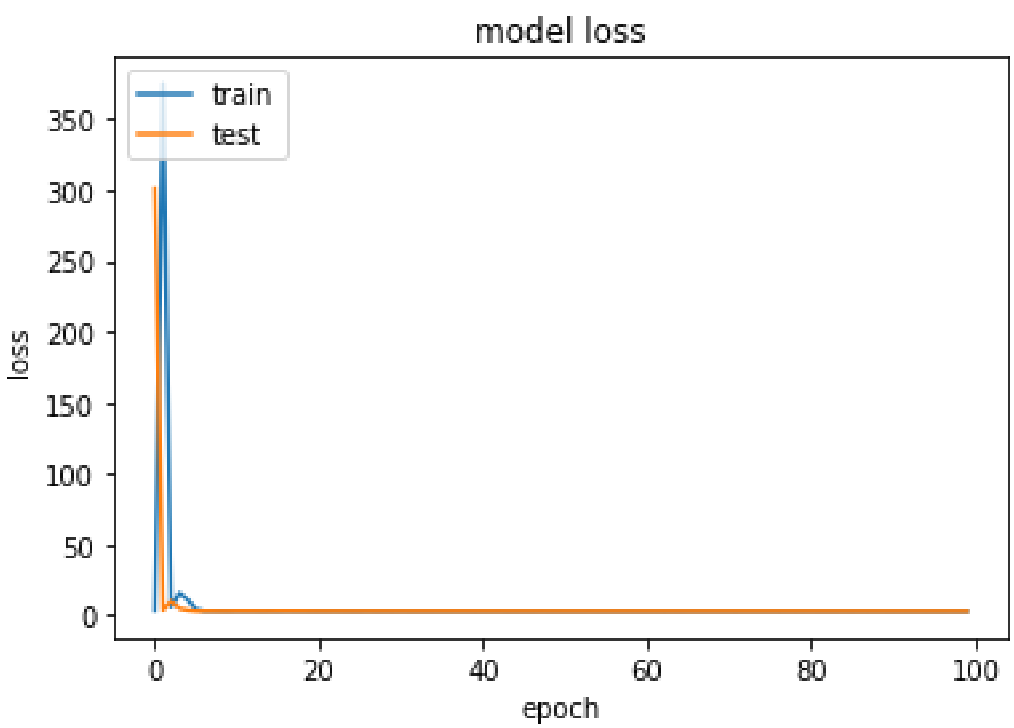
```
Epoch 92/100
1/1 - 5s - loss: 2.9974 - accuracy: 0.0417 - val_loss: 3.0544 - val_accuracy:
0.0000e+00 - 5s/epoch - 5s/step
Epoch 93/100
1/1 - 5s - loss: 2.9867 - accuracy: 0.0573 - val_loss: 3.0539 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 94/100
1/1 - 5s - loss: 3.0141 - accuracy: 0.0469 - val_loss: 3.0530 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 95/100
1/1 - 5s - loss: 2.9791 - accuracy: 0.0625 - val_loss: 3.0548 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 96/100
1/1 - 6s - loss: 2.9926 - accuracy: 0.0781 - val_loss: 3.0587 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 97/100
1/1 - 5s - loss: 2.9929 - accuracy: 0.0469 - val_loss: 3.0640 - val_accuracy:
0.0208 - 5s/epoch - 5s/step
Epoch 98/100
1/1 - 6s - loss: 3.0123 - accuracy: 0.0260 - val_loss: 3.0667 - val_accuracy:
0.0208 - 6s/epoch - 6s/step
Epoch 99/100
1/1 - 5s - loss: 2.9860 - accuracy: 0.0521 - val_loss: 3.0682 - val_accuracy:
0.0417 - 5s/epoch - 5s/step
Epoch 100/100
1/1 - 5s - loss: 2.9735 - accuracy: 0.0885 - val_loss: 3.0682 - val_accuracy:
0.0417 - 5s/epoch - 5s/step
```

[40]:

```python
print(history.history.keys())
# summarize 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()
#summarize 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='upper left')
plt.show()
```

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


model accuracy
train
test
0.10
0.08
0.06
0.04
0.02
0.00
accuracy
0
20
40
60
80
100
epoch


model loss
train
test
350
300
250
200
150
100
50
0
loss
0
20
40
60
80
100
epoch