In [95]:
```
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Dropout, Conv2D,MaxPool2D, BatchNo
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
```

In [96]:
```
#import data from .npz file
data = np.load("ORL_faces.npz")
```

In [97]:
```
#split and scale the data
trainY = data["trainY"]
trainX = data["trainX"]
testY = data["testY"]
testX = data["testX"]
scaler=MinMaxScaler()
```

In [98]:
```
trainX = scaler.fit(trainX).transform(trainX)
testX = scaler.transform(testX)
```

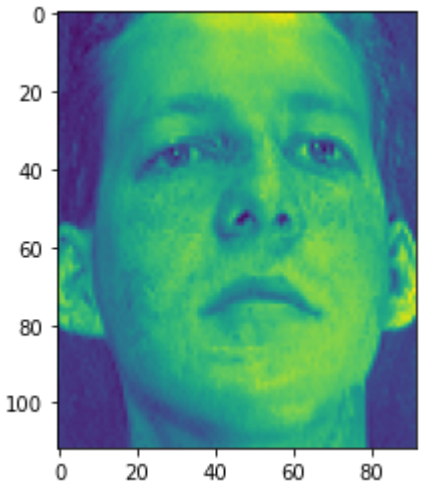
In [99]:
```
#reshape inputs, encode categories
trainY = to_categorical(trainY)
testY = to_categorical(testY)
trainX = trainX.reshape(240,112,92,1)
testX = testX.reshape(160,112,92,1)
trainY.shape,trainX.shape,"||||",testY.shape,testX.shape
```

Out[99]: ((240, 20), (240, 112, 92, 1), '||||', (160, 20), (160, 112, 92, 1))

In [100…

```
#verify correct shaping
plt.imshow(testX[2])
```

Out[100… <matplotlib.image.AxesImage at 0x1cb190fed90>



In [149…

```
#construct the model
model = Sequential()

model.add(
    Conv2D(64,input_shape=(112,92,1), kernel_size=(5,5), activation="relu")
)
model.add(Dropout(.5))

model.add(
    MaxPool2D((2,2),strides=(2,2))
)
model.add(
    Conv2D(64, kernel_size=(5,5), activation="relu")
)
model.add(Flatten())
model.add(
    Dense(20, activation="softmax")
)

model.compile(loss=tf.keras.losses.CategoricalCrossentropy(),optimizer="adam",
```

In [150…

```
model.summary()
```

```
Model: "sequential_22"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_38 (Conv2D)           (None, 108, 88, 64)       1664
_________________________________________________________________
dropout_2 (Dropout)          (None, 108, 88, 64)       0
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 54, 44, 64)        0
_________________________________________________________________
conv2d_39 (Conv2D)           (None, 50, 40, 64)        102464
_________________________________________________________________
```

```
flatten_9 (Flatten)          (None, 128000)            0
_________________________________________________________________
dense_16 (Dense)             (None, 20)                2560020
=================================================================
Total params: 2,664,148
Trainable params: 2,664,148
Non-trainable params: 0
```

In [151…
```
#train the model
model.fit(trainX,trainY, epochs=5)
```

```
Epoch 1/5
8/8 [==============================] - 5s 592ms/step - loss: 3.9538 - accurac
y: 0.1625
Epoch 2/5
8/8 [==============================] - 5s 597ms/step - loss: 2.2838 - accurac
y: 0.4875
Epoch 3/5
8/8 [==============================] - 5s 610ms/step - loss: 0.6996 - accurac
y: 0.8833
Epoch 4/5
8/8 [==============================] - 5s 632ms/step - loss: 0.1733 - accurac
y: 0.9708
Epoch 5/5
8/8 [==============================] - 5s 636ms/step - loss: 0.0429 - accurac
y: 0.9958
```

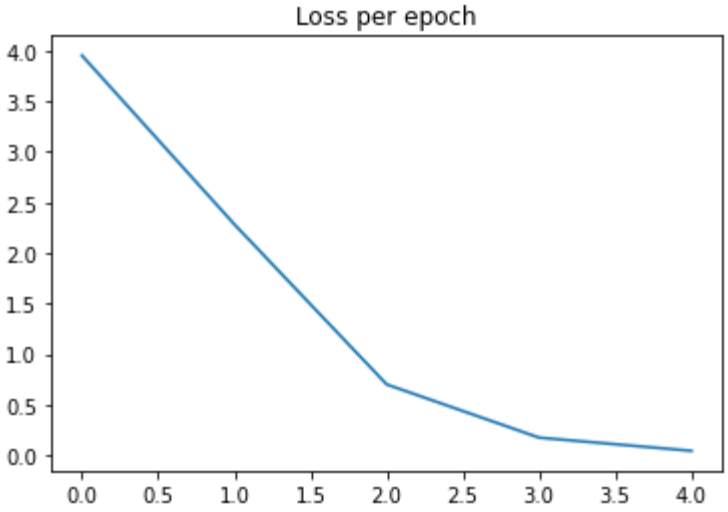
Out[151… `<tensorflow.python.keras.callbacks.History at 0x1cb1f8adf40>`

In [152…
```
results = model.history.history
```

In [153…
```
results
```

Out[153…
```
{'loss': [3.9538192749023438,
  2.283787488937378,
  0.6996336579322815,
  0.1732865869998932,
  0.042905766516923904],
 'accuracy': [0.16249999403953552,
  0.48750001192092896,
  0.8833333253860474,
  0.9708333611488342,
  0.9958333373069763]}
```

In [175…
```
#plot the resulting losses
plt.plot(results["loss"])
plt.title("Loss per epoch")
plt.show()
```



In [155…
```
# Evaluate the model
model.evaluate(testX,testY)
```

```
5/5 [==============================] - 1s 128ms/step - loss: 0.3972 - accurac
y: 0.9438
```

Out[155… `[0.3972092270851135, 0.9437500238418579]`

In [174…
```
#save the model
model.save("facial_recognition_.9438_accuracy.hd5")
```

```
WARNING:tensorflow:From C:\Users\perry\anaconda3\lib\site-packages\tensorflow\
python\training\tracking\tracking.py:111: Model.state_updates (from tensorflo
w.python.keras.engine.training) is deprecated and will be removed in a future
version.
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied aut
omatically.
WARNING:tensorflow:From C:\Users\perry\anaconda3\lib\site-packages\tensorflow\
python\training\tracking\tracking.py:111: Layer.updates (from tensorflow.pytho
n.keras.engine.base_layer) is deprecated and will be removed in a future versi
on.
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied aut
omatically.
INFO:tensorflow:Assets written to: facial_recognition_.9438_accuracy.hd5\asset
s
```

In [ ]: