In [1]:
```
#importing librabry
from tensorflow.keras.layers import Conv2D,Dense,Flatten,MaxPooling2D,Dropout
from tensorflow.keras.models import Sequential
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import cv2
from sklearn.model_selection import train_test_split
from keras_visualizer import visualizer
from sklearn.metrics import *
```

In [2]:
```
path1=r"C:\Users\venka\Downloads\oxford_102\dataset\train"
lt=os.listdir(path1)
faces=[]
label=[]
n1=0
for i in lt:
    path = f"{path1}/{i}"
    n=0
    for j in os.listdir(path):
        img=cv2.imread(f"{path}/{j}")
        img=cv2.resize(img,(64,64))
        faces.append(img)
        label.append(int(i))
```

In [3]:
```
faces=np.array(faces)
label=np.array(label)
#Normalization
# faces=faces/225
# label=label/225
print(faces.shape,label.shape)
```

```
(6552, 64, 64, 3) (6552,)
```

In [4]:
```
x_train,x_test,y_train,y_test = train_test_split(faces,label,test_size=0.30,random_
```

In [6]:
```
#creating deeplearning model
model = Sequential()
#adding convolutional layer acts as input layer
model.add(Conv2D(64,5,input_shape=(64,64,3),padding="same",activation="relu"))
model.add(MaxPooling2D())
model.add(Conv2D(32,3,padding="same",activation="relu"))
model.add(MaxPooling2D())
model.add(Conv2D(32,3,padding="same",activation="relu"))
model.add(MaxPooling2D())
model.add(Flatten())
model.add(Dense(512,activation="relu"))
model.add(Dense(256,activation="relu"))
model.add(Dense(103,activation="Softmax"))
```

In [10]:
```
#view model summary
model.compile(optimizer="adam",loss="sparse_categorical_crossentropy",
              metrics=['accuracy'])
history=model.fit(x_train,x_test,epochs=20)
```

```
Epoch 1/20
205/205 [==============================] - 20s 97ms/step - loss: 0.7011 - accurac
y: 0.8396
Epoch 2/20
205/205 [==============================] - 40s 196ms/step - loss: 0.5530 - accurac
y: 0.8771
Epoch 3/20
205/205 [==============================] - 36s 176ms/step - loss: 0.4996 - accurac
y: 0.8874
Epoch 4/20
205/205 [==============================] - 39s 190ms/step - loss: 0.5920 - accurac
y: 0.8730
Epoch 5/20
205/205 [==============================] - 35s 173ms/step - loss: 0.6578 - accurac
y: 0.8622
Epoch 6/20
205/205 [==============================] - 36s 175ms/step - loss: 0.5830 - accurac
y: 0.8747
Epoch 7/20
205/205 [==============================] - 35s 170ms/step - loss: 0.5580 - accurac
y: 0.8819
Epoch 8/20
205/205 [==============================] - 38s 185ms/step - loss: 0.5631 - accurac
y: 0.8777
Epoch 9/20
205/205 [==============================] - 35s 171ms/step - loss: 0.4426 - accurac
y: 0.8979
Epoch 10/20
205/205 [==============================] - 35s 170ms/step - loss: 0.5262 - accurac
y: 0.8874
Epoch 11/20
205/205 [==============================] - 37s 180ms/step - loss: 0.4364 - accurac
y: 0.9035
Epoch 12/20
205/205 [==============================] - 52s 253ms/step - loss: 0.4140 - accurac
y: 0.9159
Epoch 13/20
205/205 [==============================] - 41s 202ms/step - loss: 0.4863 - accurac
y: 0.9016
Epoch 14/20
205/205 [==============================] - 49s 238ms/step - loss: 0.4057 - accurac
y: 0.9083
Epoch 15/20
205/205 [==============================] - 48s 234ms/step - loss: 0.4942 - accurac
y: 0.8980
Epoch 16/20
205/205 [==============================] - 38s 183ms/step - loss: 0.6498 - accurac
y: 0.8900
Epoch 17/20
205/205 [==============================] - 37s 181ms/step - loss: 0.4720 - accurac
y: 0.9066
Epoch 18/20
205/205 [==============================] - 36s 175ms/step - loss: 0.3870 - accurac
y: 0.9164
Epoch 19/20
205/205 [==============================] - 36s 174ms/step - loss: 0.4569 - accurac
y: 0.9057
Epoch 20/20
205/205 [==============================] - 44s 217ms/step - loss: 0.4600 - accurac
y: 0.9035
```

In [11]:
```
#testing with 50 samples
model.evaluate(x_test,y_test)
model.save(r"C:\Users\venka\Desktop\models\anand_model.h5")
```

```
62/62 [==============================] - 2s 33ms/step - loss: 0.3511 - accuracy:
0.9217
```

In [12]:
```
y_pred=model.predict(x_test)
```

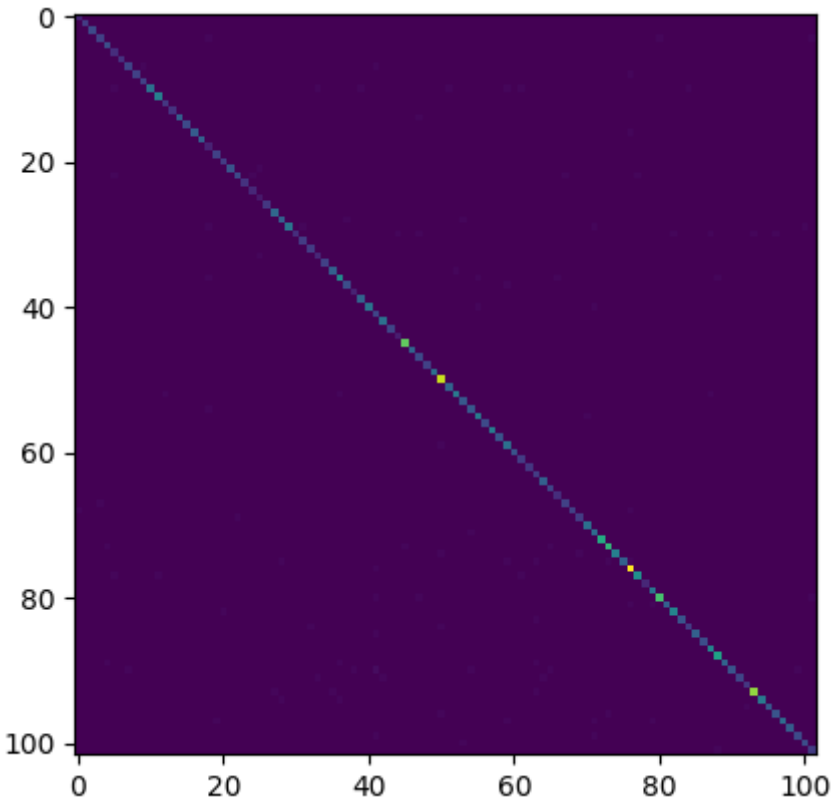
```
62/62 [==============================] - 3s 23ms/step
```

In [13]:
```
predicted_labels = np.argmax(y_pred, axis=1)
```

In [14]:
```
confusion_mat=confusion_matrix(predicted_labels,y_test)
```

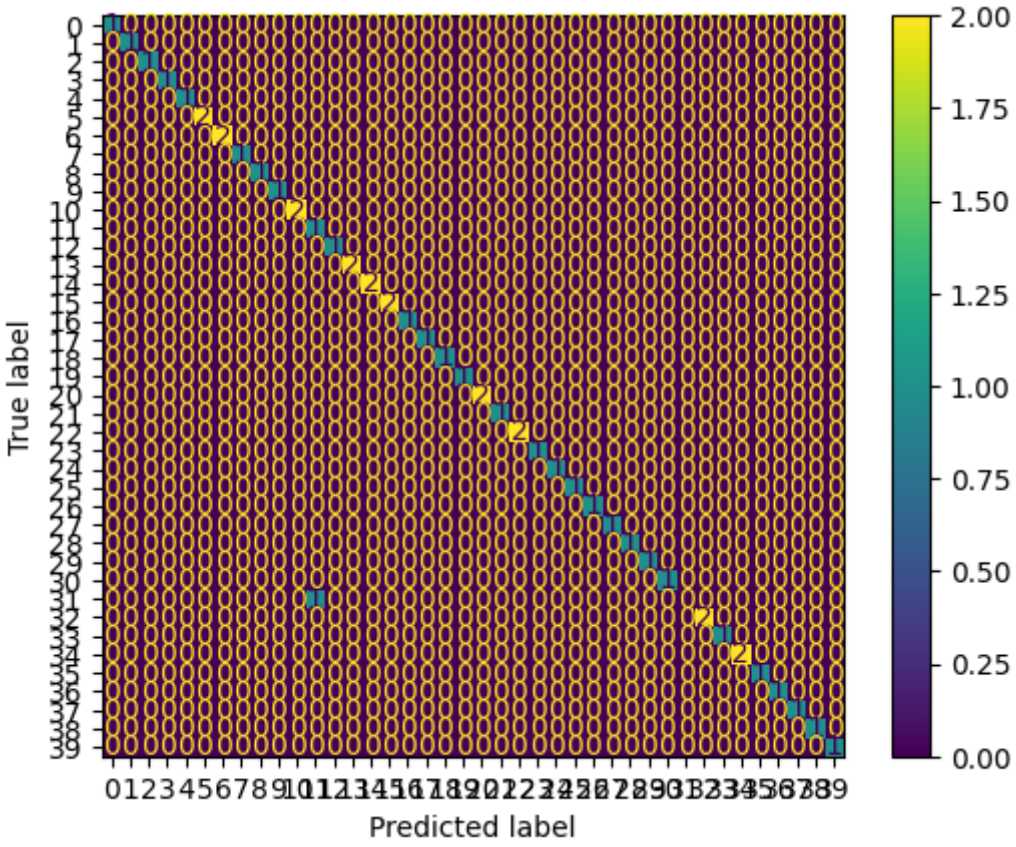
In [18]:
```
plt.imshow(confusion_mat)
```

Out[18]:
```
<matplotlib.image.AxesImage at 0x236db66ef20>
```



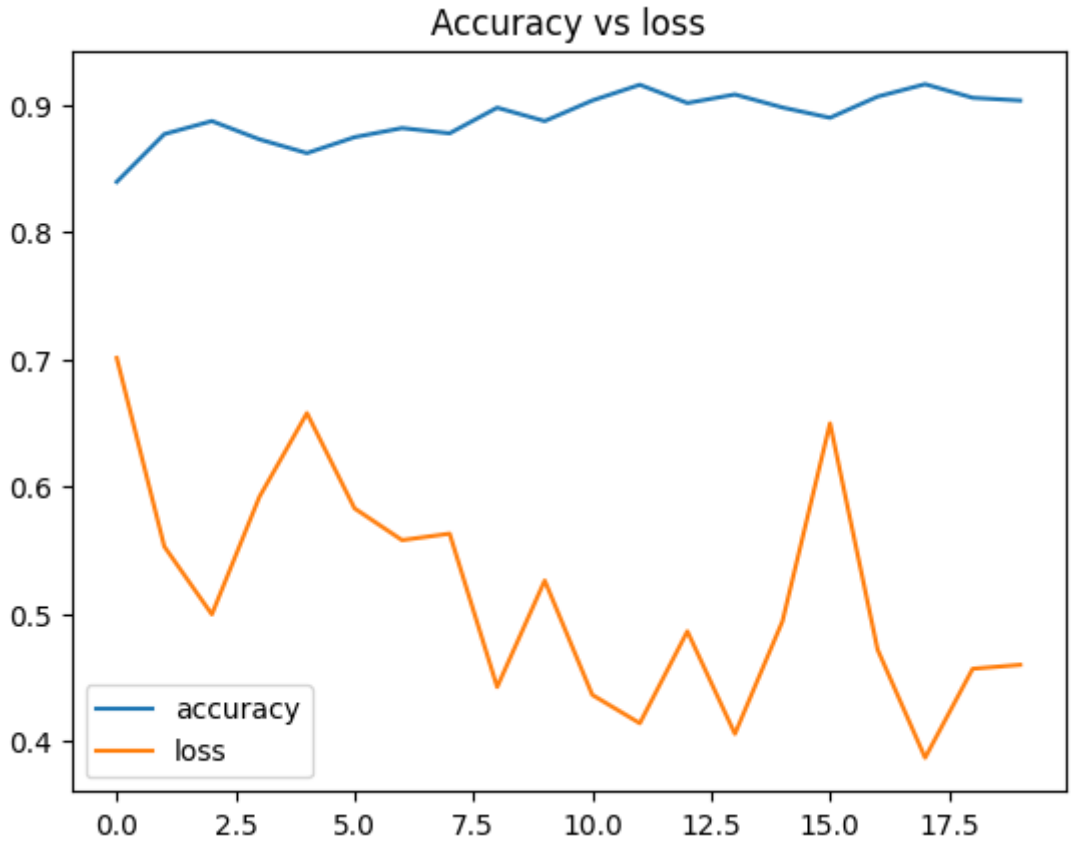
In [26]:
```
disp = ConfusionMatrixDisplay(confusion_matrix=confusion_mat)
disp.plot()
```

Out[26]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2155a6b4c40>
```



In [17]:
```
accuracy=np.array(history.history["accuracy"])
```

In [18]:
```
plt.plot(accuracy,label="accuracy")
plt.plot(history.history["loss"],label="loss")
plt.title("Accuracy vs loss")
plt.legend()
plt.show()
```



In [9]:
```
m1 = model.save(r"C:\Users\venka\Desktop\models\anand_model.h5")
```

In [ ]:
```

```