Exercise 6: Quantize the network using calibration data

```
%% Quantize the pretrained network
qLocal = dlquantizer(net, ExecutionEnvironment="MATLAB");
calResults = calibrate(qLocal, xTrain) %#ok<NASGU>
```

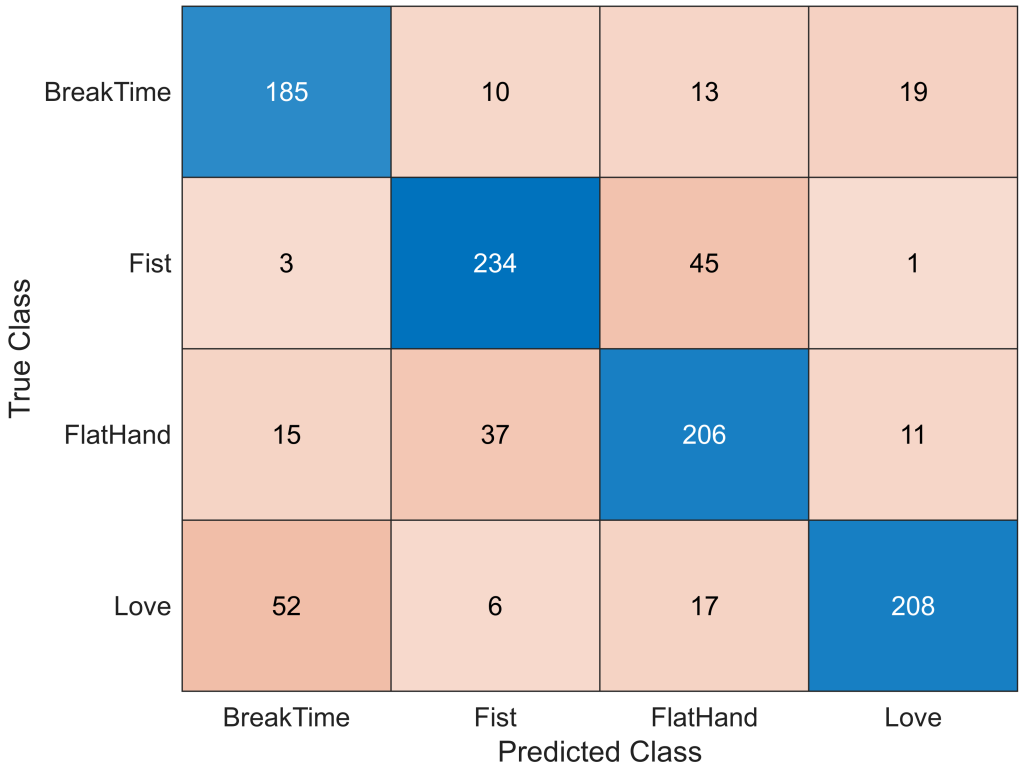
```
calResults = 23×5 table
```

...

	Optimized Layer Name	Network Layer Name	Learnables / Activations
1	"conv_1_Weights"	'conv_1'	"Weights"
2	"conv_1_Bias"	'conv_1'	"Bias"
3	"conv_2_Weights"	'conv_2'	"Weights"
4	"conv_2_Bias"	'conv_2'	"Bias"
5	"conv_3_Weights"	'conv_3'	"Weights"
6	"conv_3_Bias"	'conv_3'	"Bias"
7	"fc_Weights"	'fc'	"Weights"
8	"fc_Bias"	'fc'	"Bias"
9	"imageinput"	'imageinput'	"Activations"
10	"imageinput_normalization"	'imageinput'	"Activations"
11	"conv_1"	'conv_1'	"Activations"
12	"batchnorm_1"	'batchnorm_1'	"Activations"
13	"relu_1"	'relu_1'	"Activations"
14	"maxpool_1"	'maxpool_1'	"Activations"
15	"conv_2"	'conv_2'	"Activations"
16	"batchnorm_2"	'batchnorm_2'	"Activations"
17	"relu_2"	'relu_2'	"Activations"
18	"maxpool_2"	'maxpool_2'	"Activations"
19	"conv_3"	'conv_3'	"Activations"
20	"batchnorm_3"	'batchnorm_3'	"Activations"
21	"relu_3"	'relu_3'	"Activations"
22	"fc"	'fc'	"Activations"
23	"softmax"	'softmax'	"Activations"

```
qNet = qLocal.quantize();

%% test the Quantized network
scoresQuantized = minibatchpredict(qNet,xTest);
YQuantized = onehotdecode(scoresQuantized,classNames,2);
confusionchart(label_Test,YQuantized);
```




```
accuracyQuantized = mean(YQuantized == label_Test)
```

```
accuracyQuantized = 0.7844
```

```
%% Export
% will encounter warning
exportONNXQuantizedNetwork(qNet, "net_Quantized.onnx");
```

```
Start exporting Quantized Network to ONNX
```

Try in Dev Cloud

```
web("https://stm32ai-cs.st.com/home");
```