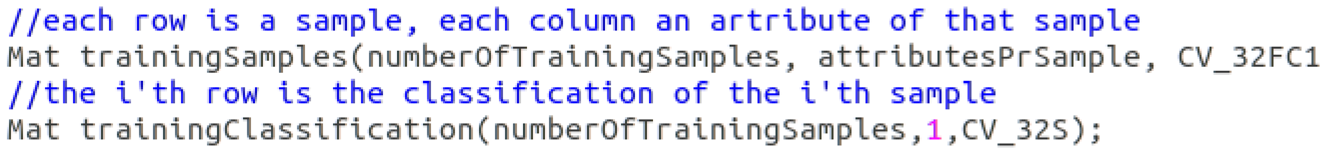
Exercise 8: Image classification

In this exercise, you will use statistical pattern recognition in OpenCv to recognise the digits in a sudoku.

- Find the contours in the training images of single digits. Compute relevant features for the digits such as the perimeter of the contours (see cv::arcLength) and the mu12 moment (see cv::Moments).
- Plot the features using your preferred plotting software and confirm that the classes are separable
- Perform classification using cv::ml::NormalBayesClassifier in OpenCV. I have provided a well commented solution you can use as reference
- Use OpenCV classification on a problem of your own

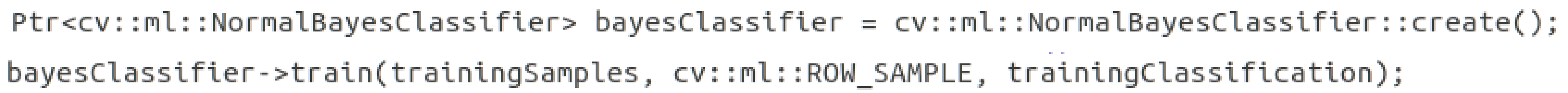
Using NormalBayesClassifier

- Fill matrixes with the computed features for each training sample



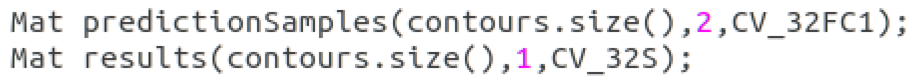
```
//each row is a sample, each column an artribute of that sample
Mat trainingSamples(numberOfTrainingSamples, attributesPrSample, CV_32FC1
//the i'th row is the classification of the i'th sample
Mat trainingClassification(numberOfTrainingSamples,1,CV_32S);
```

- Create and train a classifier



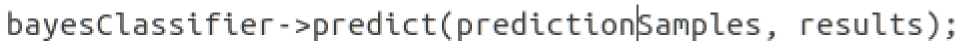
```
Ptr<cv::ml::NormalBayesClassifier> bayesClassifier = cv::ml::NormalBayesClassifier::create();
bayesClassifier->train(trainingSamples, cv::ml::ROW_SAMPLE, trainingClassification);
```

- Fill the matrix with the computed features for the shapes to be classified. Also create the matrix to hold the classification results.



```
Mat predictionSamples(contours.size(),2,CV_32FC1);
Mat results(contours.size(),1,CV_32S);
```

- Use the trained classifier to predict the results



```
bayesClassifier->predict(predictionSamples, results);
```