LUT Computer Vision and Pattern Recognition Laboratory 2021-11-10

BM40A0702 Pattern Recognition and Machine Learning

Lasse Lensu

Practical Assignment

Digits 3-D

1 Problem Statement

Your task is to develop a learning system for hand-written digits 0, 1, ..., 9. The digits are "written" as free-hand strokes in the air with the index finger, and the 3-D location information of the finger tip has been read by using a LeapMotion sensor. The purpose of the system is to recognise the hand-written digits, that is, to classify them correctly based on the 3-D location time series.

The index finger location data has been collected from a human experiment with the LeapMotion sensor, driver and software development kit, matleap Leap Motion Controller Interface by Jeff Perry, and a Matlab function by the author of this description. The location information of each stroke of the experiment has been stored into separate Matlab and comma-separated values (CSV) files. The file name contains the information on the digit and a number identifying the stroke as follows: `stroke_DIGIT_NUM.{mat|csv}` where `DIGIT` is the digit label and `NUM` is a running identification number for the stroke. For example, `stroke_3_0001.mat` would mean that the instruction to the human subject was to draw digit 3 and the Matlab file is the first sample. The notes are as follows:

- The digit labels assigned to the strokes come from the instruction given to a human subject of the experiment about what to draw.
- No identification information about the human subjects is given.
- The samples in the training set are randomly selected from the whole data set, thus, the order of the files does not correspond to the order of strokes during the experiment.

The data has been divided into training and test sets. The training data is published for the method development and it has a uniform distribution over the classes.

Your task is to implement a Matlab function `C = digit_classify(testdata)` that classifies a single unknown example given as an $N \times 3$ matrix `testdata`. The output class `C` should be an integer corresponding to the recognised digit.

Using another programming language or environment $\mathcal{X}$ than Matlab for the implementation is possible. In this case, the implementation with $\mathcal{X}$ must comply with the following requirements:

- The implementation must run on a standard operating system distribution without any special additional installations, or the submitted package including the implementation must be self-contained with the necessary packages.
- There must be a Matlab wrapper in the implementation in the form described above.

PO BOX 20, FI-53851 LAPPEENRANTA, FINLAND
E-mail: lasse.lensu@lut.fi

2 Requirements

The practical assignment is meant to be done in freely selected groups of at most three students.

To carry out the *programming task*, you must obey the following rules:

- Allowed: The use of "low-level" Matlab toolbox/$\mathcal{X}$ library functions available in a standard installation. In addition, the method implementations you or one of your practical assignment group members has prepared for this course can be used.
- Not allowed: The use of "high-level" Matlab/$\mathcal{X}$ library functions such as `classify` or another similar function doing most or all the necessary work, or any other source code, for example, from the Internet.

To prepare the *documentation* of your work, you must obey the following rules:

- Allowed: You can use references for the documentation if you acknowledge them (with a proper citation to the reference) and do not directly copy any text from a reference.
- Not allowed: Use of any material prepared by others (without properly acknowledging the source), or direct copying of sentences or their parts from a reference.

By returning the assignment results you assure that i) you acknowledge all sources (no plagiarism), and ii) you have not used any forbidden material. Any signs of plagiarism will be further studied, and when the evidence is clear, an administrative process will be started.

2.1 Matlab function or wrapper

The Matlab function that performs the classification must have the name `digit_classify` and it must take parameters and return values in the required way. The following details must be taken into account:

- Feature extraction: Possible function(s) responsible for feature extraction must be called within the `digit_classify` function.
- Training: You may use the training data as you see fit. If the classifier requires training, it is not a good idea to train the method every time the function `digit_classify` is called. Therefore, the parameters set by the training process need to be loaded or hardcoded into the implementation. If your classification technique requires extra parameters, the parameter values must be fixed or determined inside the `digit_classify` function.
- Inference: The classification function only needs to take in one sample ($N \times 3$ matrix) to be classified, possibly extract the features and produce a single class label. The computational complexity of your implementation must be practical so that testing its performance with hundreds of samples can be performed in a reasonable time.
- Environment for experiments: The `digit_classify` function and possible other related functions must run in the Matlab version available in the course's computer classroom. Both the filename and function name have to be correct.
- High-level functions: The use high-level functions of Matlab or $\mathcal{X}$ such as `classify` is not allowed in your solution. You must make your own classifier implementing a classification method.
- Visualisations or debug information: the submitted code should not produce any visualisations or provide profound debug information. (It is good practice to include a debugging flag into the code to enable or disable producing such info.)

Remember to properly comment your codes. Write also a help section to your codes that tells the purpose of the function, usage, and explanation of the parameters. In Matlab, comments following the first line of a function will show when the `help` command is used with the name of the function. You can see an example, if you type following command in Matlab:

```
>> help mean
>> type mean
```

2.2 Documentation

Write a report in English about your project. The documentation should contain a cover page where you give the following information:

1. the course number and name,
2. project title,
3. date, and
4. the student numbers, names and roles of each group member[1].

Describe the methods used for the feature extraction and classification in such detail that a reader would be able to implement the same kind of functions for the feature extraction and classification just based on your documentation and the cited references. Presenting an algorithm and explaining it in words at a higher level is a good way to describe the principles of methods. Justify your choices, that is, present the reasons to select the feature extraction technique and classifier, and the procedure to select the possibly needed parameter values for your solution.

Include in the report your classification results with the given (training) data by performing suitable validation. Standard approaches can be used for this purpose.

At the end of your documentation, you should list all the references used. Note that you are allowed to use any references/information you want, but all source code must be written by you or your practical assignment group member for this project or for this course earlier. The "low-level" functions of the official Matlab toolboxes/$\mathcal{X}$ library can be used.

3 Deadline and submission

The deadline of submitting the results of your work to Moodle is **Monday, 13 December 2021 at 08:00 EET**. The results consisting of the pdf document and all relevant codes (feature extraction, training, classification and analysis of results) must be packed into a single file using `zip` for the submission to Moodle. The file name of the package must be *STNUM*`.zip` where *STNUM* is the student number of one of the authors. When *STNUM*`.zip` is extracted, it should create a single directory *STNUM*. This directory should contain your report, the `digit_classify` Matlab function/wrapper and all the other files (except the standard Matlab/$\mathcal{X}$ functions) needed run the classification function.

4 Evaluation

The work will be evaluated based on the submitted implementation and report according to the following criteria:

1. Methods: The documentation contains grounds for method and parameter value selections and such descriptions that enable understanding the solution.
2. Successful implementation:
   (a) The implementation solves the problem and it is possible to run a classification test as instructed.

[1] Describe the contribution of each student in the group.

(b) The submitted classifiers will be ranked according to their classification performance, that is, the smaller the classification error, the better classifier. The classification error will be determined using separate test data available only to the organisers. All classes have approximately equal number of samples in the test set. The classifier should perform better than the "baseline", that is, a random classifier.

(c) The code is properly commented so that the grader can understand the implementation.

3. Results and analysis: The produced results are appropriately presented and analysed in depth to support the conclusions in the report.

4. Documentation: The report includes all the typical sections and relevant references.

In addition to the "standard" criteria described above, the following criteria will be taken into account in the evaluation:

- Novelty of the approach. (If the idea comes from somebody else's work, you must acknowledge the source.)
- Difficulty of implementing the selected approach. (However, the result of the work cannot be empty even in the case of a complicated approach.)
- Well-designed demonstration or visualisation.

Based on the evaluation, each submission will be graded using the standard scale 0, 1,..., 5. The practical assignment grade affects the final grade of the course. To pass the course, the grade of the practical assignment must be at least 1.

5 Notes and tips

The data has been gathered using a LeapMotion sensor. For more information, see LeapMotion.

The location data can contain spurious samples or sample sequences in a stroke. Therefore, data pre-processing can be beneficial.

It is likely that the feature extraction part has a significant effect on the classification performance. Try to either develop a good method yourself, or look for more information in the related literature (and cite the references whenever necessary).

If there are any questions or problems with the assignment description, the data or if you aim to use $\mathcal{X}$ instead of Matlab, contact the person supervising the practical assignment before inventing your own interpretations about the instructions or making too radical assumptions.