

Mustererkennung - **TODO** ÜbungX

Samuel Gfrörer

TODO 2016-1-15

1 **TODO**

1.1 Ausführen des Programms

Das Programm kann folgendermaßen ausgeführt werden:

```
$ ./scripts/run_test.sh
```

1.2 Implementierung

Der Code wird als Github-Repository verwaltet (siehe). Um das Programm selbst zu installieren und zu kompilieren, siehe unten.

1.2.1 Ordnerstruktur

```
.
|-- test
|   |-- __TODO__.hs
|-- resource
|   |-- ...
|-- src
|   |-- PatternRecogn
|   |   |-- __TODO__.hs
|   |-- ...
|-- app
|-- plots
|   |-- ...
```

Das Programm teilt sich auf in eine eine Bibliothek (siehe Verzeichnis “./src”) und Tests (siehe Verzeichnis “./test”). Für diese Übung sind folgende Dateien im Quellcode relevant:

TODO

1. Eingabe- und Ausgabedateien

Der Test kann mit folgendem Befehl ausgeführt werden:

```
$ ./scripts/run_test.sh
```

Im Ordner “./resource” befinden sich die Trainingsdatensätze und der Testdatensatz. Die Ausgabe erfolgt über die Standardausgabe.

1.2.2 Die Funktionalität des Tests

Der Trainingsdatensatz für die Tests befindet sich im Verzeichnis “./resource/train.*”. Es handelt sich um einen Datensatz zur Erkennung der Ziffern 0, ..., 9 aus Handschrift. Der Datensatz stammt von der Webseite zum Buch:

- Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani: An Introduction to Statistical Learning with Applications in R

Das Programm testet die Güte der Klassifizierung anhand des **TODO**-Algorithmus.

1.2.2.1 Ausgabe des Programms

TODO

Erklärung der Ausgabe:

1.2.3 Ergebnis

Die entsprechenden Plots finden sich im Verzeichnis “./examplePlots/{1,2,3}”. Die Standardausgabe des Programms findet sich unter “./examplePlots/{1,2,3}/log”

Die erzeugten Plots der Testläufe finden sich auch im Anhang dieses Dokuments (siehe unten). Die folgenden Tabellen^[questionMark] fassen die Ergebnisse dreier Testdurchläufe zusammen (siehe “examplePlots/*”):

1.2.3.1 Diskussion der Ergebnisse

TODO

1.3 Kompilieren des Programms

1.3.1 Abhängigkeiten

- git (siehe <https://git-scm.com/>)
- stack (siehe <https://docs.haskellstack.org/>)

1.3.2 Kompilieren

```
$ git clone https://github.com/EsGeh/pattern-recognition
$ git checkout exercise8-release
$ stack setup
$ stack build
```

2 Anhang: Plots

siehe “./examplePlots”