



UNIVERSITY OF APPLIED SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Medien Projekt

Gesichtserkennung

Gesichtserkennung mittels Neuronalen Netzen

Eingereicht am:

1 September 2018

Eingereicht von:

Jakob Lambert
Süldorfer Mühlenweg 31
22123 Hamburg
Tel.: +49 17632643656
E-mail: minf101027@fh-wedel.de

Referent:

Prof. Dr. Dennis Säring
Fachhochschule Wedel
Feldstraße 143
22880 Wedel
Phone: (041 03) 80 48-43
E-mail: sdg@fh-wedel.de

Inhaltsverzeichnis

Abbildungsverzeichnis	III
List of Listings	IV
1 Einleitung	1
1.1 Aufgabe	1
1.2 Einführung in die Gesichtserkennung	1
2 Entwicklungsumgebung	2
2.1 Software	2
2.1.1 Installation	2
2.1.2 Anwendung	2
2.2 Bild Datensatz	2
2.2.1 Datensätze für Tensorbox bereitstellen	3
2.3 Tensorbox	3
2.3.1 Installation	3
2.3.2 Training	3
2.3.3 Evaluation	4
3 Methoden	5
3.1 Verwendete Technologien	5
3.2 Struktur	5
3.3 Training der Gesichtserkennung	5
3.3.1 Installation von Tensorbox	5
3.3.2 Vorbereiten der WIDER Face Datenbank	5
3.3.3 Training mittels Tensorbox	5
4 Ist-Analyse	6
5 Zusammenfassung	7
6 Ausblick	8
7 Literaturverzeichnis	9
Literaturverzeichnis	10
8 Eidesstattliche Erklärung	11

Abbildungsverzeichnis

1.1	Automatische Gesichtserkennung mit OpenCV.	1
2.1	Beispiele aus dem WIDER Face Datensatz	3

List of Listings

1

Einleitung

1.1 Aufgabe

Es ist eine Applikation zu entwickeln welches Gesichter mit mit einem Neuronalen Netz erkennt. Die gefundenen Gesichter sollen ausgeschnitten werden und in einem Ordner zur Weiterverarbeitung gespeichert werden.

1.2 Einführung in die Gesichtserkennung

Bei der Gesichtserkennung unterscheidet man zwischen Lokalisation von Gesichtern und der Zuordnung der Gesichter zu einer bestimmten Person.

¹



Abbildung 1.1: Automatische Gesichtserkennung mit OpenCV.

¹https://de.wikipedia.org/wiki/Gesichtserkennung#/media/File:Face_detection.jpg

2

Entwicklungsumgebung

2.1 Software

2.1.1 Installation

1. Download https://github.com/DevJakobL/Face_detection
2. Am speicherungsart folgende befehle ausführen

```
cd TENSORBOX/utis && python setup.py install && ../../  
pip install -r path/to/requirements.txt
```

2.1.2 Anwendung

Die App kann mittels des Befehls:

```
python app.py [-h] [--weights WEIGHTS] --input_image INPUT_IMAGE  
[--output_dir OUTPUT_DIR]
```

- `-h` : Ausgabe der Hilfe
- `--weights WEIGHTS`: Optional kann angegeben werden welche Datei mit den Gewichtungen verwendet werden soll.
- `--input_image INPUT_IMAGE` : Datei in der die Gesichter erkannt werden sollen.
- `--output_dir OUTPUT_DIR` : Optionale kann auch der Ordner in dem die gefundenen Gesichter gespeichert werden soll angegeben werden.

2.2 Bild Datensatz

Für das Training des Neuronalen Netzes wird der WIDER Face ¹ Datensatz verwendet. Der Datensatz beinhaltet 32.203 Bilder und 393.703 markierte Gesichter. Der WIDER Face Datensatz wird in 61 Klassen Organisiert. Aus den einzelnen Klassen werden wiederum zufällig 40% Trainings und 10% Validation Datensätze erstellt. Die 50% Test Datensätze können nicht verwendet werden da die Gesichter nicht gegenzeichnet sind.

¹<http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/>



Abbildung 2.1: Beispiele aus dem WIDER Face Datensatz

2.2.1 Datensätze für Tensorbox bereitstellen

1. Downloaden der einzelnen Datensätze Wider Face Training Images, Wider Face Validation Images und Wider Face Testing Images von der Webseite <http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/>
2. Speichern der Projekt Ordner für die Test Bilder, *data/WIDER_train* für die Trainings Bilder und *data/WIDER_val* für die Bilder zum Validieren.
3. Der von Tensorbox benötigten Jsen Dateien mit folgendem Konsolenbefehl:

```
python wider_parser.py
```
4. Test Datensatz zusammenstellen in dem aus den zuvor erstellten Jsen Dateien die gewünschten Datensätze ausgewählt werden und in die *wider_test.json* kopiert werden.

2.3 Tensorbox

2.3.1 Installation

1. Download https://github.com/DevJakobL/Face_detection
2. Am speicherungsart folgende befehle ausführen

```
cd TENSORBOX/utils && python setup.py install && ../../  
pip install -r path/to/requirements.txt
```
3. WIDER-Face Datenbank fürs Training vorbereiten, wie in Abschnitt 2.2 beschrieben.

2.3.2 Training

```
python train.py --hypes hypes/overfeat_rezoom.json --gpu 0 --logdir output
```

Mit Hilfe der *hypes/overfeat_rezoom.json* Datei kann das Training für den eigenen Datensatz eingestellt werden. Mehr dazu in Abschnitt 3.3.

2.3.3 Evaluation

```
python tensorbox/evaluate.py --weights WEIGHTS_FILE --test_boxes data/  
wider_val.json
```

WEIGHTS_FILE beinhaltet die Gewichte des zu validierenden neuronalen Netzes z.B. `output/overfeat_rezoom_2018_07_10_00.50/save.ckpt-1000`. Nach dem die Validierung abgeschlossen ist wird ein Ordner erstellt z.B. `output/overfeat_rezoom_2018_07_10_00.50/images_val_boxes_1000`. in diesem befindet sich die Ergebnis Datei "results.png" erstellt.

3

Methoden

3.1 Verwendete Technologien

- Python 3.7
- Tensorbox

3.2 Struktur

3.3 Training der Gesichtserkennung

3.3.1 Installation von Tensorbox

3.3.2 Vorbereiten der WIDER Face Datenbank

3.3.3 Training mittels Tensorbox

4

Ist-Analyse

5

Zusammenfassung

6

Ausblick

7

Literaturverzeichnis

Literaturverzeichnis

- [Knu98] Donald E. Knuth. *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.



Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Ort, Datum

Jakob Lambert