

Logo 1

Logo2

Goethe-Universität Frankfurt
Fachbereich 12 - Informatik und Mathematik

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science

Thema: Deep Neural Networks for occluded Image Recognition

Autor: Julius Taylor <s8423760@stud.uni-frankfurt.de>
MatNr. 5210444

Version vom: 26. Oktober 2017

Betreuer: Prof. Dr. Jochen Triesch

Sperrvermerk

Die vorliegende Arbeit beinhaltet interne und vertrauliche Informationen der Firma <Firmenname>. Die Weitergabe des Inhalts der Arbeit im Gesamten oder in Teilen sowie das Anfertigen von Kopien oder Abschriften - auch in digitaler Form - sind grundsätzlich untersagt. Ausnahmen bedürfen der schriftlichen Genehmigung der Firma <Firmenname>.

Zusammenfassung

Convolutional Neural Networks sind haben in den letzten Jahren in vielen Machine Learning Problemen bisher ungeahnte Benchmarks ermöglicht, die in einigen Domänen sogar Menschliche Performance übersteigen. Dabei sind rekurrente Verbindungen in solchen Netzen bisher selten beachtet worden, obwohl die Struktur des menschlichen Gehirns solche nahelegt. In dieser Arbeit versuche ich die Ergebnisse von Spoerer und Kriegeskorte zu reproduzieren, welche Hinweise fanden, dass das Inkorporieren solcher rekurrenten Verbindungen Vorteile bei der Bilderkennung in schwierigen Szenarien, wie der teilweisen Verdeckung zu erkennender Objekte bieten kann.

like whaat?

Abstract

Inhaltsverzeichnis

Abbildungsverzeichnis	5
Tabellenverzeichnis	5
Listingverzeichnis	5
Abkürzungsverzeichnis	5
1 Einleitung	9
2 Kapitel 1	9
3 Kapitel 2	9
4 Ausblick	9
5 Fazit	9
Literaturverzeichnis	12
Anhang	13
Eidesstattliche Erklärung	13

Abbildungsverzeichnis

1	Beispiel einer Bildbeschreibung	9
2	Beschreibung	9

Tabellenverzeichnis

Listingverzeichnis

1	Die Datei <code>data-config.xml</code> dient als Beispiel für XML Quellcode . . .	9
2	Das Listing zeigt Java Quellcode	10

Abkürzungsverzeichnis

CMS	
---------------	--

CSS

ERM

GNU

GPL

GUI

HTML

IM

JS

JSON

KPI

LGPL

OCR

RSS

SQL

TDD

UGC

WWW

XMPP

1 Einleitung

2 Kapitel 1

3 Kapitel 2

4 Ausblick

5 Fazit

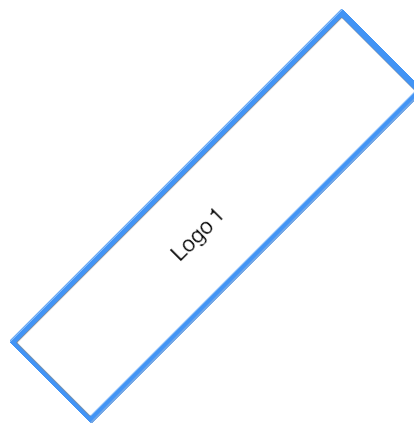


Abbildung 1: Beispiel einer Bildbeschreibung¹



Abbildung 2: Beschreibung

Abbildung 2 [S.9]

Überschrift 1	Überschrift 2
Info 1	Info 2
Info 3	Info 4

```
1 <dataConfig>
2   <dataSource type="JdbcDataSource"
3       driver="com.mysql.jdbc.Driver"
4       url="jdbc:mysql://localhost/bms_db"
5       user="root"
6       password="" />
7 </document>
```

¹Bildquelle: Beispielquelle

```

8      <entity name="id"
9          query="select id, htmlBody, sentDate, sentFrom, subject, textBody
10             from mail">
11      <field column="id" name="id"/>
12      <field column="htmlBody" name="text"/>
13      <field column="sentDate" name="sentDate"/>
14      <field column="sentFrom" name="sentFrom"/>
15      <field column="subject" name="subject"/>
16      <field column="textBody" name="text"/>
17      </entity>
18  </document>
19 </dataConfig>

```

Listing 1: Die Datei data-config.xml dient als Beispiel für XML Quellcode

```

1  /* generate TagCloud */
2  Cloud cloud = new Cloud();
3  cloud.setMaxWeight(_maxSizeOfText);
4  cloud.setMinWeight(_minSizeOfText);
5  cloud.setTagCase(Case.LOWER);
6
7  /* evaluate context and find additional stopwords */
8  String query = getContextQuery(_context);
9  List<String> contextStoplist = new ArrayList<String>();
10 contextStoplist = getStopwordsFromDB(query);
11
12 /* append context stoplist */
13 while(contextStoplist != null && !contextStoplist.isEmpty())
14     _stoplist.add(contextStoplist.remove(0));
15
16 /* add cloud filters */
17 if (_stoplist != null) {
18     DictionaryFilter df = new DictionaryFilter(_stoplist);
19     cloud.addInputFilter(df);
20 }
21 /* remove empty tags */
22 NonNullFilter<Tag> nnf = new NonNullFilter<Tag>();
23 cloud.addInputFilter(nnf);
24
25 /* set minimum tag length */
26 MinLengthFilter mlf = new MinLengthFilter(_minTagLength);
27 cloud.addInputFilter(mlf);
28
29 /* add taglist to tagcloud */
30 cloud.addText(_taglist);
31
32 /* set number of shown tags */
33 cloud.setMaxTagsToDisplay(_tagsToDisplay);

```

Listing 2: Das Listing zeigt Java Quellcode

Die Zuordnung aller möglichen Werte, welche eine Zufallsvariable annehmen kann nennt man *Verteilungsfunktion* von X .

Die Funktion $F: \mathbb{R} \rightarrow [0,1]$ mit $F(t) = P(X \leq t)$ heißt Verteilungsfunktion von X .²

Für eine stetige Zufallsvariable $X: \Omega \rightarrow \mathbb{R}$ heißt eine integrierbare, nicht-negative reelle Funktion $w: \mathbb{R} \rightarrow \mathbb{R}$ mit $F(x) = P(X \leq x) = \int_{-\infty}^x w(t)dt$ die *Dichte* oder *Wahrscheinlichkeitsdichte* der Zufallsvariablen X .³

²Konen, vgl. [?] [S.55]

³Konen, vgl. [?] [S.56]

Literaturverzeichnis

Anhang

Eidesstattliche Erklärung

Eidesstattliche Erklärung zur <-Arbeit>

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Unterschrift :

Ort, Datum :

