Evaluation - OCR

Projekt: DigitalSchoolNotes

Projekt Team: Adler, Brinnich, Hohenwarter, Karic, Stedronsky

Version 3.0 **15.12.2015** **Status: [RELEASE]**

	Datum	Name	Unterschrift
Erstellt	01.10.2015	Adin Karic	
Geprüft	15.11.2015	Niklas Hohenwarter	
Freigegeben			
Git-Pfad: */doc/evaluation*		**Dokument:** *evaluation_ocr.doc*	

1 Changelog

Version	Datum	Status	Bearbeiter	Kommentar
0.1	2015-10-01	Erstellt	Adin Karic	Dokument erstellt
1.0	2015-10-07	Geprüft	Selina Brinnich	QA
1.1	2015-10-12	Bearbeitet	Adin Karic	Formatierung, Fehler ausgebessert, Abbildungsverzeichnis
2.0	2015-10-12	Geprüft	Thomas Stedronsky	QA
2.1	2015-12-15	Bearbeitet	Philipp Adler	Codeverzeichnis hinzugefügt, Abbildungsverzeichnis bearbeitet
3.0	2015-12-15	Geprüft	Niklas Hohenwarter	QA

2 Vergleich

	Tesseract OCR (pytesseract)	OCRopus 0.3	OmniPage Ultimate (Testversion)
Installation	8/10 Nach einigen Problemen eigentlich simpel	1/10 Nicht geglückt, fragmentartige Anleitungen	10/10 Sehr einfach (mit Wizard)
Komplexität/Handhabung	8/10 Intuitive Bedienung, Komplexität nicht anspruchsvoll	5/10 Preprocessing bei der Bildanalyse, mittlere Komplexität	6/10 Programm wirkt ein wenig komplex und sehr mächtig
Dokumentation	6/10 partiell vorhanden [1]	3/10 Wenig vorhanden, Entwicklungsteam plant mehr Dokumentation zu machen [3]	8/10 User-Manual ist vorhanden [5]
(Lizenz-)kosten	10/10 Apache-Lizenz (freie Software)	10/10 Apache-Lizenz (freie Software)	0/10 kostenpflichtig (Client-Windows-Version) 199€
Community	8/10 Forum ist vorhanden [2] 4013 Fragen bei StackOverflow	6/10 Forum ist vorhanden [4] 105 Fragen bei StackOverflow	8/10 42 Fragen bei StackOverflow Firmen-Support ist vorhanden
Maschinenschrift-erkennung	6/10 Liegt etwa bei 50% (ist aber trainierbar)	0/10 Konnte nicht getestet werden (Installation fehlgeschlagen)	9/10 Hervorragend

Handschrift-erkennung	1/10 So gut wie garnicht	0/10 Konnte nicht getestet werden(Installation fehlgeschlagen)	0/10 Kein Erfolg
Prototyp	9/10 Erfolgreich	0/10 Nicht erfolgreich	7/10 Erfolgreich, jedoch nicht für uns nutzbar
Python-Anbindung	9/10 Ist gegeben durch "pytesseract"	9/10 ist vorhanden	0/10 Nicht gegeben, eigenständiges Programm
Gesamtpunktzahl	**65/90**	**34/90**	**48/90**

3 Tesseract OCR (pytesseract)

„tesseract-ocr ist ein Kommandozeilenprogramm zur Texterkennung. Ursprünglich von Hewlett Packard zwischen 1984 und 1995 als kommerzielles Programm entwickelt, wurde der Code 2005 freigegeben. Die Entwicklung wird von Google unterstützt, da eine Open Source-Lösung zur Erstellung von E-Books benötigt wurde. Das Programm unterstützt etliche westeuropäische und asiatische Sprachen wie z.B. vietnamesisch.

tesseract-ocr ist ein reines Zeichenerkennungs-Programm, es liefert keine Layout-Analyse, und gibt unformatierten Text, ab Version 3.00 auch hOCR, ab 3.03 zudem als PDF aus. Die Texterkennung kann theoretisch auch 'trainiert' werden." [6]

3.1 Installation des Prototyps (tesseract-engine only)

Für die Installation dieses Prototyps wurden folgende Anleitungen benutzt: [6][7][9]

Um *Tesseract-OCR* zu installieren, installiere ich die folgenden Pakete in meiner Debian-VM:

- `tesseract-ocr`
- `tesseract-ocr-deu`

```
root@eval:/home/eval# sudo apt-get install tesseract-ocr tesseract-ocr-deu
```

Nun kopiere ich mir ein paar Bilder in die VM:

```
eval@eval:~/test$ ls
drei.jpg   eins.jpg   fuenf.jpg   vier.jpg   zwei.jpg
```

Nun teste ich tesseract indem ich *test3.jpg* analysieren lasse und in *out3.txt* schreibe:

Hier das Originalbild *test3.jpg*

Figure 1: Beispielbild OCR

Ausführung von tesseract *test3.jpg* out3

```
root@eval:/home/eval/grimhack# tesseract test3.jpg out3
Tesseract Open Source OCR Engine v3.03 with Leptonica
```

less-Ausgabe von *out3.txt*:

```
#THE_FRESH_STEP
Adamski, Coric, Schwertberger
Entwicklung einer neuen Webseite.
die alle Abteilungen. Werlstatten und A / m
```

→ funktioniert

3.2 Installation Prototyp mit Python-Anbindung (pytesser-Modul)

Für die Installation dieses Prototyps wurden folgende Anleitungen benutzt:[6][7][10]

Zuerst installiere ich Python3:

```
root@eval:/home/eval/test# apt-get install python3
```

Als Nächstes installiere ich *pip*(pip installs python) um *pil*(python imaging library) installieren zu können:

```
root@eval:/home/eval/test# apt-get install python3-pip
```

Nun installiere ich folgende libraries:

```
sudo apt-get install python-dev libjpeg-dev libfreetype6-dev zlib1g-dev
```

Schließlich installiere ich mittels *pip3 pillow*:

```
pip3 install pillow
Successfully installed pillow
```

Beim Installieren von pil tritt folgende Fehlermeldung auf:

```
Command python setup.py egg_info failed with error code 1 in /tmp/pip-
build-4o34d8gq/PIL
```

```
Storing debug log for failure in /root/ .pip/pip.log
```

Ich überspringe den Fehler und lade mal das *pytesser* Paket [9] herunter und entpacke es:

```
root@eval:/home/eval# unzip pytesser_V0.0.1.zip
```

Nun (zum converten von jpg zu tif) installiere ich ein Paket namens: *imagemagick*

```
apt-get install imagemagick
```

Jetzt konvertiere ich das Bild von *png* zu *tif*:

```
root@eval:/home/eval# convert fonts_test.png -auto-level -compress none
myimage.tif
```

Schließlich erstelle ich ein python-File (*test.py*) mit folgendem Inhalt:

```
from PIL import Image
from pytesser import *
image_file = 'myimage.tif'
im = Image.open(image_file)
text = image_to_string(im)
text = image_file_to_string(image_file)
text = image_file_to_string(image_file, graceful_errors=True)
print "=====output=======\n"
print text
```

Code 1 Image zu Text

Nun führe ich das python-Programm durch den Befehl "`python test.py`" aus. (Natürlich muss es im Ordner der Bilddatei ausgeführt werden)

```
root@eval:/home/eval# python test.py
Traceback (most recent call last):
   File "test.py", line 3, in <module>
      from pytesser import *
   File "/home/eval/pytesser.py", line 6, in <module>
      import Image
ImportError: No module named Image
root@eval:/home/eval# vim pytesser.py
```

Der folgende Fehler tritt auf und nach längerer Recherche breche ich an dieser Stelle die Installation von pytesser ab.

3.3 Installation Prototyp mit Python-Anbindung (pytesseract)

Für die Installation dieses Prototyps wurden folgende Anleitungen benutzt: [6][11][12][13]

Die ersten Schritte sind ähnlich wie beim oberen Prototypen:

```
apt-get install python3
apt-get install python3-pip
pip3 install pytesseract
pip3 install pillow
```

Folgender Fehler tritt bei der Installation von *pillow* auf:

```
ValueError: --enable-jpeg requested but jpeg not found, aborting
```

Als Lösung präsentiert sich das Paket *python3-pil*:

```
apt-get install python3-pil
```

Nun wird wie oben die *tesseract-engine* installiert:

```
apt-get install tesseract-ocr (siehe oben)
```

Ich lege nun folgendes Testscript an:

```
import pytesseract
from PIL import Image
print (pytesseract.image_to_string(Image.open('test3.jpg')))
```

Code 2 Pytesseract Bildumwandlung

Es funktioniert! Hier das Bild für die Eingabe(*test3.jpg*):

Figure 2: Beispielbild OCR

Hier der Output nach Ausführen des *test.py* –Scripts mittels python3 *test.py*:

```
#THE_FRESH_STEP
Adamski, Coric, Schwertberger
Entwicklung einer neuen Webseite.
die alle Abteilungen. Werlstatten und A / m
```

Nun ein Test mit einem Bild mit Handschrift:

Figure 3 Beispielbild OCR Handschrift

Wie man sieht kein Erfolg:

```
root@eval:/home/eval/grimhack# python3 test.py
root@eval:/home/eval/grimhack#
```

4 OCRopus

„OCRopus is a collection of document analysis programs, not a turn-key OCR system. In order to apply it to your documents, you may need to do some image preprocessing, and possibly also train new models." [13]

Für die Installation dieses Prototyps wurden folgende Anleitungen benutzt: [6][14][15][16]

Folgende Befehle sind notwendig:

```
apt-get install g++
apt-get install scons
apt-get install subversion
```

Nun lade ich das *iulib* Packet herunter [17]

```
git clone https://github.com/tmbdev/iulib
```

Fehlende Bibliotheken nachladen:

```
apt-get install libpng12-dev libjpeg62-turbo-dev libtiff5-dev libavcodec-
dev libavformat-dev libsdl-gfx1.2-dev libsdl-image1.2-dev
cd uilib
scons install
make install → FEHLER (es ist überhaupt kein Make-File vorhanden)
```

```
root@eval:/home/eval/ocropus2/iulib# make install
make: *** No rule to make target 'install'.   Stop.
root@eval:/home/eval/ocropus2/iulib#
```

Diesen Fehler überspringe ich mal und fahre fort.

Als Nächstes ist *OCRopus* zu installieren, die Anleitung verweist jedoch auf eine Seite die bei Aufruf "Forbidden 403" liefert. Die Installation von OCRopus wird zu diesem Zeitpunkt abgebrochen.

5 OmniPage Ultimate

„OmniPage Ultimate ist das Flaggschiff von Nuance zum Scannen und Konvertieren von Dokumenten. Es ist ideal für Geschäftsanwender, kleinere Büros und Arbeitsgruppen zur Verarbeitung, Verteilung und Archivierung von Papier- und PDF-Dokumenten." [18]

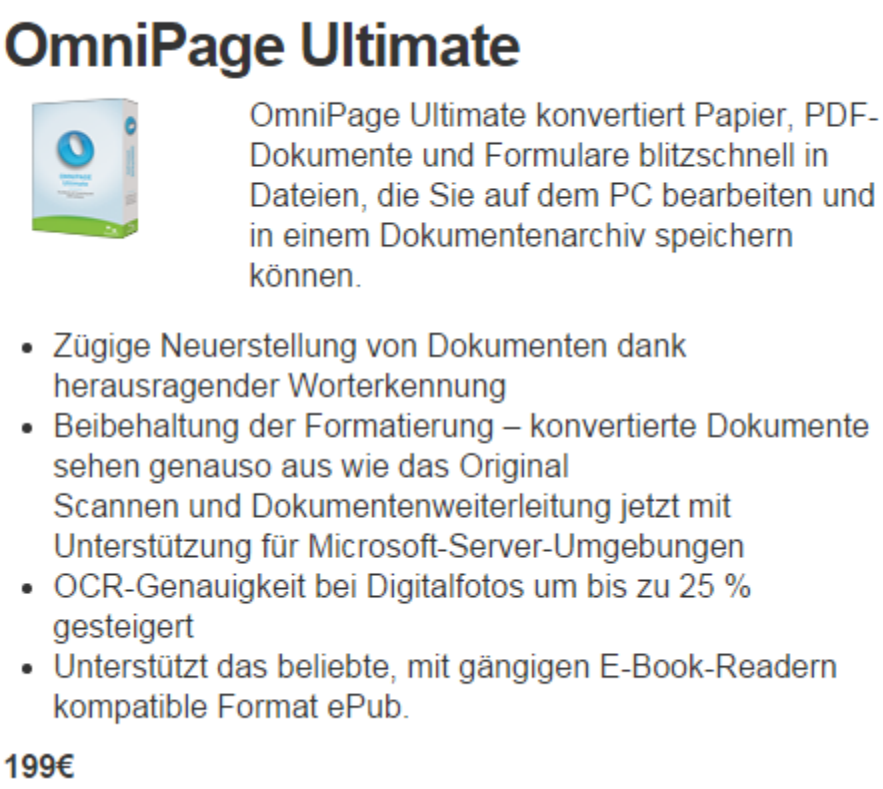

Figure 4: OmniPage Zusammenfassung

Das es sich nicht um ein kostenloses Produkt handelt lade ich eine 15-tägige Testversion herunter. Ich erhalte eine Mail in der steht welche Umgebungen unterstützt werden:

- Windows XP mit SP3, 32 Bit
- Windows 7, 32/64 Bit
- Windows 8, 32/64 Bit
- Windows Server 2008 R2
- Windows Server 2012

Nach dem Herunterladen folge ich den Installationsanweisungen der Software:

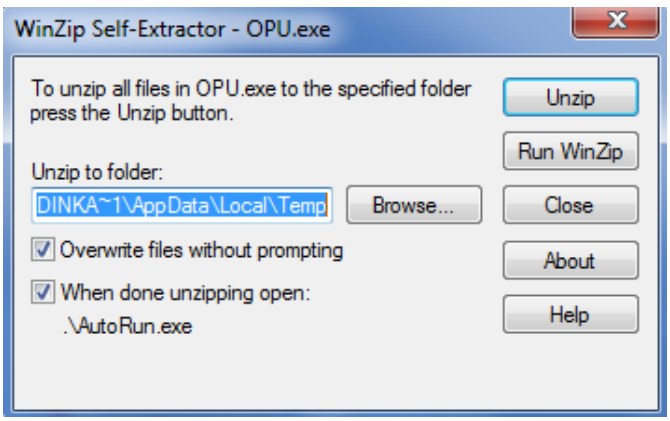

Figure 5: Omnipage Installationsanweisung

(Zusatzsoftware "*Cloud Connector*" wird auch installiert)

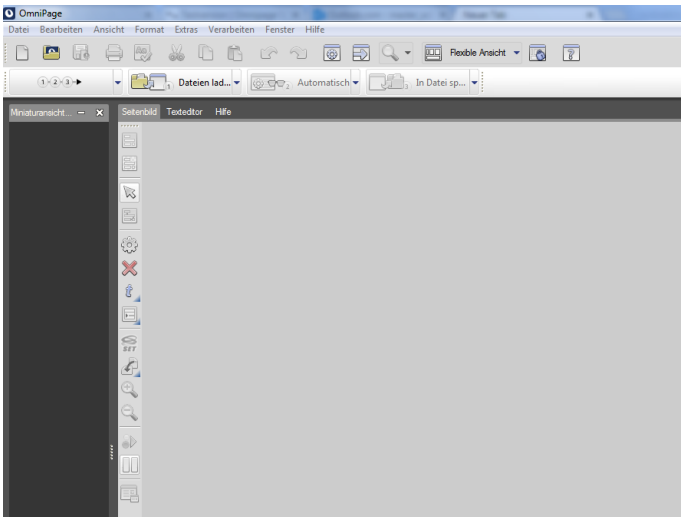

Figure 6: OmniPage GUI

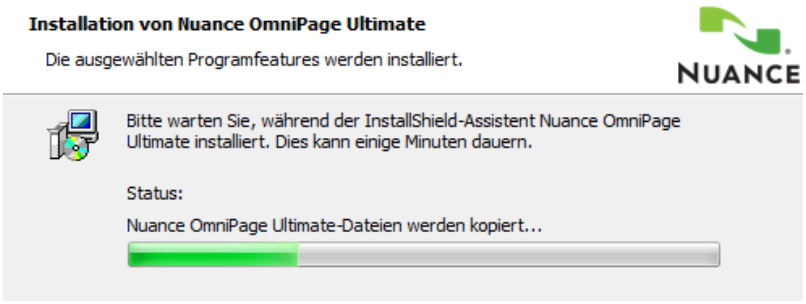

Figure 7: OmniPage Installationsscreen

Da es nun installiert ist probiere ich gleich die Funktionalität aus.

Untersuchen eines Bildes mit Maschinenschrift:

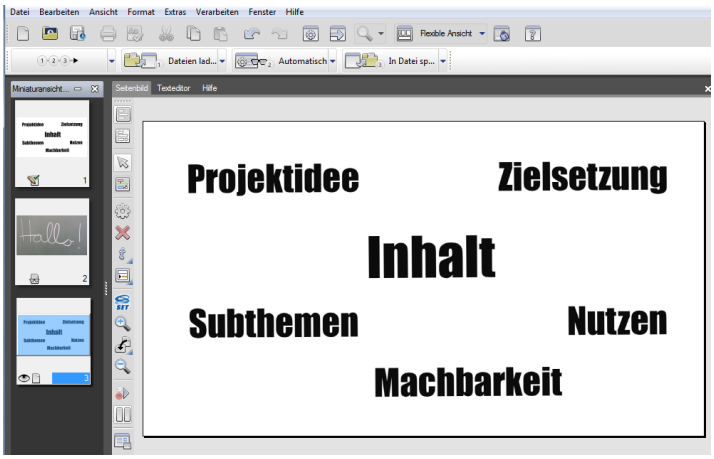

Figure 8: OmniPage Maschinenschrift Bild

Das Ergebnis ist einwandfrei und gleich bearbeitbar:

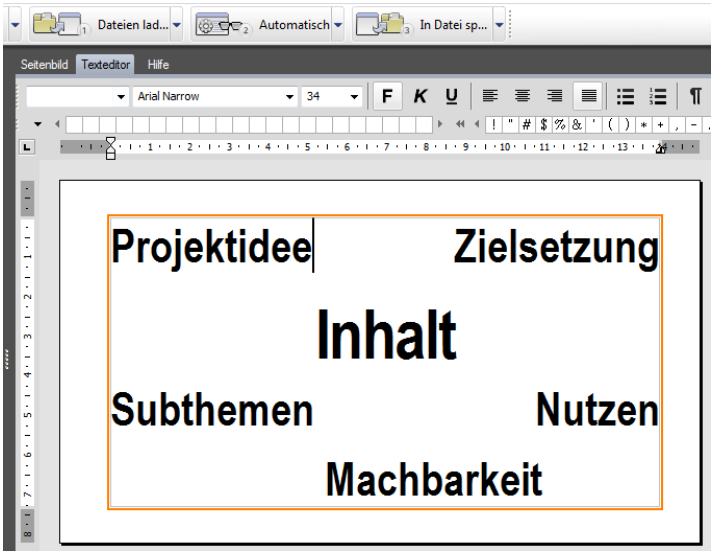

Figure 9: OmniPage Maschinenschrift Ergebnis

Als Nächstes ein Test mit einem Bild mit Handschrift:

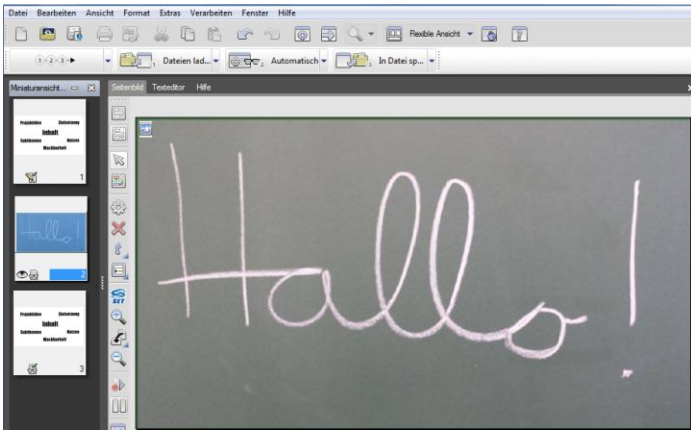

Figure 10: OmniPage Handschrift Bild

Wie man erkennen kann kein Erfolg beim Analysieren von Handschrift-Bildern:

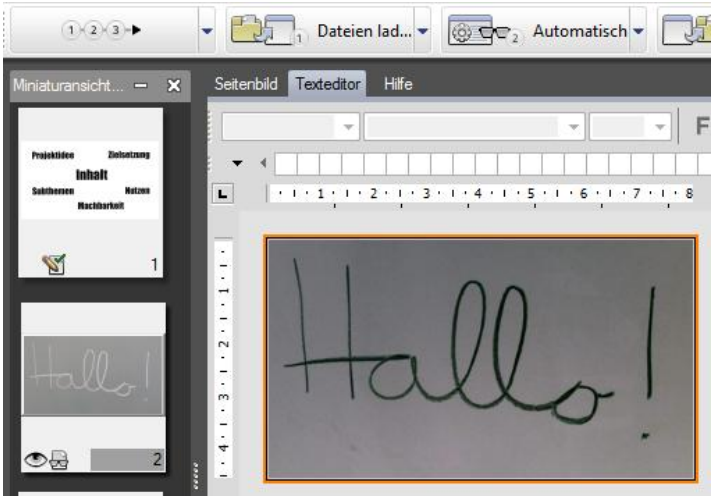

Figure 11: OmniPage Handschrift Ergebnis

6 Endergebnis

Aufgrund der oben angeführten Evaluierung und des Vergleichs der verschiedenen Softwarepakete entscheidet sich das Team für die *tesseract-ocr*-engine in Verbindung mit dem *pytesseract*-Modul.

7 Quellen

[1] Google Inc., "python-tesseract", https://code.google.com/p/python-tesseract/8/10, zuletzt besucht: 05.10.2015

[2] Google Inc., "tesseract-ocr", https://groups.google.com/forum/#!forum/tesseract-ocr, zuletzt besucht: 06.10.2015

[3] Google Inc., " API documentation of Ocropus", https://groups.google.com/forum/#!topic/ocropus/ojnJAADA4q8, zuletzt besucht: 07.10.2015

[4] Google Inc., "ocropus", https://groups.google.com/forum/#!forum/ocropus, zuletzt besucht: 07.10.2015

[5] Nuance, "Support User's Guide", http://support.nuance.com/usersguides/default.asp?UsersGuidesProduct=omnipage,zuletzt besucht: 07.10.2015

[6] Heinrich Schwietering, "tesseract-ocr", http://wiki.ubuntuusers.de/tesseract-OCR, zuletzt besucht: 07.10.2015

[7] fosshelp, "How to convert jpg to tiff for OCR with tesseract", http://fosshelp.blogspot.co.at/2013/04/how-to-convert-jpg-to-tiff-for-ocr-with.html, zuletzt besucht: 07.10.2015

[8] Bobby Grayson, "Setting Up a Simple OCR Server", https://realpython.com/blog/python/setting-up-a-simple-ocr-server/, zuletzt besucht: 07.10.2015

[9] Google Inc., "PyTesser version 0.0.1", https://code.google.com/p/pytesser/downloads/detail?name=pytesser_v0.0.1.zip&can=2&q, zuletzt besucht: 07.10.2015

[10] Quora, "How do I use PyTesser and Tesseract OCR in Ubuntu with Python?", https://www.quora.com/How-do-I-use-PyTesser-and-Tesseract-OCR-in-Ubuntu-with-Python, zuletzt besucht: 07.10.2015

[11] Python Software Foundation, "pytesseract 0.1.6", https://pypi.python.org/pypi/pytesseract, zuletzt besucht: 07.10.2015

[12] Delimitry, "Installing tesseract for python on Ubuntu 14.04", http://delimitry.blogspot.co.at/2014/10/installing-tesseract-for-python-on.html, zuletzt besucht: 07.10.2015

[13] Grimhacker, "INSTALLING PYTESSERACT – PRACTICALLY PAINLESS", http://grimhacker.com/wordpress/2014/11/23/installing-pytesseract-practically-painless/, zuletzt besucht: 07.10.2015

[14] tmbdev, "Python-based tools for document analysis and OCR", https://github.com/tmbdev/ocropy, zuletzt besucht: 07.10.2015

[15] Rui Maximo, "Installing OCRopus on Ubuntu", https://ruimaximo.wordpress.com/2010/06/06/installing-ocropus-on-ubuntu/, zuletzt besucht: 07.10.2015

[16] Fluid, "Installing OCRopus 0.3", https://wiki.fluidproject.org/display/fluid/Installing+OCRopus+0.3, zuletzt besucht: 07.10.2015

[17] tmbdev, "The iulib Image Understanding and colib Data Structure Libraries.", https://github.com/tmbdev/iulib, zuletzt besucht: 07.10.2015

[18] Nuance, "OmniPage Ultimate", http://www.nuance.de/for-business/by-product/omnipage/ultimate/index.htm, zuletzt besucht: 07.10.2015

8 Abbildungsverzeichnis

9 Codeverzeichnis