
DOCUMENTS
Michael Eismann
DOPAK 2016
DOCUMENTS
Cloud
UserExits
Mobile
BigData
ScriptExtension
Chart
Responsive
Deployment
EnterpriseJa
InvoicePlugin
JSON
HTML5
Dashboard
LDAP

Motivation

Idee:

- Bereitstellung eines einheitlichen OCR API zur herstellerunabhängigen Implementierung der:
  - Text-Extraktion aus gescannten Dateien,
  - Layout-Analysen von gescannten Dokumenten,
  - Erkennung von Scan PDFs und Umwandlung in schrifterkannte PDFs
  - Umwandlung von Bild Dateien in schrifterkannte PDFs.

Realisierung:

- Mit Hilfe von GhostScript, PDFBox und Tesseract
- Als Javascript API für die Integration in die Documents Scripting Engine (otrOCR).
- Als Commandline Utility für die Integration in das WPL und den Archiv-Index (ocr.cmd)

Historie

Verschiedenen Projektlösungen

- Vorgelagertes OCR
- Migration von TIF Archiven in PDF
- Migration von PDF Scan Archiven in schrifterkannte PDFs
- Notwendigkeit einer Plattformunabhängigen Lösung (Linux / Windows)
- Integration alternativer OCR Engines zur bisher verwendeten Abbyy FRE 10 Lösung

Stand heute

- alpha Version eines plattformunabhängigen OCR API
- auf Basis von tesseract, pdfbox und GhostScript

tesseract

Tesseract Historie

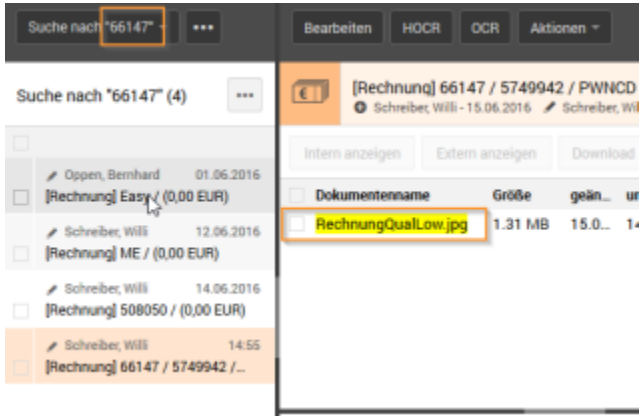
- Texterkennungssoftware die ursprünglich von HP (1985 – 1995) entwickelt wurde.
- Heute wird das Projekt von Google betreut und auf GitHub weiterentwickelt.
- Google verwendet tesseract unter anderem für das Projekt google books

Tesseract Features

- Eingabeformate: JPEG, PNG, TIFF, MULTITIFF … (kein PDF)
- Ausgabeformate: PDF, HOCR, TEXT
- Unterstützt derzeit ca. 40 Sprachen
- Layoutanalyse
- Hohe Erkennungsrate
- Apache Lizenz

Anwendungsbeispiel WPL

ocr CLI für WPL nutzen



Anforderung

- Textinhalte aus Bilddateien dem Volltextindex hinzufügen.
- Unterstützte Formate: JPEG, PNG, TIFF, MULTITIFF … (kein PDF)

Umsetzung

- Ergänzung des CLI in der Datei wpl.ini
- Auch für das Archiv anwendbar

```
20  [filter]
21  pdf=pdf\pdftotext -q -enc UTF-8 %d -
22  doc=tika\tika.bat -eUTF8 "%d"
23  xls=tika\tika.bat -eUTF8 "%d"
24  ppt=tika\tika.bat -eUTF8 "%d"
25  rtf=tika\tika.bat -eUTF8 "%d"
26  docx=tika\tika.bat -eUTF8 "%d"
27  xlsx=tika\tika.bat -eUTF8 "%d"
28  pptx=tika\tika.bat -eUTF8 "%d"
29  html=html\lynx -cfg=html\lynx.cfg -force_html -dump "%d"
30  jpg=..\addon\otrOcr\ocrBridge\ocr.cmd -mode textstrip -if "%d" -c -it jpg
31  txt=
32  log=
33  ini=
```

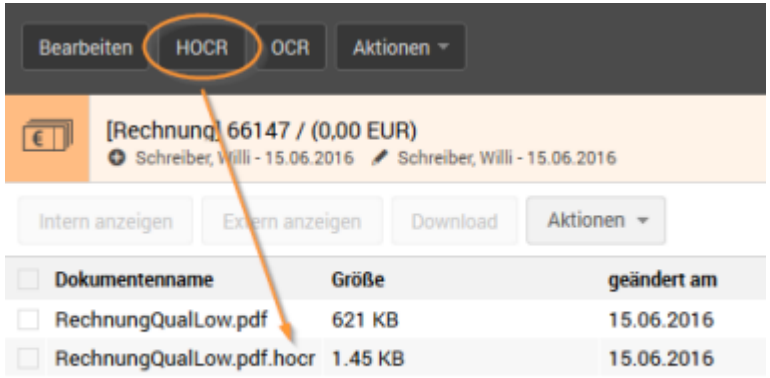
otris software

Anwendungsbeispiel HOCR

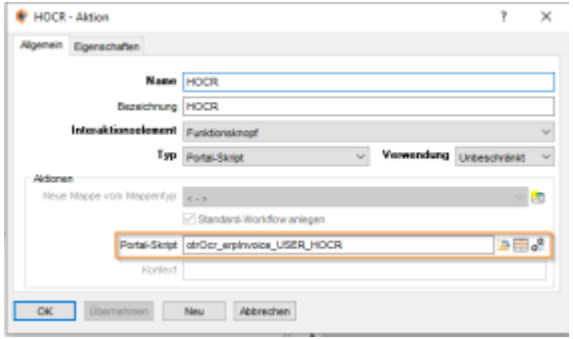
HOCR XML Datei erstellen

Anwendungsfall

- Nutzung der ocr API zum Erstellen eines XML Layout Dokuments zur Positionsbestimmung von Textinhalten.
- Zu Demo-Zwecken als Benutzerdefinierte Aktion an der Rechnungsmappe.
- HOCR Datei wird sichtbar parallel zur original Datei mit der Endung .hocr abgelegt.



HOCR XML bereitstellen




```
1  //#import "otrOcr"
2  /**
3   * @filetype erpInvoice
4   * @action HOCR
5   *
6   * Benutzerdefinierte Aktion - Erstellt HOCR Datei parallel zu allen
7   * hochgeladenen PDF Dateien
8   *
9   * @author Michael Eismann <eismann@otris.de>
10  * @copyright 2016 otris software AG
11  */
12 var docFile = context.file;
13 if(docFile){
14     otrOcr.doOcr(
15         docFile // current docfile
16         ,"deu"  // ocr language
17         ,"pdf"  // handle only PDF files
18         ,"hocr" // output format home
19         ,null   // Source Register / null = all Documents Register
20         ,null   // Target register / null = same register as source register
21         ,false); // true, only Documents with new Flag
22 }
```

otris software
vereinfacht Verantwortung.

Vielen Dank

Michael Eismann
Bereichsleiter Consulting

eismann@otris.de
www.otris.de

otris software AG
Königswall 21
44137 Dortmund