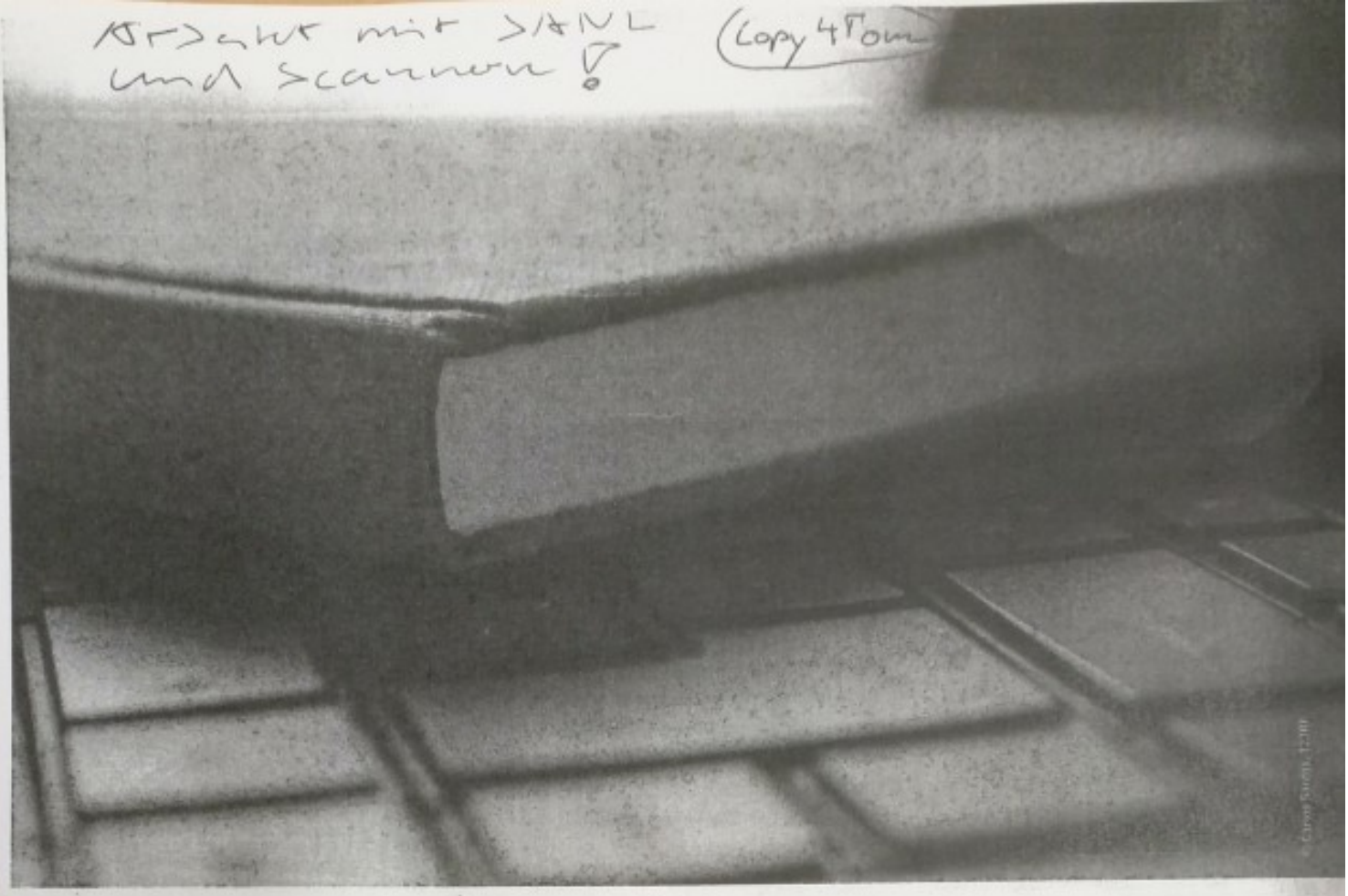


Mit Tesseract und GImageReader Texte digitalisieren

Reichlich Lesefutter

Gute OCR-Programme für den Desktop sind Mangelware. Mit GImageReader steigt jetzt ein vielversprechender Neuling in den Ring. Erik Bärwaldt

Lange Zeit gestaltete sich der Einsatz von Scannern unter Linux aufgrund fehlender oder fehlerhafter Treiber als Glücksspiel. Inzwischen hat sich die Situation durch das Sane-Projekt und das kommerzielle Paket Vuescan deutlich entspannt: Beide unterstützen zuverlässig mehrere Hundert Scanner-Modelle jeder Bauart. Doch nicht jeder Anwender begnügt sich damit, Papiervorlagen zu scannen und danach als Bilddatei auf dem Massenspeicher abzulegen.

Möchte man die digitalisierten Texte weiterverarbeiten, setzt das eine leistungsfähige Texterkennung voraus. Bei sogenannten OCR-Software („Optical Character Recognition") variiert die Qualität extrem, wobei Linux jedoch inzwischen in der ersten Liga mitspielt. Die Entwicklung von entsprechender Software gestaltet sich alles andere als trivial: Da Scanner lediglich Rastergrafiken liefern, liegt es am Programm, zunächst die grafischen Elemente auf der Vorlage vom eigentlichen Text zu unterscheiden. Im nächsten Schritt muss die Software Fehler in der Rastergrafik eliminieren.

Schlechte und schief eingescannte Vorlagen oder handschriftlich in einen gedruckten Text eingefügte Symbole, Anmerkungen und Linien verursachen vor allem bei Schwarz-Weiß oder Graustufen-Vorlagen fehlerhafte Pixel, die zu Problemen bei der Erkennung führen. Moderne OCR-Programme verwenden Algorithmen, die Pixel in der Umgebung

README

Nur wenige Programme beherrschen das automatisierte Erkennen von Texten unter Linux. Mit der OCR-Engine Tesseract und dem darauf abgestimmten grafischen Frontend GImageReader sparen Sie sich jedoch eine Menge Tipparbeit.

prüfen und den Scan korrigieren, falls die Muster nicht übereinstimmen.

Anschließend beginnt die eigentliche Arbeit: Die Software gleicht die gefundenen Elemente gegen eine im Programm vorhandene Datenbank mit Mustern ab. Hier gilt: Je umfangreicher diese Datenbank ausfällt und je mehr Schriftschnitte sie enthält, desto treffsicherer arbeitet die Software im ersten Durchlauf.

In einem weiteren Durchlauf prüft das OCR-Programm dann mögliche Inkonsistenzen: So treten bei weniger ausgereiften Programmen Verwechslungen von ähnlichen Zeichen und Ziffern (S / 5, B / ß / 8) auf. Anhand von Wörterbüchern und linguistischen Prüfverfahren korrigiert die Software solche Resultate. Gute OCR-Programme ermöglichen zudem noch die manuelle Korrektur der erkannten Texte. Dafür bringt die Software idealerweise eine Lernfunktion mit, die anhand der manuellen Korrekturen neue Prüfmuster generiert.

Eine besondere Herausforderung ergibt sich für entsprechende Programme bei Einsatz verschiedener Sprachen und Schriftschnitte. Insbesondere mehrsprachige Vorlagen verlangen der OCR-Engine hohe Erkennungsraten auch bei vielfältigen Sonderzeichen ab. Darüber hinaus wirken sich Schriftschnitte auf das Ergebnis aus, wie Fettung oder Kursivierung. Speziell bei Dokumenten mit alten Schriftarten wie Fraktur, Textura, Schwabacher oder Rotunda gelingt nur wenigen Programmen eine befriedigende Digitalisierung.

Unter Linux

Unter Linux hat sich die für die Kommandozeile konzipierte OCR-Engine Tesseract als außerordentlich leistungsfähig erwiesen. Sie arbeitet oft im Duo mit einer grafischen Oberfläche und eignet sich so für den Einsatz auf dem Desktop. Tesseract stammt ursprünglich vom US-amerikanischen Computerhersteller Hewlett-Packard, der sie zwischen 1985 und 1995 weiterentwickelte. Von 1995 bis 2005 lag die Software brach; der Konzern hatte dieses Marktsegment inzwischen aufgegeben. 2005 nahm sich Google der Software an und gab sie nach einer Code-Revision als freie Software unter der Apache-Lizenz an die Entwicklergemeinde weiter. Das führte dazu, dass sie sich im Linux-Universum verbreitete und dank entsprechender Module sogar deutsche Frakturschriften erkennt. Das modular aufgebaute Programm kann dank entsprechender Erweiterungen rund 100 Sprachen erkennen, wobei sich allein drei Module der Fraktur widmen.

Oberflächliches

Da OCR-Engines in aller Regel als Programme für die Kommandozeile vorliegen, die von Haus aus über keine grafische Oberfläche verfügen, haben sich freie Entwickler dieses Defizits angenommen. Als Ergebnis dieser Bemühungen entstand eine stattliche Anzahl von Frontends für unterschiedliche Zwecke.

Diese GUIS zielen meist auf eine spezielle OCR-Engine ab, deren Fähigkeiten sie nach Möglichkeit durch entsprechende Optionen komplett abbilden. Neben Tesseract und dem in Russland entwickelten Cuneiform sind hier die Engines Gocr und Ocrad zu nennen, die ebenfalls als freie Software bereitstehen.

Die grafischen Programme unterstützen dabei meist sämtliche von den OCR-Engines genutzten Formate für Bilder. Über entsprechende Sane-Module eignen sie sich zudem für den direkten Einsatz mit einem Scanner. Die Integration von Sane erhöht den Bedienkomfort merklich, da in diesem Fall das Erkennen des Texts direkt nach dem Scannen startet. Andere Programme speichern zunächst die digitalisierten Daten als Bilddatei ab und laden diese anschließend ins Texterkennungsprogramm.

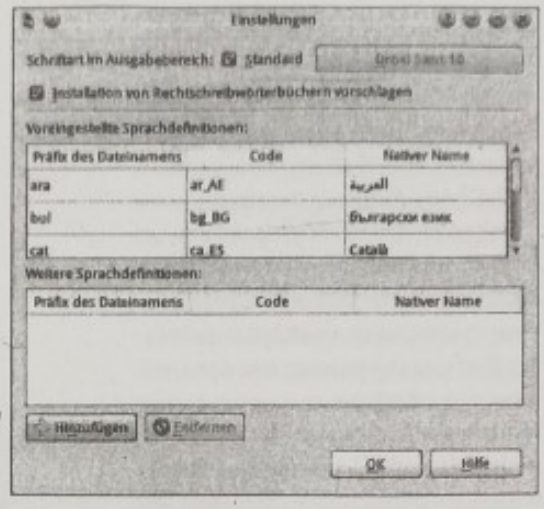
Unter Linux spielen insbesondere YAGF (Yet Another Graphical Frontend) und OCRFeeder als grafische Oberflächen für OCR-Engines eine Rolle. Beide kooperieren zwar anstandslos mit Sane und verfügen über recht eingängig zu bedienende Dialoge, weisen aber spezielle beim Erkennen von Bereichen nach wie vor gravierende Schwächen auf. Der Einsatz der Engines klappt nicht immer reibungslos, da beide Programme mehrere parallel installierte OCR-Lösungen unterstützen und dabei gelegentlich aus dem Tritt geraten.

GImageReader 3.2.1
LU/ocr/

Erste Schritte

Als relativ unbekanntes Frontend fristet der GImageReader ein Schattendasein. Er schreibt sich neben einer einfachen Oberfläche vor allem ein schlankes Design auf die Fahnen. Die meisten gängigen Distributionen führen die Software inzwischen im Repository, sodass sie sich problemlos über die Paketverwaltung installieren lässt.

Zusätzlich stehen für GImageReader der Quellcode zur manuellen Integration



1 Der Konfigurationsdialog bietet unter anderem die Möglichkeit, zu überprüfen, ob Sie alle benötigten Sprachpakete für die OCR-Engine installiert haben.

in ein noch nicht unterstütztes Derivat und ältere Varianten der Software auf der Projektseite bei Sourceforge zum Herunterladen bereit. Den Quellcode der derzeit aktuellen Version finden Sie auf der Heft-DVD dieser Ausgabe.

Vorab befördern Sie jedoch die Engine Tesseract auf die Platte, samt den zugehörigen Paketen für die benötigten Sprachen, da GImageReader ausschließlich mit dieser Engine zusammenarbeitet. Tesseract finden Sie in aller Regel ebenfalls im Repository der Distributionen.

2 Die Software erlaubt es, manuell oder automatisch mit Text gefüllte Bereiche zu identifizieren. Diese nummeriert sie entsprechend durch.

Nach der Installation von GImageReader finden Sie im Untermenü *Grafik* des Startmenüs einen entsprechenden Eintrag. Das Frontend begrüßt Sie mit einer zunächst fast leeren Oberfläche. Neben einer horizontalen Schalterleiste am oberen Rand gibt es zwei nebeneinander angeordnete, zu Beginn noch leere Fenster.

Zunächst konfigurieren Sie die Software. Dazu klicken Sie rechts oben in der Leiste auf den Schalter mit dem Zahnrad-Symbol und wählen im daraufhin erscheinenden Kontextmenü den Eintrag *Einstellungen*. Im entsprechenden Dialog werfen Sie einen Blick in die Tabelle der Sprachdefinitionen: Hier sollte in der linken Spalte *Präfix des Dateinamens* der Eintrag *deu* auftauchen, um deutsche Texte mit ihren Umlauten und Sonderzeichen korrekt einzulesen.

Planen Sie, Texte aus älteren Publikationen zu verarbeiten, die Frakturschriften enthalten, so sollten in der Präfix-Spalte zusätzlich die beiden Einträge *deu-f* und *deu-frak* auftauchen. Fehlen diese, so installieren Sie die passenden Pakete über das Paketmanagement nach.

Weiterer Einstellungen bedarf es zunächst nicht, sodass Sie den Dialog mit einem Klick auf *OK* unten rechts schließen. Danach steht die Software zum ersten Einsatz bereit 1.

Quellen

Beim Erkennen von Vorlagen erweist sich das Programm als äußerst flexibel: Es liest sowohl PDF-Dokumente als auch Bilddateien verschiedener Formate ein und versucht, deren Inhalte zu erkennen.

Dazu klicken Sie links im Programmfenster im Bereich *Quellen* bei aktivem Reiter *Dateien* auf das symbolisierte Ordner-Icon und wählen im anschließend geöffneten Dialog die gewünschte Datei aus. Enthält diese mehrere Seiten, besteht die Möglichkeit, durch das Dokument zu blättern und eine bestimmte Seite aufzurufen. Dazu nutzen Sie die Elemente, die die Software in der Leiste oben mittig einblendet.

Ist ein Scanner an den Computer angeschlossen, und möchten Sie von diesem direkt Texte einlesen, so klicken Sie im Bereich *Quellen* auf den Reiter *Erwerben*. Sie sehen nun einen Dialog, der Parameter des Scanners anzeigt. Haben Sie mehrere Scanner im Einsatz, so wählen Sie zunächst das gewünschte Gerät aus der Auswahl aus.

Als Nächstes legen Sie den Pfad für die Ausgabe fest. Die beiden darunter befindlichen Auswahlfelder definieren den Modus zum Scannen (Graustufen oder Farbe) sowie die Auflösung. Bei schlechten Vorlagen sollte diese höher ausfallen, bei guten Vorlagen oder großen Schriften belassen Sie die Auflösung auf den voreingestellten 200 dpi.

Haben Sie alle gewünschten Parameter eingestellt, klicken Sie unten mittig auf *Scannen*. Der Scanner liest nun die Vorlage ein und zeigt ein unbearbeitetes Bild im rechten Bereich an. Zusätzlich speichert die Software das Bild im angegebenen Verzeichnis, wobei sie mehrere Dateien fortlaufend nummeriert. Diese Dateien eignen sich bei späteren Durchläufen als Vorlage.

Markierung

Nach Öffnen einer Datei oder Einlesen der Vorlage ziehen Sie mit der Maus einen Rahmen um den Bereich, den Sie mit der OCR-Engine bearbeiten möchten. Der Zeiger wandelt sich dabei zu einem kleinen Kreuz. Alternativ nutzen Sie den Schalter *Layout automatisch erkennen*.

Die Software ermittelt nun vorhandene Textbereiche und markiert diese mit

fortlaufend nummerierten Rahmen. Dabei erkennt sie unter Umständen sogar Spalten und nummeriert diese ebenfalls in der korrekten Reihenfolge. Für Bereiche, deren Inhalt Sie nicht benötigen, entfernen Sie die Markierungen durch einen Rechtsklick auf den Rahmen und die Auswahl der Option *Löschen* im Kontextmenü. Die Software zählt die verbleibenden Elemente anschließend automatisch neu durch 2.

Nach dem Markieren der Bereiche überlassen Sie das Erkennen dem Backend Tesseract: Klicken Sie dazu auf *Auswahl erkennen*. Sollte das falsche Sprachmodul eingestellt sein, klicken Sie zunächst rechts daneben auf den kleinen Pfeil und wählen im geöffneten Auswahlfeld das korrekte aus. Die Software öffnet nun rechts ein neues überlagerndes Fenster und zeigt darin die Ergebnisse an. Der in Abbildung 3 angezeigte Text erfordert je nach Qualität der Vorlage und dem daraus resultierenden Ergebnis noch etwas Arbeit. Korrekturen nehmen Sie direkt in der Textanzeige vor.

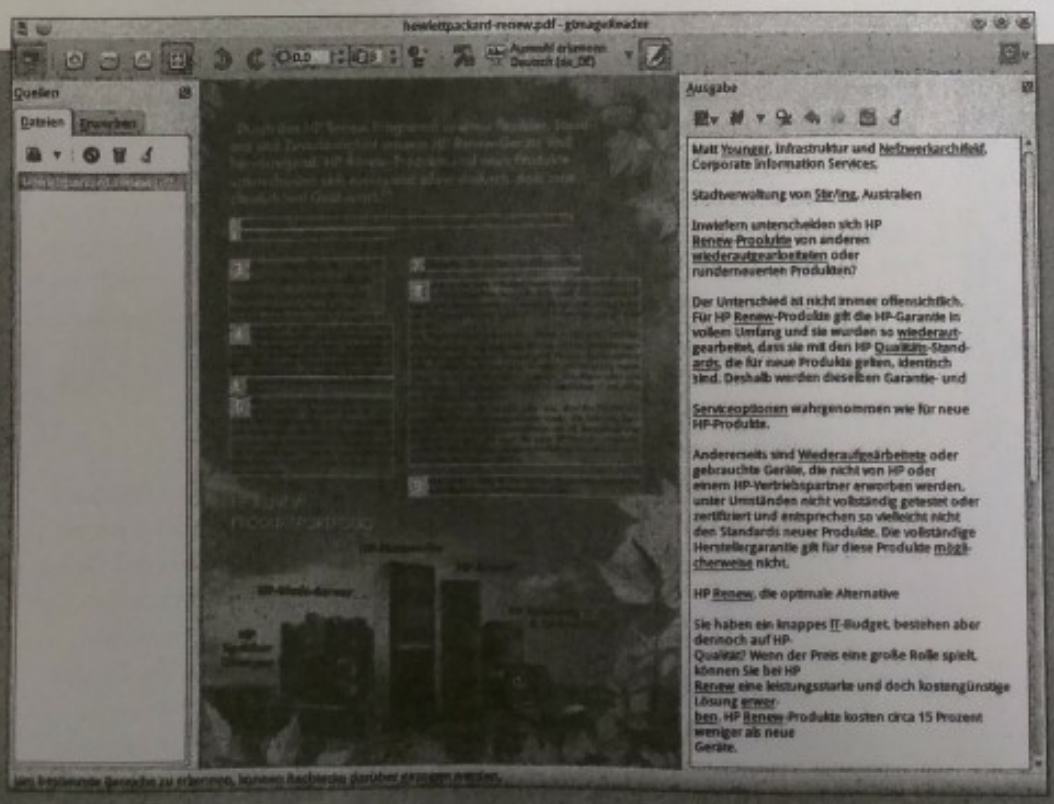
Das Fenster bietet zudem einige grundlegende Funktionen zum Editieren: So besteht die Möglichkeit, überschüssige Umbrüche automatisch zu entfernen, indem Sie nach dem Markieren des entsprechenden Textes auf den Schalter *Zeilenumbrüche im markierten Text entfernen* klicken. Eine Suchfunktion gestattet es außerdem, fehlerhaft eingelesene identische Zeichenfolgen zu ersetzen.

Haben Sie alle Modifikationen am Rohtext vorgenommen, so sichern Sie diesen mit einem Klick auf *Ausgabe speichern* und legen im anschließenden Dialog Pfad und Dateinamen fest. GImageReader legt die Daten in Form einer unformatierten Textdatei ab, die Sie mit problemlos mit einem beliebigen Editor weiterverarbeiten können. Daneben vermeidet die Beschränkung auf ein solch simples Format ansonsten unvermeidliche Konvertierungsfehler.

Licht und Schatten

In unserem Test musste das Duo Tesseract/gImageReader verschiedene Vorlagen in deutscher und englischer Sprache einlesen. Diese reichten von aktuellen gedruckten Texten über alte, mit der Schreibmaschine getippten Blättern bis hin zu Buchseiten mit Frakturschrift.

Mit herkömmlichen Schreibmaschinenseiten auf weißem Papier hatte die



3 Erkannte Texte zeigt die Software rechts in einem eigenen Fenster an. Sie erfordern meist etwas Nacharbeit.

Software weder in deutscher noch in englischer Sprache Probleme. Selbst gedruckte Seiten mit verschiedenen Schriftgrößen und Schriftschnitten zeigten erstaunlich gute Ergebnisse: Hier fiel kaum Nacharbeit an, die Erkennungsrate lag bei nahezu 100 Prozent. Weniger erfreulich fielen die Ergebnisse mit Frakturschriften aus: Hier versagte Tesseract vor allem bei Vorlagen mit weniger als 10 Punkt Schriftgröße. Flecken auf altem Papier brachten die Applikation ebenfalls gelegentlich aus dem Tritt.

Erstaunlich gut fielen die Ergebnisse beim Scannen und Auswerten aus, wenn im Spaltensatz das automatische Erkennen von Textbereichen aktiviert war: Selbst bei farbigen Hintergründen identifizierte das Programm die Bereiche und nummerierte sie in der korrekten Reihenfolge, sodass Tesseract anschließend den Fließtext vollständig wiedergab 4.

Je nach der Beschaffenheit der Vorlage verbessern Sie das Ergebnis beim Erkennen unter Umständen, indem Sie schlicht die Auflösung des Scanners anpassen. Dazu experimentieren Sie im Reiter *Erwerben* im Auswahlfeld *Auflösung:* mit unterschiedlichen Werten. Bei hohen Auflösungen beansprucht das Scannen der Vorlagen allerdings auch entsprechend mehr Zeit.

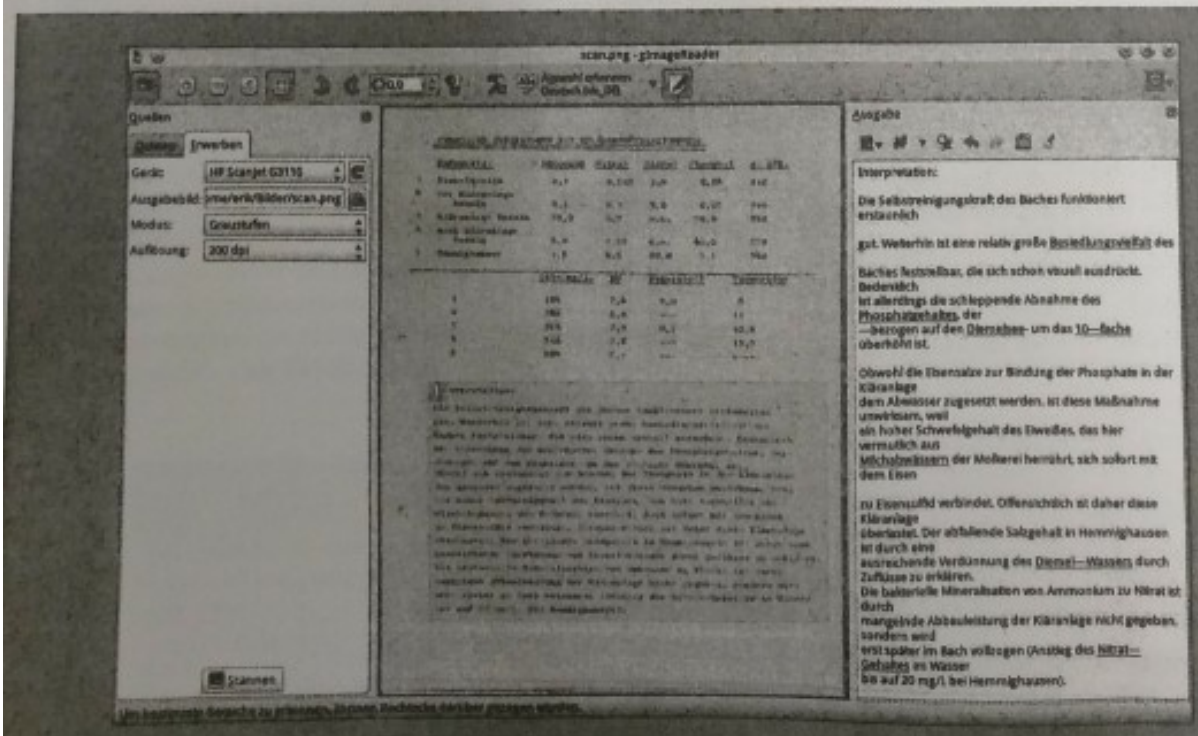
Das Anpassen der Sprachoptionen führt manchmal ebenfalls zu besseren Resultaten: So differenzieren sowohl das Modul für Deutsch als auch jenes für Englisch zwischen verschiedenen Dialekten. Im deutschen Modul wählen Sie zwischen deutscher, österreichischer und schweizer Vorlage, im englischen zwischen Amerikanisch, Britisch und Kanadisch.

Fazit

Gute Programme für die Texterkennung unter Linux gibt es bereits seit längerer Zeit. Allerdings haperte es bislang noch an praxistauglichen grafischen Frontends. GImageReader behebt dieses Manko: Das Programm konzentriert sich mit einer eingängigen Oberfläche ohne Gimmicks auf die wirklich relevanten Arbeitsschritte und kommt daher schnell zu Ergebnissen.

Dank der ausgereiften OCR-Engine Tesseract genügen dabei Vorlagen von nur durchschnittlicher Qualität, und das Software-Duo beherrscht außerdem viele Fremdsprachen. Vor allem Heimanwender, die sporadisch einzelne Seiten scannen und die so gewonnenen Texte weiterverarbeiten möchten, gelangen mit Tesseract und GImageReader schnell zu befriedigenden Ergebnissen. (agr/jlu) ■

Weitere Infos und interessante Links
www.linux-user.de/qr/38977



4 Auch weniger gute Vorlagen, wie hier eine alte Schreibmaschinenseite, meistert Tesseract problemlos.

EDIT: ein Artikel vom gleichen Autor aus dem Heft 05/2011 zu Tesseract http://www.linux-community.de/Internal/Artikel/Print-Artikel/LinuxUser/2011/05/Texterkennung-mit-Tesseract . Und noch ein Artikel Eingescannte "Texte automatisch erkennen: Alphabetisierung" http://www.linux-community.de/Internal/Artikel/Print-Artikel/LinuxUser/2011/04/Eingescannte-Texte-automatisch-erkennen aus dem Heft 04/2011.