

Module 2

Image acquisition & preprocessing

Uwe Springmann

Centrum für Informations- und Sprachverarbeitung (CIS)
Ludwig-Maximilians-Universität München (LMU)



2015-09-14

Motivation

- remember: the complete OCR workflow consists of several steps:
 - 1 image acquisition
 - 2 preprocessing
 - 3 (ground truth production, model training)
 - 4 recognition
 - 5 evaluation
 - 6 postprocessing: annotation, error correction, tagging, ...
- “a chain is only as strong as its weakest link”:
bad images/preprocessing will severely limit the quality of your end result
- trade-off: fast result against quality result (requires some manual processing)
- make an informed decision based on your objectives

Image acquisition

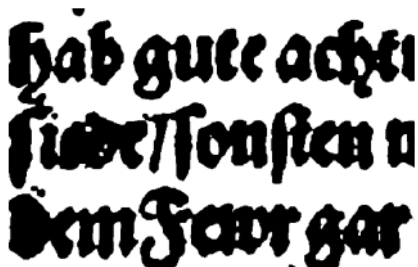
Where to look for digitized books

- look for scans at [HathiTrust](#), [archive.org](#), [Europeana](#), [The European Library](#), [DDB](#), [Wikisource](#), [BSB](#), or [Google books](#)
- try to find the best scan (Google books are often the worst); larger file sizes point to higher resolution
- especially good scans can be found in DFG-funded projects ([VD16](#), [VD17](#), [VD18](#))
- if you cannot find a scan:
 - have it scanned from an institution (can be expensive)
 - your local research library may be able to help you
 - or do-it-yourself:
 - procure your own copy, take the pages apart and scan them
 - scan either in color or (at least) grayscale
 - resolution: preferably 300-400 dpi; higher resolution may not be better (connected components in letter shapes may fall apart)
 - the [DFG digitisation guidelines](#) may be helpful

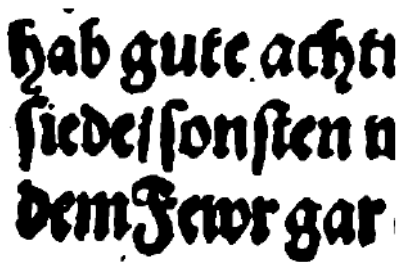
Some tips for image acquisition

- often books found at Google are also available at a higher resolution at BSB (search BSB first)
- use the [BSB OPACplus catalog](#) to search for volumes (results can be filtered for online resources)
- at [archive.org](#), download “single page processed JP2 zip” file rather than pdf or djvu files (the latter are downgraded in resolution)
- avoid binarized images, do your own binarization later on
- publicly available images tend to be downsized 150 dpi “service copies” (pdf or jpg); you can ask for higher resolution original png or tiff images
- you can still OCR 150 dpi material, but if the results are not good enough for you, get 300 dpi scans before you do heavy postcorrection

Effect of image quality on recognition

A scan of a handwritten note in a black, stylized script. The text is arranged in three lines: 'Hab gute achi', 'siede/sonsten n', and 'dem Feuer gar'. The scan quality is lower, with some pixelation and less sharp edges on the letters.

(a) Google

A scan of the same handwritten note as in (a), but from a higher resolution source (BSB). The letters are sharper, more defined, and the overall image quality is superior, with less noise and better contrast.

(b) BSB

- the same scan with lower (Google) and higher (BSB) resolution
- after model training, the accuracy on test pages is 94% (Google) and 97% (BSB)

Preprocessing

Preprocessing tasks

- preprocessing consists of (some of) the following tasks:
 - splitting: split double-side images into single pages, or several columns into single-column images
 - cropping: get rid of (black) boundaries
 - deskewing: bring image to horizontal orientation
 - dewarping: “flatten” image, if scanned from warped pages
 - despeckle: noise reduction, suppress black spots (“speckles”)
 - binarization: separate signal (characters, black) from noise (background, white)
 - zoning: separate text zones from non-text (images, graphs etc.); separate semantically different text zones (running heads, page numbers, footnotes, columns, ...)
 - line segment: cut text zones in single text lines
- all OCR engines have some kind of built-in preprocessing facility
- however, for optimal results it is often better to do some manual tool-assisted preprocessing

Example: Gart der Gesundheit (printing of 1487)

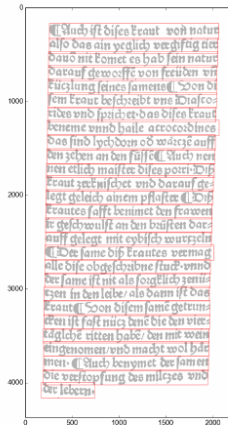
Johann Wonnecke von Kaub (Johannes von Cuba), Gart der Gesundheit (1487)



original image

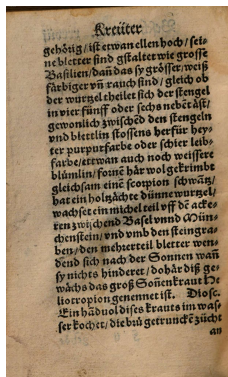
¶ Auch ist dīses kraut von nature also das ain yeglich vergifft die daus mit komet es hab sein natur darauf gewoiffe von freuden vñ künzlung seines samens. ¶ Von dīsem kraut beschreibet vñs Diascorides vñd sprichet das dīses kraut beneme vñnd haille atrocordines das sind lychedorn oð wärzē auff den zehen an den füßē. ¶ Auch nen nen etlich maister dīses port. Dīf kraut zet mischet vñd darauf gelegt geleich ainem pflaster. ¶ Dīf krautes safft benimet den frawen ir geschwulst an den brüsten dar auff gelegt mit eybisch wurzeln. ¶ Dē same dīf krautes vermag alle dīse obgeschribne stuch vñnd der same ist nit als sorglich zenußzen in den laibe als dann ist das kraut. ¶ Son dīsem same getruncken ist soft nūc denē die den vier täglich ritten habē den mit wein eingenomen vñd macht wol hāc men. ¶ Auch benymet der same die verstopfung des milczes vñd der lebēn.

binarized text zone



line segmented

Effect of preprocessing on recognition (Bodenstein 1557)



Kreüter
 gehözig / ist erwan ellen hoch / se-
 ne bletter sind gskaltee wie groesse
 Basilien / dan das sy grösser / weisse
 färbiger vñ rauch sind / gleich ob
 der wurzel theilet sich der stengel
 in vier fünff oder sechs nebet äst /
 gewonlich zwischen den stengeln
 vñ bletterlin stossens herfür bey-
 ter purpurfarbe oder schier leib-
 farbe / erwan auch noch weisse
 blümlin / formē hār wol getrimbe
 gleichsam einē scorpion / schwāg /
 hat ein holgächte dünne wurzel /
 wachset ein mibel teil vff dē acke-
 ren zwischē basel vñ dē Mü-
 chenstein / vñ vmb den steingra-
 ben / den mehrteil bletter wens
 den sich nach der Sonnen wā-
 sy nichts hinderet / dohār diß ge-
 wächs das groß sonentraut He-
 liotropion genennet ist. Diosc.
 Ein hād uol dises krauts im was-
 ser kochet / die bñ getrunckē zucht
 an

Character accuracy:

OCR engine	orig.	prepr.
Tesseract (Fraktur)	35%	71%
Abbyy (Fraktur + hist. lexicon)	78%	79%

Preparing the document

- to begin preprocessing, we need single page images in tif or png format
- often you will start from images contained in a single large pdf file or in other formats (jpg, JP2)
- document splitting and format conversion can be done by these open source tools:
 - pdf splitting: [PDFtk](#) (Linux: pdftk package)
 - format conversion (choose one of these for batch processing):
 - convert from [ImageMagick](#) suite
 - convert from [GraphicsMagick](#) suite
 - pdftoppm, pdftimages from [Xpdf](#) tools, or (Linux) from poppler-utils package
- if your image is blurred, has an unusual perspective, etc., you can get some help on image preprocessing here:
 - Fred's ImageMagick Scripts (ready-made scripts for a wide variety of tasks)
 - Dan Bloomberg's [leptonica](#) package (look at the [dewarping](#) example!)
- further preprocessing will be done by [ScanTailor](#)

Example: Goethe, Wahlverwandtschaften (1809)

- available at BSB: [Wahlverwandtschaften, vol. 1](#)
- download and rename as `goethe.pdf`
- the following commands assume:
 - a Linux / MacOS system, but similar tools exist for Windows (see above)
 - that you have installed the necessary software (for Debian-flavored Linux variants, this is as easy as step 0)
- step 0: install software (Debian-flavored Linux)

```
$ sudo apt-get install pdftk poppler-utils \
    imagemagick scantailor
```
- step 1: split pdf in single pages

```
$ mkdir pdf
$ pdftk goethe.pdf burst output pdf/%04d.pdf
```

Example (Goethe): pixel size, convert to png

- step 2: find pixel size of images in pdf
 - for scanned books, pdf is just a container format for included images
 - as a vector format, a pdf does not have a pixel size

```
$ pdfimages -list 0100.pdf
page  num  type  width height color comp bpc  enc
-----
    1     0 image   714  1283  rgb     3   8  jpeg
```

- the included jpeg image has 714x1283 pixels
 - for jpeg images in pdf, step 1 is just `pdfimages -j gdg.pdf gdg`
- step 3: convert pdf (or other format) to png

```
$ mkdir png
$ cd pdf
$ for f in pdf; do convert "$i" "${i}/.pdf/.png"; done
$ mv *.png ../png
```

Example (Goethe): resolution

- step 4: find resolution of image (needed as input for ScanTailor)
 - sometimes the scanning resolution (dpi) is given in metadata (archive.org)
 - if you know the physical size of your page:
divide pixel height (or width) by height (or width) in inch (1 in = 2.54 cm)
 - png image has 714x1283 pixels (same as jpeg;
otherwise use convert with `-density` option)
 - take pixel measurements from png image with ruler (last page) at 100% image size (okular or other viewer)
 - rule of thumb: height of 6 text lines ca. 1 inch
 - pixels per inch (ppi, used in imaging) correspond to dots per inch (dpi, used in printing)

Example (Goethe): resolution (cont'd)

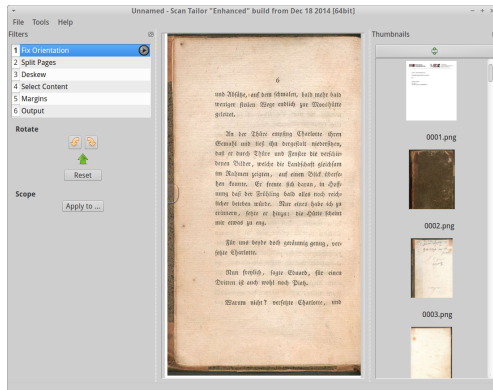
in DFG scans, a ruler was scanned with one of the last pages:
measure ruler size in pixels



- here: $355 \text{ pixels} / (5/2.54) \text{ inch} = 180 \text{ ppi}$
- not ideal resolution, but this is what we got
- resolution of 150 .. 180 dpi to be expected for downloadable files (lower size saves bandwidth)

Example (Goethe): ScanTailor

Convert png image into binarized tif using ScanTailor



ScanTailor with png of original image

6
und Abfäße, auf dem schmalen, bald mehr bald weniger steilen Wege endlich zur Mooschütte geleitet.

An der Thüre empfing Charlotte ihren Gemahl und ließ ihn dergestalt niederstehen, daß er durch Thüre und Fenster die verschiedenen Bilder, welche die Landschaft gleichsam im Rahmen zeigten, auf einen Blick übersehen konnte. Er freute sich daran, in Hoffnung daß der Frühling bald alles noch reichlicher beleben würde. Nur eines habe ich zu erinnern, setzte er hinzu: die Thüre scheint mir etwas zu eng.

Für uns beyde doch geräumig genug, versetzte Charlotte.

Nun freysich, sagte Eduard, für einen Dritten ist auch wohl noch Platz.

Wacum nicht? versetzte Charlotte, und

tif image as result of preprocessing

Example (Goethe): recognition compared

character vs. word accuracy in %:

	characters		words	
OCR engine	png	tif	png	tif
Tesseract	86.42	96.06	68.18	84.55
OCROPUS	95.33	96.06	82.73	89.09
Abbyy FR 11	96.79	95.33	92.73	91.82

Conclusion

- for 19th century Fraktur printings, ca. 95% character accuracy can be achieved by any engine (without training)
- separate preprocessing makes a difference for character (Tesseract) and word accuracies (Tesseract, OCRopus)
- Abbyy has very good automatic preprocessing, separate preprocessing is unnecessary