

# 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

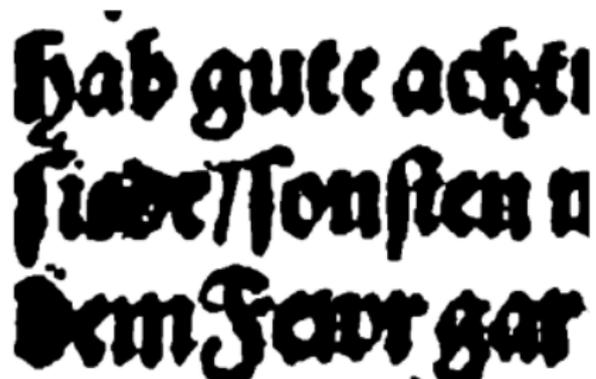
## Where to look for digitized books

- 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

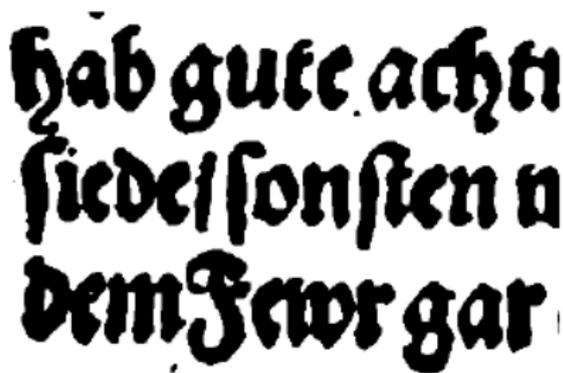
- often books found at Google are also available at a higher resolution at BSB (search BSB first)
- use the [BSB OPACplus catalog](#) to search for volumes (results can be filtered for online resources)
- at [archive.org](#), download “single page processed JP2 zip” file rather than pdf or djvu files (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 low-resolution scan of a German proverb. The text is in a black, stylized font on a white background. The letters are somewhat blurry and lack fine detail, characteristic of a lower resolution scan.

Hab gute aechti  
siede/sonsten n  
dem Feuer gar

(a) Google

A high-resolution scan of the same German proverb. The text is in the same black, stylized font on a white background. The letters are much sharper and more detailed than in the low-resolution scan, showing clear edges and fine texture.

Hab gute aechti  
siede/sonsten n  
dem Feuer gar

(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

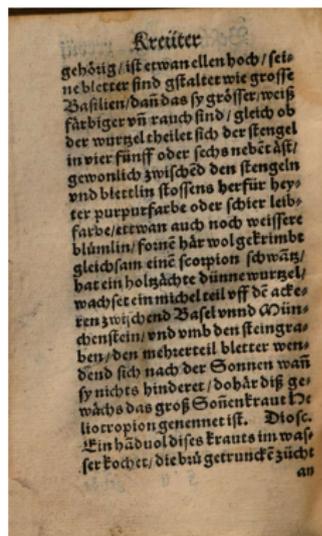
¶ Auch ist dieses Kraut von natur also das ain yeglich verghfug die daus mit Komet es hab sein natur darauf gewoiffe von freuden vñ Kücklung seines samens ¶ Von diesem Kraut beschreibet vns Diascorides vnd spricht das dieses Kraut beneme vñnd haille atrocordines das sind lychdom oð wärzge auff den zehen an den füße ¶ Auch nen nen etlich maister dieses port-Dif Kraut zett mischet vñnd darauf gelegt gleich ainem pflaster ¶ Dief Krautes sofft benimet den frauen fr geschwulst an den brüsten dar auff gelegt mit eybisch wurtzeln ¶ Die same dif Krautes vermag alle dise obgeschribne stuch vñnd der same ist nit als sorglich zenuetzen in den leibe als dann ist das Kraut ¶ Son diesem same getruncken ist soft nütz den die den vier täglich ritten habē den mit wein ingenomen vñnd macht wol hde men ¶ Auch benymet der same die verstopfung des milczes vñnd der lebern.

binarized text zone



line segmented

## Effect of preprocessing on recognition (Bodenstein 1557)



Kreütter  
 gehöbig / ist erwan ellen hoch / sei-  
 ne bletter sind gskalter wie grosse  
 Basilien / dan das sy grösser / weiß  
 färbiger vñ rauch sind / gleich ob  
 der wurzel theilet sich der stengel  
 in vier fünff oder sechs nebēt äst/  
 gewonlich zwischē den stengeln  
 vñ bletterin stossens herfür bey-  
 ter purpurfarbe oder schier leib-  
 farbe / erwan auch noch weissere  
 blümlin / formē hār wol getrimbe  
 gleichsam einē scorpion / schwāg/  
 hat ein holzächte dünne wurzel/  
 wachset ein mibel teil vff dē acke-  
 ren zwischēnd Basēl vñnd Mün-  
 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 Sönettraut He-  
 liotropion genemmet ist. Diose.  
 Ein hād uol dieses Frauts im was-  
 ser Kochet / die bü getrunckē zücht  
 an

	char. acc.	
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, pdfimages 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, Wahlverwandschaften (1809)

- available at BSB: [Wahlverwandschaften, 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)

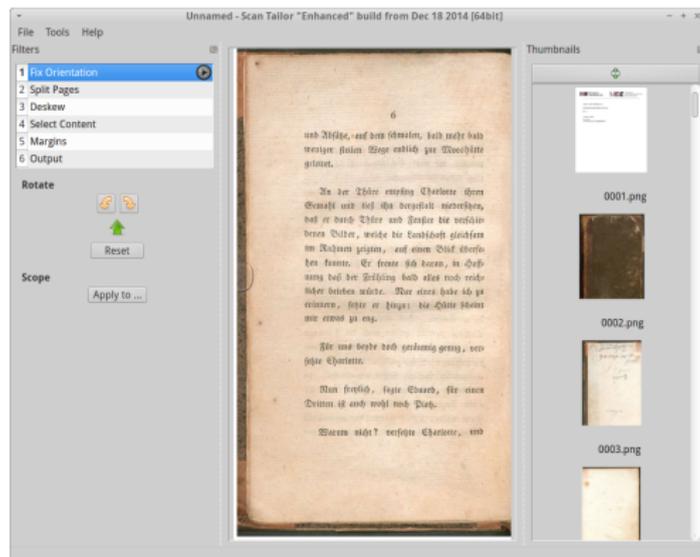
in DFG scans, a ruler was scanned with one of the last pages:  
measure ruler size in pixels



- 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 Hütte scheint mir etwas zu eng.

Für uns beyde doch geräumig genug, versetzte Charlotte.

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

Warum nicht? versetzte Charlotte, und

tif image as result of preprocessing

## Example (Goethe): recognition compared

character vs. word accuracy in %:

OCR engine	char.		word	
	png	tif	png	tif
Tesseract	86.42	96.06	68.18	84.55
OCRopus	95.33	96.06	82.73	89.09
Abby 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