Tesseract OCR

Text from images

CONCEPTS

The call is tesseract <input> <outputbase> [options] [formats]. The output base gets its extension added; use stdout to print instead.

Image 300 dpi, clean, upright
↓ `segment (--psm)`
Layout blocks, lines, words
↓ `recognize (--oem, -l)`
Text words + confidence
↓ `write (config names)`
Output txt, pdf, hocr, tsv

Options such as -l and --psm go before the output format names. Image quality (about 300 dpi, high contrast, straight) matters more than any flag.

BASICS

`tesseract <img> <out>`
OCR an image into <out>.txt

`tesseract <img> stdout`
Print recognized text

`tesseract stdin stdout`
Read image from a pipe, print text

`tesseract <list.txt> <out>`
OCR images listed one per line

`tesseract --version`
Show version

LANGUAGES

`tesseract --list-langs`
List installed languages

`tesseract <img> <out> -l <lang>`
Recognize one language

`tesseract <img> <out> -l <lang>+<lang>`
Combine several languages

A language needs its <lang>.traineddata file in the tessdata folder. Distros ship extras as separate packages.

LAYOUT (--PSM)

`tesseract <img> <out> --psm 3`
Fully automatic layout (default)

LAYOUT (--PSM) (cont.)

`tesseract <img> <out> --psm 6`
One uniform block of text

`tesseract <img> <out> --psm 7`
One single text line

`tesseract <img> <out> --psm 11`
Sparse text, any order

Wrong results on a tiny crop or a form field? Try 7, 8 or 6 before anything else. Use 11 for scattered labels.

ENGINE & TUNING

`tesseract <img> <out> --oem 1`
Use the LSTM neural engine

`tesseract <img> <out> --dpi <n>`
Set input resolution

`tesseract <img> <out> -c <name>=<value>`
Set a parameter

`-c tessedit_char_whitelist=0123456789`
Digits only

OUTPUT FORMATS

`tesseract <img> <out> txt`
Plain text, <out>.txt (default)

`tesseract <img> <out> pdf`
Searchable PDF with text layer

`tesseract <img> <out> hocr`
hOCR HTML with word boxes

`tesseract <img> <out> tsv`
Tab-separated words, boxes, conf

`tesseract <img> <out> txt pdf hocr`
Write several formats at once

PREPARING IMAGES

`magick <in> -colorspace Gray <out>.png`
Convert to grayscale

`magick <in> -resize 200% <out>.png`
Enlarge small text

`pdftoppm -r 300 -png <in>.pdf <prefix>`
Rasterize a PDF to pages