Tesseract OCR Text from images

START HERE

tesseract *<img>* **stdout**
OCR an image to the terminal

tesseract *<img>* *<out>*
Write <out>.txt

tesseract *<img>* *<out>* **-l** *<lang>*
Pick the language

tesseract *<img>* *<out>* **--psm 6**
Treat as one text block

tesseract *<img>* *<out>* **pdf**
Make a searchable PDF

tesseract --list-langs
Show installed languages

tesseract *<img>* *<out>* **tsv**
Words with boxes and confidence

tesseract --version
Show version and libraries

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

CONCEPTS

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

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
All results are combined into one output per format.

tesseract --version
Show version

tesseract --help
Show usage

tesseract --help-psm
List page segmentation modes

tesseract --help-oem
List engine modes

tesseract --help-extra
Show advanced help

LANGUAGES

tesseract --list-langs
List installed languages

tesseract *<img>* *<out>* **-l** *<lang>*
Recognize one language
Default is eng. Codes are three letters, e.g. deu, fra.

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

tesseract *<img>* *<out>* **-l script/***<Script>*
Use a script model

tesseract --tessdata-dir *<dir>* **--list-langs**
List languages in another folder

TESSDATA_PREFIX=*<dir>* **tesseract** *<img>* *<out>*
Point to the tessdata parent

LANGUAGES (cont.)

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)

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

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

tesseract *<img>* *<out>* **--psm 8**
One single word

tesseract *<img>* *<out>* **--psm 10**
One single character

tesseract *<img>* *<out>* **--psm 4**
One column of variable sizes

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

tesseract *<img>* *<out>* **--psm 12**
Sparse text plus orientation

tesseract *<img>* *<out>* **--psm 0**
Detect orientation and script only

tesseract *<img>* *<out>* **--psm 1**
Automatic with orientation detection

tesseract *<img>* *<out>* **--psm 13**
Raw line, no extra processing

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>* **--oem 0**
Use the legacy engine
Needs traineddata files that include the legacy models.

ENGINE & TUNING (cont.)

tesseract *<img>* *<out>* **--dpi** *<n>*
Set input resolution
Use when the image has no or a wrong dpi in its metadata.

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

-c tessedit_char_whitelist=0123456789
Digits only

-c preserve_interword_spaces=1
Keep column spacing

tesseract *<img>* *<out>* **--user-words** *<file>*
Add a word list

tesseract *<img>* *<out>* **--user-patterns** *<file>*
Add patterns for expected text

tesseract --print-parameters
List all -c parameters

OMP_THREAD_LIMIT=1 tesseract *<img>* *<out>*
Limit to one thread
Faster for many small images run in parallel.

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>* **alto**
ALTO XML

tesseract *<img>* *<out>* **page**
PAGE XML

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

tesseract *<img>* *<out>* **quiet**
Suppress warnings and messages

tesseract *<img>* *<out>* **logfile**
Write debug messages to tesseract.log

OUTPUT FORMATS (cont.)

tsv columns: level, page_num, block_num, par_num, line_num, word_num, left, top, width, height, conf, text. Filter low conf to find shaky words.

PREPARING IMAGES

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

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

magick *<in>* **-density 300** *<out>***.png**
Set 300 dpi metadata

magick *<in>* **-threshold 50%** *<out>***.png**
Binarize to black and white

magick *<in>* **-deskew 40%** *<out>***.png**
Straighten a slightly tilted scan

magick *<in>* **-trim +repage** *<out>***.png**
Cut away empty margins

pdftoppm -r 300 -png *<in>***.pdf** *<prefix>*
Rasterize a PDF to pages
Tesseract reads images, not PDFs. pdftoppm is from poppler-utils.

Dark background or light text? Run -negate first. Tesseract wants dark text on a light background.

RECIPES

Searchable PDF from a scan

```
tesseract scan.png scan -l eng pdf
# writes scan.pdf with an invisible text layer
```

Scanned PDF to text

```
pdftoppm -r 300 -png doc.pdf page
ls page-*.png > pages.txt
tesseract pages.txt doc -l eng txt
```

Clean up, then OCR from a pipe

```
magick in.jpg -colorspace Gray -resize 200% png:- \
  | tesseract stdin stdout --psm 6
```

Confident words only

```
tesseract img.png stdout tsv \
  | awk -F'\t' '$1==5 && $11>80 {print $12}'
```