OCR

scan pdf to file. first extract the pages and ocr them, then make one doc

```
pdfimages -tiff input.pdf plaatje
for i in *.tif; do tesseract $i tempje-$i; done
cat tempje-plaatje-0*.txt >> docje.txt
```

Use tesseract to OCR a multi-page PDF file

First, convert the PDF to multiple TIFF files, because tesseract does not work with PDF.

Number the files with 2 digits at the end, remove the alfa-channel:

```
# convert -density 300 inputfile.pdf -depth 8 -alpha off outputfile_%02d.tiff
```

Then, use tesseract to make it into text:

```
# tesseract inputfile.tiff outputfile
```

if you do not provide and extension for the outputfile, it will become .txt

Creating an overlay with the OCRed text

The newer version of Tesseract (3.03 RC at the time of writing this) can do this:

- free, opensource and cross-platform
- starting from version 3.03 PDF output is available
- CLI software
- multiple languages support
- unfortunately, single image input, so to make a complete document, one must create a batch script to convert each page image to searchable PDF. After that PDF pages should be combined to a single PDF using tools like pdftk.

This is the command:

`tesseract -l  input.tif output pdf`

Note that in order to use this approach, the input PDF has to be rasterized first, since tesseract will not get PDF as input.

To combine multiple PDF files into one

`# pdfunite output_*.pdf result.pdf`

I have created a script

This script will do the work for you. Place the script in a directory together with the PDF to be processed, and run it.

```
#!/bin/bash

# converts a PDF containing scanned pages into a
# new PDF file in which the OCRed text is overlayed,
# making the PDF searchable on text strings.

# use:
#       doit.sh nice.PDF

# requires: tesseract-ocr, convert (ImageMagick), pdfunite, pdfinfo

# 20170417      1.0     PvdM    first version
# 20170418      1.1     PvdM    minor adjustment and improvements, mainly in the counter

bestand="$1"
newbestand=$(echo $bestand | cut -d"." -f1)_searchable.pdf
teller="0"
teller2="000"
aantpaginas=$(pdfinfo "$bestand" | grep 'Pages:' | awk '{ print $2 }')
RESTORE='\033[0m'
RED='\033[00;31m'
GREEN='\033[00;32m'
```

```
YELLOW='\033[00;33m'
BLUE='\033[00;34m'
PURPLE='\033[00;35m'
CYAN='\033[00;36m'
LIGHTGRAY='\033[00;37m'

function check_input {
if [ -z "$bestand" ]; then
        echo - Error. Usage:
        echo "         ./doit.sh input.pdf"; echo
        exit 1
fi
}

function check_error {
        if [ $? != 0 ]; then
                echo == Error! There was a problem in the command.
                exit 1
        fi
}

clear
echo "Converting PDF to searchable (overlay) PDF."
echo -e "-----------------------------------------\n"
check_input
echo -e " $bestand contains $RED $aantpaginas $RESTORE pages.\n"
echo " - (1/3) Extracting scanned PDF to images........"
convert -density 300 "$bestand" -depth 8 -alpha off temp_%03d.tiff
check_error
echo -e " - Done.\n"
echo

echo " - (2/3) Doing OCR on the images.........."
for i in temp_*.tiff; do
        tesseract -l eng $i temp_pdf_$teller2.pdf pdf
        check_error
        ((teller++))
        teller2=$(printf "%05d" $teller)
        echo -e " - (2/3) Doing OCR on the images. $RED Page $teller/$aantpaginas
done.$RESTORE"
```

```
done
echo -e " - Done.\n"
echo

echo " - (3/3) Combining the result into 1 (searchable) PDF"
pdfunite temp_pdf_*.pdf "$newbestand"
check_error
echo -e " - Done. $RED'$newbestand'$RESTORE created.\n"
rm temp*
```

examples

how to extract images from pdf

`pdfimages -all sm_td20a_very_detailed.pdf .`

`pdfimages -f 40 -l 41 -tiff hfe_teac_x-300_300r_service.pdf power-pcb`

Revision #3
Created 2026-04-01 17:13:55 CEST by Philip
Updated 2026-04-13 19:24:56 CEST by Philip