

Preface

بۇ [Tesseract] دەرسلىكى پۈتۈنلەي ئىنگلىزچە نۇسخىسى، ئۇيغۇر تىلى دەرسلىكى پات ئارىدا چىقىدۇ.
ئۇيغۇر تىلى دەرسلىكى ئېلان قىلىنىشىنى كۈتۈپ تۇرالمىسا، ئېلخەت يېزىپ ئىلتىماس قىلىڭ ياكى ياردەم سوراڭ.
بۇلارنىڭ ھەممىسى ھەقسىز ياردەم قىلىدۇ .
ئېلخەت : com_at_dlno@126.com

*This is Tesseract Tutorials is fully English version, and Uyghur Language tutorials coming soon.
What if can't wait Uyghur language tutorials release, please write E-mail: com_at_dlno@126.com
to ask request OR any help. That all help free of charge.*

Tesseract Tutorials

1 OCR Profile

What is OCR? And how the working principle of OCR?

OCR (Optical Character Recognition) refers the OCR goes to find any printable character area, which detect light, dark spot and its shape, size, then based on above properties translate into printable character from an image.

Terminology			
#	EN	UY	CN
1	Feature extraction	ئالاھىدىلىكنى ئېلىش	特征提取
2	Orientation and Segmentation	ئۆرۈن بەلگىلەش ۋە پارچىلاش	定位和分割
3			
4			

2 Tesseract Profile

2.1 Tesseract Quick Intro

Tesseract is Open Source OCR Engine, which is current stable version is Tesseract 5.x. Before Tesseract 4.x (exclude), Tesseract recognition is based character patterns which calls Tesseract legacy OCR engine mode.

But begins from Tesseract 4, Tesseract adds a new Neural Net (LSTM) based OCR engine which is focused on line recognition, but also still supports the legacy Tesseract OCR engine of Tesseract 3. Compatibility with Tesseract 3 is enabled by using the Legacy OCR Engine mode with (--oem 0) options. But have to know that, the traineddata files also need to support the legacy engine, for example those from the tessdata repository.

Tesseract Trained-data Repository table		
#	Repository	Profile
1	tessdata_fast	Fast trained LSTM models; Speed high, but NOT so high accurate percentage. <u>ONLY LSTM Engine mode;</u>
2	tessdata	Trained models with fast variant of the "best" LSTM models + legacy models <u>LSTM Engine mode + legacy Engine mode;</u>
3	tessdata_best	Best (most accurate) trained LSTM models; Speed low, but high accurate percentage. <u>ONLY LSTM Engine mode;</u>

Tesseract support cross platform, Windows, Mac OS, Linux all available。Tesseract support multiple image format such as JPEG、PNG、TIFF etc. And the Tesseract output also supports various output formats: plain text, HTML, PDF, TSV.

Tesseract use multiple image handle operation, feature extraction and machine learning tech. to implement OCR procedure. Tesseract work principle is using trained-data to recognize character first, and refer to the context and language model to make adjustment crude recognition result to empower recognition percentage.

Tesseract OCR Repository profile table		
#	Item	Value
1	Tesseract official website	N/A
2	Tesseract official Repository	https://github.com/tesseract-ocr/tesseract
3	GitHub repository Info	<p>Languages</p> <p> ● C++ 96.5% ● Java 1.0% ● Makefile 0.9% ● CMake 0.8% ● C 0.5% ● Shell 0.3% </p>
4	Supported Compilers	<u>GCC</u> 4.8 and above <u>Clang</u> 3.4 and above <u>MSVC</u> 2015, 2017

The Tesseract 4 (LSTM based) stable version is [4.0.0](#), released on October 29, 2018.

The Tesseract 5 (LSTM based) stable version is [5.4.1](#), released on June 11, 2024.

Tesseract can be used directly via [command line](#), or (for programmers) by using an [API](#) to extract printed text from images.

Tesseract OCR Related Repository Profile Table			
#	Item	Value	Profile
1	Tesseract OCR DOC	https://github.com/tesseract-ocr/tessdoc https://github.com/tesseract-ocr/tessdoc.git https://tesseract-ocr.github.io/tessdoc/	Tesseract OCR Engine complete documentation
2			Data used for LSTM model training

2.2 Tesseract Windows

Tesseract for Windows maintains `libtesseract`, which is OCR engine library file; And `tesseract`, which is command line program. Developers can use `libtesseract` [C](#) or [C++](#) API to build their own application.

```
Volkan@DESKTOP-LCD3FQ6 MINGW64 /c/Program Files/Tesseract-OCR/TESSERACT_4
$ ls -hl *tesseract*
-rwxr-xr-x 1 Volkan 197121 63M Mar 14 2019 libtesseract-4.dll*
-rwxr-xr-x 1 Volkan 197121 127K Sep 7 07:19 tesseract-uninstall.exe*
-rwxr-xr-x 1 Volkan 197121 835K Mar 14 2019 tesseract4.exe*
```

Command line

```
$ tesseract imagename outputbase [-l lang] [--oem ocrenginemode] [--psm pagesegmode] [configfiles...]
```

Argument intro:

imagename: Name of image to be OCR.

This argument value can be "image name", "image list" OR "stdin", means single image, OR image list, even stdin.

outputbase: OCR output result.

This argument value can be "output name" OR "stdout", means single image.

-l lang: Specify OCR recognition natural language.

--oem: QCR Engine Mode

This argument specifies OCR Engine type.

--psm: Page Segmentation Mode

This argument page segmentation type.

configfile: NOT so common use.

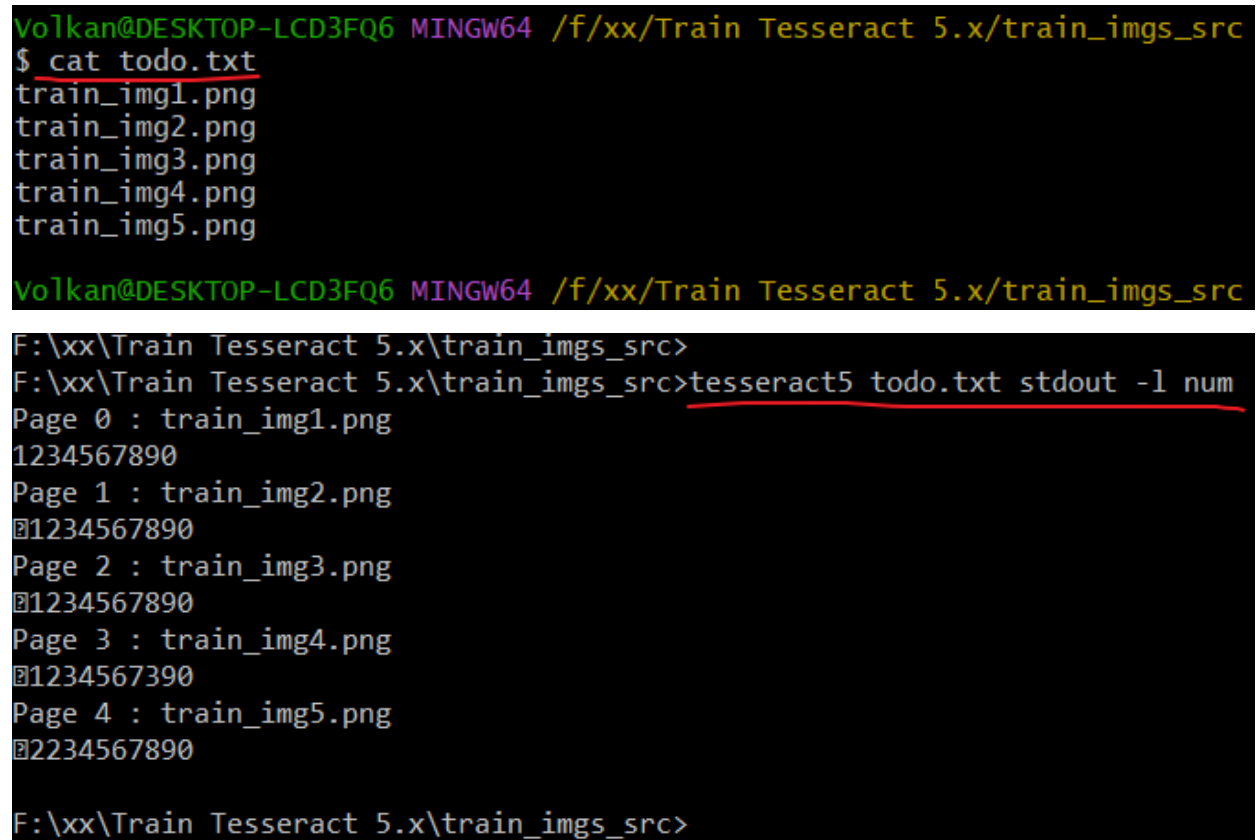
Using One Language

```
$ tesseract images/eurotext.png file_out -l eng // recognize English
```

Using Multiple Languages

```
$ tesseract images/eurotext.png -l eng+uig // recognize English & Uyghur
```

Tesseract command line also support multiple image do OCR at once. Just crate a file, append OCR image target list, and use that image list file as an input.





















```
Vo1kan@DESKTOP-LCD3FQ6 MINGW64 /f/xx/Train Tesseract 5.x/train_imgs_src
$ cat todo.txt
train_img1.png
train_img2.png
train_img3.png
train_img4.png
train_img5.png

Vo1kan@DESKTOP-LCD3FQ6 MINGW64 /f/xx/Train Tesseract 5.x/train_imgs_src
F:\xx\Train Tesseract 5.x\train_imgs_src>
F:\xx\Train Tesseract 5.x\train_imgs_src>tesseract5 todo.txt stdout -l num
Page 0 : train_img1.png
1234567890
Page 1 : train_img2.png
1234567890
Page 2 : train_img3.png
1234567890
Page 3 : train_img4.png
1234567390
Page 4 : train_img5.png
2234567890
F:\xx\Train Tesseract 5.x\train_imgs_src>
```

Have to notice that the "TODO.txt" image list file, every image line by line, and the line separator with "\n". If line operator is "\r\n", the command line occurs an error. In other word, use "Git Bash" tools "vi" command to create that "TODO.txt" file.

Windows Tesseract installation path would have lots of executables, which is helpful for main command line Tesseract works.

 ambiguous_words.exe	12/14/2022 10:25	Application	1,120 KB
 classifier_tester.exe	12/14/2022 10:25	Application	5,538 KB
 cntraining.exe	12/14/2022 10:25	Application	5,203 KB
 combine_lang_model.exe	12/14/2022 10:25	Application	4,028 KB
 combine_tessdata.exe	12/14/2022 10:25	Application	1,308 KB
 dawg2wordlist.exe	12/14/2022 10:26	Application	598 KB
 lstmeval.exe	12/14/2022 10:26	Application	9,582 KB
 lstmtraining.exe	12/14/2022 10:26	Application	10,656 KB
 merge_unicharsets.exe	12/14/2022 10:26	Application	425 KB
 mftraining.exe	12/14/2022 10:26	Application	5,873 KB
 set_unicharset_properties.exe	12/14/2022 10:26	Application	7,386 KB
 shapeclustering.exe	12/14/2022 10:26	Application	5,540 KB
 <u>tesseract5.exe</u> Tesseract exe. file	12/14/2022 10:26	Application	1,369 KB
 tesseract-uninstall.exe	09/07/2024 05:43	Application	147 KB
 <u>text2image.exe</u>	12/14/2022 10:26	Application	11,200 KB
 <u>unicharset_extractor.exe</u>	12/14/2022 10:26	Application	4,063 KB
 winpath.exe	12/14/2022 09:23	Application	19 KB
 wordlist2dawg.exe	12/14/2022 10:27	Application	1,109 KB

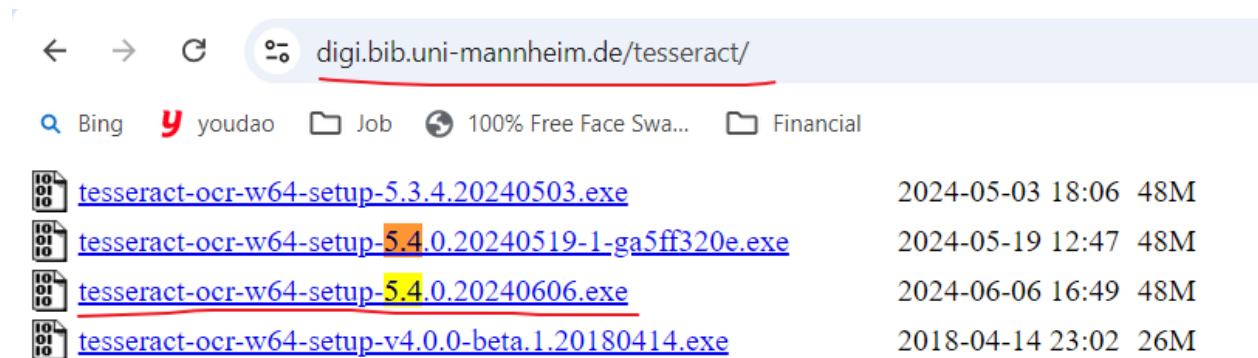
The helpful sub-command line intro would be coming soon.

3 Tesseract Install

3.1 Tesseract Windows

Go to website download latest Tesseract OCR Windows 64.

URI: <https://digi.bib.uni-mannheim.de/tesseract/>



OR download directly tesseract-ocr-w64-setup-5.4.0.20240606.exe:

<https://digi.bib.uni-mannheim.de/tesseract/tesseract-ocr-w64-setup-5.4.0.20240606.exe>

After finish install, remember that append Tesseract bin path, and TESSDATA_PREFIX add into Environment variable. Environment variable TESSDATA_PREFIX refers {TESSERACT_OCR}/tessdata directory, which stores *.traineddata directory.

After configuration, if could output like below, means Tesseract 5 install succeeded.

```
C:\Users\Volkan>echo %TESSDATA_PREFIX%
C:\Program Files\Tesseract-OCR\TESSERACT_5\tessdata

C:\Users\Volkan>tesseract
Usage:
  tesseract5 --help | --help-extra | --version
  tesseract5 --list-langs
  tesseract5 imagename outputbase [options...] [configfile...]

OCR options:
  -l LANG[+LANG]          Specify language(s) used for OCR.
NOTE: These options must occur before any configfile.

Single options:
  --help                  Show this help message.
  --help-extra            Show extra help for advanced users.
  --version               Show version information.
  --list-langs            List available languages for tesseract engine.

C:\Users\Volkan>
```

3.2 Tesseract Fedora

Pre-install: build dependencies

```
$ sudo dnf install gcc gdb gcc-c++ autoconf automake libtool libjpeg-devel libpng-devel libtiff-devel  
zlib-devel
```

And also need to install **Leptonica** with source code (**Run dependency**). About **Leptonica** look at Appendix I.

The screenshot shows the GitHub release page for Leptonica version 1.84.0. The page title is "Leptonica version 1.84.0". It indicates that DanBloomberg released this version on Dec 24, 2023, with 70 commits to master since this release. The release is described as a "configure-ready release, derived from the master on 23 Dec 2023". Under the "Assets" section, there are three items: "leptonica-1.84.0.tar.gz" (13.4 MB, Dec 24, 2023), "Source code (zip)" (Dec 24, 2023), and "Source code (tar.gz)" (Dec 24, 2023). The "leptonica-1.84.0.tar.gz" asset is highlighted with a red underline. At the bottom, it shows "3 people reacted".

Install **Leptonica** commands:

```
$ wget https://github.com/DanBloomberg/leptonica/releases/download/1.84.0/leptonica-  
1.84.0.tar.gz  
$ tar -xf leptonica-1.84.0.tar.gz -C install_ leptonica-1.84.0  
$ cd install_ leptonica-1.84.0  
$ ./autogen.sh  
$ ./configure --prefix=/usr/local/leptonica  
$ make  
$ sudo make install
```



```
[atie@localhost ~/AtieSpace]
$ tar -xf leptonica-1.84.0.tar.gz -C install_leptonica-1.84.0
[atie@localhost ~/AtieSpace]
$ cd install_leptonica-1.84.0/
[atie@localhost ~/AtieSpace/install_leptonica-1.84.0]
$ ll
total 4
drwxr-xr-x. 8 atie atie 4096 Sep 19 18:13 leptonica-1.84.0
[atie@localhost ~/AtieSpace/install_leptonica-1.84.0]
$ cd leptonica-1.84.0/
[atie@localhost ~/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0]
$ ll
total 1144
-rw-r--r--. 1 atie atie 52323 Dec 24 2023 aclocal.m4
-rwxr-xr-x. 1 atie atie 326 Sep 5 2018 autogen.sh
drwxr-xr-x. 2 atie atie 118 Dec 24 2023 autom4te.cache
drwxr-xr-x. 3 atie atie 46 Dec 24 2023 cmake
-rw-r--r--. 1 atie atie 12518 Jun 17 2023 CMakeLists.txt
drwxr-xr-x. 2 atie atie 162 Dec 24 2023 config
-rwxr-xr-x. 1 atie atie 518395 Dec 24 2023 configure ./configure --prefix=/usr/local/leptonica
-rw-r--r--. 1 atie atie 9373 Sep 1 2023 configure.ac
-rw-r--r--. 1 atie atie 106620 Dec 24 2023 Doxyfile
-rw-r--r--. 1 atie atie 132996 Dec 24 2023 doxygen.log
-rw-r--r--. 1 atie atie 1521 Jul 30 2020 leptonica-license.txt
-rw-r--r--. 1 atie atie 414 Mar 2 2023 lept.pc.cmake
-rw-r--r--. 1 atie atie 425 Jun 30 2019 lept.pc.in
-rw-r--r--. 1 atie atie 3473 Sep 5 2018 lok.lua
drwxr-xr-x. 2 atie atie 131 Dec 24 2023 m4
-rw-r--r--. 1 atie atie 2322 Sep 24 2022 Makefile.am
-rw-r--r--. 1 atie atie 32545 Dec 24 2023 Makefile.in
-rwxr-xr-x. 1 atie atie 178 Sep 5 2018 make-for-auto
-rwxr-xr-x. 1 atie atie 165 Sep 5 2018 make-for-local
-rw-r--r--. 1 atie atie 9610 Sep 5 2018 moller52.jpg
drwxr-xr-x. 5 atie atie 20480 Dec 24 2023 prog
-rw-r--r--. 1 atie atie 57775 Dec 24 2023 README.html
-rw-r--r--. 1 atie atie 6319 Feb 6 2023 README.md
drwxr-xr-x. 2 atie atie 8192 Dec 24 2023 src
-rw-r--r--. 1 atie atie 8518 Aug 6 2020 style-guide.txt
-rw-r--r--. 1 atie atie 17604 Aug 29 2023 sw.cpp
-rw-r--r--. 1 atie atie 91780 Dec 24 2023 version-notes.html
[atie@localhost ~/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0]
```

```
[atie@localhost ~/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0]
$ make
(CDPATH="${ZSH_VERSION+}:" && cd . && /bin/sh '/home/atie/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0/bin/sh' -f stamp-h1
touch config/config.h.in
cd . && /bin/sh ./config.status config_auto.h
config.status: creating config_auto.h
config.status: config_auto.h is unchanged
make all-recursive
make[1]: Entering directory '/home/atie/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0'
Making all in src
make[2]: Entering directory '/home/atie/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0/src'
CC      adaptmap.lo
```

```
[atie@localhost ~/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0]
$ sudo make install
[sudo] password for atie:
Making install in src
make[1]: Entering directory '/home/atie/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0/src'
make[2]: Entering directory '/home/atie/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0/src'
/usr/bin/mkdir -p '/usr/local/leptonica/lib'
```

Check the Letonica installation.

```
[atie@localhost ~/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0]
$ ls -l /usr/local/leptonica/
total 4
drwxr-xr-x. 2 root root 4096 Sep 19 18:23 bin
drwxr-xr-x. 3 root root 23 Sep 19 18:23 include
drwxr-xr-x. 4 root root 158 Sep 19 18:23 lib
[atie@localhost ~/AtieSpace/install_leptonica-1.84.0/leptonica-1.84.0]
```

Confirm the `Leptonica` version use by Tesseract with below method.

```
$ wget https://github.com/tesseract-ocr/tesseract/archive/refs/tags/5.4.1.tar.gz
$ tar -xf tesseract-5.4.1.tar.gz -C install_tesseract/
$ ./autogen
$ ./configure --enable-debug --prefix=/usr/local/tesseract
```

```
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ ./configure --enable-debug
checking for g++... g++
checking whether the C++ compiler works... yes
checking for C++ compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o

.....

checking for brew... false
checking for asciidoc... false
checking for xsltproc... true
checking for wchar_t... yes
checking for long long int... yes
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for libcurl... no
checking for lept >= 1.74... no
configure: error: Leptonica 1.74 or higher is required. Try to install libleptonica-dev package.
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$
```

After finish installation, don't forget append Leptonica paths to `/etc/profile` file below variables, which used by `Tesseract OCR`.

```
#####
#####Atie ADD#####
#####
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/leptonica/lib/pkgconfig
export PKG_CONFIG_PATH
CPLUS_INCLUDE_PATH=$CPLUS_INCLUDE_PATH:/usr/local/leptonica/include/leptonica
export CPLUS_INCLUDE_PATH
C_INCLUDE_PATH=$C_INCLUDE_PATH:/usr/local/leptonica/include/leptonica
export C_INCLUDE_PATH
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/leptonica/lib
export LD_LIBRARY_PATH
LIBRARY_PATH=$LIBRARY_PATH:/usr/local/leptonica/lib
export LIBRARY_PATH
LIBLEPT_HEADERSDIR=/usr/local/leptonica/include/leptonica
export LIBLEPT_HEADERSDIR
#####
#####Atie END#####
#####
```

```
#####
#####Atie ADD#####
#####
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/leptonica/lib/pkgconfig
export PKG_CONFIG_PATH
CPLUS_INCLUDE_PATH=$CPLUS_INCLUDE_PATH:/usr/local/leptonica/include/leptonica
export CPLUS_INCLUDE_PATH
C_INCLUDE_PATH=$C_INCLUDE_PATH:/usr/local/leptonica/include/leptonica
export C_INCLUDE_PATH
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/leptonica/lib
export LD_LIBRARY_PATH
LIBRARY_PATH=$LIBRARY_PATH:/usr/local/leptonica/lib
export LIBRARY_PATH
LIBLEPT_HEADERSDIR=/usr/local/leptonica/include/leptonica
export LIBLEPT_HEADERSDIR
#####
#####Atie END#####
#####
```

Tesseract Compile & Install

github.com/tesseract-ocr/tesseract/releases/tag/5.4.1

stweil released this Jun 11 · 34 commits to main since this release · 5.4.1 · b5f279e

What's Changed

This release fixes a regression with legacy or mixed models (issue [#4257](#)).

- Avoid FP overflow in NormEvidenceOf (fixes issue [#4257](#)) by [@stweil](#) in [#4259](#)
- Update deprecated Node.js 16 GitHub actions by [@stweil](#) in [#4262](#)
- Fix code style issues which were reported by Codacy by [@stweil](#) in [#4263](#)
- Fix some issues which were reported by Codacy by [@stweil](#) in [#4266](#)
- Fix more Codacy issues by [@stweil](#) in [#4267](#)
- Several build fixes by [@zdenop](#)

Full Changelog: [5.4.0...5.4.1](#)

Contributors

zdenop and stweil

Assets

Source code (zip)	Jun 11
Source code (tar.gz)	Jun 11

```
$ wget https://github.com/tesseract-ocr/tesseract/archive/refs/tags/5.4.1.tar.gz
$ tar -xf tesseract-5.4.1.tar.gz -C install_tesseract/
$ ./autogen
$ ./configure --enable-debug --prefix=/usr/local/tesseract
$ make
$ sudo make install
$ sudo make ldconfig
```

```
[atie@localhost ~/AtieSpace/install_tesseract]
$ ll
total 2424
drwxr-xr-x. 13 atie atie   4096 Jun 12 02:18 tesseract-5.4.1
-rw-r--r--.  1 atie atie 2474970 Sep 19 17:41 tesseract-5.4.1.zip
[atie@localhost ~/AtieSpace/install_tesseract]
$
```

```
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ ./autogen.sh
Running aclocal
Running /usr/bin/libtoolize
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, 'config'.
libtoolize: copying file 'config/ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'm4'.
libtoolize: copying file 'm4/libtool.m4'
libtoolize: copying file 'm4/ltoptions.m4'
libtoolize: copying file 'm4/ltsugar.m4'
libtoolize: copying file 'm4/ltversion.m4'
libtoolize: copying file 'm4/lt~obsolete.m4'
Running aclocal
Running autoconf
Running autoheader
Running automake --add-missing --copy
configure.ac:379: installing 'config/compile'
configure.ac:27: installing 'config/missing'
Makefile.am: installing 'config/depcomp'
```

All done.

To build the software now, do something like:

```
$ ./configure [--enable-debug] [...other options]
```

```
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
```

```
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ ./configure --enable-debug --prefix=/usr/local/tesseract
checking for g++... g++
checking whether the C++ compiler works... yes
checking for C++ compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C++... yes
checking whether g++ accepts -g... yes
checking for g++ option to enable C++11 features... none needed
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
```

```
config.status: creating include/config_auto.h
config.status: executing depfiles commands
config.status: executing libtool commands
```

Configuration is done.
You can now build and install tesseract by running:

```
$ make
$ sudo make install
$ sudo ldconfig
```

Documentation will not be built because asciidoc or xsltproc is missing.

You cannot build training tools because of missing dependency.
Check configure output for details.

```
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$
```

```
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ make
Making all in .
make[1]: Entering directory '/home/atie/AtieSpace/install_tesseract/tesseract-5.4.1'
CXX      src/tesseract-tesseract.o
CXX      src/api/libtesseract_la-baseapi.lo
CXX      src/api/libtesseract_la-altorenderer.lo
CXX      src/api/libtesseract_la-pagerenderer.lo
CXX      src/api/libtesseract_la-capi.lo
CXX      src/api/libtesseract_la-hocrrenderer.lo
CXX      src/api/libtesseract_la-lstmboxrenderer.lo
CXX      src/api/libtesseract_la-pdfrenderer.lo
CXX      src/api/libtesseract_la-renderer.lo
CXX      src/api/libtesseract_la-wordstrboxrenderer.lo
```

```
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ sudo make install
[sudo] password for atie:
Making install in .
make[1]: Entering directory '/home/atie/AtieSpace/install_tesseract/tesseract-5.4.1'
make[2]: Entering directory '/home/atie/AtieSpace/install_tesseract/tesseract-5.4.1'
/usr/bin/mkdir -p '/usr/local/tesseract/lib'
/bin/sh ./libtool --mode=install /usr/bin/install -c libtesseract.la '/usr/local/tesseract/lib'
libtool: install: /usr/bin/install -c .libs/libtesseract.so.5.0.4 /usr/local/tesseract/lib/libtesseract.so.5.0.4
libtool: install: (cd /usr/local/tesseract/lib && { ln -s -f libtesseract.so.5.0.4 libtesseract.so.5; } )
libtool: install: /usr/bin/install -c .libs/libtesseract.lai /usr/local/tesseract/lib/libtesseract.lai
libtool: install: /usr/bin/install -c .libs/libtesseract.a /usr/local/tesseract/lib/libtesseract.a
libtool: install: chmod 644 /usr/local/tesseract/lib/libtesseract.a
libtool: install: ranlib /usr/local/tesseract/lib/libtesseract.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/var/lib/napd/s"
```

```
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ sudo ldconfig
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$
```

Check the Tesseract installation.

```

[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ ls -l /usr/local/tesseract/
total 0
drwxr-xr-x. 2 root root 23 Sep 19 19:38 bin
drwxr-xr-x. 3 root root 23 Sep 19 19:38 include
drwxr-xr-x. 3 root root 145 Sep 19 19:38 lib
drwxr-xr-x. 3 root root 22 Sep 19 19:38 share
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ tree /usr/local/tesseract/
/usr/local/tesseract/
├── bin
│   └── tesseract
├── include
│   └── tesseract
│       ├── baseapi.h
│       ├── capi.h
│       ├── export.h
│       ├── ltrresultiterator.h
│       ├── ocrclass.h
│       ├── osdetect.h
│       ├── pageiterator.h
│       ├── publictypes.h
│       ├── renderer.h
│       ├── resultiterator.h
│       ├── unichar.h
│       └── version.h
├── lib
│   ├── libtesseract.a
│   ├── libtesseract.la
│   ├── libtesseract.so -> libtesseract.so.5.0.4
│   ├── libtesseract.so.5 -> libtesseract.so.5.0.4
│   ├── libtesseract.so.5.0.4
│   ├── pkgconfig
│   └── tesseract.pc
└── share
    └── tessdata
        ├── configs
        │   ├── alto
        │   ├── ambigs.train
        │   ├── api_config
        │   └── bigram

```

Add Tesseract binary into PATH direct.


```

83 #####
84 #####Atie ADD#####
85 #####
86
87 # -----
88 # For Leptonica
89 PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/leptonica/lib/pkgconfig
90 export PKG_CONFIG_PATH
91 CPLUS_INCLUDE_PATH=$CPLUS_INCLUDE_PATH:/usr/local/leptonica/include/leptonica
92 export CPLUS_INCLUDE_PATH
93 C_INCLUDE_PATH=$C_INCLUDE_PATH:/usr/local/leptonica/include/leptonica
94 export C_INCLUDE_PATH
95 LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/leptonica/lib
96 export LD_LIBRARY_PATH
97 LIBRARY_PATH=$LIBRARY_PATH:/usr/local/leptonica/lib
98 export LIBRARY_PATH
99 LIBLEPT_HEADERSDIR=/usr/local/leptonica/include/leptonica
100 export LIBLEPT_HEADERSDIR
101
102 # -----
103 # For Tesseract
104 export PATH=$PATH:/usr/local/tesseract/bin
105
106 #####
107 #####Atie END#####
108 #####
109

```

Create soft link for Tesseract command for easy use.

```

[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ sudo ln -sf /usr/local/tesseract/bin/tesseract /usr/local/bin/tesseract
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ ll /usr/local/bin/tesseract
lrwxrwxrwx. 1 root root 34 Sep 19 19:45 /usr/local/bin/tesseract -> /usr/local/tesseract/bin/tesseract
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$ tesseract -v
tesseract 5.4.1
  leptonica-1.84.0
    libjpeg 6b (libjpeg-turbo 2.1.4) : libpng 1.6.37 : libtiff 4.4.0 : zlib 1.2.13
  Found AVX2
  Found AVX
  Found FMA
  Found SSE4.1
  Found OpenMP 201511
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$
[atie@localhost ~/AtieSpace/install_tesseract/tesseract-5.4.1]
$

```


4 Tesseract Training

4.1 Training for Tesseract5 Windows

Training outline

1. Install "jTessBoxEditor" tools.

This tool is helpful another image format converts into TIF file (*.tif), OR TIF image packet format (*.tif), and edit Box format file to correct and adjust the Box file Orientation and Segmentation result.

2. Get target image and merge into a TIF image packets.
3. Generate Box file (*.box).
4. Open TIF image packet and adjust the Box file character Orientation and Segmentation result.
5. Create image character FONT metadata file.
6. Run bat file generate trained data (*.traineddata).

If Tesseract before on Tesseract-4.x (exclude), until here all the training procedure finish. So, as long as the (*.traineddata) generated, go to use it.


7. LSTM training.

From Tesseract-4.x (include) also need another training: LSTM training procedure.

8. Run bat file generate trained data (*.traineddata).

Train data Preparation

- 1) Download a `uig.traineddata` from <https://github.com/tesseract-ocr/tessdata> for use LSTM training.

 `uig.traineddata`

- 2) Download a `jTessBoxEditor` from <https://sourceforge.net/projects/vietocr/files/jTessBoxEditor/>.
For edit, correct & check character recognition.

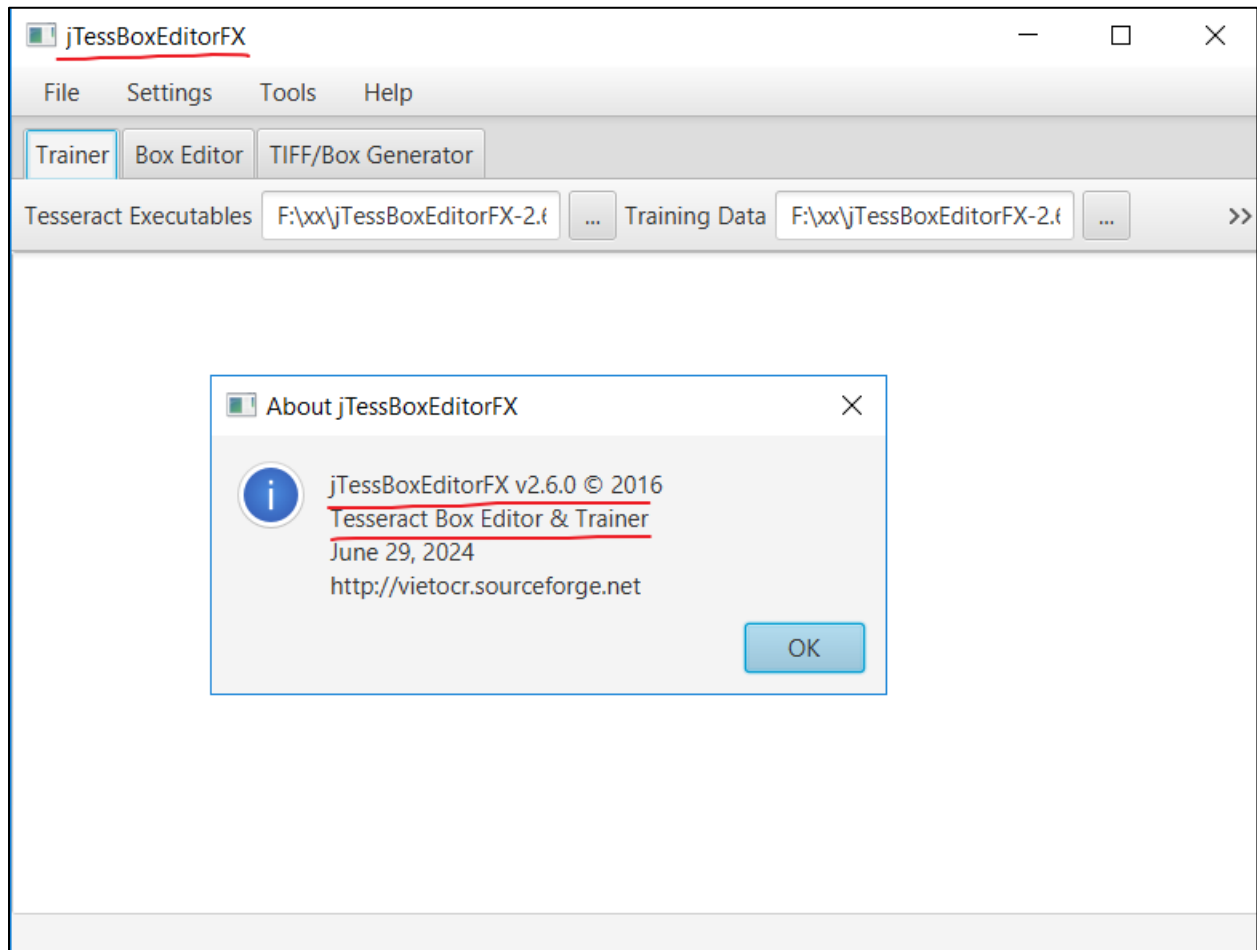
`jTessBoxEditorFX-2.6.0.zip`

This tool developed by Java, so need JRE 8+.

Do train steps

1. Image convert into TIFF format.

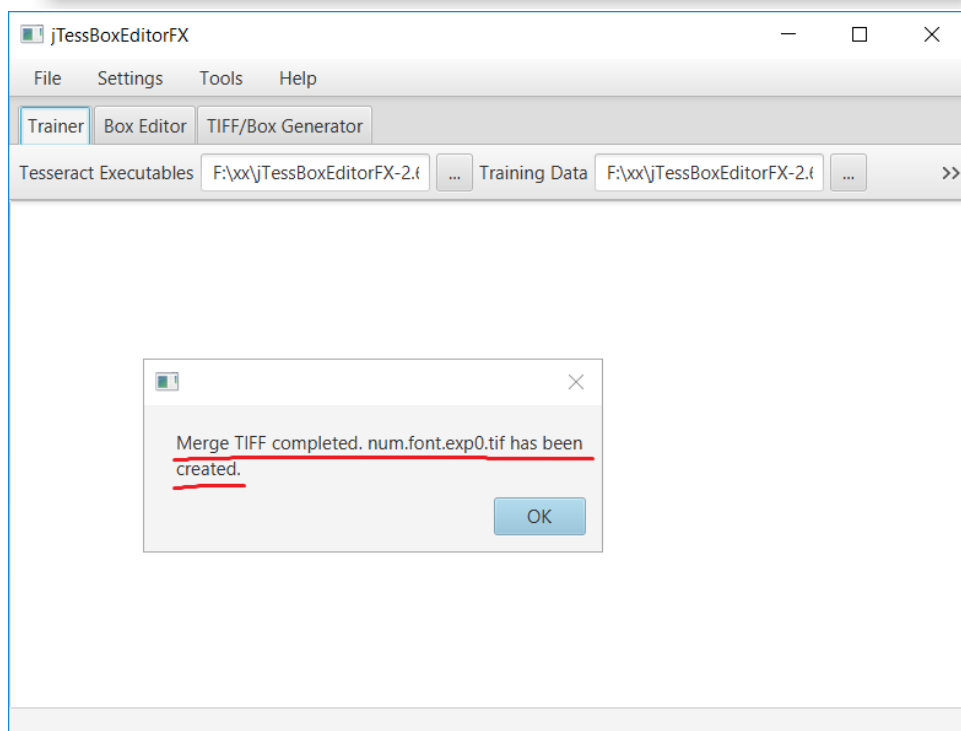
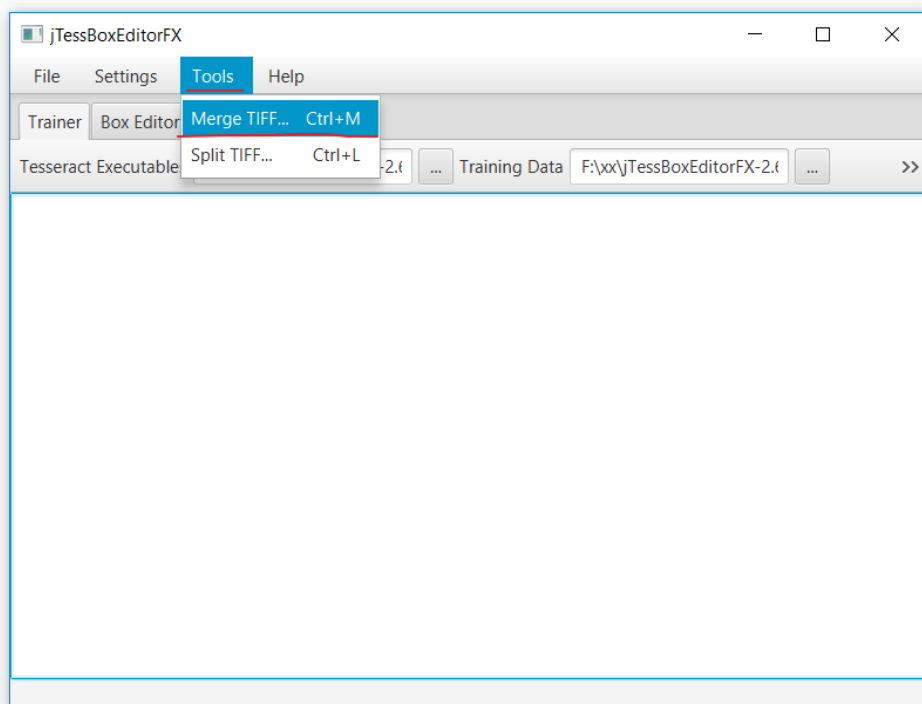
If use jTessBoxEditor DON'T need convert into TIF format, while packet into TIF image format the tool automatic convert it to TIF format.



2. Material image packet into a TIF image.

Use jTessBoxEditor tools to combine all material images into a compound TIFF packet file.

1234567890 1234567890 1234567890 1234567890 1234567890
train_img1.png train_img2.png train_img3.png train_img4.png train_img5.png



1234567890
num.font.exp0.tif

How to pick a name for generated TIF image packet, this all MUST obey Tesseract rules. The rule says that:

TIF image packet name: `[language].[font name].exp[num].tif`. A demo name " num.font.exp0.tif ".

#	Item	Profile
1	language	The TIF image contained language name like "uig";
2	font name	Font name like UKIJ, Alkatip etc.;
3	num	Experiment/Exercise/Train Seq. number

3.Generate BOX file

In the Windows command line, change to directory to current working directory and type below command to generate *.box file.

```
$ tesseract num.font.exp0.tif num.font.exp0 batch.no chop makebox // generate num.font.exp0.box file
```

Make Box File command:

```
$ tesseract [tif packet image] [box file name] batch.no chop makebox // generate "box_name.box" file
```

Note that the `*.tif` and `*.box` file picked name rule is the same. The rule shown above. And the both should the same directory.

TIF image packet file vs. BOX file

1.TIF image packet file stores lots of images page by page.

2.BOX file stores the characters location information (every character's X, Y quadrant and height, width) in TIF image packet file.

```
F:\xx\Train Tesseract 5.x>tesseract5 num.font.exp0.tif num.font.exp0 batch.no chop makebox
Page 1
Page 2
Page 3
Page 4
Page 5
F:\xx\Train Tesseract 5.x>
```

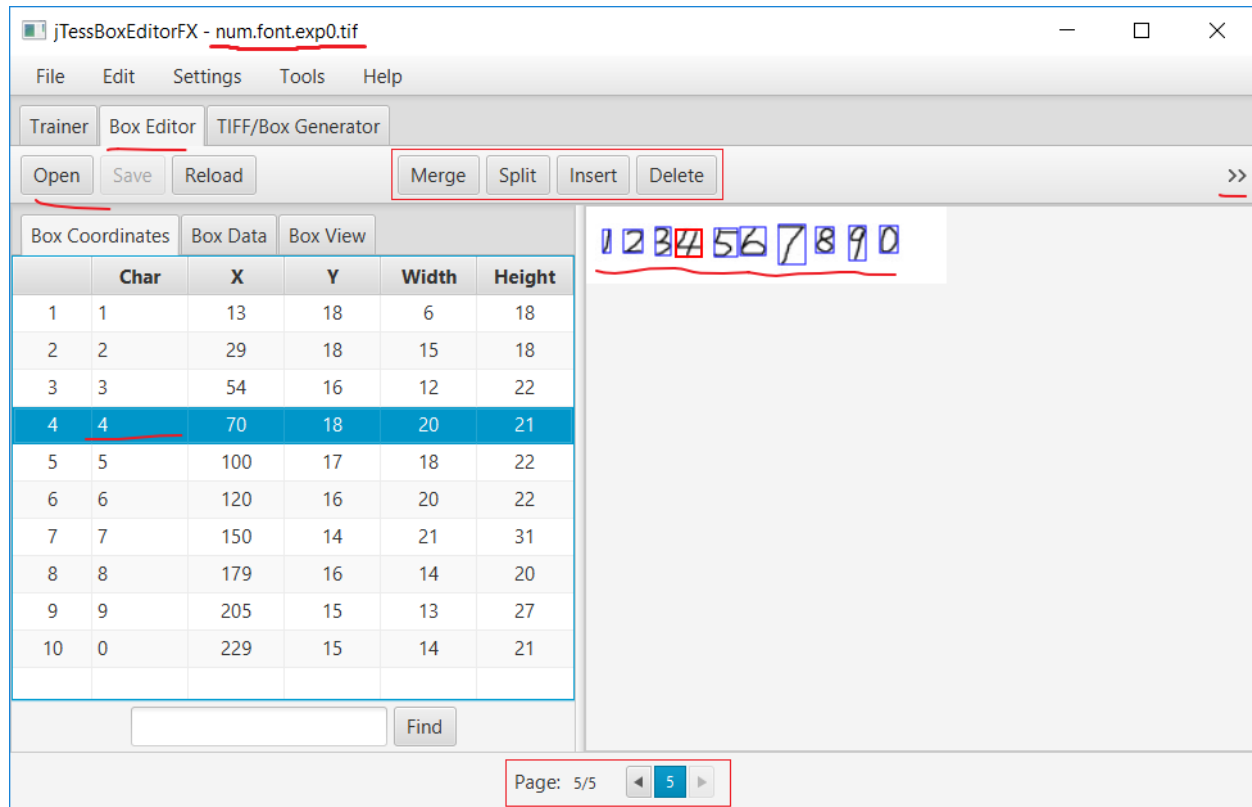
After run above command would generate a *.box file based on commpond *.tif.



4.Check & Correct chars

This is what we really do work on the whole training process.

Open `jTessBoxEditor`, click Box Editor -> Open, and open `num.font.exp0.tif` generated above procedure. And will see some of characters recognition is wrong, so need to check and correct them one by one for all images.



That window could correct character, even change the character X, Y, width, height metadata. And if don't need, also can delete OR add even a character node etc. No forget to save of course.

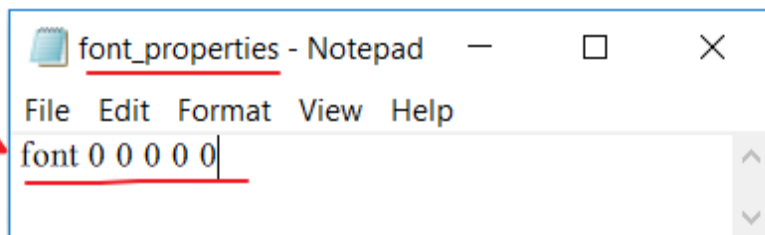
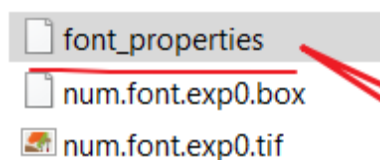
Until here, the `jTessBoxEditor` tool mission is finished.

5.Create FONT metadata file: `font_properties`

In the work directory, create a file named `"font_properties"`, and write `"font 0 0 0 0 0"`.

[Syntax] : `<fontname> <italic> <bold> <fixed> <serif> <fraktur>`

Argument			
#	Item	Value	Profile
1	fontname	font name	0: Not existed 1: existed
2	italic	specify font italic OR NOT	
3	bold	specify font bold OR NOT	
4	fixed	Default font name	
5	serif	Serif OR NOT	
6	fraktur	German bold	



6.Generate *. traineddata

Run below bat file to generate *. traineddata.

```

1 echo Run Tesseract for Training ...
2 tesseract5.exe num.font.exp0.tif num.font.exp0 nobatch box.train
3
4 echo Compute the Character Set..
5 unicharset_extractor.exe num.font.exp0.box
6 mfttraining -F font_properties -U unicharset -O num.unicharset num.font.exp0.tr
7
8 echo Clustering..
9 cntraining.exe num.font.exp0.tr
10
11 echo Rename Files ...
12 rename normproto num.normproto
13 rename inttemp num.inttemp
14 rename pffmtable num.pffmtable
15 rename shapetable num.shapetable
16
17 echo Create Tessdata..
18 combine_tessdata.exe num.
19
20 set /p input= Type any key to end ...
21

```

```

echo Run Tesseract for Training ...
tesseract5.exe num.font.exp0.tif num.font.exp0 nobatch box.train

echo Compute the Character Set..

```



unicharset_extractor.exe num.font.exp0.box mfttraining -F font_properties -U unicharset -O num.unicharset num.font.exp0.tr echo Clustering.. cntraining.exe num.font.exp0.tr echo Rename Files ... rename normproto num.normproto rename inttemp num.inttemp rename pffmtable num.pffmtable rename shapetable num.shapetable echo Create Tessdata.. combine_tessdata.exe num. set /p input= Type any key to end ...	
--	--

If lower than Tesseract 4.x (exclude), until here the train finished. But begins with Tesseract 4.x (include), also need more train with LSTM.

6. LSTM training


Use *.tif and *.box file generate *.lstmf file for training LSTM.

\$ tesseract num.font.exp0.tif num.font.exp0 -l uig--psm 6 lstm.train // generate num.font.exp0.lstmf file

Argument			
#	Item	Value	Profile
1	num.font.exp0.tif	TIF packet image name	
2	num.font.exp0	Output *.lstmf file name	
3	-l lang	LSTF train language	
4	--psm NO	Recognition mode	The mode 6 is better

After the command finish, would generate a file named "num.font.exp0.lstmf".

```
F:\xx\Train Tesseract 5.x - Copy>tesseract5 num.font.exp0.tif num.font.exp0 --psm 6 lstm.train
Page 1
Page 2
Page 3
Loaded 2/2 lines (1-2) of document num.font.exp0.lstmf
Page 4
Loaded 3/3 lines (1-3) of document num.font.exp0.lstmf
Page 5
Loaded 4/4 lines (1-4) of document num.font.exp0.lstmf
F:\xx\Train Tesseract 5.x - Copy>
```

Generated file:  num.font.exp0.lstmf

7. Extract LSTM data from already LSTM traineddata file

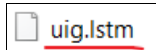
The file *.traineddata MUST get from https://github.com/tesseract-ocr/tessdata_best repository.

```
$ combine_tessdata -e uig.traineddata uig.lstm // generate uig.lstm file
```

```
F:\xx\Train Tesseract 5.x - Copy>combine_tessdata -e uig.traineddata uig.lstm
Extracting tessdata components from uig.traineddata
Wrote uig.lstm
Version:4.00.00alpha:uig:synth20170629
17:lstm:size=11779387, offset=192
18:lstm-punc-dawg:size=4506, offset=11779579
19:lstm-word-dawg:size=1249010, offset=11784085 generate uig.lstm
20:lstm-number-dawg:size=32242, offset=13033095
21:lstm-unicharset:size=8037, offset=13065337
22:lstm-recoder:size=1201, offset=13073374
23:version:size=30, offset=13074575

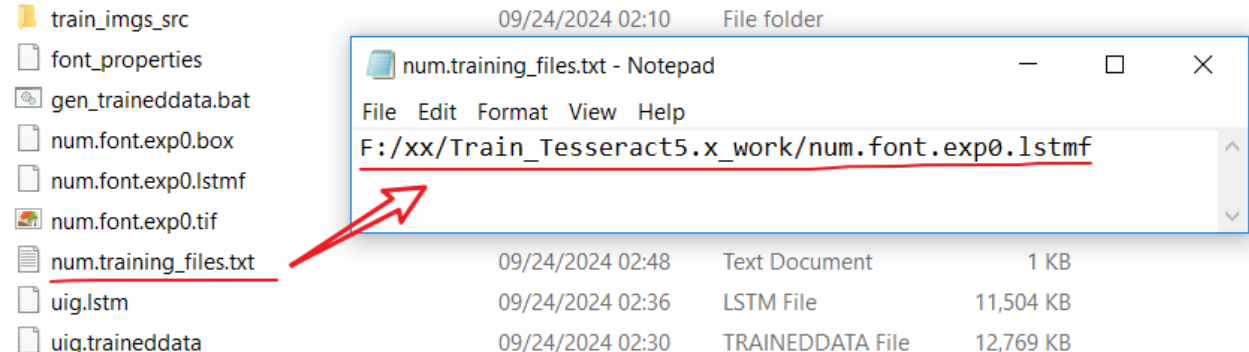
F:\xx\Train Tesseract 5.x - Copy>
```

Generated file:



8.Touch num.training_files.txt and specify *.lstmf file

```
$ echo "F:/xx/Train Tesseract 5.x/num.font.exp0.lstmf" > num.training_files.txt // generate
num.training_files.txt
```



9.LSTM train

In working directory, create an output file, and working directory execute below command.

```
$ lstmtraining \
--model_output="F:/Train_Tesseract5.x_Work/output" \
--continue_from="F:/Train_Tesseract5.x_Work/uig.lstm" \
--train_listfile="F:/Train_Tesseract5.x_Work/num.training_files.txt"
--traineddata="F:/Train_Tesseract5.x_Work/uig.traineddata" \
--debug_interval -1 \
--max_iterations 800
```


Argument			
#	Item	Value	Profile
1	--modeloutput DIR	Directory	LSTM trained data file output directory; in other word, "output_*.checkpoint " file directory.
2	--continue_from DIR	LSTM File name	LSTM module data extract from LSTM *.traineddata
3	--train_listfile DIR	File name	
4	--traineddata DIR	LSTM *. Traineddata file name	LSTM *. Traineddata file name
5	--debug_interval NO		-1 means verbose
6	--max_iterations NO	Training times with integer	

This command would generate multiple output_*.checkpoint and output_checkpoint files.

```
F:\xx\Train_Tesseract5.x_work>
F:\xx\Train_Tesseract5.x_work>lstmtraining --model output="F:/xx/Train_Tesseract5.x_work/output" --continue_from="F:/xx/
Train_Tesseract5.x_work/uig.lstm" --train_listfile="F:/xx/Train_Tesseract5.x_work/num.training_files.txt" --traineddata=
"F:/xx/Train_Tesseract5.x_work/uig.traineddata" --debug_interval -1 --max_iterations 800
Loaded file F:/xx/Train_Tesseract5.x_work/uig.lstm, unpacking...
Warning: LSTMTrainer deserialized an LSTMRecognizer!
Continuing from F:/xx/Train_Tesseract5.x_work/uig.lstm
Loaded 5/5 lines (1-5) of document F:/xx/Train_Tesseract5.x_work/num.font.exp0.lstmf
Iteration 0: GROUND TRUTH : |234567390
Iteration 0: ALIGNED TRUTH : |23456739390
Iteration 0: BEST OCR TEXT : | 234567 05
File num.font.exp0.lstmf line 0 :
Mean rms=4.312%, delta=31.579%, train=90%(100%), skip ratio=0%
Iteration 1: GROUND TRUTH : |234.567890
Iteration 1: ALIGNED TRUTH : |234.5678990
Iteration 1: BEST OCR TEXT : |2545660
File num.font.exp0.lstmf line 0 :
Mean rms=3.861%, delta=27.039%, train=85.909%(100%), skip ratio=0%
Iteration 2: GROUND TRUTH : |234567890
Iteration 2: ALIGNED TRUTH : |234567890
Iteration 2: BEST OCR TEXT : | 234567 0
File num.font.exp0.lstmf line 2 :
Mean rms=3.973%, delta=28.443%, train=90.606%(100%), skip ratio=0%
Iteration 3: GROUND TRUTH : |234567890
Iteration 3: BEST OCR TEXT : |235670
File num.font.exp0.lstmf line 0 :
Mean rms=3.745%, delta=25.355%, train=80.455%(100%), skip ratio=0%
Iteration 4: GROUND TRUTH : |234567890
Iteration 4: ALIGNED TRUTH : |2345678990
Iteration 4: BEST OCR TEXT : |234 56 7 0
```










```

Iteration 793: GROUND TRUTH : [234567890
File num.font.exp0.lstmf line 0 (Perfect):
Mean rms=0.413%, delta=1.01%, train=3.994%(15.491%), skip ratio=0%
Iteration 794: GROUND TRUTH : 1234567890
File F:/xx/Train_Tesseract5.x_work/num.font.exp0.lstmf line 4 (Perfect):
Mean rms=0.413%, delta=1.008%, train=3.989%(15.472%), skip ratio=0%
Iteration 795: GROUND TRUTH : |234567390
File num.font.exp0.lstmf line 0 (Perfect):
Mean rms=0.412%, delta=1.007%, train=3.984%(15.452%), skip ratio=0%
Iteration 796: GROUND TRUTH : |234.567890
File num.font.exp0.lstmf line 0 (Perfect):
Mean rms=0.412%, delta=1.006%, train=3.979%(15.433%), skip ratio=0%
Iteration 797: GROUND TRUTH : 1234567890
File num.font.exp0.lstmf line 2 (Perfect):
Mean rms=0.411%, delta=1.005%, train=3.974%(15.414%), skip ratio=0%
Iteration 798: GROUND TRUTH : [234567890
File num.font.exp0.lstmf line 0 (Perfect):
Mean rms=0.411%, delta=1.003%, train=3.969%(15.394%), skip ratio=0%
Iteration 799: GROUND TRUTH : 1234567890
File F:/xx/Train_Tesseract5.x_work/num.font.exp0.lstmf line 4 (Perfect):
Mean rms=0.41%, delta=1.002%, train=3.964%(15.375%), skip ratio=0%
2 Percent improvement time=0, best error was 6.342 @ 113
At iteration 113/800/800, Mean rms=0.410000%, delta=1.002000%, BCER train=3.964000%, BWER train=15.375000%, skip ratio=0.000000%, New best BCER = 3.964000 wrote best model:F:/xx/Train_Tesseract5.x_work/output_3.964000_113_800.checkpoint wr
ote checkpoint.

```

Finished! Selected model with minimal training error rate (BCER) = 3.964

F:\xx\Train_Tesseract5.x_work>

 output_3.663000_74_800.checkpoint	09/26/2024 20:16	CHECKPOINT File	34,548 KB
 output_4.186000_74_700.checkpoint	09/26/2024 20:16	CHECKPOINT File	34,548 KB
 output_4.883000_74_600.checkpoint	09/26/2024 20:16	CHECKPOINT File	34,548 KB
 output_5.860000_74_500.checkpoint	09/26/2024 20:15	CHECKPOINT File	34,548 KB
 output_7.325000_74_400.checkpoint	09/26/2024 20:15	CHECKPOINT File	34,548 KB
 output_9.733000_73_300.checkpoint	09/26/2024 20:15	CHECKPOINT File	34,548 KB
 output_14.600000_73_200.checkpoint	09/26/2024 20:14	CHECKPOINT File	34,548 KB
 output_28.500000_69_100.checkpoint	09/26/2024 20:14	CHECKPOINT File	34,548 KB
 <u>output_checkpoint</u>	09/26/2024 20:16	File	69,096 KB

10.Generate LSTM mode traineddata

Use last command generated "output_checkpoint" file and old LSTM "uig. traineddata", combine together and generate new LSTM " num.traineddata" file.

```

$ lstmtraining \
  --stop_training \
  --continue_from="F:/Train_Tesseract5.x_Work/output_checkpoint" \
  --traineddata="F:/Train_Tesseract5.x_Work/uig.traineddata" \
  --model_output="F:/Train_Tesseract5.x_Work/num.traineddata"


```

Argument			
#	Item	Value	Profile
1	--stop_training	NA	Default needs
2	--continue_from FILE	Output_checkout file name with direcotry	
3	--traineddata FILE	Mother *.traineddata File name	
4	--model_output DIR	Output *.traineddata File name	Target LSTM *. Traineddata file name

The command would generate LSTM *.traineddata file.

```
F:\xx\Train_Tesseract5.x_work>
F:\xx\Train_Tesseract5.x_work>lstmtraining --stop training --continue from="F:/xx/Train Tesseract5.x work/output checkpo
int" --traineddata="F:/xx/Train_Tesseract5.x work/uig.traineddata" --model output="F:/xx/Train_Tesseract5.x_work/num.tr
aineddata"
Loaded file F:/xx/Train_Tesseract5.x_work/output_checkpoint, unpacking...      Generate num.traineddata
F:\xx\Train_Tesseract5.x_work>
```


Generate

 num.traineddata

The generated LSTM *.traineddata copy to {TESSERACT_INSTALL_PATH}/tessdata. Until here all mission is done.

Appendix I: Leptonica

Leptonica is an open source library containing software that is broadly useful for image processing and image analysis applications.

#	Item	Value
1	Leptonica official website	http://www.leptonica.org/
2	Leptonica official Repository	https://github.com/DanBloomberg/leptonica
3	GitHub repository Info	<div><p>Languages</p><p>● C 97.9% ● HTML 0.8% ● C++ 0.6% ● CMake 0.3% ● Shell 0.2% ● Makefile 0.1% ● Other 0.1%</p></div>

Open Source Projects that use Leptonica:

- [tesseract](#) (optical character recognition)
- [OpenCV](#) (computer vision library)
- [jbig2enc](#) (encodes multipage binary image documents with jbig2 compression)