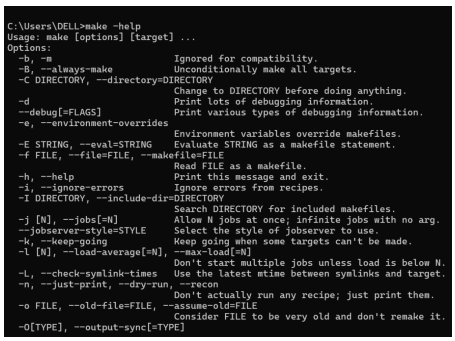
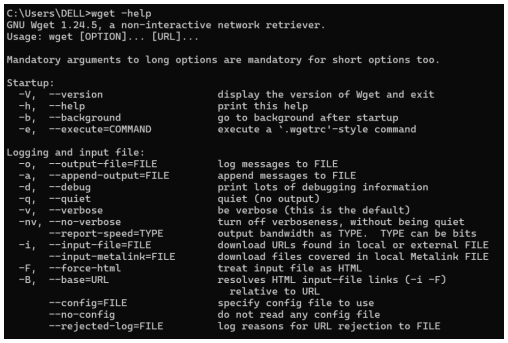
Step 1: set up environment

- Install make, wget, bash, this is the most important step in this tutorial
- You must set up bash to follow it



```
C:\Users\DELL>make -help
Usage: make [options] [target] ...
Options:
  -b, -m                      Ignored for compatibility.
  -B, --always-make           Unconditionally make all targets.
  -C DIRECTORY, --directory=DIRECTORY
                              Change to DIRECTORY before doing anything.
  -d                          Print lots of debugging information.
  --debug[=FLAGS]             Print various types of debugging information.
  -e, --environment-overrides
                              Environment variables override makefiles.
  -E STRING, --eval=STRING    Evaluate STRING as a makefile statement.
  -f FILE, --file=FILE, --makefile=FILE
                              Read FILE as a makefile.
  -h, --help                  Print this message and exit.
  -i, --ignore-errors         Ignore errors from recipes.
  -I DIRECTORY, --include-dir=DIRECTORY
                              Search DIRECTORY for included makefiles.
  -j [N], --jobs[=N]          Allow N jobs at once; infinite jobs with no arg.
  --jobserver-style=STYLE     Select the style of jobserver to use.
  -k, --keep-going            Keep going when some targets can't be made.
  -l [N], --load-average[=N], --max-load[=N]
                              Don't start multiple jobs unless load is below N.
  -L, --check-symlink-times   Use the latest mtime between symlinks and target.
  -n, --just-print, --dry-run, --recon
                              Don't actually run any recipe; just print them.
  -o FILE, --old-file=FILE, --assume-old=FILE
                              Consider FILE to be very old and don't remake it.
  -O[TYPE], --output-sync[=TYPE]
```



```
C:\Users\DELL>wget -help
GNU Wget 1.24.5, a non-interactive network retriever.
Usage: wget [OPTION]... [URL]...

Mandatory arguments to long options are mandatory for short options too.

Startup:
  -V,  --version                   display the version of Wget and exit
  -h,  --help                      print this help
  -b,  --background                go to background after startup
  -e,  --execute=COMMAND           execute a `.wgetrc'-style command

Logging and input file:
  -o,  --output-file=FILE          log messages to FILE
  -a,  --append-output=FILE        append messages to FILE
  -d,  --debug                     print lots of debugging information
  -q,  --quiet                     quiet (no output)
  -v,  --verbose                   be verbose (this is the default)
  -nv, --no-verbose                turn off verboseness, without being quiet
       --report-speed=TYPE         output bandwidth as TYPE.  TYPE can be bits
  -i,  --input-file=FILE           download URLs found in local or external FILE
       --input-metalink=FILE       download files covered in local Metalink FILE
  -F,  --force-html                treat input file as HTML
  -B,  --base=URL                  resolves HTML input-file links (-i -F)
                                     relative to URL
       --config=FILE               specify config file to use
       --no-config                 do not read any config file
       --rejected-log=FILE         log reasons for URL rejection to FILE
```

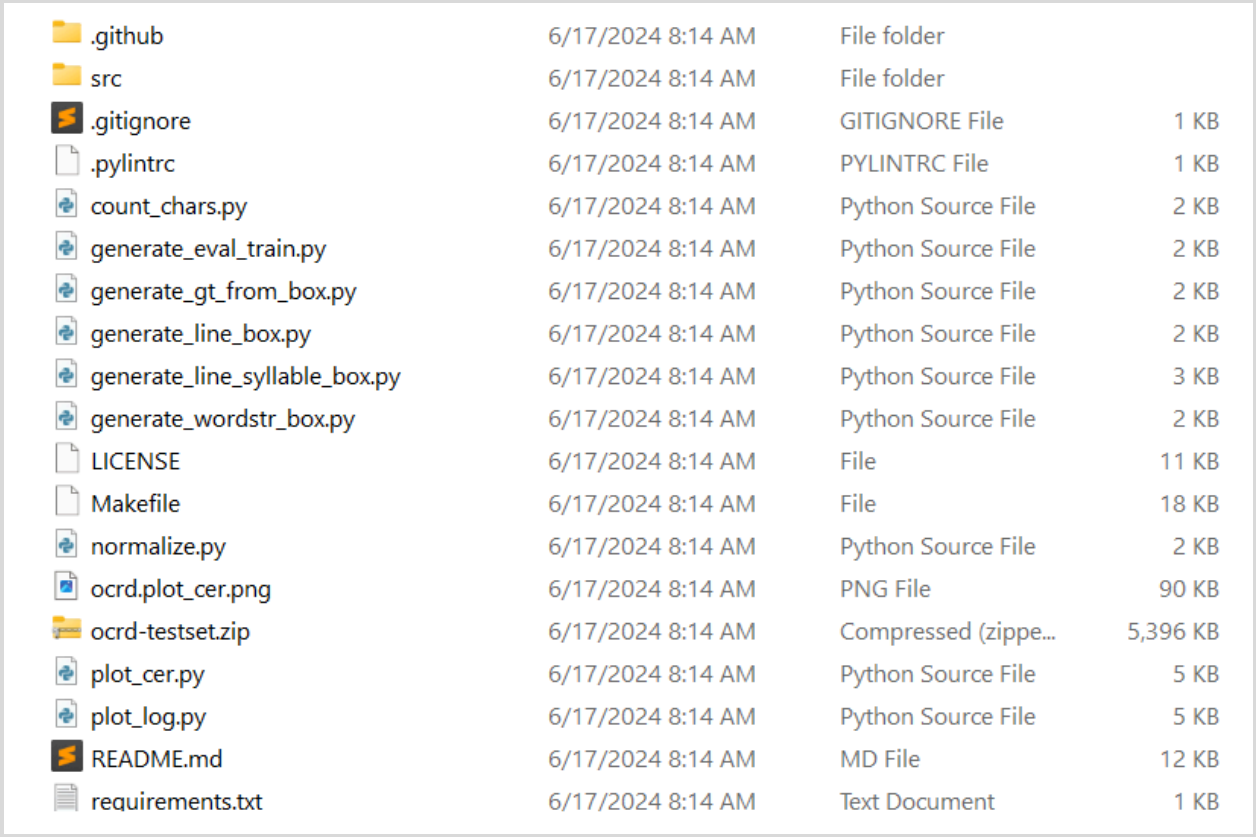
```
C:\Users\DELL>bash --help
GNU bash, version 5.1.16(1)-release-(x86_64-pc-linux-gnu)
Usage:  /bin/bash [GNU long option] [option] ...
        /bin/bash [GNU long option] [option] script-file ...
GNU long options:
        --debug
        --debugger
        --dump-po-strings
        --dump-strings
        --help
        --init-file
        --login
        --noediting
        --noprofile
        --norc
        --posix
        --pretty-print
        --rcfile
        --restricted
        --verbose
        --version
Shell options:
        -ilrsD or -c command or -O shopt_option         (invocation only)
        -abefhkmnptuvxBCHP or -o option
Type `/bin/bash -c "help set"' for more information about shell options.
Type `/bin/bash -c help' for more information about shell builtin commands.
Use the `bashbug' command to report bugs.

bash home page: <http://www.gnu.org/software/bash>
```

Step 2: create tesstrain(anywhere)

`git clone https://github.com/tesseract-ocr/tesstrain.git`

```
C:\TestCode>git clone https://github.com/tesseract-ocr/tesstrain.git
Cloning into 'tesstrain'...
remote: Enumerating objects: 1110, done.
remote: Counting objects: 100% (467/467), done.
remote: Compressing objects: 100% (59/59), done.
remote: Total 1110 (delta 434), reused 410 (delta 408), pack-reused 643
Receiving objects: 100% (1110/1110), 13.52 MiB | 2.81 MiB/s, done.
Resolving deltas: 100% (653/653), done.
```



.github	6/17/2024 8:14 AM	File folder	
src	6/17/2024 8:14 AM	File folder	
.gitignore	6/17/2024 8:14 AM	GITIGNORE File	1 KB
.pylintrc	6/17/2024 8:14 AM	PYLINTRC File	1 KB
count_chars.py	6/17/2024 8:14 AM	Python Source File	2 KB
generate_eval_train.py	6/17/2024 8:14 AM	Python Source File	2 KB
generate_gt_from_box.py	6/17/2024 8:14 AM	Python Source File	2 KB
generate_line_box.py	6/17/2024 8:14 AM	Python Source File	2 KB
generate_line_syllable_box.py	6/17/2024 8:14 AM	Python Source File	3 KB
generate_wordstr_box.py	6/17/2024 8:14 AM	Python Source File	2 KB
LICENSE	6/17/2024 8:14 AM	File	11 KB
Makefile	6/17/2024 8:14 AM	File	18 KB
normalize.py	6/17/2024 8:14 AM	Python Source File	2 KB
ocrd.plot_cer.png	6/17/2024 8:14 AM	PNG File	90 KB
ocrd-testset.zip	6/17/2024 8:14 AM	Compressed (zippe...	5,396 KB
plot_cer.py	6/17/2024 8:14 AM	Python Source File	5 KB
plot_log.py	6/17/2024 8:14 AM	Python Source File	5 KB
README.md	6/17/2024 8:14 AM	MD File	12 KB
requirements.txt	6/17/2024 8:14 AM	Text Document	1 KB

Step 3: make langdata

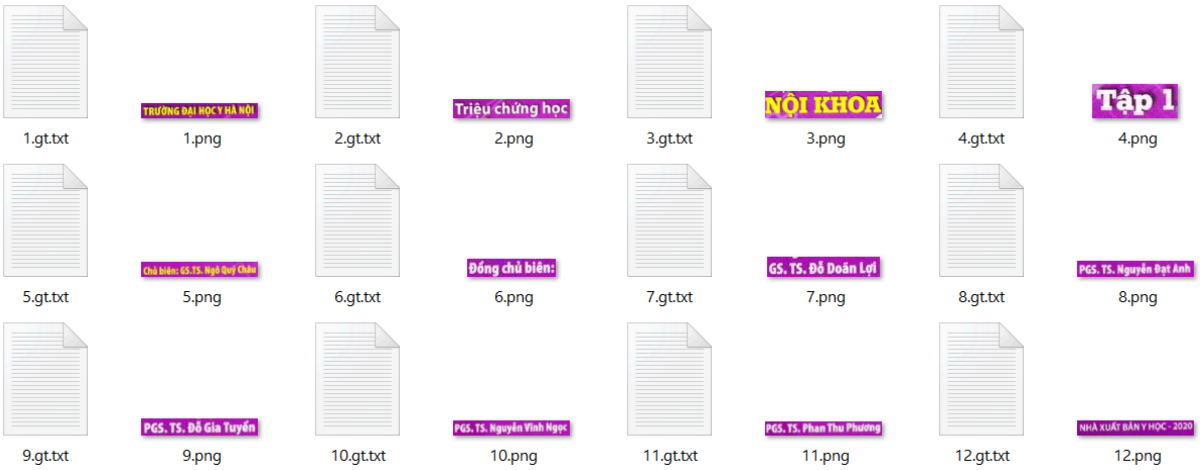
Point to tesstrain and run this:
`make tesseract-langdata`
You need to have `wget` and `make` to run this command. It may take a while

.github	6/17/2024 8:14 AM	File folder	
data	6/17/2024 8:19 AM	File folder	
src	6/17/2024 8:14 AM	File folder	
.gitignore	6/17/2024 8:14 AM	GITIGNORE File	1 KB
.pylintrc	6/17/2024 8:14 AM	PYLINTRC File	1 KB

Now you have a subdir named data in your tesstrain

Step 4: create ground truth data

- Crop image by line and label it. You need to create the label file with **gt.txt** extension and corresponding name to the image file. See below:



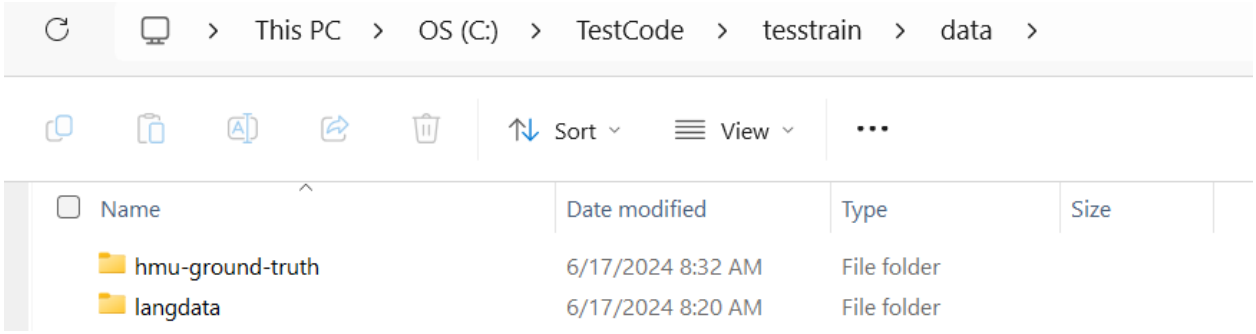
- You can install R and run this to make labeling process faster:

```
install.packages("remotes") # only if `remotes` is not installed
remotes::install_github("arcruz0/tesseractgt")

library(tesseractgt)
create_gt_txt(folder = "alg-ground-truth", # folder with images
              extension = "png",           # extension of image
files
              engine = tesseract::tesseract(language = "eng"))
```

  See also about the packages
  See also about cropping text line

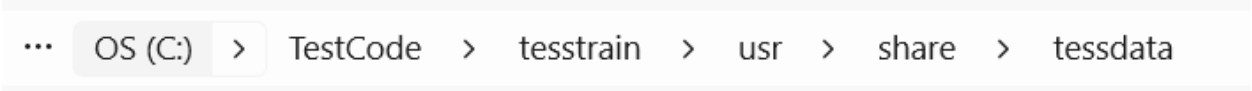
- Move your lan-ground-truth to your data folder
  Attention: you need to to set the name for your ground truth folder with this format `{langname}-ground-truth`



Step 5: create base data for fine-tuning

- Create folder like this in your tesstrain

```
└── usr/
    └── share/
        └── tessdata/
```



Download base data and save it in `tessdata` folder

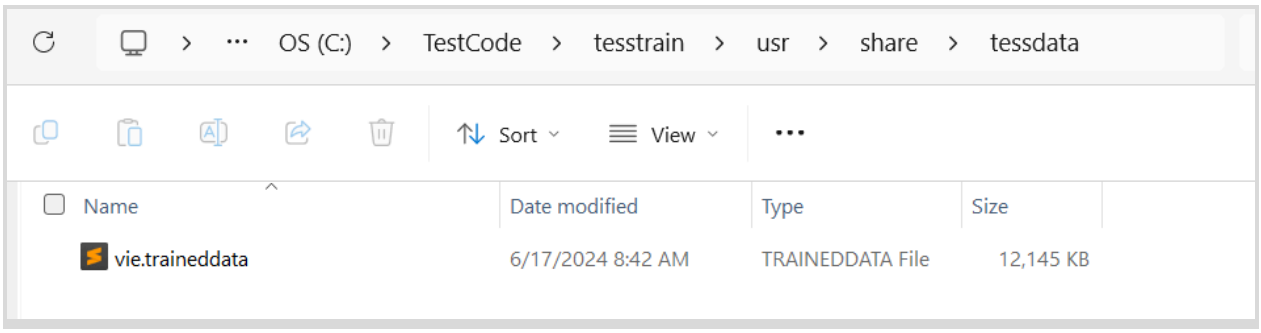
OR you can run this command

`wget -P usr/share/tessdata`
https://github.com/tesseract-ocr/tessdata_best/raw/main/eng.traineddata

```
C:\TestCode\tesstrain>wget -P usr/share/tessdata https://github.com/tesseract-ocr/tessdata_best/raw/main/vie.traineddata
--2024-06-17 08:42:35--  https://github.com/tesseract-ocr/tessdata_best/raw/main/vie.traineddata
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/tesseract-ocr/tessdata_best/main/vie.traineddata [following]
--2024-06-17 08:42:37--  https://raw.githubusercontent.com/tesseract-ocr/tessdata_best/main/vie.traineddata
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 2606:50c0:8001::154, 2606:50c0:8003::154, 2606:50c0:80
00::154, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|2606:50c0:8001::154|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12435550 (12M) [application/octet-stream]
Saving to: 'usr/share/tessdata/vie.traineddata'

vie.traineddata               100%[==============================================>]  11.86M  16.7MB/s    in 0.7s

2024-06-17 08:42:38 (16.7 MB/s) - 'usr/share/tessdata/vie.traineddata' saved [12435550/12435550]
```



Step 6: fine-tune

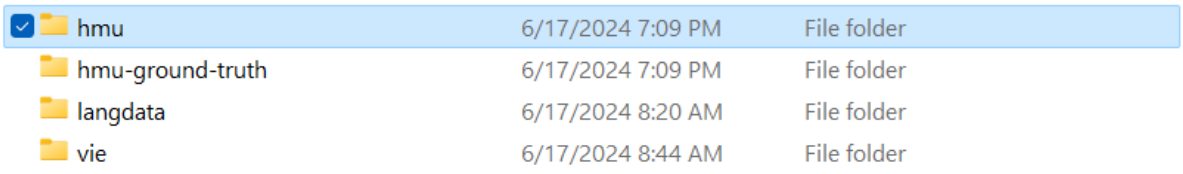
- Open your bash and run this command: `make training MODEL_NAME=alg START_MODEL=eng FINETUNE_TYPE=Impact`
- If you got this error

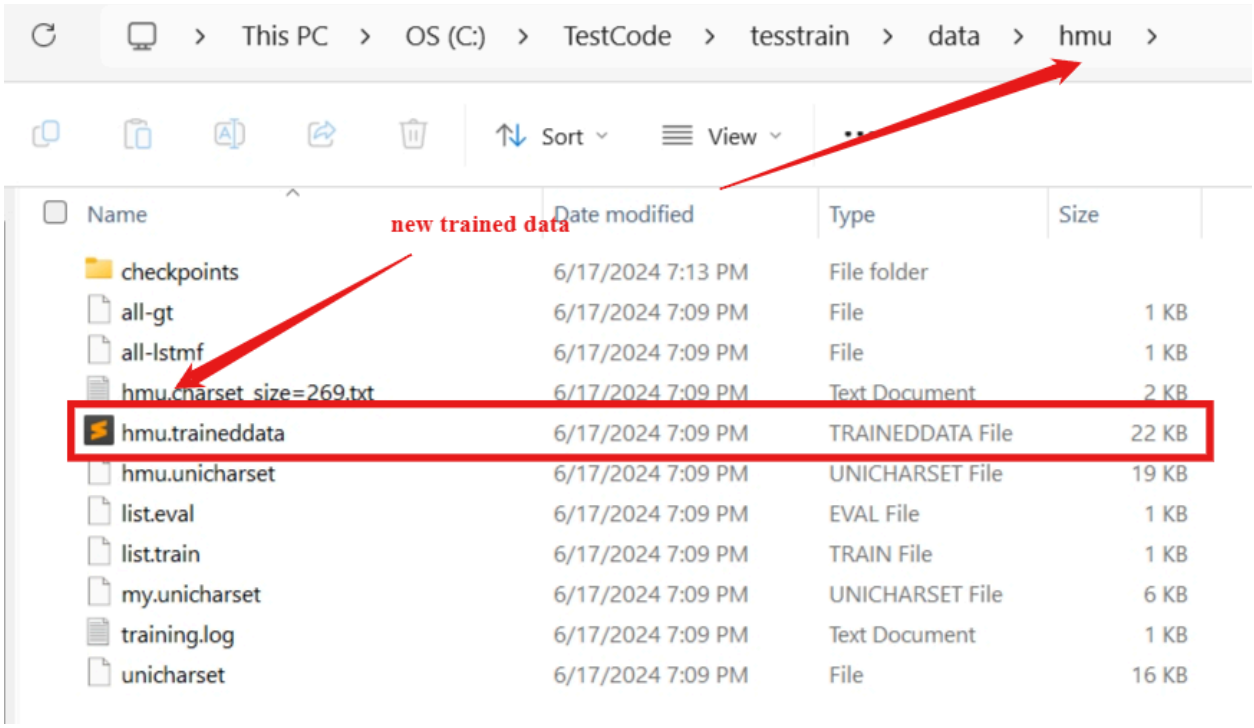
  ```
  dos2unix: command not found
  ```

  solve it using `sudo apt get dos2unix`

Step 7: using new model

- Find your `tessdata` in your base `tesseract-ocr` folder and paste the new trained data into it




This PC
OS (C:)
TestCode
tesstrain
data
hmu
Sort
View
Name
Date modified
Type
Size
new trained data
checkpoints 6/17/2024 7:13 PM File folder
all-gt 6/17/2024 7:09 PM File 1 KB
all-lstmf 6/17/2024 7:09 PM File 1 KB
hmu.charset_size=269.txt 6/17/2024 7:09 PM Text Document 2 KB
hmu.traineddata 6/17/2024 7:09 PM TRAINEDDATA File 22 KB
hmu.unicharset 6/17/2024 7:09 PM UNICHARSET File 19 KB
list.eval 6/17/2024 7:09 PM EVAL File 1 KB
list.train 6/17/2024 7:09 PM TRAIN File 1 KB
my.unicharset 6/17/2024 7:09 PM UNICHARSET File 6 KB
training.log 6/17/2024 7:09 PM Text Document 1 KB
unicharset 6/17/2024 7:09 PM File 16 KB