Study about the best open-source English OCR
Top 3 OCR that support English language

1. Tesseract OCR

- **Description**: One of the most widely used open-source OCR engines.
- **Languages Supported**: Over 100 languages, including English.
- **Features**: High accuracy, doesn't support PDF as input (we can use OCRmyPDF to convert PDF to image first). supports multiple output formats (PDF, plain text, etc.), and can be integrated with various programming languages.
- **GitHub**: Tesseract OCR

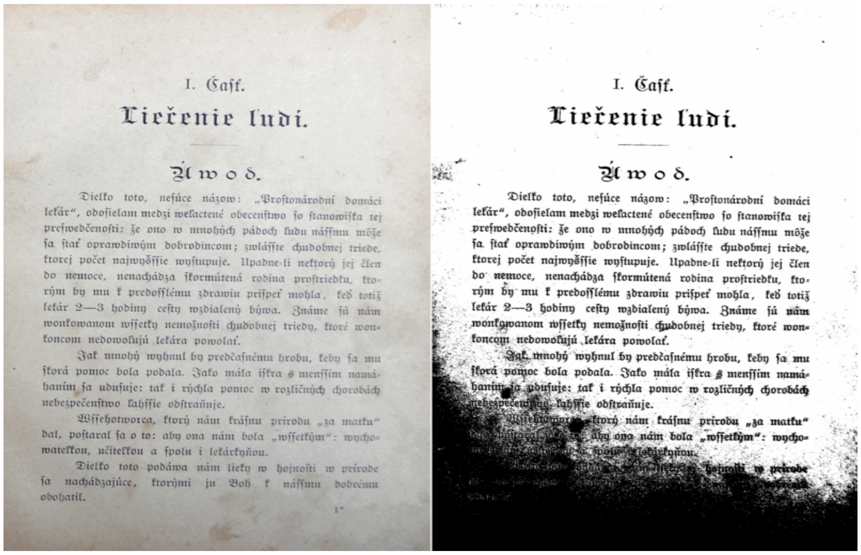
The text line recognizer of this model is based on the LSTM and we can Train Tesseract LSTM with make from Single Line Images and Ground truth Transcription, This lib has many arguments for adjusting the model for better recognition, also it has an internal image pre-processing unit (Leptonica library) which is helpful at most of the time (although sometimes we have to tune or remove them occasionally):

Some important steps for processing an image:

+**inverting image:** handle inverted image (dark background and light text)

+The best performance of this lib is active for images with at least 300 dpi resolution (so it may be beneficial to resize images.)

+ **Binarisation:** Tesseract does this internally (Otsu algorithm), but the result can be suboptimal, from Tesseract 5.0 we can choose one of these Adaptive Otsu and Sauvola algorithms with our threshold as an argument.



I. Čaſť.

Liečenie ľudí.

Úwod.

Dielko toto, neſúce názow: „Proſtonárodní domáci lekár“, odoſielam medzi weľactené obecenſtwo ſo ſtanowiſka tej preſwedčenoſti: že ono w mnohých pádoch ľudu náſſmu môže ſa ſtať oprawdiwým dobrodincom; zwláſſte chudobnej triede, ktorej počet najwyšſſie wyſtupuje. Upadne-li nektorý jej člen do nemoce, nenachádza ſkormútená rodina proſtriedku, ktorým by mu k predoſſlému zdrawiu priſpeť mohla, keď totiž lekár 2—3 hodiny ceſty wzdialený býwa. Známe ſú nám wonkowanom wſſetky nemožnoſti chudobnej triedy, ktoré wonkoncom nedowoľujú lekára powolať.

Jak mnohý wyhnul by predčaſnému hrobu, keby ſa mu ſkorá pomoc bola podala. Jako mála iſkra s menſſím namáhaním ſa uduſuje: tak i rýchla pomoc w rozličných chorobách nebezpečenſtwo ľahſſie odſtraňuje.

Wſſehotworca, ktorý nám kráſnu prirodu „za matku“ dal, poſtaral ſa o to: aby ona nám bola „wſſetkým“: wychowatelkou, učitelkou a ſpolu i lekárkyňou.

Dielko toto podáwa nám lieky w hojnoſti w prirode ſa nachádzajúce, ktorými ju Boh k náſſmu dobrému obohatil.

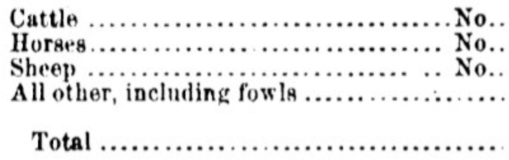
1*

Example of an image in which Otsu affects the contrast of the image

Note: Noise is a random variation of brightness or color in an image and it can not be removed using Binarisation, if we want to use this lib we have to apply some pre-processing steps (I prefer to use a deep Auto-encoder to remove them)

+ <u>Sometimes we need to apply Dilation and Erosion</u> : We need Bold characters or Thin characters (especially those with Serifs), which may impact the recognition of details and reduce recognition accuracy.

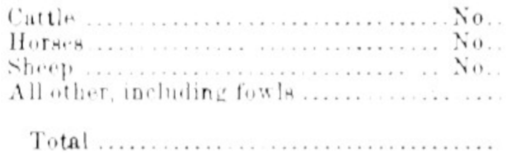
Original:



Cattle .... No..
Horses.................................... No..
Sheep .... No..
All other, including fowls ....

Total ....

Erosion applied:

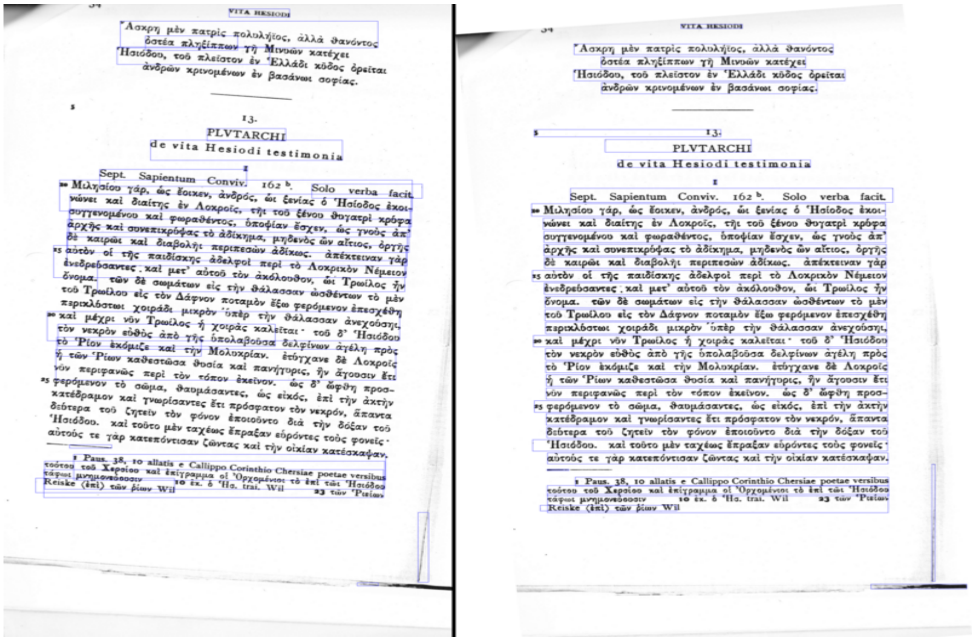


Cattle .... No..
Horses.................................... No..
Sheep .... No..
All other, including fowls ....

Total ....

Example of erosion operation (it’s good idea to apply on Heavy ink bleeding from historical documents)

+<u>Rotation (deskewing) :</u> A skewed image is when a page has been scanned when not straight. The quality of Tesseract’s line segmentation reduces significantly if a page is too skewed



34 VITA HESIODI

Ἄσκρη μὲν πατρὶς πολυλήϊος, ἀλλὰ θανόντος
ὀστέα πληξίππων γῆ Μινυῶν κατέχει
Ἡσιόδου, τοῦ πλεῖστον ἐν Ἑλλάδι κῦδος ὀρεῖται
ἀνδρῶν κρινομένων ἐν βασάνωι σοφίας.

13.

PLVTARCHI

de vita Hesiodi testimonia

1

Sept. Sapientum Conviv. 162ᵇ. Solo verba facit. Μιλησίου γάρ, ὡς ἔοικεν, ἀνδρός, ὧι ξενίας ὁ Ἡσίοδος ἐκοινώνει καὶ διαίτης ἐν Λοκροῖς, τῆι τοῦ ξένου θυγατρὶ κρύφα συγγενομένου καὶ φωραθέντος, ὑποψίαν ἔσχεν, ὡς γνοὺς ἀπ' ἀρχῆς καὶ συνεπικρύψας τὸ ἀδίκημα, μηδενὸς ὢν αἴτιος, ὀργῆς δὲ καιρῶι καὶ διαβολῆι περιπεσὼν ἀδίκως. ἀπέκτειναν γὰρ αὐτὸν οἱ τῆς παιδίσκης ἀδελφοὶ περὶ τὸ Λοκρικὸν Νέμειον ἐνεδρεύσαντες, καὶ μετ' αὐτοῦ τὸν ἀκόλουθον, ὧι Τρωίλος ἦν ὄνομα. τῶν δὲ σωμάτων εἰς τὴν θάλασσαν ὠσθέντων τὸ μὲν τοῦ Τρωίλου εἰς τὸν Δάφνον ποταμὸν ἔξω φερόμενον ἐπεσχέθη περικλύστωι χοιράδι μικρὸν ὑπὲρ τὴν θάλασσαν ἀνεχούσηι, καὶ μέχρι νῦν Τρωίλος ἡ χοιρὰς καλεῖται· τοῦ δ' Ἡσιόδου τὸν νεκρὸν εὐθὺς ἀπὸ γῆς ὑπολαβοῦσα δελφίνων ἀγέλη πρὸς τὸ Ῥίον ἐκόμιζε καὶ τὴν Μολυκρίαν. ἐτύγχανε δὲ Λοκροῖς ἡ τῶν Ῥίων καθεστῶσα θυσία καὶ πανήγυρις, ἣν ἄγουσιν ἔτι νῦν περιφανῶς περὶ τὸν τόπον ἐκεῖνον. ὡς δ' ὤφθη προσφερόμενον τὸ σῶμα, θαυμάσαντες, ὡς εἰκός, ἐπὶ τὴν ἀκτὴν κατέδραμον καὶ γνωρίσαντες ἔτι πρόσφατον τὸν νεκρόν, ἅπαντα δεύτερα τοῦ ζητεῖν τὸν φόνον ἐποιοῦντο διὰ τὴν δόξαν τοῦ Ἡσιόδου. καὶ τοῦτο μὲν ταχέως ἔπραξαν εὑρόντες τοὺς φονεῖς· αὐτούς τε γὰρ κατεπόντισαν ζῶντας καὶ τὴν οἰκίαν κατέσκαψαν.

1 Paus. 38, 10 allatis e Callippo Corinthio Chersiae poetae versibus τούτου τοῦ Χερσίου καὶ ἐπίγραμμα οἱ Ὀρχομένιοι τὸ ἐπὶ τῶι Ἡσιόδου τάφωι μνημονεύουσιν   10 ἐκ. ὁ Ἡσ. trai. Wil   23 τῶν Ῥείων Reiske (ἐπὶ) τῶν ῥίων Wil

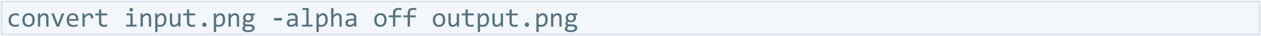
The Border problem: we have some challenges with borders: missing borders, too big borders (thicker than 10px and especially when processing a single letter/digit or one word on a large background) can cause problems).

Also, we need to pay attention to Scanned pages that often have dark borders around them. These can be erroneously picked up as extra characters, especially if they vary in shape and gradation.

+ We have to remove the Alpha channel from images:

Tesseract 4.00 removes the alpha channel with the leptonica function pixRemoveAlpha(): it removes the alpha component by blending it with a white background.

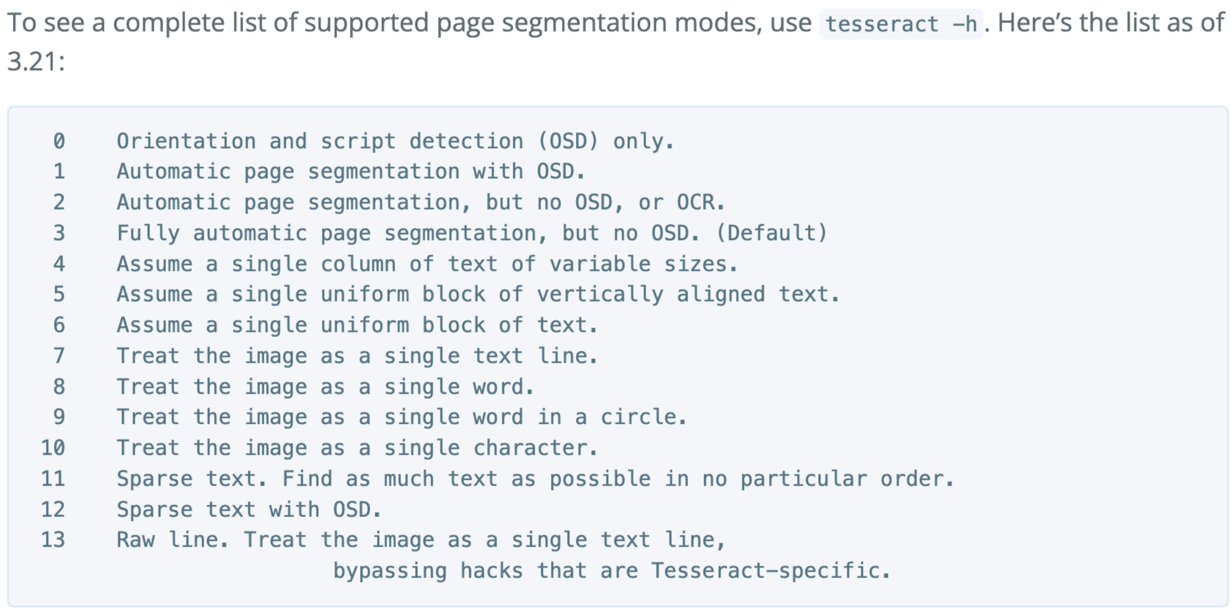
But Tesseract 3.0x expects that users remove the alpha channel using this command :



```
convert input.png -alpha off output.png
```

+ Page segmentation method:

By default Tesseract expects a page of text when it segments an image. If you're just seeking to OCR a small region, try a different segmentation mode, using the --psm argument.



To see a complete list of supported page segmentation modes, use `tesseract -h`. Here's the list as of 3.21:

```
 0    Orientation and script detection (OSD) only.
 1    Automatic page segmentation with OSD.
 2    Automatic page segmentation, but no OSD, or OCR.
 3    Fully automatic page segmentation, but no OSD. (Default)
 4    Assume a single column of text of variable sizes.
 5    Assume a single uniform block of vertically aligned text.
 6    Assume a single uniform block of text.
 7    Treat the image as a single text line.
 8    Treat the image as a single word.
 9    Treat the image as a single word in a circle.
10    Treat the image as a single character.
11    Sparse text. Find as much text as possible in no particular order.
12    Sparse text with OSD.
13    Raw line. Treat the image as a single text line,
                         bypassing hacks that are Tesseract-specific.
```

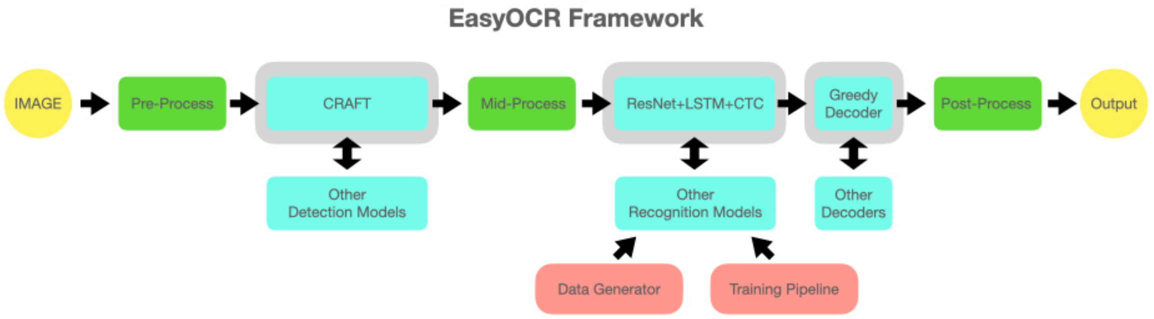
Reference 1 Reference 2 Reference 3(neural network)

Tip : Vertical text is now supported for Chinese, Japanese and Korean, and should be detected automatically. Also Instead of the file path, we can also pass an OpenCV image object (numpy array)

2. EasyOCR

- **Description**: A robust and user-friendly OCR tool built on PyTorch.
- **Languages Supported**: 80+ languages, including English.
- **Features**: Easy to use, supports various image formats, high accuracy, and fast processing.
- **GitHub**: EasyOCR

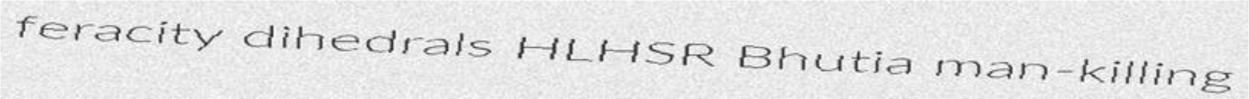
This OCR tool tries to present a simple and fast OCR rather than a complex one, it uses from CRAFT as a detection model and a trainable custom recognition model based on a neural network.

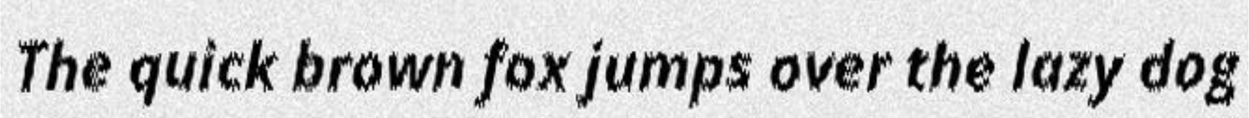


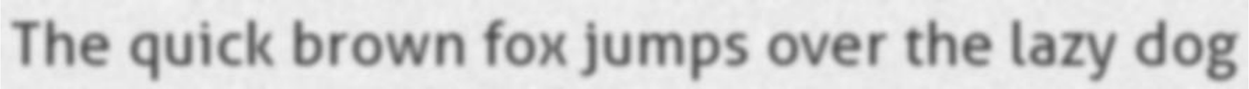
How we can fine-tune this model?

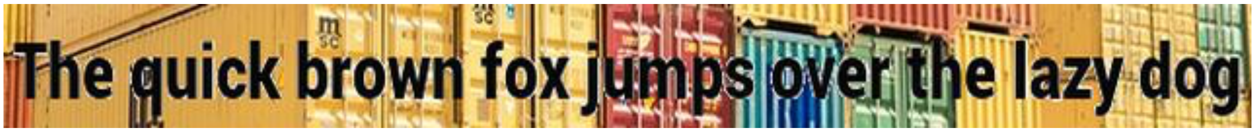
You can use your own data or generate your own dataset. To generate your own data, we recommend using TextRecognitionDataGenerator. After you have a dataset, you can train your own model by following this repository deep-text-recognition-benchmark. The network needs to be fully convolutional in order to predict flexible text length. Our current network is **'None-VGG-BiLSTM-CTC'**. Once you have your trained model (a .pth file), you need 2 additional files describing recognition network architecture and model configuration.

Note: TextRecognitionDataGenerator is a very helpful lib for generating datasets, also we can add skewing, distortion, and blurring to our text or add different backgrounds to them and finally we can make handwritten examples too, all of them help us to train robust model.

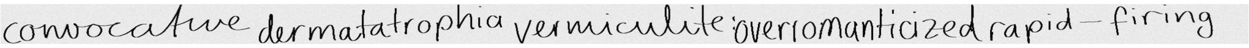

feracity dihedrals HLHSR Bhutia man-killing


The quick brown fox jumps over the lazy dog


The quick brown fox jumps over the lazy dog



The quick brown fox jumps over the lazy dog



convocative dermatatrophia vermiculite overromanticized rapid-firing

Notice: at this moment, this framework doesn't support handwritten OCR but they will add it in future versions.

Note 1: ['ch_sim','en'] is the list of languages you want to read. You can pass several languages at once but not all languages can be used together. English is compatible with every language and languages that share common characters are usually compatible with each other.

Note 2: Instead of the file path chinese.jpg, you can also pass an OpenCV image object (numpy array) or an image file as bytes. A URL to a raw image is also acceptable.

Parameters of Text Detection model (from CRAFT)

- **text_threshold** (float, default = 0.7) - Text confidence threshold
- **low_text** (float, default = 0.4) - Text low-bound score
- **link_threshold** (float, default = 0.4) - Link confidence threshold
- **canvas_size** (int, default = 2560) - Maximum image size. Image bigger than this value will be resized down.
- **mag_ratio** (float, default = 1) - Image magnification ratio

recognize method

If horizontal_list and free_list are not given. It will treat the whole image as one text box.

Parameters

- **image** (string, numpy array, byte) - Input image
- **horizontal_list** (list, default=None) - see format from output of detect method
- **free_list** (list, default=None) - see format from output of detect method
- **decoder** (string, default = 'greedy') - options are 'greedy', 'beamsearch' and 'wordbeamsearch'.
- **beamWidth** (int, default = 5) - How many beam to keep when decoder = 'beamsearch' or 'wordbeamsearch'
- **batch_size** (int, default = 1) - batch_size>1 will make EasyOCR faster but use more memory
- **workers** (int, default = 0) - Number thread used in of dataloader
- **allowlist** (string) - Force EasyOCR to recognize only subset of characters. Useful for specific problem (E.g. license plate, etc.)

- **blocklist** (string) - Block subset of character. This argument will be ignored if allowlist is given.
- **detail** (int, default = 1) - Set this to 0 for simple output
- **paragraph** (bool, default = False) - Combine result into paragraph
- **contrast_ths** (float, default = 0.1) - Text box with contrast lower than this value will be passed into model 2 times. First is with original image and second with contrast adjusted to 'adjust_contrast' value. The one with more confident level will be returned as a result.
- **adjust_contrast** (float, default = 0.5) - target contrast level for low contrast text box

Return list of results

It is good to know what the responsibility of the decoder. What is the different type of decoding here?

- **Greedy Decoder**:

  - **Description**: This is the simplest decoding method. It selects the most likely character at each step without considering future steps.
  - **Advantages**: Fast and straightforward.
  - **Disadvantages**: It can be less accurate because it doesn't consider context or future predictions.

- **Beam Search Decoder**:

  - **Description**: This method keeps multiple hypotheses (beams) at each step and selects the sequence with the highest cumulative probability. It looks ahead to optimize the entire sequence.
  - **Advantages**: More accurate than the greedy decoder because it considers multiple possible sequences.
  - **Disadvantages**: Slower than greedy decoding due to the increased computational complexity.

- **Word Beam Search Decoder**:

  - **Description**: This is an extension of the beam search decoder that uses a dictionary of known words to improve accuracy. It ensures that the decoded sequences are valid words or parts of words.
  - **Advantages**: Highest accuracy for text that matches the dictionary. Useful for languages with clear word boundaries and known vocabulary.
  - **Disadvantages**: Requires a dictionary and can be slower than regular beam search.

In this OCR framework, we have an alphabet list TXT file for each language (for Persian it contains 52)

This is Persian common word list link :

https://github.com/JaidedAI/EasyOCR/raw/master/easyocr/dict/fa.txt

As we can see the default word list for Persian is too bad, and we don’t use from many of them in daily life (also some of them is not Persian at all)

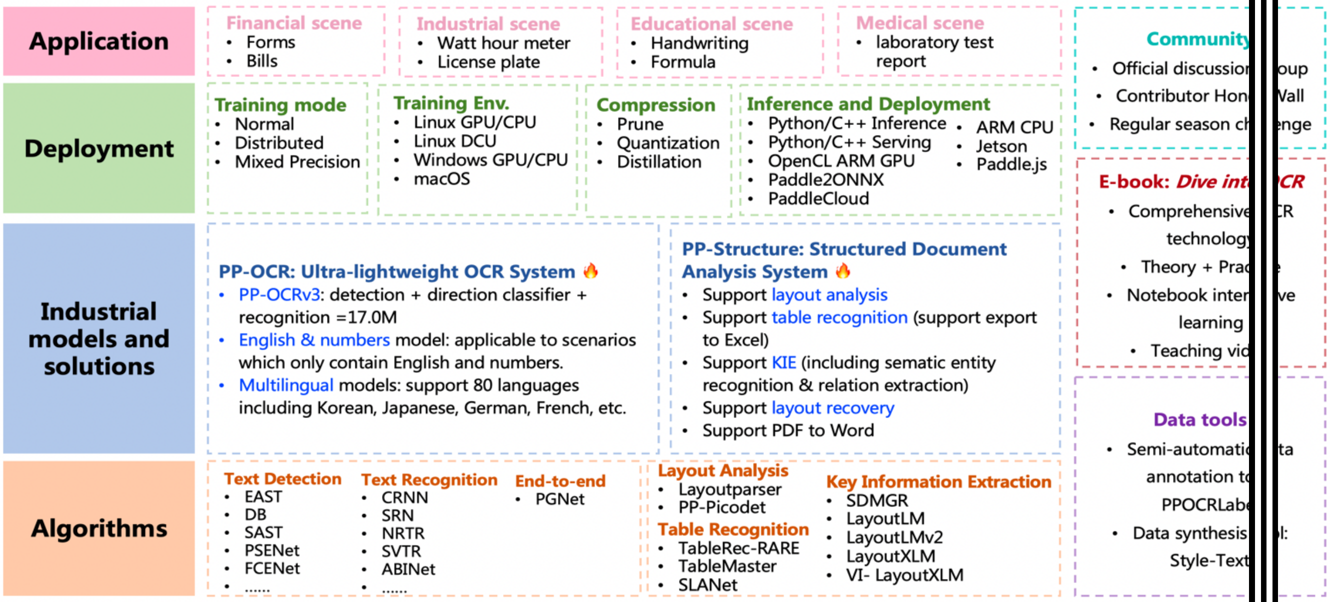
العاشی, پربلاو, سوئویاروی, جیانکشی, پاراگونیت, کاستیلفابیب...

Note : All deep learning execution is based on Pytorch

Recognition model paper: https://arxiv.org/abs/1507.05717

3. **PaddleOCR**
   - **Description**: Part of the PaddlePaddle deep learning platform, providing high-performance OCR.
   - **Languages Supported**: Multiple languages, including English.
   - **Features**: End-to-end OCR system, supports various image formats, integrates with other PaddlePaddle tools, high accuracy, and fast inference.
   - **GitHub**: PaddleOCR

Tip: this framework has a model compression feature which we can make ultra-light weight models which can deployed on embedded and IOT devices and …



Application
Financial scene
- Forms
- Bills

Industrial scene
- Watt hour meter
- License plate

Educational scene
- Handwriting
- Formula

Medical scene
- laboratory test report

Deployment
Training mode
- Normal
- Distributed
- Mixed Precision

Training Env.
- Linux GPU/CPU
- Linux DCU
- Windows GPU/CPU
- macOS

Compression
- Prune
- Quantization
- Distillation

Inference and Deployment
- Python/C++ Inference
- Python/C++ Serving
- OpenCL ARM GPU
- Paddle2ONNX
- PaddleCloud
- ARM CPU
- Jetson
- Paddle.js

Industrial models and solutions
PP-OCR: Ultra-lightweight OCR System 🔥
- PP-OCRv3: detection + direction classifier + recognition =17.0M
- English & numbers model: applicable to scenarios which only contain English and numbers.
- Multilingual models: support 80 languages including Korean, Japanese, German, French, etc.

PP-Structure: Structured Document Analysis System 🔥
- Support layout analysis
- Support table recognition (support export to Excel)
- Support KIE (including sematic entity recognition & relation extraction)
- Support layout recovery
- Support PDF to Word

Algorithms
Text Detection
- EAST
- DB
- SAST
- PSENet
- FCENet
- ……

Text Recognition
- CRNN
- SRN
- NRTR
- SVTR
- ABINet
- ……

End-to-end
- PGNet

Layout Analysis
- Layoutparser
- PP-Picodet

Table Recognition
- TableRec-RARE
- TableMaster
- SLANet

Key Information Extraction
- SDMGR
- LayoutLM
- LayoutLMv2
- LayoutXLM
- VI- LayoutXLM

Community
- Official discussion oup
- Contributor Hon Wall
- Regular season ch enge

E-book: *Dive int CR*
- Comprehensive CR technology
- Theory + Pra e
- Notebook inter ve learning
- Teaching vid

Data tools
- Semi-automati ta annotation to PPOCRLab
- Data synthesis l: Style-Text

Yml configuration method :
The PaddleOCR uses configuration files to control network training and evaluation parameters. In the configuration file, you can set the model, optimizer, loss function, and pre- and post-processing parameters of the model. PaddleOCR reads these parameters from the configuration file, and then builds a complete training process to train the model. Fine-tuning can also be completed by modifying the parameters in the configuration file, which is simple and convenient.

Evaluation Indicators for this framework :

(1) **Detection stage:** First, evaluate according to the IOU of the detection frame and the labeled frame. If the IOU is greater than a certain threshold, it is judged that the detection is accurate. Here, the detection frame and the label frame are different from the general general target detection frame, and they are represented by polygons. Detection accuracy: the percentage of the correct detection frame number in all detection frames is mainly used to judge the detection index. Detection recall rate: the percentage of correct detection frames in all marked frames, which is mainly an indicator of missed detection.

(2) **Recognition stage:** Character recognition accuracy, that is, the ratio of correctly recognized text lines to the number of marked text lines. Only the entire line of text recognition pairs can be regarded as correct recognition.

(3) **End-to-end statistics:** End-to-end recall rate: accurately detect and correctly identify the proportion of text lines in all labeled text lines; End-to-end accuracy rate: accurately detect and correctly identify the number of text lines in the detected text lines The standard for accurate detection is that the IOU of the detection box and the labeled box is greater than a certain threshold, and the text in the correctly identified detection box is the same as the labeled text.

Different types of model compression :

1-Model Quantization: Generally, a more complex model would achieve better performance in the task, but it also leads to some redundancy in the model. Quantization is a technique that reduces this redundancy by reducing the full precision data to a fixed number, so as to reduce model calculation complexity and improve model inference performance.

[https://github.com/PaddlePaddle/PaddleOCR/blob/main/deploy/slim/quantization/README_en.md]

2-model pruning: Generally, a more complex model would achieve better performance in the task, but it also leads to some redundancy in the model. Model Pruning is a technique that reduces this redundancy by removing the sub-models in the neural network model, so as to reduce model calculation complexity and improve model inference performance.
[https://github.com/PaddlePaddle/PaddleOCR/blob/main/deploy/slim/prune/README_en.md]

3-model distillation:

knowledge distillation refers to the use of teacher models to guide student models to learn specific tasks, to ensure that the small model obtains a relatively large performance improvement under the condition of unchanged parameters. In addition, in the knowledge distillation task, a mutual learning model training method was also derived.

Whether it is a large model distilling a small model, or a small model learning from each other and updating parameters, they are essentially the output between different models or mutual supervision between feature maps. The only difference is (1) whether the model requires fixed parameters. (2) Whether the model needs to be loaded with a pre-trained model. For the case where a large model distills a small model, the large model generally needs to load the pre-trained model and fix the parameters. For the situation where small models distill each other, the small models generally do not load the pre-trained model, and the parameters are also in a learnable state. In the task of knowledge distillation, it is not only the distillation between two models, but also the situation where multiple models learn from each other.

[https://github.com/PaddlePaddle/PaddleOCR/blob/main/doc/doc_en/knowledge_distillation_en.md]

Tip: in this framework, we can train every part of the framework with our data