

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství

Obor: Aplikace informatiky v přírodních vědách



**Optické rozpoznávání znaků
na naskenovaných historických
plakátech pomocí
nejmodernějších metod**

**Optical Character Recognition
on Scanned Historical Posters
Using the State-of-the-Art
Methods**

VÝZKUMNÝ ÚKOL

Vypracoval: Anna Gruberová

Vedoucí práce: Ing. Adam Novozámský, Ph.D.

Rok: 2022

České vysoké učení technické v Praze

Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství

Akademický rok 2021/2022

ZADÁNÍ VÝZKUMNÉHO ÚKOLU

Student: Bc. Anna Gruberová

Studijní program: Aplikace informatiky v přírodních vědách

Název práce česky: Optické rozpoznávání znaků na naskenovaných historických plakátech pomocí nejmodernějších metod

Název práce anglicky: Optical Character Recognition on Scanned Historical Posters Using the State-of-the-Art Methods

Pokyny pro vypracování:

1. Seznamte se s problematikou optického rozpoznávání znaků. Na základě rešerše vyberte několik metod, se kterými budete dále pracovat a vyhodnocovat úspěšnost jejich detekce.
2. Stáhněte několik volně dostupných datasetů, které jsou využívány v literatuře k porovnání jednotlivých metod na OCR. Dále vytvořte svůj vlastní dataset z obdržených dat.
3. Nastudujte techniky porovnání OCR výstupů s ground-truth.
4. U vybraných metod prostudujte jejich chování na jednotlivých datasetech při různém nastavení parametrů.
5. Navrhněte také možnosti filtrování výstupů jednotlivých metod za účelem snížení falešných detekcí.

Doporučená literatura:

- [1] R. C. Gonzalez, R. E. Woods, Digital Image Processing (4th ed.). Pearson, 2018. ISBN 9353062985.
- [2] GOODFELLOW, Ian, Yoshua BENGIO a Aaron COURVILLE. Deep learning. Cambridge, Massachusetts: The MIT Press, [2016]. ISBN 0262035618.
- [3] SMITH, R. An Overview of the Tesseract OCR Engine. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2 [online]. IEEE, 2007, 2007, s. 629-633. ISBN 0-7695-2822-8. ISSN 1520-5363. Dostupné z: doi:10.1109/ICDAR.2007.4376991
- [4] CHEN, Xiaoxue, et al. Text Recognition in the Wild. ACM Computing Surveys [online]. 2021, 54(2), 1-35 [cit. 2021-10-2]. ISSN 0360-0300. Dostupné z: doi:10.1145/3440756

Jméno a pracoviště vedoucího práce:

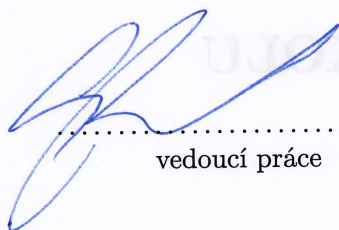
Ing. Adam Novozámský, Ph.D.

Computer Vision Lab, Institute of Visual Computing & Human-Centered Technology,
TU Wien - Faculty of Informatics

Datum zadání výzkumného úkolu: 15. 10. 2021

Termín odevzdání výzkumného úkolu: 31. 8. 2022

V Praze dne 15. 10. 2021



.....
vedoucí práce

.....
vedoucí katedry

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracovala samostatně a použila jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne

.....

Anna Gruberová

Poděkování

.

Anna Gruberová

Název práce:

Optické rozpoznávání znaků na naskenovaných historických plakátech pomocí nej

Autor: Anna Gruberová

Obor: Aplikace informatiky v přírodních vědách

Druh práce: Výzkumný úkol

Vedoucí práce: Ing. Adam Novozámský, Ph.D.
Computer Vision Lab, Institute of Visual Computing & Human-Centered Technology, TU Wien - Faculty of Informatics

Konzultant: –

Abstrakt: .

Klíčová slova: .

Title:

Optical Character Recognition on Scanned Historical Posters Using the State-of-t

Author: Anna Gruberová

Abstract: .

Key words: .

Contents

1	OCR	11
1.1	Text detection	11
1.2	Text recognition	11
2	Software	13
2.1	Scene text detection	13
2.1.1	CRAFT	13
2.2	Scene text recognition	13
2.2.1	CRNN	13
2.3	End-to-end systems	13
2.3.1	Tesseract	14
2.3.2	EasyOCR	14
2.3.3	Keras-ocr	15
2.3.4	Google Cloud Vision	16
2.3.5	Other	16
3	Datasets	17
3.1	Synthetic datasets	17
3.2	Scene text image datasets	17
3.3	Wien TU dataset	18
4	Testing methods	19
4.1	EasyOCR	19
4.2	Tesseract	19
4.3	19

Chapter 1

OCR

1.1 Text detection

1.2 Text recognition

Outline

- What is OCR
- Datasets (types(synthetic, photos, scanned documents), problems(languages, noise, nonhorizontal text))
- Text detection
 - Description
 - Methods (CRAFT)
- Text recognition
 - Description
 - Methods
- End-to-end systems (Annotating tool)
 - Reading scanned documents
 - EasyOCR
 - keras-ocr
 - Tesseract (PyTesseract)
 - (Google Cloud Vision free) paid
 - (AWS Recognition) paid
 - (Kili) paid

- Results evaluation
 - Comparison of output and ground-truth
 - Bag of words
- Testing methods on free datasets
 - Description of datasets
- Using methods on historical posters
 - Description of dataset

Chapter 2

Software

2.1 Scene text detection

Methods

2.1.1 CRAFT

Character Region Awareness for Text Detection (CRAFT) is framework for scene text detection. It uses a Convolutional Neural Network. It performs good also on curved or differently deformed texts. Its methodology is to localize individual characters then characters belonging to the same word (based on distance) can be connected into word box or polygon. After that bounding box is created around it and output contains the rectangle coordinates.[3]

2.2 Scene text recognition

Methods

2.2.1 CRNN

Convolutional Recurrent Neural Network

2.3 End-to-end systems

End-to-end system when given an image with text it can both detect and recognize the text and produces a string output of the text. The supply of such tools is wide, ranging from open source libraries for various programming languages to commercial softwares with modern GUI. New methods are still being developed as there is always

space for improvement. New methods can come from commercial background or are developed for international OCR competitions. In the next section some of the free available tools are described.

2.3.1 Tesseract

Tesseract is an open source text recognition engine. It supports over 160 languages. Originally Tesseract was created by Hewlett-Packard in late 1980s, from 2006 it is developed and maintained by Google. As it does not have a built-in GUI direct use is via command line. However, there exist a significant number of GUIs for Linux, Windows, Mac for computer usage and also for Android and iOS to use on mobile phones and few online OCR services. Another way how to use the engine is via libraries for computer languages, namely for example they exist for Java called tess4j, Python called pytesseract, R, Ruby and others. [11]

Tesseract is mainly used as tool for recognizing documents (with both computer font text or handwritten text). Best results are obtained on preprocessed images. The preprocessing includes noise reduction, horizontal alignment of text, elimination of dark borders around text region, conversion to binary black and white picture and other adjustments depending on the nature of the picture. Thus when used on scene text images it gives generally worse results than other OCR softwares.

Computations with Tesseract are supported for GPU and also CPU. Tesseract uses for recognition Long Short Term Memory (LSTM) model (kind of RNN).

Page Segmentation models By default Tesseract expects a page of text – black letters on white background grouped in horizontal lines, where font type and font size vary only slightly. To deal with differently distributed text over an image Tesseract provides thirteen page segmentation models (PSM). When selecting the right model Tesseract performance can increase from zero up to almost perfect results. Description of all the PSMs can be find directly via Tesseract help command in console application. <https://pyimagesearch.com/2021/11/15/tesseract-page-segmentation-modes-psms-explained-how-to-improve-your-ocr-accuracy/>

2.3.2 EasyOCR

EasyOCR is a product of Jaded AI for both image text detection and recognition. it supports over 80 languages and various scripts such as Latin, Chinese, Arabic etc. The company offers software with web interface for free and also prepaid version which enables usage of a new model for custom data. However, in addition to the web interface, the company also created a python package under the same name.[1]

The product is still in development and aims for wider functionality. A future idea of EasyOCR package is to provide an easy-to-use tool where one can plug-in already created state-of-art models and use them for annotating. Pipeline of EasyOCR behavior is shown in the image 2.1. As it can be seen in this image, default detection model is CRAFT and for recognition is used CRNN (Convolutional Recurrent

Neural Network) which model is composed of following components: feature extraction (Resnet is used) and VGG (Convolutional Neural Network), sequence labeling (LSTM is used) and decoding (CTC is used).[2]

EasyOCR package by default computes annotation on GPU, however there is a possibility for CPU computations (provided that the selected model supports it).

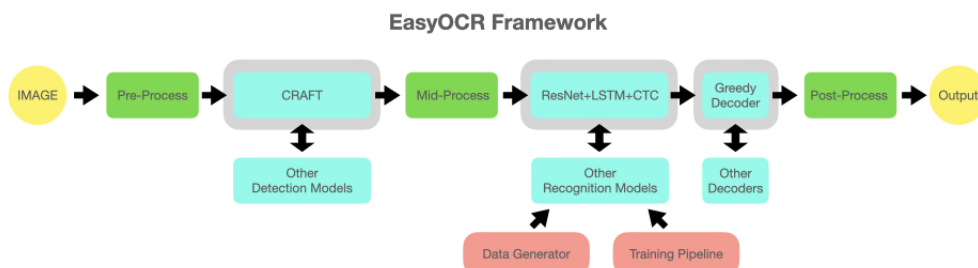


Figure 2.1: Diagram of EasyOCR pipeline. Grey slots are placeholders for models. The mentioned models are the ones used as default. [2]

2.3.3 Keras-ocr

keras-ocr is a python library used for detecting and recognizing text in images created by Fausto Morales. It works with variety of languages and with different writing scripts. It allows computing on CPU as well as on GPU. It unites the CRAFT text detection model and an implementation in Keras python library of CRNN for recognizing text, worth mentioning this is a different implementation of CRNN than in EasyOCR.[7]

On the official website¹ of the package there is a comparison of this method with two other OCR APIs – Google Cloud Vision and AWS Rekognition. Their performance was tested on 1,000 images from the COCO-Text validation set using a basic pre-trained model of each method. None of the investigated methods performed poorly; however, AWS Rekognition had the worst precision and recall results. Google’s method and keras-ocr has similar results. It is important to mention that no tuning parameters were used in any of these methods. Another candidate for comparison was Tesseract but it performed on very badly on given data, most likely due to the fact that Tesseract is suitable for scanned documents rather than for photos of real life scenery and objects with text. [7]

CRAFT already provides a pretrained model which can be used directly without modification for text detection or it is used as initial model for training a new model on new data. This model was trained on three datasets (SynthText, IC13, IC17) and supports English and multi language text detection.[9] Similarly for recognition, CRNN also has a pretrained model. This model was trained on the synthetic word dataset which consists of 9 million images with vocabulary of 90K English words.[10] To use these models in the keras-ocr library one either doesn’t specify anything and

¹<https://pypi.org/project/keras-ocr/>

use the defaults, or pass the value `clovaai-general` for the CRAFT pretrained model or `kurapan` for the CRNN model.

Keras-ocr offers preprocessing for four public datasets though any text image dataset can be examined using this tool. These four datasets are: BornDigital dataset, COCO-Text dataset, ICDAR 2013 dataset, ICDAR 2019 dataset (only Latin-only scripts).[6]

2.3.4 Google Cloud Vision

Google Cloud Vision is software from Google which consist of two products: AutoML Vision and Vision API. Vision API detects objects, faces and text from images with already pretrained model. With AutoML Vision user can train custom model from own data. It has free and paid version with a GUI.[4] API for python

2.3.5 Other

AWS Rekognition

Chapter 3

Datasets

3.1 Synthetic datasets

3.2 Scene text image datasets

SCUT-CTW dataset

SCUT-CTW1500 dataset contains exactly 1500 images of scene text in English language. The key feature of this dataset is that each image contains both horizontally aligned text and curved text. There are cases where the curvature is only slight and cases where text forms a circle with letter upside down. Recognizing multi-oriented and curved text is more of a challenge than horizontal text. This dataset is split to train and test data. Two thirds of dataset thus one thousand images for training and five hundred for testing. According to the description of this dataset on relevant GitHub repository dataset was manually labeled and lately corrected, therefore labels seem to be very accurate. However for example ground truth for image 1313.jpg misses all occurrences of letter I, as the depicted font was probably misread.[12, 8]

The labels for train data are in XML format and each file carries information about the file name of respective image file, text information – i.e., words in a text line, 14 coordinates of a bounding polygon and coordinates, height and width of a circumscribed rectangle. Later the authors added coordinates of center point of each English letter to be used as detection ground truth. The ground truth of test data is in simple text file (TXT) and contains only 14 coordinates of polygon and a text which is within that region.

KAIST Scene Text Database

This dataset contains 3000 images of photographed text. It can be divided into three major categories – text of Korean language, English language and mixed languages. As I concentrate on text in latin script in this paper further information relates to

the English language dataset. The number of images is then reduced to less than four hundred images. Photographed objects are mostly shop banners or parts of magazine front pages. Photographs were either taken by a high-resolution digital camera or a low-resolution mobile phone camera.[5] Each photography has a ground truth description and a bitmap image. In the bitmap file only text is highlighted (by white or red color) and everything else apart from text is set as black. Ground truth files are in XML format and includes a name of an image, its resolution and bounding box for each word and also a bounding box for each letter of the word.

To use this dataset for testing and training the XML ground truth needed to be converted to string and int values. I wrote a parser, that combines letters to form a word that is within a given bounding box. I changed the notation of bounding boxes from one coordinate, width and height attributes to two top left and bottom right coordinates.

Unfortunately this dataset has few errors in filenames of corresponding files or in the content of XML files. Usually these are only typos, however they prevent automatic preprocessing of dataset. Due to this problem these mistakes need to be found and manually corrected. Also there is a small number of ground truth XML file with fully missing data. Despite these shortcomings this dataset is useful because of the bitmap files. This allows to compare results of both images affected by shooting conditions and images dependent only on font and position.

3.3 Wien TU dataset

Chapter 4

Testing methods

4.1 EasyOCR

4.2 Tesseract

Potreba cernobiele a threshold tesseract to ma radsi, problem delat centralne - otsu a po castech ale stejne tezke u spousty obrazku s ruznym osvetlenim. Tesseract zkousime u bitmap obr PSM 11 na Kaist a je o hodne horsi nez PSM 6, s 11 je to 9.6. proc. jinak u normlalnich obrazku je lepsi zase 11, taky treba o 9 procent rozdil.. U CTW datasetu je to s 6 horsi.

4.3

Bibliography

- [1] JAIDED AI. **Jaided AI - Distribute the benefits of AI to the world.** Last accessed 20 May. 2022. Available from: <https://www.jaided.ai/>.
- [2] JAIDED AI. **Jaidedai/EasyOCR.** Last accessed 13 Jun. 2022. Available from: <https://github.com/JaidedAI/EasyOCR>.
- [3] YOUNGMIN BAEK, BADO LEE, DONGYOON HAN, SANGDOO YUN, AND HWALSUK LEE. **Character region awareness for text detection.** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.
- [4] **Vision AI — derive image insights via ML — cloud vision API google cloud.** Available from: <https://cloud.google.com/vision/>.
- [5] **Kaist scene text database.** Last accessed 10 Jun. 2022. Available from: http://www.iapr-tc11.org/mediawiki/index.php/KAIST_Scene_Text_Database.
- [6] **keras-ocr documentation.** Last accessed 29 Apr. 2022. Available from: <https://keras-ocr.readthedocs.io/en/latest/index.html>.
- [7] **keras-OCR.** Last accessed 29 Apr. 2022. Available from: <https://pypi.org/project/keras-ocr/>.
- [8] YULIANG LIU, LIANWEN JIN, SHUAITAO ZHANG, CANJIE LUO, AND SHENG ZHANG. **Curved scene text detection via transverse and longitudinal sequence connection.** *Pattern Recognition*, 90:337–345, 2019.
- [9] CLOVA AI RESEARCH. **Clovaai/Craft-pytorch: Official implementation of character region awareness for text detection (CRAFT).** Last accessed 29 Apr. 2022. Available from: <https://github.com/clovaai/CRAFT-pytorch>.
- [10] **Text recognition data - Visual Geometry Group - University of Oxford.** Last accessed 2022. Available from: <https://www.robots.ox.ac.uk/~vgg/data/text/#sec-synth>.
- [11] TESSERACT-OCR. **Tesseract-OCR/tessdoc: Tesseract documentation.** Last accessed 6 May. 2022. Available from: <https://github.com/tesseract-ocr/tessdoc>.

- [12] YULIANG-LIU. **Yuliang-Liu/curve-text-detector**. Last accessed 13 Jun. 2022. Available from: <https://github.com/Yuliang-Liu/Curve-Text-Detector>.