Effectiveness of Modern Text Recognition Solutions and Tools for Common Data Sources

Kirill Smelyakov, Anastasiya Chupryna, Dmytro Darahan and Serhii Midina

Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine

Abstract

In the article features of functioning of the most common optical character recognition (OCR) tools EasyOCR and TesserOCR are considered; experimental analysis of results of work of these OCR is given for the most widespread data sources, such as electronic text document, internet resource, and banner; based on analysis of the experiment results from the comparative analysis of considered OCRs by time and accuracy was made; effective algorithm of using an OCR and recommendations for their application was offered for not-distorted data, for slightly and highly distorted data.

Keywords 1

Optical character recognition (OCR), text recognition, efficiency estimation

1. Introduction

Optical character recognition (OCR) is widely used for extracting text information from printed documents, books, posters, business cards, electronic text documents, and internet resources, and also for automation of data entry [1, 2, 3, 4].

The output of recognition is a computerized text that can be easily handled [1, 5, 6, 7]. Received text may be used in cognitive computing, text mining, and text-to-speech [8]. Such text in our time often may be used in a large number of modern and perspective IT solutions and tools [9], first of all – in artificial intelligence (AI) systems [10-12] and machine learning, in ICT, robotics-based on machine vision [13, 14, 15] and computational intelligence [16-19], etc.

There are several approaches to recognize text data from images. Core algorithms of OCR systems belong to one of two basic types: matrix matching and feature extraction with different efficiency in different situations [1, 5].

Matrix matching (pattern matching, pattern recognition, image correlation) means comparing an image with stored glyph pixel-by-pixel. Feature extraction decomposes glyphs into features, that comparing with a vector-based representation of the character. More often neural networks (NN) are used for the detection of features [10, 11]. This approach more accurate than matrix matching, so most modern OCR systems using it. Regarding using NNs in the second approach, even one algorithm could show different results depending on hyperparameters, image quality, etc. So, sometimes it's a quite serious challenge to choose the appropriate OCR for a specific data source.

The purpose of this article – to do a comparative analysis of efficiency of the most widespread OCR (EasyOCR and TesserOCR) based on results of text recognition experiments in different conditions of

1COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine
EMAIL: kyrylo.smelyakov@nure.ua (K. Smelyakov); anastasiya.chupryna@nure.ua (A. Chupryna); dmytro.darahan@nure.ua (D. Darahan); serhii.midina@nure.ua (S. Midina)
ORCID: 0000-0001-9938-5489 (K. Smelyakov); 0000-0003-0394-9900 (A. Chupryna); 0000-0003-4918-4579 (D. Darahan); 0000-0002-1518-600X (S. Midina)

data acquisition; to describe features of using considered OCRs for common data sources; to offer an algorithm and recommendations for effective practical application of EasyOCR and TesserOCR for standard conditions of use. Therefore, the experiments are focused on assessing the effectiveness of the state of the art OCR application on standard computers without special support of GPU and HPC computing.

The issues of image preprocessing are not considered, since this is a rather broad topic that requires additional research. The images are considered to have been prepared and no additional preprocessing steps are taken, besides than those OCR systems perform by default.

2. Experiment planning

Two OCR frameworks were chosen for experiments to estimate text recognition performance. The main criteria of choice were open-source code, flexible API, and using Neural Networks. The framework must be written in Python and support many languages. For testing were chosen TesserOCR (Python wrapper of Tesseract) and EasyOCR by JaidedAI [20, 21].

Currently, exists a pretty big amount of text data sources, which are interested in terms of extracting recognition of a text. These are scanned books and documents, web-resources, classical information tables from streets, advertisement and information posters, banners, scoreboards, and many others (i.e. natural scene images).

Images of these resources may differ significantly in quality in terms of distortion. In the article we took into account the most common types of artifacts: distortion, blurring, skewing, adding background, which were generated by utility Text Recognition Data Generator (TRDG) [22].

The data set used in the work was specially created for testing on target devices (standard computers); since standard datasets are not well suited for this. They take too long for processing.

In the article are considered a dataset with images of the most common categories:

- scanned books;
- images from web-pages (with both text and infographics / only text);
- photos of banners, containing text;
- photos of text from banners;
- slightly distorted photos of text from banners;
- highly distorted photos of text from banners.

In total 61 images. The sites for “images from web-pages” were randomly selected. The language of the text is English.

To create a dataset, besides using real text-photos, using free-for-all utility TRDG were generated images with given distortion level, which imitates natural scene images with low quality, may be received from outdated cameras.

There are three distortion degrees of obtained images:

- soft blurred (slightly distorted with using blurring);
- hard distorted (strongly distorted with using all the distortions mentioned above);
- clear (text without any distortions).

This notation is not generally used, and introduced by us just for clarity and simplification of the following descriptions of our experiments. Further, in the article, we will use these notations to describe image distortion levels.

Simply blurred images (i.e. “soft blurred”) were taken as images with a low degree of distortion and represent photos of text with good resolution.

Blurred images with adding background and using bit distortion (i.e. “hard distorted”) were taken as images with a high degree of distortion and represent photos with low but acceptable resolution.

Images without any distortions (i.e. “clear”) represent screenshots. Using the ternary numeral system, we can describe the level of distortion as 0 for clear images, 1 for soft blurred images, and 2 for hard distorted images.

All artifacts were applied to photos using corresponding command-line keys for TRDG on the dataset generation stage. A list of keys can be found in the user manual of the TRDG utility [22].

All data, used in our work, and also results of recognition and summary table of results are in the data warehouse [23]. Next UML-diagram (Figure 1) visualizes the file structure of the using dataset.

Both services TesserOCR and EasyOCR were tested on the same input data and hardware under one test script, measuring text recognition time for both services.

To ensure a level playing field for both neural networks, based on which the tested services work, we are not accelerating them explicatively by using such common practices as batching or using multithreading; was left default settings, given by developers. Both NNs have flexible API and wide possibilities for customization, therefore, with correctly selected hyperparameters during neural network training for a specific situation, any of them may show much better results, especially in case of using GPU and HPC, but then it will be difficult to objectively compare their productivity.

Since networks accept files in different ways, this difference was also taken into consideration. EasyOCR can accept directly most image file formats, TesserOCR by default accept only a few file formats directly when transmitting their addresses on a drive, which makes it less flexible than EasyOCR.

TesserOCR relies in work on using PIL (Python Imaging Library) for transmitting images. Actually, PIL just uploads images from drive into intermediate format. For these purposes, it can also be used OpenCV, scikit, etc. The work of two libraries in conjunction is described in the user's manual and examples from developers, which can be found on the official project page. Preliminary experiments during the phase of search of researched libraries show, that image transmitting by using PIL does not significantly affect text recognition speed and quality.

Based on this, and also on the fact that this approach is actually standard for TesserOCR, it was used in NNs testing, to put them on a level playing field. For the purity of the experiment, for TesserOCR was also counted time for using PIL functions, the cause of EasyOCR should also perform image transformation in a recognition method call.

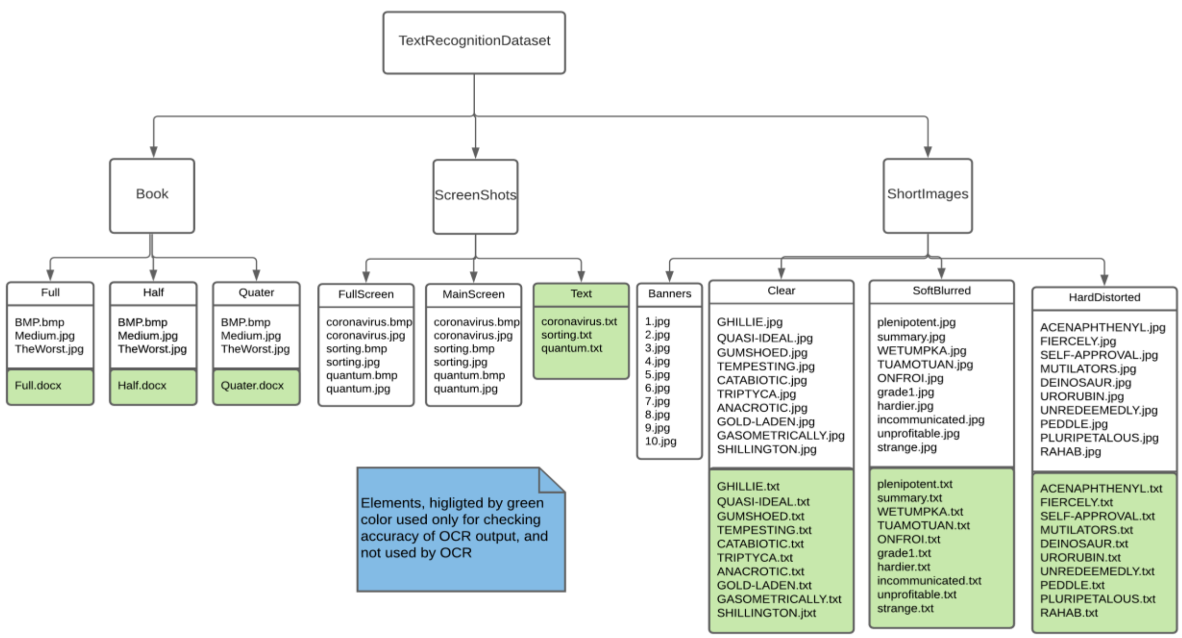

Figure 1: UML-diagram of test dataset structure

During experiment performance, test script one by one launches both NNs for recognition of every image from the dataset, marks the start and end time of text recognition, and saves time value and recognized data for each example in JSON format for further processing. For experiment purity, tests were run 3 times to reduce the possible impact of low-level processes of OS or the behavior of the hardware part of the PC. In total it is 122 recognitions per one iteration (61 per each NN), and a total of 366 recognitions was made during the experiment.

The experiment has been run on a laptop HP Elitebook 8460p with Mobile DualCore Intel Core i5-2540M CPU, frequency 3100 MHz (31 x 100) with 4 threads ant 4 Gb RAM DDR3 and Intel Sandy Bridge-MB – Integrated Graphics Controller (MB GT2 1.3GHz+) under the manage of OS Kali Linux. This computer was taken as an instance of the target device class of the experiment.

3. Experiment, results and discussion

Estimates of the results of experiments are presented in Table 1 – Table 12.

Table 1 shows recognition time for the scanned fragment of a book page in seconds, where full / half/quarter in the file name – page filling degree, high / medium / worst – image quality level.

Table 1
Recognition time of scanned fragment of a book page

Filename	Tools			
	TeserOCR		EasyOCR	
	Best Result, sec	Worst Result, sec	Best Result, sec	Worst Result, sec
full/the worst.jpg	9,75	27,07	944,14	1095,51
half/the worst.jpg	4,89	6,98	387,13	416,65
quarter/the worst.jpg	1,03	1,19	187,75	213,12
full/medium.jpg	4,98	5,09	886,79	1062,27
half/medium.jpg	2,72	2,72	341,8	383,29
quarter/medium.jpg	0,9	0,92	148,02	213,11
full/high.bmp	1,6	5,48	831,4	31980,82
half/high.bmp	3,03	3,85	374,6	375,01
quarter/high.bmp	1	1,13	155,62	177,88

Table 2 shows the probability of correct character recognition for a scanned book page fragment during a series of experiments.

Table 3 shows recognition time for screenshot in seconds, where full/main in the file name – page filling degree, quantum / coronavirus / sorting – text subject.

Table 2
Probability of correct character recognition of scanned fragment of a book page

Filename	Tools	
	TeserOCR	EasyOCR
full/the worst.jpg	0.9996	0.9906
half/the worst.jpg	1	0.9907
quarter/the worst.jpg	1	0.9926
full/medium.jpg	1	0.9902
half/medium.jpg	1	0.99
quarter/medium.jpg	1	0.9908
full/high.bmp	1	0.9909
half/high.bmp	1	0.99
quarter/high.bmp	1	0.9926

Table 3
Screenshot recognition time

Filename	Tools			
	TeserOCR		EasyOCR	
	Best Result, sec	Worst Result, sec	Best Result, sec	Worst Result, sec
full/quantum.jpg	1,69	2,07	487,94	568,1
main/quantum.jpg	1,5	1,6	234,89	281,43
full/quantum.bmp	1,62	1,85	497,94	637,37
main/quantum.bmp	1,56	1,88	230,77	301,32
full/coronavirus.jpg	1,14	1,27	260,45	319,65
main/coronavirus.jpg	1,28	1,37	222,19	271,47
full/coronavirus.bmp	1,25	1,8	260,55	304,9
main/coronavirus.bmp	1,2	1,5	216,66	282,11
full/sorting.jpg	1,12	1,27	338,76	1767,91
main/sorting.jpg	0,85	0,9	152,34	184,89
full/sorting.bmp	1,16	1,33	338,86	434,42
main/sorting.bmp	0,76	0,9	150,78	192,26

Table 4 shows the probability of correct character recognition for screenshot during a series of experiments.

Table 4
Probability of correct character recognition on screenshots

Filename	Tools	
	TeserOCR	EasyOCR
full/quantum.jpg	0,972	0,856
main/quantum.jpg	1	0,896
full/quantum.bmp	0,992	0,868
main/quantum.bmp	1	0,916
full/coronavirus.jpg	0,9903	0,8446
main/coronavirus.jpg	0,9903	0,9223
full/coronavirus.bmp	0,995	0,859
main/coronavirus.bmp	0,995	0,907
full/sorting.jpg	0,9416	0,8102
main/sorting.jpg	0,9708	0,8832
full/sorting.bmp	0,927	0,839
main/sorting.bmp	0,9708	0,839

Table 5 shows recognition time on a soft blurred photo in seconds.

Table 5
Recognition time on soft blurred photos

Filename	Tools	
	TeserOCR, sec	EasyOCR, sec
plenipotent.jpg	1,44	41,35

summary.jpg	0,95	40,23
WETUMPKA.jpg	1,02	42,2
TUAMOTUAN.jpg	0,92	47,91
ONFROI.jpg	0,82	42,3
grade1.jpg	0,88	45,72
hardier.jpg	0,8	41,4
incommunicated.jpg	0,93	49,08
unprofitable.jpg	0,81	42,48
strange.jpg	0,88	52,73

Table 6 shows the probability of correct character recognition on soft blurred photos during a series of experiments. Table 7 shows recognition time for banners in seconds. Table 8 shows the probability of correct character recognition for banners during a series of experiments. Table 9 shows recognition time on hard distorted photos in seconds. Table 10 shows the probability of correct character recognition on hard distorted photos during a series of experiments. Table 11 shows recognition time on clear photos in seconds.

Table 6
Probability of correct recognition of text characters on soft blurred photographs

Filename	Tools	
	TeserOCR	EasyOCR
plenipotent.jpg	1	0,9947
summary.jpg	0,9896	0,9896
WETUMPKA.jpg	0,9952	0,9858
TUAMOTUAN.jpg	0,9913	1
ONFROI.jpg	1	1
grade1.jpg	1	0,9905
hardier.jpg	0,9902	0,9854
incommunicated.jpg	1	0,9914
unprofitable.jpg	1	1
strange.jpg	0,9952	1

Table 7
Time to recognize text on banners

Filename	Tools	
	TeserOCR, sec	EasyOCR, sec
1.jpeg	0,17	3,79
2.jpg	0,18	9,11
3.jpg	0,18	2,71
4.jpg	0,18	31,5
5.jpg	0,17	10,03
6.jpg	0,2	7,32
7.jpg	0,18	8,13
8.jpg	0,19	5,68
9.jpg	0,17	18,23
10.jpg	0,24	10,03

Table 8
Probability of correct recognition of text characters on banners

Filename	Tools	
	TeserOCR	EasyOCR
1.jpeg	0	0,0909
2.jpg	0	0,3888
3.jpg	0	0
4.jpg	0	0,3625
5.jpg	0	0,0263
6.jpg	0	0,2142
7.jpg	0	0,4166
8.jpg	0	0,28
9.jpg	0	0,1891
10.jpg	0,50	1

Table 9
Recognition time on hard distorted photos

Filename	Tools	
	TeserOCR, sec	EasyOCR, sec
ACENAPHTHENYL.jpg	0,76	54,11
FIERCELY.jpg	0,83	44
SELF-APPROVAL.jpg	0,25	66,26
MUTILATORS.jpg	0,22	52,06
DEINOSAUR.jpg	0,61	49,89
URORUBIN.jpg	0,69	18,15
UNREDEEMEDLY.jpg	0,68	2,92
PEDDLE.jpg	0,81	61,2
PLURIPETALOUS.jpg	0,6	48,15
RAHAB.jpg	0,26	54,99

Table 10
Probability of correct recognition of text characters on highly distorted photographs

Filename	Tools	
	TeserOCR	EasyOCR
ACENAPHTHENYL.jpg	1	0,9957
FIERCELY.jpg	1	1
SELF-APPROVAL.jpg	0	0,9683
MUTILATORS.jpg	0	0,9904
DEINOSAUR.jpg	0,9357	1
URORUBIN.jpg	0	0
UNREDEEMEDLY.jpg	0	0
PEDDLE.jpg	0	0,1845
PLURIPETALOUS.jpg	0,9018	0,9953
RAHAB.jpg	0,639	0,7441

Table 11
Recognition time on clear photos

Filename	Tools	
	TeserOCR, sec	EasyOCR, sec
GHILLIE.jpg	0,67	69,43
QUASI-IDEAL.jpg	0,91	49,38
GUMSHOED.jpg	0,87	48,96
TEMPESTING.jpg	1,02	48,83
CATABIOTIC.jpg	0,86	48,83
TRIPTYCA.jpg	0,83	47,56
ANACROTIC.jpg	0,89	48,55
GOLD-LADEN.jpg	0,57	30,6
GASOMETRICALLY.jpg	0,72	79,08
SHILLINGTON.jpg	0,68	34,26

Table 12 shows the probability of correct character recognition on clear photos during a series of experiments.

Table 12
Probability of correct recognition of text characters on clear photos

Filename	Tools	
	TeserOCR	EasyOCR
GHILLIE.jpg	0,9955	0,9955
QUASI-IDEAL.jpg	1	0,9953
GUMSHOED.jpg	1	1
TEMPESTING.jpg	1	0,9902
CATABIOTIC.jpg	1	1
TRIPTYCA.jpg	0,9946	1
ANACROTIC.jpg	0,9953	1
GOLD-LADEN.jpg	0,9797	0,7538
GASOMETRICALLY.jpg	0,9916	0,9957
SHILLINGTON.jpg	0,9892	1

We have found that TesserOCR much faster than EasyOCR in all test cases. Also, TesserOCR shows quite better accuracy in most cases, but for the recognition of small hard distorted images, EasyOCR shows better accuracy. So, for images of average big plain text, such as book pages the best OCR is TesserOCR, but for short texts with a lot of noise speeded up EasyOCR may be preferred.

Was found a direct relationship between the accuracy of recognition within one format and a time of recognition, and also direct dependence on the quality of recognizable image, regardless of file format. With lower quality, recognition is faster, at the same time, the inverse dependency of the recognition quality on the image quality is weakly expressed and does not always manifest itself. With different image formats, the BMP format often falls out of the general pattern, and corresponds to the average quality of the JPG format, slightly inferior to it in time, but superior in quality. In fact, the file format does not directly affect the recognition quality, since images are usually converted to an intermediate format, but the quality of the image stored on the disk directly affects the recognition quality.

During the control experiments, in the same conditions, and on the same inputs, was detected spread of time values between tests, most likely due not to functioning features of computer hardware and software, but due to features of algorithms of OCR. Regardless of differences of received results, results of all tests a grouped around mean, the spread relative to which is the greater, the higher the quality, and the larger the volume of the recognized text. For each test on EasyOCR, the spread is much larger than for TesserOCR.

TesserOCR is much faster than EasyOCR and better recognize regular book text, but if natural scene images are taken, then EasyOCR works better. Of course, EasyOCR is still inferior to the opponent in time, but not in quality, recognizing text fully or partially, where TesserOCR doesn't recognize anything. In Table 13 given superiority rates for TesserOCR and EasyOCR by recognition time and quality and symbols per second ratio. Constructing this table, we calculated average indicators of time and accuracy by given according to the tables above and divided corresponding indicators for EasyOCR into indicators for TeserOCR. Their quality (accuracy) is expressed by the ratio of correct recognition probability. Our calculations may be expressed by formula

$$R = \frac{\sum_{i=1}^{n} e}{n} \Big/ \frac{\sum_{i=1}^{n} t}{n}, \quad (1)$$

where R – superiority rate for corresponding indicators; n – numbers of tests; e – indicators, obtained in experiments for EasyOCR; t – indicators, obtained in experiments for TesserOCR. Figures 2 and 3 provide a visualized presentation of obtained superiority rates. Figure 4 visualizes the superiority rate for TesserOCR and EasyOCR by symbols per second ratio.

Table 13
Superiority rate of work of TeserOCR and EasyOCR by time and quality

Data	Tools		
	Runtime	Quality	Symbols per second
Books	374,41	0,9908	160,12
Screenshots	273,83	0,8951	257,72
SoftBlurred	47,13	0,9975	unknown
Banners	57.27	5,936	unknown
HardDisorted	79,11	1,7631	unknown
Clear	56,6	0,9783	unknown

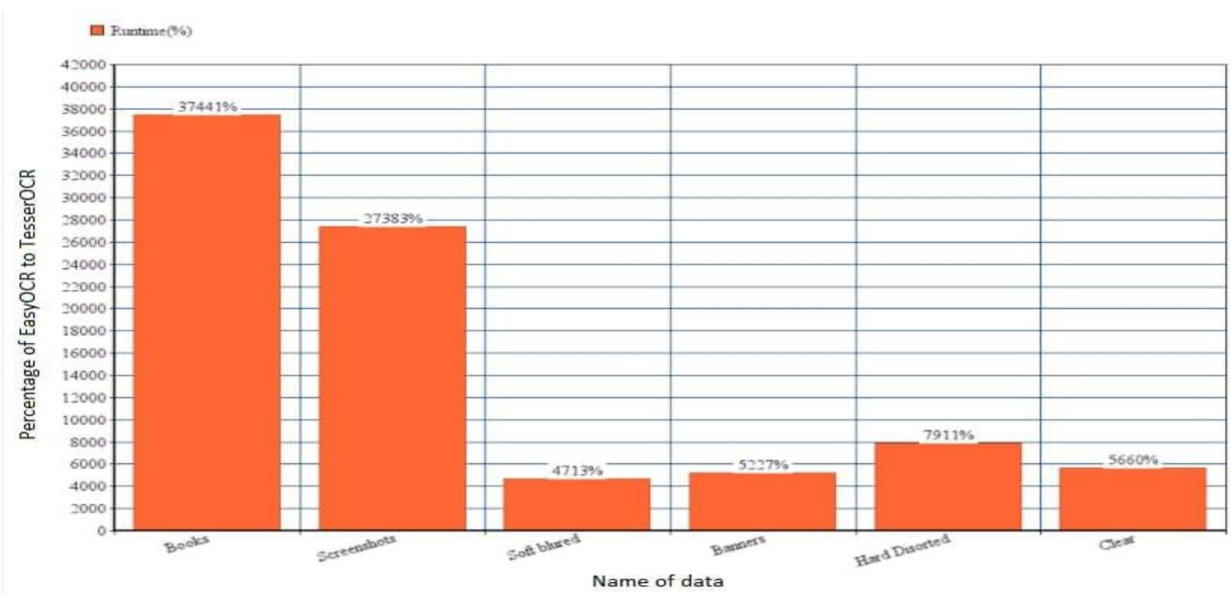

Figure 2: Runtime superiority rate

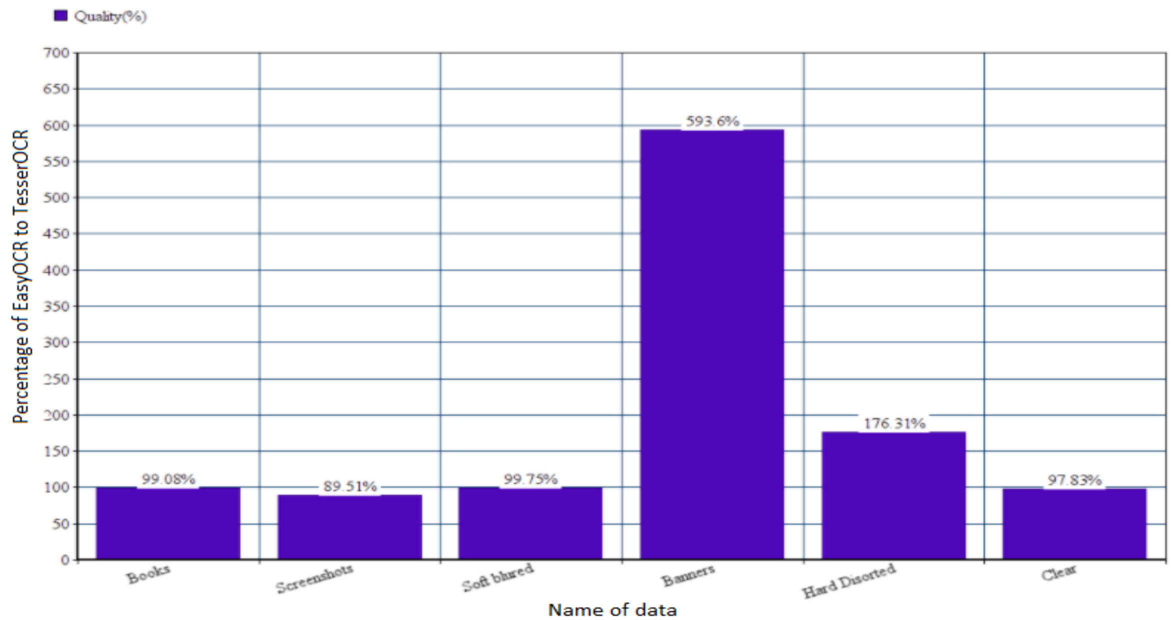

Figure 3: Quality superiority rate

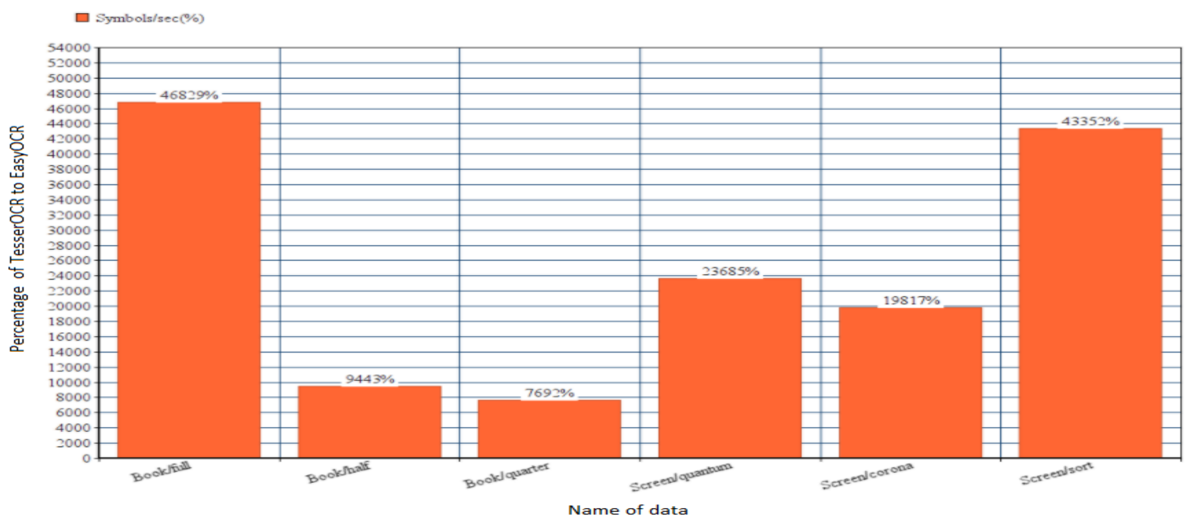

Figure 4: Symbols per second superiority rate

4. Algorithm and recommendations

Analyzing obtained experimental results it is possible to formulate algorithms and recommendations for practical use of reviewed text recognition tools. The common algorithm of text recognition consists of 2 stages: 1) preprocessing and 2) recognition of prepared text. In the preprocessing stage, it is advisable to convert images into bitmaps before submitting them to the neural network, e.g. using PIL (Python Imaging Library) or any other appropriate image processing library (OpenCV, scikit, etc), if the image was originally not in format BMP. It will give the most balanced ratio of time to quality among other formats and also will provide more effective work with images. Converting into bitmap gives unification of image processing code in the final application, besides, in some situations, bitmap capture from a screen is the fastest way of image feed. Nevertheless, if the priority parameter of recognition is speed, then it is advisable to convert images into lighter formats with lower quality. In this case, the quality of recognition without preliminary retraining or hyperparameter tuning will dropdown. For effective use of both frameworks, it is necessary to do hyperparameter tuning, to train a neural network for specific input data types, and also to convert images into bitmaps before recognition. In this case, performance will be much better, than the one described as a result of the experiment. According to experiment results, the TesserOCR library can be recommended for recognition of scanned books, screenshots, soft blurred and clear photos. While the EasyOCR library can be recommended for recognition of advertisements, banners, and hard distorted photos (i.e. natural scene images). Based on the results of the analysis it should be noted that nowadays the EasyOCR library is actively developed and in the coming years by certain parameters can surpass TesserOCR, especially for the conditions of neural network optimization and overclocking.

5. Conclusions

In the work was set up the experiment and made a comparative analysis of the performance of the most common OCRs (TesserOCR and EasyOCR) by time and by the accuracy of recognition in equals conditions (with default configuration). Described application features of both services for different data sources, which can appear during the real use of neural networks. Received results describe only neural network performance without training stage and give only a general understanding of the functioning of each of them and about their field of application at the time of writing of the article. A complete analysis of the capabilities of each NN is needed deeper researches, which, however, was not the aim of this work. They imply serious modification and optimization of the code, obtained just after installation, programming on the lower level, retraining of NN, provided by developers or image preprocessing. According to the analysis of the experimental results, proposed algorithms and recommendations for effective practical application of reviewed OCRs. Besides, it is also recommended to do configuration and training of NN for each specific case (data source) for obtaining the best performance.

It is also important to remember that the results of the experiments carried out are relevant only to standard computer without the support of GPU and HPC calculations. This is done specifically to evaluate the performance of real applications that will run on the same standard computers.

6. References

[1] What is OCR and OCR Technology, 2020. URL: https://pdf.abbyy.com/learning-center/what-is-ocr.

[2] D. Berchmans and S. S. Kumar, ”Optical character recognition: An overview and an insight”, in: Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, 2014, pp. 1361-1365. doi: 10.1109/ICCICCT.2014.6993174.

[3] A. M. Sabu and A. S. Das, ”A Survey on various Optical Character Recognition Techniques”, in: Proceedings of the 2018 Conference on Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, 2018, pp. 152-155. doi: 10.1109/ICEDSS.2018.8544323.

[4] M. Rahman Majumder, B. Uddin Mahmud, B. Jahan and M. Alam, ”Offline optical character recognition (OCR) method: An effective method for scanned documents”, in: Proceedings of the 2019 22nd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2019, pp. 1-5. doi: 10.1109/ICCIT48885.2019.9038593.

[5] What is OCR? An Introduction to Optical Character Recognition, 2020. URL: https://anyline.com/news/what-is-ocr.

[6] M. Brisinello, R. Grbić, D. Stefanovič and R. Pečkai-Kovač, ”Optical Character Recognition on images with colorful background”, in: Proceedings of the 2018 IEEE 8th International Conference on Consumer Electronics, Berlin, 2018, pp. 1-6. doi: 10.1109/ICCE-Berlin.2018.8576202.

[7] An Introduction to OCR for Beginners, 2020. URL: https://towardsdatascience.com/an-introduction-to-optical-character-recognition-for-beginners-14268c99d60.
[8] What is Cognitive Computing? How are Enterprises benefitting from Cognitive Technology, 2019. URL: https://towardsdatascience.com/what-is-cognitive-computing-how-are-enterprises-benefitting-from-cognitive-technology-6441d0c9067b.
[9] Kyrychenko I., Gruzdo I., Tereshchenko G., Cherednichenko O. Applıcatıon of Paragraphs Vectors Model for Semantıc Text Analysıs, in: Proceedings of the 4th International Conference on Computational Linguistics and In-telligent Systems (COLINS-2020), Lviv, Ukraine, April 23-24, 2020. – Volume I, pp. 283-293.
[10] Ian Goodfellow, Yoshua Bengio, Aaron Courville Deep Learning, 3rd. ed., MIT Press, 2016, 787p. DOI/ISBN: 0262035618.
[11] Peter Norvig , Stuart Russell Artificial Intelligence: A Modern Approach, 3rd. ed., Pearson Education Limited, 2016, 1152p. DOI/ISBN: 9780136042594.
[12] Kyrychenko I., Proniuk G., Geseleva N., Tereshchenko G. Spatial interpretation of the notion of relation and its application in the sys-tem of artificial intelligence, in: Proceedings of the 3rd International Conference on Computational Linguistics and In-telligent Systems (COLINS-2019), Kharkiv, Ukraine, April 18-19, 2019. – pp. 266-276.
[13] Rafael C. Gonzalez, Richard E. Woods Digital Image Processing, 4th. ed., Pearson/Prentice Hall, 2018, 1168p. DOI/ISBN: 9780133356724.
[14] Chupryna A., Smelyakov K., Hvozdiev M., Sandrkin D. Gradational Correction Models Efficiency Analysis of Low-Light Digital Image, in: Proceedings of the 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), 25 April 2019, Vilnius, Lithuania. – P. 1-6. doi: 10.1109/eStream.2019.8732174.
[15] Datsenko A., Skrypka V., Akhundov A., Smelyakov K. Efficiency of Image Reduction Algorithms with Small-Sized and Linear Details, in: Proceedings of the 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 8-11 Oct. 2019, Kyiv, Ukraine. – P. 745-750. doi: 10.1109/PICST47496.2019.9061250.
[16] Sharonova, N., Kanishcheva, O., Image and video tag aggregation // CEUR Workshop Proceedings, Vol. 2268, 2017, pp. 161-172, 2017.
[17] Kanishcheva, O.. Cherednichenko, O., Sharonova, N. Image Tag Core Generation // Proceedings of the 1st International Workshop on n Digital Content & Smart Multimedia (DCSMart 2019). Lviv, Ukraine, December 23-25, 2019, pp. 35-44.
[18] Cherednichenko, O., Yanholenko, O., Vovk, M., Sharonova, N. Towards structuring of electronic marketplaces contents: Items normalization technology // CEUR Workshop Proceedings. Vol. 2604, 2020, P. 44-55.
[19] Smelyakov K., Yeremenko D., Sakhon A., Polezhai V., Chupryna A. Braille character recognition based on neural networks, in: Proceedings of the IEEE Second International Conference on Data Stream Mining & Processing (DSMP), August 21-25. – 2018. – P. 509-513.
[20] Tesserocr, 2020. URL: https://github.com/sirfz/tesserocr.
[21] EasyOCR, 2021. URL: https://www.jaided.ai/easyocr.
[22] TextRecognitionDataGenerator, 2020. URL: https://github.com/Belval/TextRecognitionDataGenerator.
[23] Data storage with our data and results, 2021. URL: https://drive.google.com/drive/folders/1dy2gEDBf38t_TY1933g57JyDJ2J4MmWA?usp=sharing.