Hybrid Training Data for Historical Text OCR

Jiří Martínek*†, Ladislav Lenc*†, Pavel Král*†, and Anguelos Nicolaou‡, and Vincent Christlein‡
* Dept. of Computer Science & Engineering
University of West Bohemia
Plzeň, Czech Republic
† NTIS - New Technologies for the Information Society
University of West Bohemia
Plzeň, Czech Republic
{jimar,llenc,pkral}@kiv.zcu.cz
‡ Pattern Recognition Lab
Friedrich-Alexander University Erlangen-Nrnberg
{anguelos.nikolaou,vincent.christlein}@fau.de

***Abstract*—Current optical character recognition (OCR) systems commonly make use of recurrent neural networks (RNN) that process whole text lines. Such systems avoid the task of character segmentation necessary for character-based approaches. A disadvantage of this approach is a need of a large amount of annotated data. This can be solved by using generated synthetic data instead of costly manually annotated ones. Unfortunately, such data is often not suitable for historical documents particularly for quality reasons.**

This work presents a hybrid approach for generating annotated data for OCR at a low cost. We first collect a small dataset of isolated characters from historical document images. Then, we generate historical looking text lines from the generated characters.

Another contribution lies in the design and implementation of an OCR system based on a convolutional-LSTM network. We first pre-train this system on hybrid data. Afterwards, the network is fine-tuned with real printed text lines. We demonstrate that this training strategy is efficient for obtaining state-of-the-art results. We also show that the score of the proposed system is comparable or even better in comparison to several state-of-the-art systems.

***Index Terms*—CNN, Hybrid Training, Historical Documents, LSTM, Neural Networks, OCR**

I. Introduction

Optical character recognition (OCR) of historical documents is a very challenging task, because the scans are often affected by noise, deformations, and artifacts such as sparkles, translucents, etc.

Most current state-of-the-art OCR systems are based on recurrent neural networks (RNNs) which use whole unsegmented text lines as an input [1]. It has been also demonstrated that a pipeline of convolutional neural networks (CNNs) and long short-term memory networks (LSTM) brings even better OCR scores [2]. These approaches are significantly superior to the previous ones which usually processed characters independently [3] and corrected the errors with a subsequent language model [4]. Another important benefit of the RNN based approaches is that the segmentation to characters is not necessary. On the other hand, this complex topology needs a significant amount of training data. Manual annotation is a very time consuming and expensive task, however synthetic data are relatively easy to obtain.

This work proposes a novel hybrid approach for creation of annotated OCR data at a low cost. This method is composed of two steps: 1) a small dataset of isolated characters is collected from historical document images; 2) historical looking text lines are composed from these characters.

Another contribution is the determination of optimal conditions for training of a CNN-LSTM OCR system with synthetic data and a limited number of real annotated text lines in order to achieve state-of-the-art OCR results. First, the network is trained on a large amount of synthetic data. These data should be sufficient to initialize the implicit statistical language model [5] and to provide a basic glyph representation. For the step stage, we use a relatively small dataset of real text lines for additional training, which fine-tunes the classifier. It is also worth noting that the created corpus as well as the tool for generating the hybrid data are freely available for research purposes[1].

The rest of the paper is organized as follows. After presenting the related work in Section II, we present the proposed classifier. Section IV briefly describes our proposed data generation approaches. After the detailed evaluation of its effect in Section VI, the paper is concluded in the last section.

II. Related Work

OCR methods are nowadays mostly based on recurrent neural networks. The systems are trained end-to-end and are able to process unsegmented text lines. A seminal work on using RNNs for text recognition was written by Graves and Schmidhuber [6] They introduced a globally trained handwriting recognizer that takes raw pixel data as an input. It can be used unchanged for

This work has been partly supported by Cross-border Cooperation Program Czech Republic - Free State of Bavaria ETS Objective 2014-2020 (project no. 211), ERDF "Research and Development of Intelligent Components of Advanced Technologies for the Pilsen Metropolitan Area (InteCom)" (no.: CZ.02.1.01/0.0/0.0/17_048/0007267) and by Grant No. SGS-2019-018 Processing of heterogeneous data and its specialized applications.

[1] http://ocr-corpus.kiv.zcu.cz/

any language. A crucial part of the network is a connectionist temporal classification (CTC) loss function [7] that allows recognizing unsegmented sequences.

Another OCR approach based on recurrent neural networks was proposed by Breuel [8]. The author combined CNNs with an LSTM network and analysed the results of combinations of several network architectures and line normalization approaches. He demonstrated that hybrid CNN with 1D LSTM outperforms LSTM alone. Using 2D LSTM network instead of a 1D one did not increase the performance. Another observation by Breuel et al. [9] was that line normalization is beneficial also for CNN-LSTM networks.

OCR methods utilizing RNNs require a significant amount of annotated data. The best way to obtain such data is the annotation of the real-world examples. However, it is a costly process and thus synthetic data are utilized in many cases.

Jaderberg et al. [10] discussed methods for synthetic data generation for natural scene text recognition. They generated images with three layers: background, foreground and an optional border / shadow layer. The fonts were randomly selected from a large catalogue to ensure variability. Noise was also added so that the images were more realistic.

Creating synthetic documents without scanning for Arabic OCR was introduced by Margner et al. [11]. After typesetting Arabic pages, the bitmap representation and corresponding ground truth was generated and it was used as an OCR database. Another synthetic data generation approach was described by Gaur et al. [12]. It aimed at handwritten Indian texts, which were created from fonts that are similar to handwriting. Various distortions were applied to enhance the script appearance.

Simistira et al. [1] utilized LSTMs for processing of historical Greek polytonic scripts. They generated synthetic data using old Greek fonts and the Ocropus line generation utility. The training data comprised the synthetic data complemented with a small amount of the real data. The character error rates (CER) reported on such data lies between $5.5\,\%$ and $14.7\,\%$. The authors have shown that without the use of synthetic data the results were much worse.

A semi-automatic approach for data annotation was described by Clausner et al. [13]. They utilized Aletheia, a tool capable of preparing ground truth data for layout analysis of documents images. The annotations were propagated from larger text regions to single glyphs. The bounding polygons estimated by the tool were manually corrected if needed. The data were prepared for the Gamera toolkit, but it can be used directly in the other OCR systems.

The whole processing pipeline for historical documents is implemented in several state-of-the-art systems. Tesseract [14] is a well-known open-source OCR system that achieves excellent results. ABBYY FineReader[2] is a popular commercial tool for OCR. It is used as a backend in Transkribus[3] [15], [16] which aims at processing of historical documents. Another state-of-the art system is OCRopus [17].

[2]https://www.abbyy.com/en-ee/finereader/

[3]https://transkribus.eu/Transkribus/

III. Convolutional Recurrent Neural Network Classifier

Our classifier is a combination of a convolutional neural network and a recurrent neural network and shares similarities with the work of Shi et al. [2]. We use the connectionist temporal classification (CTC) loss as proposed by Graves et al. [7]. Binarized images are used as an input to the CNN and this network is used for feature extraction. These vectors are passed to the RNN, which uses LSTM cells [18]. Following Graves et al. [19], we use a bidirectional LSTM architecture.

The output of the LSTM is given to a set of dense layers with a softmax activation function. Its output represents a probability distribution of characters per each time frame. Let $\mathcal{A}$ be a set of symbols that the classifier recognizes ($|\mathcal{A}| = 83$). Then $p_t^{a_i}$ is a probability of observing the character a_i at a given time t. At each time t the sum of the probabilities of all symbols is equal to 1.

$$\sum_{i=1}^{|\mathcal{A}|} p_t^{a_i} = 1 \tag{1}$$

The most probable symbol $\hat{a}_t$ of each time frame t is then determined as:

$$\hat{a}_t = \underset{a_i \in \mathcal{A}}{\operatorname{argmax}}\, p_t^{a_i} \tag{2}$$

The last part of the classifier is a transcription layer, which decodes the predictions for each frame into the output sequence. To be able to distinguish each individual characters the blank-symbol (-) is inserted. It is also necessary to remove any duplicates in the sequence. The architecture of the classifier is depicted in Figure 1.

IV. Synthetic Data Generation

The processed documents originate from the second half of the nineteenth century. The utilized script is German Fraktur, which can be easily generated using a modern text processor. However, attention must be paid to the different appearance in comparison to historical printings. Another important issue is the implicit language model learned by the neural network. It has been shown [5] that the implicitly learned model can significantly improve the accuracy. Therefore, the texts used for data generation must conform to the language of that time, which differs significantly from present-day German.

To handle the above mentioned issues we propose a novel hybrid approach for synthetic data generation. As a text source, we use historical German corpora [20] from which we picked 25 000 sentences for text line generation.

To obtain the glyph images, we have employed a semi-automatic data annotation method. We have chosen a small amount of automatically extracted text line images (one page, 130 lines) that were fed to a simple, projection profile based character segmenter. The output of the segmenter was presented to a human annotator who corrected possible faulty segmentation and annotated the characters. The program then extracted the isolated characters images. In this way, we have obtained several examples of each character together with an initial set of line images with ground truths.

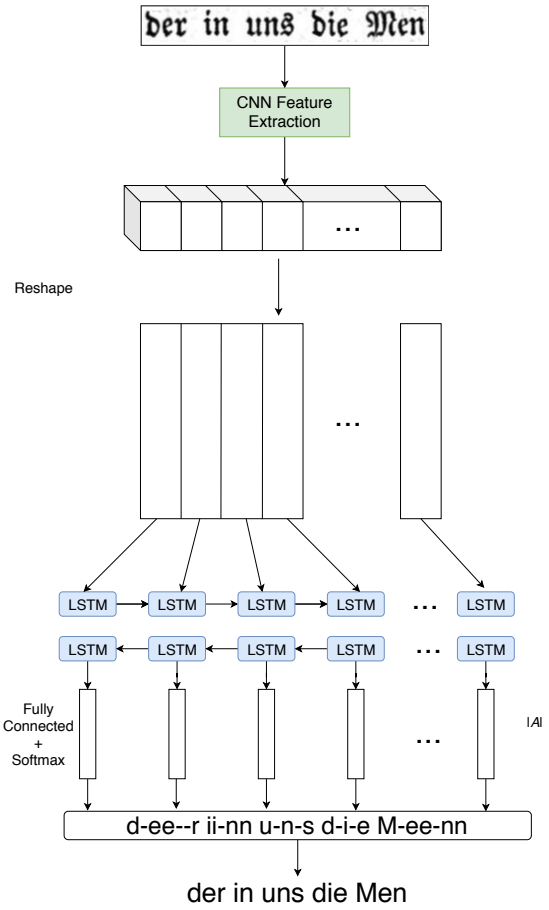

Fig. 1. CRNN classifier architecture

The character images are used to compose line images according to the prepared sentences from the historical corpora. Each character of the text line is picked randomly from available samples. The glyphs are not necessarily aligned to the image center, which ensures a diversity of the synthetic dataset. We propose three approaches to handle spacing between the images:

1) Constant space (CS) - fixed sized gap of 4 pixels.
2) Random space (RS) - random value between 1 and 5 pixels.
3) Precise space (PS) - a value computed from the annotated page[4].

Figure 2 shows an example line images generated by the three approaches.

For comparison, we also utilize a dataset generated by the standard tool TextRecognitionDataGenerator[5]. This tool has many parameters that influence generated images. We created

[4]During the annotation process, we automatically measured the gap between the each pair of the characters available and we use this value as a base to a gap size. If a given character pair was not seen before, a random space is used.

[5]https://github.com/Belval/TextRecognitionDataGenerator

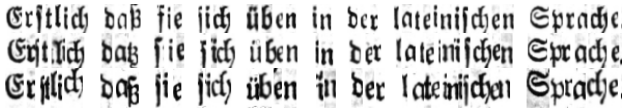

Fig. 2. Examples of hybrid data generated by the proposed approaches: constant space (CS), random space (RS), precise space (PS) - top to bottom

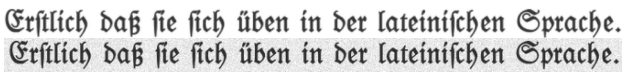

Fig. 3. Examples of generated data by TextRecognitionDataGenerator: white background (G) - top, background with Gaussian noise (GN) - bottom

two variants: The first one is a clear text on white background (G) while the second one has added Gaussian noise to the background (GN). The mean value is 235 and the standard deviation is equal to 10. Figure 3 shows the two variants of a line image generated using this tool.

All data described in this section (the three proposed hybrid approaches and the two reference text generators) will be referred to as synthetic data.

V. Experimental Setup

A. Real Dataset

This dataset serves mainly for evaluation of the proposed approaches. It consists of ten manually annotated pages of a historical German newspaper from 1866. The line images are automatically extracted from the scanned pages. We developed a simple text segmentation method, which utilizes pre-processing methods provided by open-source tools Leptonica[6] and opencv[7]. The height of the line images is approximately 40 pixels, while the width is variable with respect to the text content. All images are binarized using Otsu's method [21]. Figure 4 shows three line image examples from this dataset.

The total of 1368 manually annotated line images are split to three non-overlapping sets. Two pages serve as a test set, one is used for validation and the remaining seven pages are used for training. Note, that this corpus as well as the hybrid data generator are freely available for research purposes at[1].

B. Evaluation Metrics

For evaluation we use standard word error rate (WER) and character error rate (CER) metrics averaged over all lines. Additionally, we employ the average edit (AE) distance, also known as the Levenshtein distance. The accuracy (ACC) metric expresses the success of perfectly recognized lines.

[6]http://www.leptonica.com/
[7]https://opencv.org/

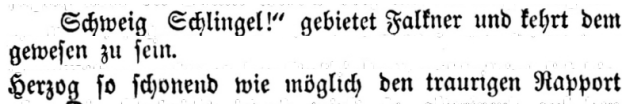

Fig. 4. Three line examples from the real dataset which are annotated manually

C. Network Settings

The network has two convolutional layers containing 40 kernels of the size 3×3. Each convolutional layer is followed by a max-pooling layer with 2×2 filters and stride 2. The output of the CNN is reshaped and fed to a fully connected layer with 128 nodes. Then there are two bi-directional LSTM layers with 256 cells. The final dense layers conform to the number of time frames and the size of the alphabet, which is 120 values.

VI. Evaluation

This section presents the experiments we performed to show that our network is able to learn from the available data.

A. Training on Real Data Only

This experiment analyzes whether it is possible to train our model from scratch solely on the real data we have at our disposal. We learn the models on the training set (7 pages) and evaluate it on the validation set (one page).

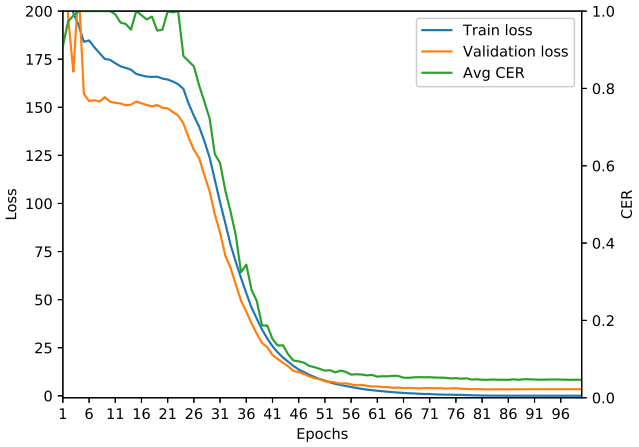

Fig. 5. Training from scratch using only the 7 pages of real data

Figure 5 shows that this training is very slow with respect to the number of epochs (X-axis). (For example, compare the graphs for synthetic data in Figure 6.) However, the network is still able to learn from this amount of data. The best obtained CER is $4.6\,\%$.

B. Training on Synthetic Data

The goal of the second set of experiments is to examine the behaviour of the network when trained only on different kinds of synthetic data. In all cases we train the network only on particular synthetic data and evaluate it, as previously, on the real validation set. The X-axis in all figures in this experiment represents the number of epochs.

Figure 6 shows the training progress on the five synthetic data types (three ones generated by the proposed hybrid approaches and two remaining ones by TextRecognitionDataGenerator - see Section IV for details). We can observe that the training loss decreases very rapidly in all cases. This experiment

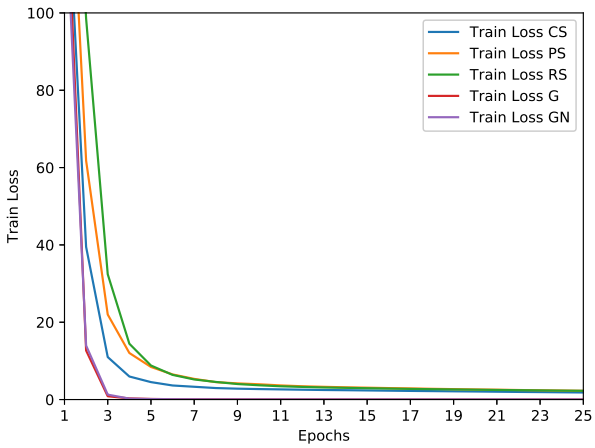

Fig. 6. Training progress on different kinds of synthetic data

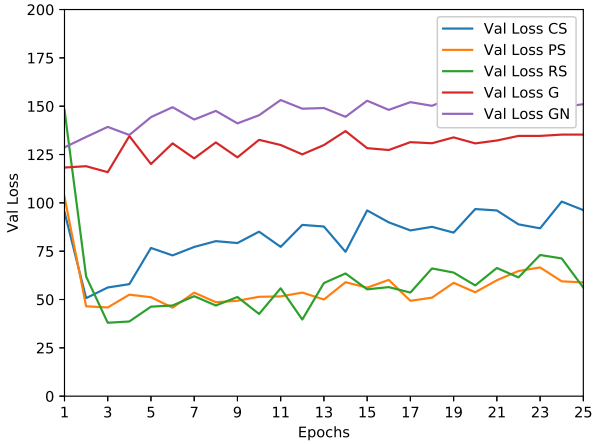

Fig. 7. Validation loss curves on different kinds of synthetic data

further shows that the training on generated data using TextRecognitionDataGenerator (G and GN) is faster and the loss value is lower than for hybrid data by the proposed methods (CS, RS, PS). Only a few epochs are needed to train the models.

The second important aspect is how the models trained on synthetic data can generalize to the real data from the validation set. Therefore, Figure 7 analyses the dependency of the validation loss on the different number of training epochs using different synthetic training data. This figure shows that validation loss begins to rise relatively early indicating that further training brings no further improvements. Therefore we set the number of epochs to five which is a compromise for all data types.

We further compare the different synthetic data generation approaches from the point of view character recognition accuracy on real validation data. The results of this experiments are depicted in Figure 8). This figure shows that the models trained on the data created by our hybrid generation technique (CS, RS,

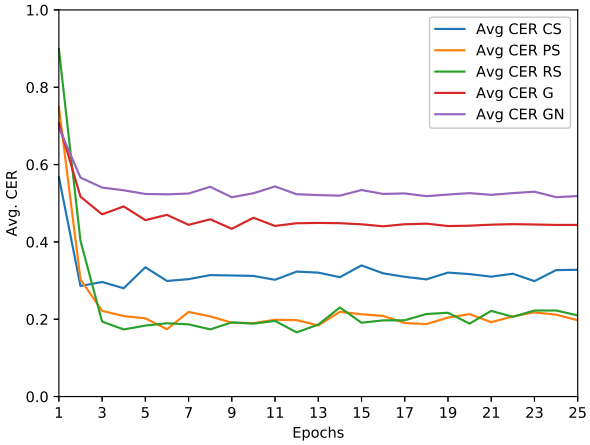

Fig. 8. Average CER curves on different kinds of synthetic data

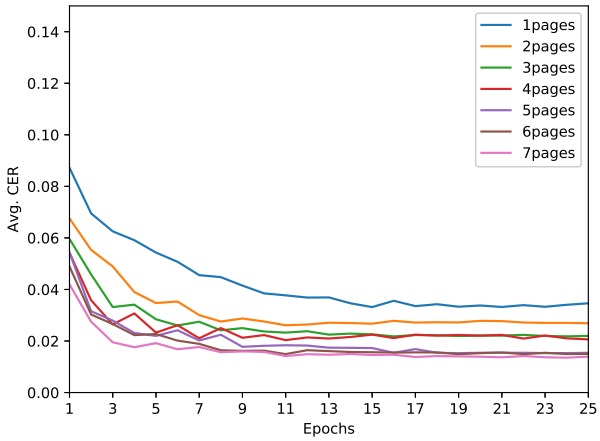

Fig. 9. Additional training with varying number of pages, the model was pre-trained on PS synthetic data.

PS) perform significantly better than the the models learned on the data generated by the standard tool (G, GN). The best obtained CER value is around $18\,\%$ for PS and RS while the best CER for generated data by TextRecognitionDataGenerator is only $46\,\%$. This figure also shows that the variable sized gaps (RS and PS) perform better than the constant ones. We will thus consider only these types of data in the following experiments.

C. Fine-tuning of the CRNN

Our classifier is trained in two stages. This experiment aims at showing how additional training with a small amount of real data influences the overall performance of the model. Based on the previous experiment, we use the data with variable sized gaps (RS and PS) for the pre-training. Five epochs of pre-training are used in all cases in accordance with our previous experiment. For additional training, we use the training set of our real corpus. The models are evaluated on the validation set.

First, we examine the influence of the amount of real data used for fine-tuning the network. Thus, we train the model with varying amount of data (1 to 7 pages from the training data). Figures 9 and 10 show the influence of the amount of additional training data on the overall performance of the models pre-trained on RS and PS synthetic data.

The curves in both figures show similar behaviour and bring very similar results. We can thus conclude that both PS and RS data generation methods are comparable. Moreover, we confirm that it is necessary to have a sufficient amount of real training data. The training on only one page has clearly the worst performance with CER around $4\,\%$. On the other hand the differences are not significant when using 5 to 7 pages for additional training. We can also conclude that using the minimal amount of 5 pages of real data for training brings CER below $2\,\%$ on validation data.

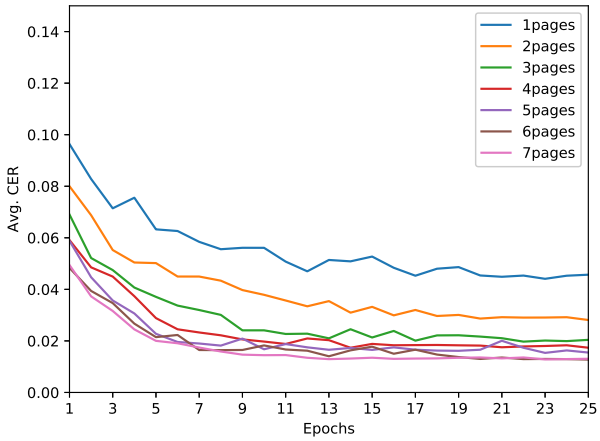

Fig. 10. Additional training with varying number of pages, the model was pre-trained on RS synthetic data.

D. Comparison with Other OCR Systems

We compare the proposed approach with Tesseract [14], Transkribus [15], [16] and OCRopus [17]. In the case of Tesseract, we used two types of tessdata: 1) *deu_frak.traineddata* - Tess[a] and 2) *Fraktur.traineddata* - Tess[b]. Both of them are trained on Fraktur skript. However, the data show considerable differences and thus we report results for both. We ran all OCR Systems on whole pages (Table II) as well as on extracted lines (Table I). We compare our system with both types of approaches. The reported results of our system are the same in both tables as we used our segmentation algorithm to extract the utilized line images. The significantly worse results reached by OCRopus on whole pages are caused mainly by the incorrect segmentation. It missed several lines completely which results in high values of the edit distance.

TABLE I
RESULTS OF DIFFERENT OCR SYSTEMS ON EXTRACTED LINES

	CRNN	Tess[a]	Tess[b]	Transkribus	OCRopus
ACC	**0.465**	0.295	0.185	0.242	0.179
Avg. ED	**1.276**	1.792	3.433	2.237	2.657
Avg. WER	**0.124**	0.165	0.257	0.184	0.248
Avg. CER	**0.028**	0.037	0.073	0.049	0.059

TABLE II
RESULTS OF OCR SYSTEMS ON WHOLE PAGES

	CRNN	Tess[a]	Tess[b]	Transkribus	OCRopus
ACC	**0.465**	0.343	0.241	0.328	0.212
Avg. ED	**1.276**	1.595	2.401	1.420	10.511
Avg. WER	**0.124**	0.157	0.188	0.153	0.362
Avg. CER	**0.028**	0.037	0.056	0.034	0.207

VII. CONCLUSIONS

In this paper, we have proposed an efficient hybrid approach for generating annotated data for historical OCR at a low cost. We have first manually collected a small dataset of isolated characters from historical document images. Then, we have generated text lines from these characters. We have further proposed and implemented an OCR system based on a CNN-LSTM network. We have firstly pre-trained this system on a large set of the the artificially generated data. Then, this network was fine-tuned using the real printed text lines.

We have shown that this learning strategy is efficient to obtain state-of-the-art OCR results. We have also demonstrated that the score of the proposed system is comparable or even better than several state-of-the-art systems.

REFERENCES

[1] F. Simistira, A. Ul-Hassan, V. Papavassiliou, B. Gatos, V. Katsouros, and M. Liwicki, "Recognition of historical greek polytonic scripts using lstm networks," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 766–770. 1, 2

[2] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017. 1, 2

[3] A. Yuan, G. Bai, L. Jiao, and Y. Liu, "Offline handwritten english character recognition based on convolutional neural network," in *2012 10th IAPR International Workshop on Document Analysis Systems*. IEEE, 2012, pp. 125–129. 1

[4] L. Zhuang, T. Bao, X. Zhu, C. Wang, and S. Naoi, "A chinese ocr spelling check approach based on statistical language models," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 5. IEEE, 2004, pp. 4727–4732. 1

[5] E. Sabir, S. Rawls, and P. Natarajan, "Implicit language model in lstm for ocr," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 7. IEEE, 2017, pp. 27–31. 1, 2

[6] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in neural information processing systems*, 2009, pp. 545–552. 1

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376. 2

[8] T. M. Breuel, "High performance text recognition using a hybrid convolutional-lstm implementation," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1. IEEE, 2017, pp. 11–16. 2

[9] T. M. Breuel, A. Ul-Hasan, M. A. Al-Azawi, and F. Shafait, "High-performance ocr for printed english and fraktur using lstm networks," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 683–687. 2

[10] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *arXiv preprint arXiv:1406.2227*, 2014. 2

[11] V. Margner and M. Pechwitz, "Synthetic data for arabic ocr system development," in *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, 2001, pp. 1159–1163. 2

[12] S. Gaur, S. Sonkar, and P. P. Roy, "Generation of synthetic training data for handwritten indic script recognition," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 491–495. 2

[13] C. Clausner, S. Pletschacher, and A. Antonacopoulos, "Efficient ocr training data generation with aletheia," *Proceedings of the International Association for Pattern Recognition (IAPR), Tours, France*, pp. 7–10, 2014. 2

[14] R. Smith, "An overview of the tesseract ocr engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2. IEEE, 2007, pp. 629–633. 2, 5

[15] T. Strau, M. Weidemann, J. Michael, G. Leifert, T. Grning, and R. Labahn, "System description of citlab's recognition & retrieval engine for icdar2017 competition on information extraction in historical handwritten records," 2018. 2, 5

[16] G. Leifert, T. Strau, T. Grning, and R. Labahn, "Citlab argus for historical handwritten documents," 2016. 2, 5

[17] T. M. Breuel, "The ocropus open source ocr system," in *Document Recognition and Retrieval XV*, vol. 6815. International Society for Optics and Photonics, 2008, p. 68150F. 2, 5

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 2

[19] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009. 2

[20] M. Federico, N. Bertoldi, and M. Cettolo, "Irstlm: an open source toolkit for handling large scale language models," in *Ninth Annual Conference of the International Speech Communication Association*, 2008. 2

[21] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. 3