Recognition of Historical Greek Polytonic Scripts Using LSTM Networks

Fotini Simistira*, Adnan Ul-Hassan†, Vassilis Papavassiliou*, Basilis Gatos§, Vassilis Katsouros* and Marcus Liwicki†‡
*Institute for Language and Speech Processing, Athena Research and Innovation Center, Athens, Greece
Email: {fotini, vpapa, vsk}@ilsp.athena-innovation.gr
†Department of Computer Science, University of Kaiserslautern, Germany
Email: adnan@cs.uni-kl.de
§Institute of Informatics and Telecommunications, NCSR Demokritos, Athens, Greece
Email: bgat@iit.demokritos.gr
‡DIVA Research Group, University of Fribourg, Switzerland
Email: marcus.liwicki@unifr.ch

***Abstract*—This paper reports on high-performance Optical Character Recognition (OCR) experiments using Long Short-Term Memory (LSTM) Networks for Greek polytonic script. Even though there are many Greek polytonic manuscripts, the digitization of such documents has not been widely applied, and very limited work has been done on the recognition of such scripts. We have collected a large number of diverse document pages of Greek polytonic scripts in a novel database, called Polyton-DB, containing $15,689$ textlines of synthetic and authentic printed scripts and performed baseline experiments using LSTM Networks. Evaluation results show that the character error rate obtained with LSTM varies from 5.51% to 14.68% (depending on the document) and is better than two well-known OCR engines, namely, Tesseract and ABBYY FineReader.**

I. Introduction

The main particularity of Greek polytonic scripts (usage started in the Hellenistic period, i.e., 3^{rd} century BC), is the appearance of various diacritics in Greek orthography notating Ancient Greek phonology: (i) the acute accent (**oxeia** – *sharp* or *high*), (ii) the grave accent (**bareia** – *heavy* or *low*), (iii) the circumflex (**perispomene** – *twisted around*), (iv) the rough breathing **dasi pneuma**, (v) the smooth breathing **psilon pneuma**, (vi) the **diaeresis** to indicate diphthong, and (vii) the **iota** subscript (hypogegrammene *written under*).

These diacritics and their combinations can be associated with the 14 vowel characters (7 upper-case and 7 lower-case letters) according to several phonologic and orthographic rules that have been differentiated from period to period, following the language changes through time. This situation results in many groups of different symbols for each vowel character that look similar. These groups of very similar characters, resulting in a very large character-set (more than 200), makes the OCR of Greek polytonic scripts a very challenging task. In addition, there is a lack of collections with ground-truthed data which hinders the development of robust recognition systems for such scripts.

In contrast, the simple monotonic orthography introduced in 1982 corresponds to modern Greek phonology, and requires only two diacritics: *tonos* to indicate stress, *diaeresis* to indicate a diphthong, i.e., the sound of two adjacent vowels, and their combination. Therefore, digitizing documents in this modern Greek language is relatively easier.

Our research is part of the OldDocPro project[1] which aims towards the recognition of Greek machine-printed and handwritten polytonic documents. In OldDocPro, we strive toward research that can assist the content holders in turning an archive of old Greek documents into a digital collection with full-text access capabilities using novel OCR methods. Our aim is to advance the frontiers and facilitate current and future efforts in old Greek document digitization and processing.

The contribution of this paper is two-fold. First, we present the Polyton-DB[2] – a novel database containing printed Polytonic Greek script. Note that Polyton-DB is an extension of GRPOLY-DB [1] consisting of scanned pages only. In this paper we show that the generation of synthetic data significantly boosts the performance. Second, a high-performance recognition system (which is based on the recently introduced LSTM networks of the OCRopus framework [2]) has been adapted to the specifics of the Greek polytonic script.

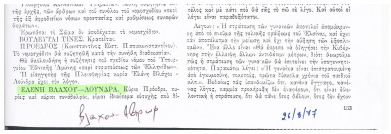
The organization of the rest of the paper is as follows. In Section II we describe in detail the Polyton-DB collection. The LSTM-based recognizer is discussed in Section III. Evaluation experiments on recognizing Polytonic Greek scripts and comparison with the OCR engines of ABBYY FineReader and Tesseract are described in Section IV. A conclusion and an outlook future work are given in Section V.

II. Polyton-DB — A Greek Polytonic Database

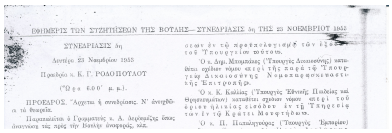
Polyton-DB includes printed polytonic Greek scripts from different periods. In particular, it contains three datasets which are described in the following subsections. Note that the first two small datasets base on the GRPOLY-DB [1], while a considerable effort for this paper and the high performance recognition has been spend on a proper generation of synthetic

[1]http://www.iit.demokritos.gr/ nstam/GRPOLY-DB
[2]the collection is available from http://media.ilsp.gr/PolytonDB

978-1-4799-1805-8/15/$31.00 ©2015 IEEE



(a) Vlahou (1977)



(b) Markezinis (1953)

Figure 1. Sample images of the Greek Parliament Proceedings.

Table I
DETAILS ABOUT VARIOUS DATASETS IN THE POLYTON-DB.

Set	Pages	Textlines
Greek Parliament Proceedings		
Vlahou	4	373
Markezinis	18	1,666
Saripolos	6	642
Venizelos	5	522
Greek Official Government Gazette	5	687
Appian's Roman History		
Synthetic data	315	11,799
Total	353	15,689

data. With such a large dataset it is feasible to train LSTM neural networks and reach a high performance. Quantitative information about the collection is presented in Table I.

A. Greek Parliament Proceedings

The first dataset consists of 3,203 textline images. that were extracted from 33 scanned pages of the Greek Parliament Proceedings (see Figure 1). These pages correspond to speeches of four Greek politicians (Vlahou in 1977, Markezinis in 1953, Saripolos in 1864 and Venizelos in 1931). For the creation of the Polyton-DB, we used the original grayscale images of all pages together with the corresponding texts. We first binarized the grayscale images [3] and then applied layout analysis and segmentation processes [4] to extract textlines and words. In order to assign the text information to the corresponding text lines an automatic transcript mapping procedure was applied [5]. Finally, the segmentation results and the transcripts' alignment were verified and corrected manually using the Aletheia framework [6].

B. Greek Official Government Gazette

The second part of POLYTON-DB includes 687 textline images (and their transcriptions), which were extracted from five scanned pages of the Greek Official Government Gazette following the processing steps described above, in Section II-A.

C. Synthetic data from Appian's Roman History

The third dataset contains 11,799 textline images of synthetic data generated by using the transcription of 315 scanned pages from Appian's Roman History written in Greek language before AD 165. This work more closely resembles a series of monographs than a connected history. It gives an account of various peoples and countries from the earliest times down to their incorporation into the Roman Empire, and survives in complete books and considerable fragments.

By comparing the document images of Appian's Roman History with the images of Greek Parliament Proceedings and Greek Official Government Gazette in terms of the readableness, we conclude that the former ones were in much better condition, i.e., clean scans without broken characters. Since this is not the case in processing historical documents, we decided to generate synthetic data in such a way that we could influence the characters' degradation with the aim of approximating the type of noise of historical scripts (see Fig. 2).

In order to simulate the common typefaces of the Greek Polytonic script we use *GFS Didot Classic* available from Greek Font Society[3]. Note that we have initially tried other fonts as well. However, the most realistic images have been achieved with this font. For the actual text line generation, we used OCRopus system's utility (*ocropus-linegen*) to generate synthetic text-line images. This utility is based on the degradation models proposed by Baird et al. [7] and uses a Python-PIL module to convert text into image. There are many parameters that can be altered to make the artificially generated text-line images resemble closely to those obtained from a scanning process. Some of the significant parameters are:

Blur: It is the pixel-wise spread in the output image, and is modeled as circular Gaussian filter.

Threshold: It is used in the binarization process. If a pixel value is greater than this threshold, then it is a black pixel.

Size: It is the height and width of individual characters in the image. It is modeled by image scaling operations.

Skew: It is the rotation angle of the output symbol.

The aforesaid OCRopus utility requires *utf-8*-encoded textlines to generate the corresponding textline images along with the *ttf*-type fontfiles. The user can specify the parameter values or use the default values. An example of a single textline image rendered with the *ocropus-linegen* utility is shown in Figure 2. In the current work, we used the default values as they generate textline images similar to our authentic data (first two textline images in Figure 2).

The number of unique character classes contained in Polyton-DB is 211, including Greek characters, numbers, special characters, hyphenation marks, etc. (see Table II).

[3]http://www.greekfontsociety.gr

ποιήσειν καὶ φιλανθρώπου, Ῥωμαίους τε ἀγαπή-
ποιήσειν καὶ φιλανθρώπου, Ῥωμαίους τε ἀγαπή-
ποιήσειν καὶ φιλανθρώπου, Ῥωμαίους τε ἀγαπή-
ποιήσειν καὶ φιλανθρώπου, Ῥωμαίους τε ἀγαπή-
ποιήσειν καὶ φιλανθρώπου, Ῥωμαίους τε ἀγαπή-
ποιήσειν καὶ φιλανθρώπου, Ῥωμαίους τε ἀγαπή-

Figure 2. Sample textline images of synthetic data. The first two textline images correspond to low distortion values (degradation values were used to generate synthetic data for our experiments), the third and fourth textline images correspond to medium distortions and the last two correspond to high values of distortion.

Table II
CHARACTERS CONTAINED IN POLYTON-DB

0 1 2 3 4 5 6 7 8 9	
Β Γ Δ Ζ Θ Κ Λ Μ Ν Ξ Π Σ Τ Φ Χ	Α ᾽Α ῾Α ᾽Α ῾Α ῞Α
β γ δ ζ θ κ λ μ ν ξ π ρ ῤ ῥ ς σ τ φ χ ψ	Ε ᾽Ε ῾Ε ᾽Ε ῾Ε ῞Ε ῾Ε
ααάὰᾶᾳᾴᾲᾷἀἁἂἃἄἅἆἇᾀᾁᾂᾃᾄᾅᾆᾇ	ε έ ὲ ἐ ἑ ἒ ἓ ἔ ἕ
ηὴήῆῃῄῂῇἠἡἢἣἤἥἦἧᾐᾑᾒᾓᾔᾕᾖᾗ	Η ᾽Η ῾Η ᾽Η ῾Η ῞Η ῾Η
ιἰἱἲἳἴἵἶἷῐῑῒΐῖῗ	Ι ᾽Ι ῾Ι ᾽Ι ῾Ι
υὐὑὒὓὔὕὖὗῠῡῢΰῦῧ	Υ ῾Υ Ρ ῾Ρ
ωὠώὼᾠᾡᾢᾣᾤᾥᾦᾧῲῳῴῶῷ	Ω ᾽Ω ῾Ω ῞Ω ῾Ω
Ο ᾽Ο ῾Ο ῞Ο ῾Ο ῞Ο	ο ὀ ὁ ὀ ὁ ὃ ὅ ὄ

III. RECOGNITION SYSTEM

A. Bidirectional LSTM Neural Networks

LSTM Networks are a modern variant of Recurrent Neural Networks (RNN). Traditional RNNs suffer from the problem of vanishing and exploding gradients, which implies that during the training process the gradient becomes either too small (vanishing) or too large (exploding) resulting, thus, in poor training. Hochreiter and Schmidhuber [8] replaced the basic unit of computation (sigmoid or tanh) with a computer-memory like cell and three multiplicative gates – *input*, *output*, and *forget*. These gates behave similar to read, write, and refresh functions in a computer memory. In whis way, the network can retain the contextual information as long as forget gate is ON. On the other hand, the output gate allows writing out the contained information and the input gate allows the network to read new information [9].

To process the contextual information in both forward and backward directions, bidirectional LSTM (BLSTM) network was proposed by Graves and Schmidhuber [10]. In this network, there are two hidden layers that process the input data in both forward (left-to-right) and backward (right-to-left) directions. This configuration allows the LSTM network to have complete contextual information about any time-step (past and future) during the processing. Both these layers are connected to the output layer.

Any standard neural network requires a segmented input data, so that its cost functions can be defined for each point. This requirement renders RNNs (or any of its variant) unusable for sequence learning tasks. Hybrid networks, such as HMM-RNNs, emerged as a possible solution to this challenge. In such an HMM-RNN architecture, the HMM part is used to segment the input data implicitly and the RNN is used for classification. However, this combination failed to utilize the full capabilities of recurrent nets. Graves [11] added a layer in an LSTM network that performs a forward-backward algorithm, called Connectionist Temporal Classification (CTC), on the output and enables LSTM networks to be used as sequence-learning machines, that is, there is no need to segment the input sequence.

Depending on how the input is presented at the input layer, LSTM networks can be categorized as 1D-LSTM networks or 2D-LSTM Networks. For 1D variant (see Figure 3), the input is in the form of a single dimensional sequence and in case of 2D, the input is given as a 2D patch. In both variants, bidirectional is present. For 2D case, the bidirectional mode means scanning the input in four directions, namely, right-to-left, left-to-right, top-to-bottom and bottom-to-top. For printed OCR tasks, we found that 1D-LSTM networks performs better than their 2D siblings [2]. To use 1D-LSTM for OCR tasks, the input textline image is scanned by a fixed-height window of 1-pixel width to convert the 2D-image into an one dimensional sequence. This 1-pixel width slice is termed as a ‘frame’.

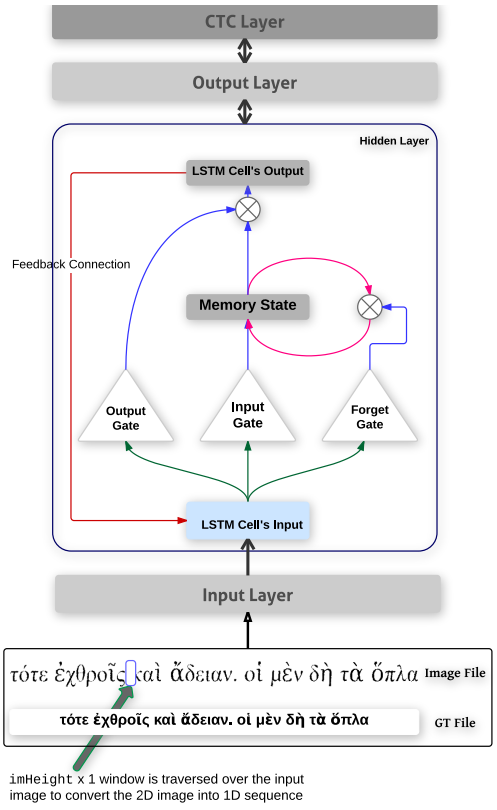


Figure 3. Simplified 1D-LSTM architecture. The Hidden layer is shown with 1 LSTM memory block. Each memory cell is connected to its surroundings with *input* and *output* gates. The input gate allows the input to be read, the output gate allows outputs to be written and the forget gate allows retention of the information within the memory cell. The CTC layer aligns the output activation with the ground-truth sequence using a forward-backward algorithm [11]. The input image is traversed by a $X \times 1$ (X being the height of the image) window to convert the 2D image into an 1D sequence. The ‘GT File’ here refers to the ground-truth labels associated with the input textline.

The training starts by choosing a random textline along with its transcription from training data. The textline is converted

into an one dimensional sequence (as described above) and each frame is fed to the LSTM network, where a forward pass is performed through the hidden and output layer. Then the forward-backward algorithm (CTC) aligns the output activation with ground-truth labels and subsequently the error is back-propagated (backward pass). During this process, the LSTM network learns to classify each frame into a target class (including space and 'reject' classes).

From the above discussion, it is clear that we require textline images of equal heights, so that they can be converted to an equal depth sequence[4]. The process of making heights of text-line images is termed as "normalization". This is also important from OCR point of view [12] as for Latin and Greek scripts, the absolute position and scale along the vertical axes are essential for distinguishing many common characters.

For our experiments, we used the open-source OCR system OCRopus [13]. The OCRopus comprises of many document image analysis modules including modules for binarization, page segmentation, textline normalization, line recognition, etc. We used the LSTM line recognizer module for the Greek polytonic script training.

B. Network structure

The number of iterations that the LSTM network would run is defined by N/f, where N is the total number of iterations (default=$1M$) and f is the mini-batch size (default=1000). In the current work, we normalized the size of the textline images to a height of $l = 48$ pixels (default values) and trained the LSTM based recognizer up to $N = 150000$ using the mini-batch size of $f = 1000$.

IV. Experimental Evaluation and Results

In order to evaluate the LSTM-based recognizer we run three series of experiments, using different combinations of the datasets described in Section II for training and testing. For comparison reasons, we use two well-known OCR engines: (i) Tesseract, an open source publicly available OCR system and (ii) ABBYY FineReader, a commercial OCR product.

In the first experiment, we used the synthetic data of Appian's Roman History and the 687 images of the Greek Official Government Gazette to train the LSTM-engine, while the textlines of the Greek Parliament Proceedings were used as the test set. In this way we ended up with a training set of $12,486$ textlines and a test set of $3,203$ textlines. It is worth mentioning that in this setting the fonts which are met in the training set are different from the four fonts of the test set. However, the OCRopus recognizer yielded a character error rate of 14.68%, after $125,000$ iterations in the training phase, as detailed in Figure 4. Note that the use of synthetic training data was important, i.e., without synthetic training data the performance was dramatically worse.

In the second configuration, the training set includes the synthetic data of Appian's Roman History, the textline images of the Greek Official Government Gazette, and the textline images of the three subsets (Saripolos, Markezinis and Vlahou) of the Greek Parliament Proceedings, while the textlines of Venizelos were the test images. In this way, we end up with a training set of $15,167$ textlines and a test set of 522 textlines. Note that, in this experiment, the training data contain text written in five different fonts, while the test set includes one font, unseen during training. We observe a character error rate that was significantly decreased to 5.67% (see Figure 5).

[4]the depth of sequence is equal to the height of the image.

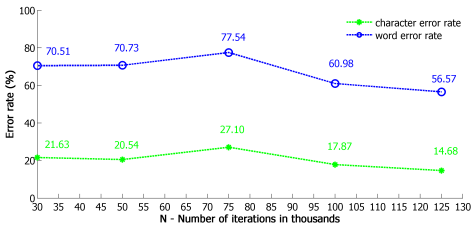


Figure 4. Error rates of LSTM in the first experiment for various iterations.

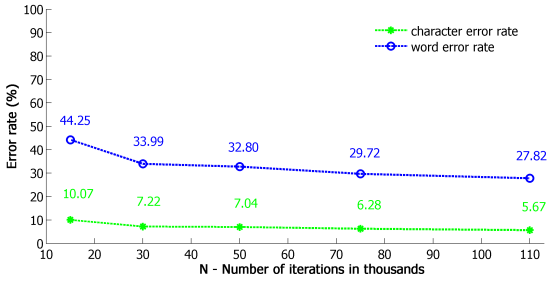


Figure 5. Error rates of LSTM in the second experiment for various iterations.

In the last experiment we tried to setup a configuration for comparing the proposed recognizer with the two aforementioned OCR systems. In the case of Tesseract the decision was straightforward, since we would like to examine the performance of a state-of-the-art tool, as it is available (i.e. no training or adaptation was applied); we used the training model for Greek polytonic script built by Nick White (http://eutypon.gr). Regarding the ABBYY FineReader engine, we adapted it to the recognition of Greek polytonic scripts by adopting the following procedure: First, we found the symbols which occur more than five times in each dataset of the Greek Parliament Proceedings. Then we randomly selected textline images with the purpose of creating a subset in which the targeted symbols occur at least five times. We ended up with a set including 367 textlines (54 from Vlahou, 136 from Markezinis, 108 from Saripolos and 69 from Venizelos). Finally, we semi-automatically segmented the images into characters and used the training utility of the ABBYY FineReader engine SDK to create the respective characters' models. Moreover, we make use of the Thesaurus Linguae Graecae corpus[5] to build a dictionary (in ABBYY FineReader's format) of *Kathareuousa*[6] and make it available

[5]http://www.tlg.uci.edu/

[6]a version between Ancient and Modern Greek, which was widely used both for literary and official purposes and was written in polytonic script

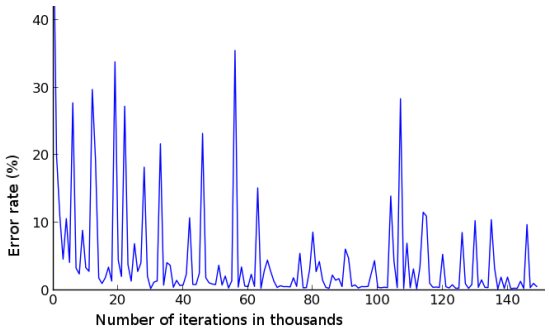


Figure 6. Training error rate using the third experiment.

Table III
CHARACTER (CER) AND WORD ERROR RATES (WER) OF THE THIRD EXPERIMENT

OCR Engine	**CER** (%)	**WER** (%)
Tesseract	30.37	71.43
ABBYY	19.20	48.60
OCRopus	**5.51**	**24.13**

to the engine with the aim of supporting recognition. These 367 textlines and the datasets of Appian's Roman History and Greek Official Government Gazette compose the train data for the LSTM-based recognizer, while the remaining $2,836$ textlines of Greek Parliament Proceedings were included in the test data. The results are presented in Table III. It is worth mentioning that the poor performance of Tesseract is mainly explained by the fact that the characters' degradation in the test set is too high, that the character segmentation introduces too many mistakes that are propagated in the recognition stage. Consequently, the use of synthetic data for training the engine could help in overcoming this shortcoming.

Regarding the LSTM-based recognizer, the training model with the lowest error rate (0.16%) was the one produced after $138,000^{th}$ iterations. By carefully examining Figure 6 we conclude that the training curve is still very unstable at the regions of 0.16% and it becomes more stable at 0.35%. As a result by using the training model produced after $148,000$ iterations, with corresponding error rate of 0.35%, results in reducing the character recognition error rate on the test set from 6.05% to 5.51%. The most frequent errors for the LSTM-recognizer are illustrated in Table IV. In particular, there are 318 deletion errors and 273 insertion errors out of $9,351$ errors in total. Furthermore, there is a great number of errors where a letter is misclassified with the same letter but different accent. For example 94 occurrences of the letter ῍Ε are erroneously classified as the letter ῾Ε .

V. CONCLUSIONS & FUTURE WORK

In this work we have presented Polyton-DB, a novel database containing $15,689$ textlines of synthetic and authentic printed Greek Polytonic script. We used this collection to train and test an LSTM-based recognizer using the OCRopus framework and achieved promising results. The LSTM-based recognizer for the Greek Polytonic script can be further improved by adding a post-processing procedure. In particular, by observing carefully the misclassified letters, we conclude that most of them could be fixed by reducing the number of different classes contained in the Polyton-DB. This can be achieved in a post processing procedure, by merging the letters that are the same but they have different accents (e.g. ἔ, ἒ).

Table IV
MOST FREQUENT ERRORS OF THE LSTM-RECOGNIZER

No. of Errors	**OCR result**	**GT character**
109	–	′
104	.	,
100	ι	τ
100	ο	σ
94	῍Ε	῾Ε

ACKNOWLEDGEMENTS

This work has been supported by the OldDocPro project funded by the GSRT. Further support was given by the EU funded project LangTerra, (FP7- REGPOT-2011-1) and by the SNF funded HisDoc 2.0 project.

REFERENCES

[1] B. Gatos, N. Stamatopoulos, and B. Louloudis, "GRPOLY-DB: An Old Greek Polytonic Document Image Database," in *ICDAR 2015*.
[2] T. M. Breuel, A. Ul-Hasan, M. Al Azawi, F. Shafait, "High Performance OCR for Printed English and Fraktur using LSTM Networks," in *ICDAR*, Washington D.C. USA, aug 2013.
[3] B. Gatos, I. Pratikakis, and S. J. Perantonis, "Adaptive Degraded Document Image Binarization," *Pattern Recognition*, vol. 39, pp. 317–327, 2006.
[4] B. Gatos, G. Louloudis, and N. Stamatopoulos, "Segmentation of Historical Handwritten Documents into Text Zones and Text Lines," in *ICFHR*, Creta, Greece, 2014, pp. 464–469.
[5] N. Stamatopoulos, G. Louloudis, and B. Gatos, "Efficient Transcript Mapping to Ease the Creation of Document Image Segmentation Ground Truth with Text–Image Alignment," in *ICFHR*, Kolkata, India, 2010, pp. 226–231.
[6] C. Clausner, S. Pletschacher, and A. A., "Aletheia – An Advanced Document Layout and Text Ground–Truthing System for Production Environments," in *ICDAR2011*, Beijing, China, 2011, pp. 48–52.
[7] H. S. Baird, "Document Image Defect Models," in *Structured Document Image Analysis*, H. S. Baird, H. Bunke, and K. Yamamoto, Eds. Springer-Verlag, 1992.
[8] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[9] F. A. Gers, N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2002.
[10] A. Graves and J. Schmidhuber, "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures," *Neural Networks*, vol. 18, pp. 602–610, 2005.
[11] A. Graves, S. Fernandez, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks." in *ICML*, 2006, pp. 369–376.
[12] M. R. Yousefi, M. R. Soheili, T. M. Breuel, and D. Stricker, "A Comparison of 1D and 2D LSTM Architectures for Recognition of Handwritten Arabic," in *DRR–XXI*, San Francisco, USA, 2015.
[13] "OCRopus – Open Source Document Analysis and OCR System." [Online]. Available: https://github.com/tmbdev/ocropy