Automatic Transcription of Ottoman Documents Using Deep Learning

Chapter · September 2024

DOI: 10.1007/978-3-031-70442-0_26

Esma Fatima Bilgin-Tasdemir
Istanbul Medeniyet University

Berrin A. Yanikoglu
Sabancı University

Automatic Transcription of Ottoman Documents Using Deep Learning

Esma F. Bilgin Tasdemir[1], Zeynep Tandoğan[1], S. Doğan Akansu[1], Fırat Kızılırmak[1], M. Umut Sen[2,3], Aysu Akcan[4], Mehmet Kuru[5], and Berrin Yanikoglu[2,3](✉)

[1] Department of Information and Document Management, Istanbul Medeniyet University, 34704 Istanbul, Turkey
esmabilgin.tasdemir@medeniyet.edu.tr

[2] Faculty of Eng. and Natural Sciences, Sabanci University, 34956 Istanbul, Turkey
{umut.sen,berrin}@sabanciuniv.edu

[3] VERIM, Sabanci University, 34956 Istanbul, Turkey

[4] Department of Near Eastern Studies, University of Vienna, 1090 Vienna, Austria
aysu.akcan@univie.ac.at

[5] Faculty of Arts and Sciences, Sabanci University, 34956 Istanbul, Turkey
mehmet.kuru@sabanciuniv.edu

Abstract. With the accelerated pace of digitization, a vast collection of Ottoman documents has become accessible to researchers and the general public. However, most users interested in these documents are unable to read them, as the text is Turkish written in the Arabic-Persian script. Manual transcription of such a massive amount of documents is also beyond the capacity of human experts. With the advancements in deep learning, we have been able to provide a solution to the long-standing problem of automatic transcription of printed Ottoman documents. We evaluated three decoding strategies including Word Beam Search that allows to use a recognition lexicon and n-gram statistics during the decoding phase. Furthermore, the effect of lexicon size and coverage and language modelling via character or word n-grams are also evaluated. Using a general purpose large lexicon of the Ottoman era (260K words and 86% test coverage), the performance is measured as 6.59% character error rate and 28.46% word error rate on a test set of 6,828 text lines.

Keywords: Ottoman Document Recognition · Turkish · Deep Learning

1 Introduction

Ottoman Turkish was the language used for administrative and literary purposes in the Ottoman Empire, from the early 15th century to the early 20th century.

Berrin Yanikoglu—Part of this work was done when Z. Tandoğan, S. D. Akansu and F. Kızılırmak were students at Sabancı University.

G. Sfikas and G. Retsinas (Eds.): DAS 2024, LNCS 14994, pp. 422–435, 2024.
https://doi.org/10.1007/978-3-031-70442-0_26

(a) (b) (c) (d)

Fig. 1. Examples of printed documents written in Naskh style and various formats. The current system works automatically for documents such as the two on the left.

Although Ottoman Turkish is based on Turkish language, it contains a considerable amount of Arabic and Persian words, loan-words and grammatical features [19,40]. The alphabet is an extended Arabic alphabet with 28 Arabic and five Persian letters.

The Ottoman Turkish language was employed in manuscripts by scribes in earlier times and in printed works from 1729 onwards, until the alphabet reform of the Turkish Republic in 1928. With the ever-increasing speed of the digitization process, a large collection of old Ottoman documents is now accessible to researchers and the general public. Unfortunately, the majority of the users interested in these documents can not read the Ottoman script. In fact, researchers in the fields of social sciences and humanities devote most of their time to scanning sources written in Ottoman Turkish and transcribing the references they reach. Consequently, there has long been a desire for an automated transcription system for Ottoman documents.

We present a novel transcription system that takes a printed Ottoman document written in the Arabic-Persian alphabet and returns its transcription in modern Turkish which is based on the Latin alphabet[2] A wide range of writing styles have been used in Ottoman Turkish documents, ranging from relatively simple Naskh style to very ornamental ones. We limit the scope to the Naskh style, which is not only less ornamental, but also the most widely used style in printed materials, as exemplified in Fig. 1.

2 Challenges in Ottoman Turkish Recognition

There are many challenges associated with the transcription of Ottoman documents. Problems associated with the cursive nature of the Arabic and Persian script are well-documented [11]. Character segmentation is quite more difficult compared to Latin-based alphabets. Furthermore, connected characters which

[2] A demo of the current system is available at https://demos.sabanciuniv.edu.

take multiple forms depending on their position in a word and diacritics and rich ligatures in certain fonts complicate line and character segmentation.

The orthography of the Ottoman script presents challenges as well. In particular omission of vowels, which is adopted from Arabic, leads to multiple heteronyms –words with the same spelling but different pronunciation. Transcription of such words requires contextual knowledge for accurate recognition. For example, Ottoman word can be transcribed as the four different Turkish words ("avlu" , "ölü", "evli" or "ulu") depending on the context.

Another challenge is the incorporation of Arabic and Persian vocabulary, along with some borrowed grammatical structures. This necessitates a significantly larger recognition lexicon compared to Turkish alone, which itself has a larger lexicon than English due to its agglutinative nature [39,41].

3 Related Work

Modern OCR systems are very good at recognizing text and symbols across diverse environments, including handwritten or printed documents, scenes, screenshots, and historical manuscripts. Prior to the advent of deep learning, popular OCR methods included Neural Networks (NNs) and Hidden Markov Models (HMMs) [4,31]. These techniques have since been replaced with Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Long Short-Term Memory units (LSTMs), and their derivatives, including bidirectional LSTMs (BiLSTMs), and more recently, transformers [14,20,22,23,28].

While the majority of document recognition research is done for Latin alphabets and especially English, there is a sizeable research towards recognition of non-Latin alphabets, including Arabic, Chinese, Japanese, and Cyrillic texts [4,5,15]. Research relevant to the recognition of Ottoman documents is summarized below.

Systems for Turkish. The Turkish alphabet is based on the Latin alphabet, with six special characters added to the English alphabet (çğıöşü) and three characters (qwx) being used only in some recent loanwords and derivations. While the extended alphabet introduces additional complexities, the primary challenge in recognizing Turkish arises from its agglutinative morphology. In brief, while 30,000-word lexicons are common in document recognition systems for English, the different word forms in daily Turkish can easily exceed one million due to its morphological structure [41]. In a recent study, Tasdemir et al. developed an HMM-based online Turkish handwriting recognition system that achieved a 91.7% word recognition rate within a vocabulary of approximately 2,000 words. However, when the vocabulary size increased to 12,500 words, the recognition rate sharply decreased to 67.9 [39].

Systems for Arabic and Persian. Research in OCR and HTR for Arabic has gained momentum in the last few decades [6,7,29]; however, success rates for Arabic recognition systems are considerably lower than those of Latin script-based systems. Much of the Arabic machine-printed OCR work is conducted on the APTI dataset which contains *synthetically* created Arabic word images rendered using several fonts [37]. Varying character recognition error rates between 0.5–2.5% are reported for Naskh and Naskh-like styles, based on the portion of that dataset used. In contrast to synthetic images in APTI, the P-KHATT dataset contains real data obtained from scanned Arabic printed line images. As expected, results on this dataset are lower, with a 3.1% Character Error Rate (CER) reported in [4] and 2.4% reported in [33].

Systems for Ottoman Turkish. There are a limited number of studies on recognition of Ottoman Turkish, most of which were conducted before the deep learning era [8,12,13,18]. In a recent work, Dolek et al. developed an Ottoman OCR system for printed Naskh line images using a CNN-LSTM network trained with both synthetic and real data [17]. The system's accuracy is reported as 88.86% letter recognition and 64% word recognition rate on a small test set comprising 21 pages [3]. An open vocabulary system for recognition of printed Naskh Ottoman texts reported 11% CER on synthetic data and 16% CER on a real data comprising of 1,200 line images from a printed historical Ottoman book [38]. As for handwritten documents, Aydemir et al. proposed an RNN-based system for recognizing word images obtained from population registration documents [9]. They reported a 12.4% character error rate and a 22.1% word error rate on a small test set of 1,000 different words.

A number of commercial tools have emerged recently for keyword search and transcription of Ottoman documents. An automatic transcription system for printed Ottoman text is realized using Transkribus which is a well-known platform specifically designed for the transcription, recognition and analysis of historical documents and handwritten texts, including RTL (right-to-left) texts [16]. The system is generated by fine-tuning a pretrained model in Transkribus platform using Ottoman printed text manually annotated at word level [2]. Another application is designed for keyword search in a predefined collection of documents [1]. A through evaluation of these commercial systems is infeasible due to the usage restrictions applied in their free versions.

4 Methodology

Unlike previous approaches in the literature, we use a single-stage approach to produce Turkish transcription directly from Ottoman Turkish documents. In two-stage approaches, the system first performs character recognition in the Arabic-Persian alphabet, followed by word recognition to obtain the corresponding Turkish word. In Fig. 2, this corresponds to recognizing the letters **k, t, a, p** first and then mapping the recognized character string to the most likely word in the lexicon (e.g. "kitap"). In our single-stage approach, we go directly from the text line image to the corresponding Turkish transcription using a CNN-BiLSTM model, as described in Sect. 4.2. Our approach has the advantage of saving time and effort in data annotation. For Turkish annotators, it is faster and easier to use Turkish characters instead of Ottoman letters when annotating the images, due to the familiarity with the Turkish letters and keyboard layout.

In addition, the accuracy of the labels produced can be checked more efficiently for similar reasons.

4.1 Dataset

There is no publicly available Ottoman document dataset. In all previous work, which is very limited in both number and scope, small-sized proprietary datasets were used. In this work, we first collected and annotated a document image dataset, which is required for training a deep network.

The dataset contains pages extracted from 13 books, written in the Ottoman script and printed between years 1870 and 1928. The books belong to various genre, namely novels, history, travelogue and epistolography books. Scanned page images of the books are automatically segmented into lines using a deep learning based segmentation method [27], resulting in 74,036 text lines, 595,144 words and 70,218 unique words in the dataset.

The text lines are manually annotated using a special transcription scheme that is designed to represent mappings between Arabic alphabet-based Ottoman letters and the Latin-alphabet based Turkish letters at a sufficient level. Uppercase characters are converted to lowercase since the Arabic alphabet has no distinct uppercase and lowercase letter forms. As a result, there are 70 unique characters that appear in the transcription text, as listed in Table 1. The characters that appear in the Ottoman document images are the 33 letters of the Arabic- alphabet and 10 Arabic digits. Additionally, the English letters, punctuation and special symbols of Table 1 appear in the Ottoman text images as well.

Table 1. The character set of the transcriptions.

Group	Characters
Modern Turkish Letters	a–z
Older Turkish Letters	â î û
Borrowed English Letters	q x w
Digits	0–9
Punctuation	, . : ; ? !’
Special Symbols	( ) * + − - / _ = [ ] & §% \ >< Space

4.2 Corpus and Lexicon

In document recognition, a large text corpus is often used to extract a lexicon of valid words and an n-gram word statistics. In this work, we collected a text corpus from a large set of novels, historical works, and periodicals created between 1888–1927, reflecting the linguistic features and the lexicon of the late Ottoman period. The corpus text is written using the modern Turkish alphabet and consists of approximately 1,761K words and 260K unique words.

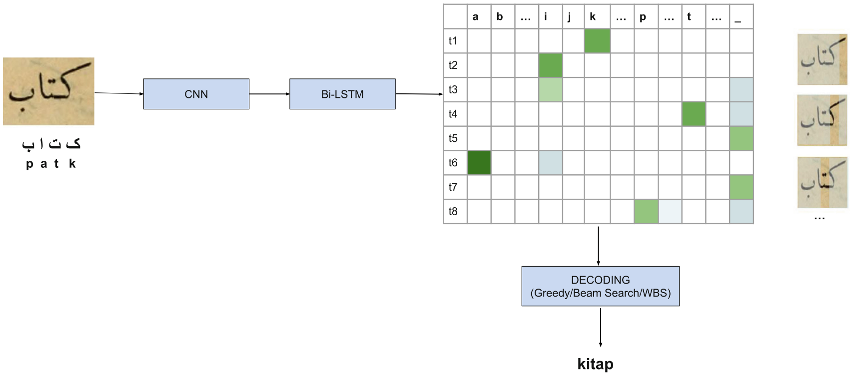

Fig. 2. System overview for the word "kitap" (*book*) written in four Arabic letters, from left to right (on the left). The vertical patches corresponding to the time frames are shown on the right (without overlap for clarity). The last character in the $T \times (C+1)$ conditional probability matrix is the special blank token used in the CTC decoding.

Additionally, we use the BOUN corpus [34] that contains raw modern Turkish text collected mainly from the web, in order to evaluate the importance of using a language model of the period. There are 4.1M unique tokens including words, punctuation marks and numbers in this corpus. Morphological analysis is used to clean the raw corpus [39], resulting in 1,578,553 unique items.

4.3 CNN-BiLSTM Model

Our CNN-BiLSTM model combines a convolutional neural network (CNN) for feature extraction and a bi-directional Long Short-Term Memory (LSTM) model for sequence modelling. The LSTM is a special type of recurrent neural network that can learn what to remember or forget from the context and the bi-directional connections allow the model to capture dependencies in both the past and future contexts.

Hybrid models of convolutional and recurrent neural networks are frequently used for handwritten text recognition in the literature [25,30,32,36] and the model used in this work is based on the model proposed in [26] for recognizing English handwritten text.

The input to the model are text line images resized to a fixed height (64 pixels) keeping the aspect ratio and padded to 2,000 pixels (based on the length of the longest line in the dataset). The only other preprocessing applied to line images is binarization using a deep learning based method [10].

The network includes 12 convolutional layers using 3×3 kernels to extract features from each image frame, and two bi-directional LSTM layers with 256 hidden neurons each to encode the sequence information. The output of the network is a sequence of probability distributions over the predefined alphabet, using the CTC loss function [21]. The CTC loss allows the model to train end-to-end, without needing to know character boundaries.

The training parameters, which are empirically decided, are a batch size of 4 and a learning rate of 1e−4. The network weights are initialized randomly and optimized using the Adam optimization algorithm. The model is trained with the CTC loss function until there is no remarkable improvement in the CTC loss.

The CNN-BiLSTM enables the single-stage approach, as it learns not only to recognize the written characters but also the vowelization. More specifically, the system learns to map some of the input frames to the missing vowels, as illustrated in Fig. 2.

4.4 Decoding

A number of strategies can be employed for word-level decoding. In the *greedy* or *best path* approach, the symbol with the highest probability is chosen at each time frame. This simple approach is often not optimal as each frame is labelled independently. *Beam search* tries to overcome this limitation by extending the best path with the highest rated k alternatives [24]. It is also possible to use a character-level language model with the Beam search, in order to integrate the contextual knowledge to the decoding process [24]. The greedy and beam search are general purpose search alternatives that do not use a lexicon.

For text recognition, Scheidl et al. [35] proposed the *Word Beam Search (WBS)*, which is a beam search that uses a lexicon to guide the search towards valid words. Specifically, while selecting the next best character alternatives, the algorithm selects the characters that result in valid prefixes in the given lexicon (all sub-strings that are valid word beginnings in the language). This is done efficiently by representing the lexicon that is learned from a training corpus, as a trie. The WBS approach can further employ a 2-gram language model (LM) trained over a given corpus to incorporate bi-gram word statistics.

In this work, we use the WBS with different lexicons and language model settings offered in WBS and evaluate their effectiveness. We experiment with two modes proposed for WBS [35]: *word* mode and *n-gram* mode. In the *word* mode, there is no language model applied during the decoding, only a list of words is used to contstraint the search. In the *n-gram* mode, a word level 2-gram language model trained over a given corpus is employed.

5 Experiments

We conducted a series of experiments. The initial experiments evaluated different decoding techniques, while subsequent experiments explored the effects of lexicon size and coverage, as well as language modeling.

The dataset is split into training, test and validation sets randomly, with 59,233 lines in the training set, 7,403 lines in the validation and 6,828 in the test set[4] The Character Error Rate (CER) and Word Error Rate (WER) metrics based on the Levenshtein distance are conventionally used as error measures in text recognition, as well as this work.

[4] Test subset is publicly available at https://github.com/verimsu/Akis-Dataset.

5.1 Evaluating Decoding Strategies

We first evaluated alternative decoding strategies, where the same CTC output matrix is decoded using a Greedy decoder, Beam search decoder and the Word Beam Search decoder in a number of experiments. We experimented with different beam sizes whenever a beam search decoder is used.

As can be seen in Table 2, the best results are obtained with Word Beam Search with a beam width of 50 using a large lexicon of the Ottoman period (see Sect. 4.2). Increasing the beam size improves the recognition performance at the expense of increased decoding time; yet, the average time spent per document is acceptable for beam size of 50 (around 0.3 s).

Beam search exhibits slightly worse performance and significantly longer inference times, as it maintains an exponentially large set of decoding path alternatives. In contrast, Word Beam Search eliminates many of these paths by assigning zero probability to them. Based on these results, we decided to use WBS with a beam width of 50 in future experiments.

Table 2. Recognition results with different decoding approaches. WBS decoding is used in word mode without using a language scoring. The decoding lexicon is the 260K-word general purpose lexicon of the era. Time indicates total time for the whole test set.

Method	Beam Width	CER%	WER%	Time
Greedy	-	7.09%	33.16%	6 min
Beam Search	10	6.94%	32.56%	17 h
Word Beam Search (word mode)	10	7.63%	29.19%	13 min
	30	6.77%	28.70%	24 min
	50	**6.59%**	**28.46%**	**38 min**

5.2 Effect of Lexicon Size and Coverage

In the first set of experiments (Table 2), the Word Beam Search in *word* mode was found to be the best decoding strategy. In this section, we report on the effects of the size and test-set coverage of the used lexicon, to understand the effect of Out-of-Vocabulary (OOV) words, or words that are missing from the decoding lexicon, on the overall performance.

We first use WBS with the *test set* lexicon, in order to measure the best case performance under the closed-vocabulary assumption with zero OOV rate. For a more realistic evaluation, we then merge the test lexicon with the 260-K large lexicon to obtain a larger lexicon with still a 100% coverage of test set words, Finally, we use a different lexicon which is derived from a modern corpus, to observe the importance of using a lexicon from the correct period. The BOUN corpus [34] originally contains 1.5 million words, however we used the most frequent 20% of the words for a fair comparison with the 260-K lexicon (Sect. 4.2). The resulting lexicon, which is referred as the *Modern corpus* in Table 3, contains 267-K words with a coverage rate of 66.39% for the test set.

The results are given in Table 3 where the results with the 260-K lexicon from Table 2 are given in the first row for ease of comparison. As expected, the lowest error rates are obtained using the test set lexicon, with 5.48% and 21.73% CER and WER respectively with a beam width of 50. The results obtained with the merged lexicon with 267,333 words and 100% coverage rate results are closer to those obtained with test-set lexicon rather than the large lexicon, implying that the OOV rate is more of a concern than a larger lexicon size. When the lexicon extracted from the modern corpus is used, the error rates are much higher, underlining the importance of using a lexicon from the appropriate era.

Table 3. Effect of lexicon size and coverage. The first row is from Table 2. Best results in each lexicon are obtained with the largest beam size of 50 (shown in bold)

Corpus	Lexicon Size	Test Coverage (%)	Beam width	CER%	WER%
Large	260,070	86.14	10	7.63	29.19
			30	6.77	28.70
			50	**6.59**	**28.46**
Test set	22,809	100	10	6.56	23.01
			30	5.69	22.11
			50	**5.48**	**21.73**
Large+test set	267,333	100	10	6.55	25.57
			30	5.89	24.84
			50	**5.78**	**24.67**
Modern corpus	267,518	66.39	10	9.43	35.03
			30	8.55	34.94
			50	**8.37**	**34.99**

5.3 Effect of Language Modelling

In this experiment, we evaluated the *n-gram* mode of the Word Beam Search against the *word* mode. While the *word* mode only considers whether a given string appears in the lexicon, *n-gram* mode takes into account n-gram word occurrence statistics when finding the best paths. We used 2-gram language modelling in this work (i.e. $n = 2$).

Results given in Table 4 show that the *n-gram* mode of the Word Beam Search obtains significantly higher errors, compared to the *word* mode (6.59% vs 9.40%

CER). We think that this is due to not having a large enough text corpus to learn the word co-occurrence statistics. The average number of occurrence of a word in the large corpus is 6.7, which is clearly low to derive reliable 2-gram statistics. A larger corpus can help in integrating reliable n-gram statistics to decoding process; this is especially needed in the case of agglutinative languages that are afflicted with the vocabulary explosion problem [39].

We also applied *character-level* 2-gram language modelling to the Beam Search method, which reduced the CER slightly from 6.94% to 6.92%. We attribute this improvement to sufficient n-gram statistics due to the lower dimensionality of the character vocabulary, as compared to the word vocabulary.

Table 4. Effect of language modelling using the Ottoman-era corpus and lexicon. Best results shown in bold.

Method	Beam Width	LM	CER%	WER%	Time
WBS	50	word mode	**6.59**%	**28.46**%	38 min
	50	word 2-gram	9.40%	30.95%	45 min
Beam Search	10	-	6.94%	32.56%	$\sim$ 1 day
	10	char 2-gram	6.92%	32.55%	$\sim$ 1 day

6 Error Analysis

When we analyzed the system output with respect to the ground truth, we noticed that there are some common patterns in the errors made by the system. One of the most frequent errors is confusing punctuation characters, while another is the addition of superfluous space characters that are introduced in the middle of a word (e.g. "an kara" vs "ankara"). As these are often not crucial in terms of the semantic understanding of the text, we also analyzed the predictions more leniently, ignoring these two types of errors. The results given in Table 5 show that these types of small errors reduces CER and WER to to 6.31% and 27.19%, respectively.

Table 5. Recognition results with WBS (word mode, beam width of 50) when ignoring less important errors. First row is from Table 2.

Method	CER%	WER%
Strict evaluation	6.59	28.46
Ignoring punctuation errors	6.35	28.34
Ignoring punctuation errors and split words	6.31	27.19

One other interesting source of error is related to the auxiliary verb'etmek' (to do/ to make/to perform), which is generally used to form compound verbs. Some of these compound verbs are spelled in adjoint form by dropping the last vowel, as in the case with *lütuf + etmek → lütfetmek* (to oblige) and *hüküm + etmek → hükmetmek* (to rule). Others are simply spelled as two separate words, for example *terk etmek* (to leave). The system often recognizes the components of the compound verbs, but without capturing the adjoint form.

7 Summary and Discussion

In this work, we present an automatic transcription system for printed Ottoman documents using a CNN-BiLSTM model. Our system obtains 6.59% CER and 28.46% WER on a test set of 6.8K line images using the Word Beam Search decoder with a 260K-word lexicon with a 86.14% test set coverage.

The error analysis showed that despite the high WER, the text output is actually quite readable, with a good portion of errors involving punctuation and space characters, or single letter substitutions during the vowelization process.

The performance of the system improves with increasing test set coverage; yet blindly increasing the decoding lexicon size is not a feasible solution for Turkish. For future work, we plan to modify the WBS method to represent words as stems and suffixes to alleviate the coverage problem and use a deep learning based language model. We will also incorporate page-level decoding which will fix some of the errors by giving context to the decoder.

Acknowledgement. This study was supported by Scientific and Technological Research Council of Turkey (TUBITAK) under the Grant Number 122E399. The authors thank TUBITAK for their support.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ottoman Turkish discovery portal. https://www.muteferriqa.com/en. Accessed 10 May 2024
2. Transkribus Ottoman Turkish print. https://readcoop.eu/model/ottoman-turkish-print/. Accessed 10 May 2024
3. https://www.osmanlica.com/. Accessed 13 Nov 2022
4. Ahmad, I., Mahmoud, S.A., Fink, G.A.: Open-vocabulary recognition of machine-printed Arabic text using hidden markov models. Pattern Recognit. **51**, 97–111 (2016)
5. Ahmed, I., Mahmoud, S., Parvez, M.: Printed Arabic text recognition. In: Märgner, V., El Abed, H. (eds.) Guide to OCR for Arabic Scripts, pp. 147–168. Springer, London (2012). https://doi.org/10.1007/978-1-4471-4072-6_7
6. Al-Badr, B., Mahmoud, S.A.: Survey and bibliography of Arabic optical text recognition. Signal Process. **41**(1), 49–77 (1995)

7. Al-Helali, B.M., Mahmoud, S.A.: Arabic online handwriting recognition (AOHR): a survey. ACM Comput. Surv. **50**(3), 33:1–33:35 (2017)
8. Arifoglu, D., Sahin, E., Adiguzel, H., Duygulu, P., Kalpakli, M.: Matching Islamic patterns in Kufic images. Pattern Anal. Appl. **18**(3), 601–617 (2015)
9. Aydemir, M.S., Aydin, B., Kaya, H., Karliaga, I., Demir, C.: Tübitak Turkish - Ottoman handwritten recognition system. In: 2014 22nd Signal Processing and Communications Applications Conference (SIU), Trabzon, Turkey, April 23-25, 2014, pp. 1918–1921. IEEE (2014)
10. Baierer, K., Büttner, A., Engl, E., Hinrichsen, L., Reul, C.: OCR-D & OCR4all: two complementary approaches for improved OCR of historical sources. In: Sumikawa, Y., Ikejiri, R., Doucet, A., Pfanzelter, E., Hasanuzzaman, M., Dias, G., Milligan, I., Jatowt, A. (eds.) Proceedings of the 6th International Workshop on Computational History (HistoInformatics 2021) co-located with ACM/IEEE Joint Conference on Digital Libraries 2021 (JCDL 2021), Online event, September 30-October 1, 2021. CEUR Workshop Proceedings, vol. 2981. CEUR-WS.org (2021)
11. Biadsy, F., El-Sana, J., Habash, N.: Online Arabic handwriting recognition using hidden Markov models (2006)
12. Can, E.F., Duygulu, P.: A line-based representation for matching words in historical manuscripts. Pattern Recognit. Lett. **32**(8), 1126–1138 (2011)
13. Can, E.F., Duygulu, P., Can, F., Kalpakli, M.: Redif extraction in handwritten Ottoman literary texts. In: 20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23–26 August 2010, pp. 1941–1944. IEEE Computer Society (2010)
14. Carbune, V., et al.: Fast multi-language LSTM-based online handwriting recognition. Int. J. Document Anal. Recognit. **23**(2), 89–102 (2020)
15. Clanuwat, T., Lamb, A., Kitamoto, A.: Kuronet: pre-modern Japanese Kuzushiji character recognition with deep learning. In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20–25, 2019, pp. 607–614. IEEE (2019)
16. Colutto, S., Kahle, P., Hackl, G., Mühlberger, G.: Transkribus. a platform for automated text recognition and searching of historical documents. In: 15th International Conference on eScience, eScience 2019, San Diego, CA, USA, September 24–27, 2019, pp. 463–466. IEEE (2019)
17. Dolek, I., Kurt, A.: A deep learning model for Ottoman OCR. Concurr. Comput. Pract. Exp. **34**(20) (2022)
18. Duygulu, P., Arifoglu, D., Kalpakli, M.: Cross-document word matching for segmentation and retrieval of Ottoman divans. Pattern Anal. Appl. **19**(3), 647–663 (2016)
19. Ergin, M.: Türk Dil Bilgisi. Boğaziçi Yayınları, İstanbul (2020)
20. Fujitake, M.: DTrOCR: decoder-only transformer for optical character recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 8025–8035 (2024)
21. Graves, A., Fernández, S., Gomez, F.J., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Cohen, W.W., Moore, A.W. (eds.) Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25–29, 2006. ACM International Conference Proceeding Series, vol. 148, pp. 369–376. ACM (2006)
22. Graves, A., Fernández, S., Liwicki, M., Bunke, H., Schmidhuber, J.: Unconstrained on-line handwriting recognition with recurrent neural networks. In: Platt, J.C.,

Koller, D., Singer, Y., Roweis, S.T. (eds.) Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007, pp. 577–584. Curran Associates, Inc. (2007)
23. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(5), 855–868 (2009)
24. Hwang, K., Sung, W.: Character-level incremental speech recognition with recurrent neural networks. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016, pp. 5335–5339. IEEE (2016)
25. Jain, M., Mathew, M., Jawahar, C.V.: Unconstrained scene text and video text recognition for Arabic script. In: 1st International Workshop on Arabic Script Analysis and Recognition, ASAR 2017, Nancy, France, April 3-5, 2017, pp. 26–30. IEEE (2017)
26. Kizilirmak, F., Yanikoglu, B.: CNN-BiLSTM model for english handwriting recognition: Comprehensive evaluation on the IAM dataset. arXiv preprint arXiv:2307.00664 (2023)
27. Kodym, O., Hradiš, M.: Page layout analysis system for unconstrained historic documents. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12822, pp. 492–506. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86331-9_32
28. Li, M., et al.: TrOCR: transformer-based optical character recognition with pre-trained models. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 13094–13102 (2023)
29. Lorigo, L.M., Govindaraju, V.: Offline Arabic handwriting recognition: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **28**(5), 712–724 (2006)
30. Martínek, J., Lenc, L., Král, P., Nicolaou, A., Christlein, V.: Hybrid training data for historical text OCR. In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019, pp. 565–570. IEEE (2019)
31. Memon, J., Sami, M., Khan, R.A., Uddin, M.: Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR). IEEE Access **8**, 142642–142668 (2020)
32. Puigcerver, J.: Are multidimensional recurrent layers really necessary for handwritten text recognition? In: 14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017, pp. 67–72. IEEE (2017)
33. Rahal, N., Tounsi, M., Hussain, A., Alimi, A.M.: Deep sparse auto-encoder features learning for Arabic text recognition. IEEE Access **9**, 18569–18584 (2021)
34. Sak, H., Güngör, T., Saraclar, M.: Resources for Turkish morphological processing. Lang. Resour. Eval. **45**(2), 249–261 (2011)
35. Scheidl, H., Fiel, S., Sablatnig, R.: Word beam search: a connectionist temporal classification decoding algorithm. In: 16th International Conference on Frontiers in Handwriting Recognition, ICFHR 2018, Niagara Falls, NY, USA, August 5-8, 2018, pp. 253–258. IEEE Computer Society (2018)
36. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2017)

37. Slimane, F., Zayene, O., Kanoun, S., Alimi, A.M., Hennebert, J., Ingold, R.: New features for complex Arabic fonts in cascading recognition system. In: Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012, pp. 738–741. IEEE Computer Society (2012)
38. Tasdemir, E.F.B.: Printed Ottoman text recognition using synthetic data and data augmentation. Int. J. Document Anal. Recognit. **26**(3), 273–287 (2023)
39. Tasdemir, E.F.B., Yanikoglu, B.A.: Large vocabulary recognition for online Turkish handwriting with sublexical units. Turkish J. Electr. Eng. Comput. Sci. **26**(5), 2218–2233 (2018)
40. Timurtaş, F.K.: Osmanlı Türkçesi Grameri III. Alfa, İstanbul (2017)
41. Yanikoglu, B.A., Kholmatov, A.: Turkish handwritten text recognition: a case of agglutinative languages. In: Kanungo, T., Smith, E.H.B., Hu, J., Kantor, P.B. (eds.) Document Recognition and Retrieval X, Santa Clara, California, USA, January 22-23, 2003, Proceedings. SPIE Proceedings, vol. 5010, pp. 227–233. SPIE (2003)