Ottoman OCR: Printed Naskh Font

Ishak DOLEK
Computer Programming Dept.
Sakarya Applied Sciences University
Sakarya, Turkey
ishakdolek@subu.edu.tr

Atakan KURT
Computer Engineering Dept.
Istanbul University-Cerrahpasa
Istanbul, Turkey
atakan.kurt@iuc.edu.tr

***Abstract*—We present an OCR tool developed for printed Ottoman documents in naksh font as part of a project named *End-to-End Conversion of Ottoman Documents to Modern Turkish* This tool uses a deep learning model trained with a data set containing original and synthetic documents. We conducted an experimental comparison of this tool named *Osmanlica.com* with Tesseract Arabic, Tesseract Persian, Abby Finereader, Miletos and Google Docs OCR tools (or models) using a test data set comprised of 21 pages of original documents. With 88,64% raw, 95,92% normalized and 97,18% joined character recognition accuracy rates, Osmanlica.com outperformed the other tools with a marked difference. Osmanlica.com also achieved 58% word recognition accuracy which is the only rate over 50% among the OCR tools compared. We shared the test data set, ground truth, OCR outputs and the test program written in Python using *difflib* at osmanlica.com/test for independent verification.**

Keywords—Ottoman, OCR, printed naksh font, deep learning

I. INTRODUCTION

Ottoman was a written language used in Ottoman Empire between 13th and 20th centuries. Although Ottoman in its core is Turkish, it borrows heavily from Arabic and Persian, both have influenced Ottoman drastically. [1] Arabic was the language of the religion, while Persian was of the literature. Ottoman used Arabic script with extra letters for representing the sounds from Persian and Turkish. There is a quite large collection books, newspapers, etc. amounting to millions in total in archives and libraries. [2] It is not practical to convert them manually to Modern Turkish, as is the case so far. The OCR tools developed for Arabic and Persian are not very good at recognizing Ottoman either. Therefore it is imperative to train deep learning models for recognizing Ottoman in order to efficiently convert documents to text.

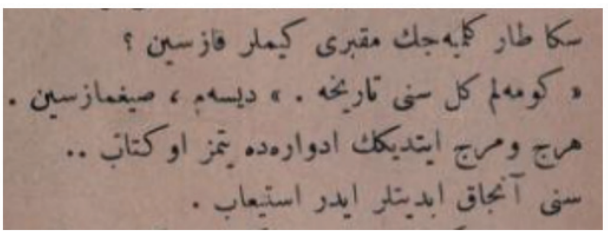
We present an OCR tool developed for printed Ottoman documents in naksh font as part of an project named *End-to-End Conversion of Ottoman Documents to Modern Turkish*[*] in this paper. The project aims to achieve its objective in four distinct steps shown in Fig. 1:

1. *Document-image conversion:* Documents are scanned, digitized and converted into image file formats. This step is not part of the project.
2. *Image-text conversion (OCR):* Images are converted to text by recognizing characters in the image using classification models trained with large data sets.
3. *Transliteration:* This step involves the orthographical conversion of Ottoman text in Arabic-based Ottoman script into Latin-based Turkish script. Since there are many-to-many correspondence between Ottoman and Turkish letters, this step is quite complex. It is further complicated since there are many-to-many mapping between the words in Ottoman (Arabic) and Turkish (Latin) script.
4. *Intra-language translation:* The translation of Ottoman text in Turkish (Latin) script to Modern Turkish is carried out in this step.

Steps in Ottoman-Turkish End-to-End Conversion

1. Document-image conversion:
   document → image file
2. Image-text conversion (OCR):
   image file → ottoman text (Arabic)
3. Transliteration:
   Ottoman text (Arabic) → Ottoman text (Latin)
4. Translation:
   Ottoman text (Latin) → Turkish text (Latin)

Ottoman-Turkish End-to-End Conversion Example



سکا طار کلمیه‌جك مقبری کیملر قازسین ؟
کومه‌لم کل سنی تاریخه دیسه‌م ، صیغمازسین.
هرج و مرج ایتدیکك ادواره ده یتمز او کتاب..
سنی آنجاق ابدیتلر ایدر استیعاب .

Sana dar gelmeyecek makberi kimler kazsın?
"Gömelim gel seni târîhe" desem, sığmazsın.
Herc ü merc ettiğin edvâra da yetmez o kitâb..
Seni ancak ebediyyetler eder istîâb.

Sana dar gelmeyecek mezarı kimler kazsın?
"Gömelim gel seni tarihe" desem, sığmazsın.
Altüst ettiğin devirlere de yetmez o kitap..
Seni ancak sonsuzluklar eder zapt.

Fig. 1 End-to-end conversion steps

This paper is organized as follows: The related work is given in Section 2. The Ottoman script is described in Section 3. The data set used in experiments are discussed in Section 4. The experimental comparison of OCR tools and the results are

This is partially project supported by TÜBİTAK 1512 program

978-1-6654-3603-8/21/$31.00 ©2021 IEEE

presented in Section 5. A summary and the future work are given in Section 6.

II. Related Work

The first studies in Ottoman documents date back to the 90's [10,11,12,13]. This effort has been continued by other groups [14,15,16]. [17,18,19,20] can be given as examples of other useful approaches for Arabic and scripts. Various methods have been tried in the optical character recognition of texts written in the Ottoman script. Studies have been carried out with Linear discriminant analysis in [21], a single layer artificial neural network in [22], support vector machines in [23], and hidden markov models in [24. In [25] a system that performs OCR for search and query purposes has been developed. However deep learning models which has been successfully applied to various problems in recent years has also been successfully applied to OCR, and satisfactory results have been obtained for Arabic script [26,27,28]

III. Characteristics of Ottoman Script

Ottoman was a written language used in Ottoman Empire between 13th and 20th centuries. Although Ottoman in its core is Turkish, it borrows heavily from Arabic and Persian, both of which had influenced Ottoman drastically. Arabic was the language of the religion, while Persian was of the literature. [1]

Ottoman script was based on the Arabic script consisting of 28 Arabic letters (ابتثجحخدذرزسشصضطظعغفقكلمنهوى) extended with 4 letters for Persian (پچژگ) and 3 letters (ﻙﺅﺀه) for Turkish sounds. Ottoman is written from right to left in cursive style. The letters are written slightly differently depending on whether they are isolated or joined with the following or preceding letter. The majority of letters (joiner) joins from both sides while only eight letters (non-joiner) joins from right side (ادذرزژوه). Thus letters have two or four forms: isolated, final, medial, initial form as shown in Table 1.

An understanding of certain characteristics of Arabic script is a prerequisite for developing better OCR models. Dot are used for differantiating among consonants (i'jam). 16 letters in Arabic have dots, the remaining 19 are dot-less (احدرسصطعكلمهوگﺀه). There are 10 letters with a single dot (بجخذزضظغفن), 3 letters with 2 dots (تقي), ve 6 letters with 3 dots (تشپچژڭ). 14 of the 19 dotted letters have the dots above (تنخذزشضظغفقنژڭ) while the remaining 5 letters have them below (بجپيچ).

The body of the letters are very smimilar or exactly the same especially in the medial or final form when dots are ignored. So the 34 lettes (excluding ء) are divided into 4 basic shape groups and 15 basic shapes when dots are dismissed: toothed (8 letters), looped (11 letters), straight (5 letters), and others (9 letters) as shown in the table.

There are number of characteristics of the Ottoman such as cursive writing style, joining of letters, frequent use dots contextual letterforms, letter similarity, overlapping letters, word segmentation, diacritics, multiple character encoding which make OCR more difficult than Latin OCR.

IV. Data Sets

We present the data set for training and testing the deep learning LSTM models of Osmanlica.com OCR tool.

Table 1 Ottoman letters & basic shape groups

Group	Letter				Basic Shape				Code
toothed	ﺐ ﺖ ﺚ ﭗ	ﺒ ﺘ ﺜ ﭙ	ﺑ ﺗ ﺛ ﭘ	ب ت ث پ	ـﺐ ـﺖ ـﺚ ـﭗ	ﺒـ ﺘـ ﺜـ ﭙـ	ﺑـ ﺗـ ﺛـ ﭘـ	ب ت ث پ	066E
	ﻦ	ﻨ	ﻧ	ن	ـﻦ	ﻨـ	ﻧـ	ن	06BA
	ﻰ	ﻴ	ﻳ	ى	ـﻰ	ﻴـ	ﻳـ	ى	0649
	ﺲ ﺶ	ﺴ ﺸ	ﺳ ﺷ	س ش	ـﺲ ـﺶ	ﺴـ ﺸـ	ﺳـ ﺷـ	س ش	0633
other	ﺪ ﮈ			د ڈ	ـﺪ			د	062F
	ﺮ ﺰ ﮋ			ر ز ژ	ـﺮ			ر	0631
	ﺞ ﺢ ﺦ ﭺ	ﺠ ﺤ ﺨ ﭽ	ﺟ ﺣ ﺧ ﭼ	ج ح خ چ	ـﺞ ـﺢ ـﺦ ـﭺ	ﺠـ ﺤـ ﺨـ ﭽـ	ﺟـ ﺣـ ﺧـ ﭼـ	ج ح خ چ	062D
looped	ﺺ ﺾ	ﺼ ﻀ	ﺻ ﺿ	ص ض	ـﺺ	ﺼـ	ﺻـ	ص	0635
	ﻂ ﻆ	ﻄ ﻈ	ﻃ ﻇ	ط ظ	ـﻂ	ﻄـ	ﻃـ	ط	0637
	ﻊ ﻎ	ﻌ ﻐ	ﻋ ﻏ	ع غ	ـﻊ	ﻌـ	ﻋـ	ع	0639
	ﻖ ﻒ	ﻘ ﻔ	ﻗ ﻓ	ق ف	ـﻖ	ﻘـ	ﻗـ	ف	066F
	ﻮ			و	ـﻮ			و	0648
	ﻢ	ﻤ	ﻣ	م	ـﻢ	ﻤـ	ﻣـ	م	0645
	ﻪ ـه	ﻬ	ﻫ	ه	ـﻪ ه	ﻬـ	ﻫـ	ه	0647
straight	ﻚ ﮎ ﮓ گ	ﻜ ﮏ ﮕ	ﻛ ﮑ ﮔ	ك ڭ گ	ـﻚ ـﮏ	ﻜـ	ﻛـ	ك ک	0643
	ﻞ	ﻠ	ﻟ	ل	ـﻞ	ﻠـ	ﻟـ	ل	0644
	ـﺎ			ا	ـﺎ			ا	0627
	ء	ء	ء	ء					0621

A. Training data set

Training data set consists of the different subsets original data collected from actual documents and synthetic data created using semi automatic methods:

Original data: This set includes pages collected from various documents and consists of roughly 1000 pages, 18,000 lines, 35,000 words, and 252,000 characters (Table 2). Pages in this set are approximately 1400 x 2000 pixels and have 300 dpi resolution. A page contain 20 lines on the average with 12 point font size and 48 points line height. Some original document samples are given in Fig. 2.

Synthetic data: Preparing original data is tedious and time consuming. Therefore we prepared synthetic documents for training in 70 different Arabic fonts using 4 different books.

The data set contains 26,000 pages, 1,300,000 words, 263,000 lines, 78,000,000 characters. Documents in this set contain 42 lines per page and are of 2500x4800 size with 300 dpi resolution 12 pixels font size, 48 pixels line height as shown in Table 2.

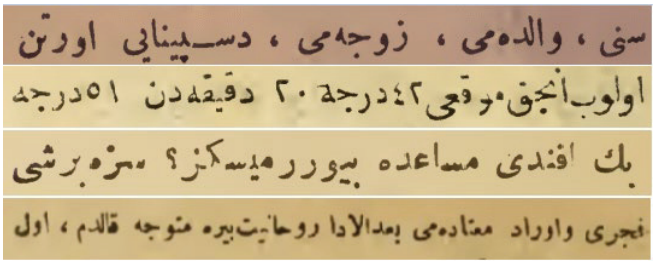


Fig. 2 Sample Ottoman documents in printed naskh

TABLE 2 FREQUENCIES IN DATA SETS

Data set	*Pages*	*Lines*	*Words*	*Characters*
Synthetic	26K	1.3M	26M	130M
Original	1K	18K	35K	252K
Training	27K	1.3M	26M	130M
Test	21	420	3K	23K

B. Test data set

We gathered 21 original pages (samples shown in Fig 2) from different sources for the testing and comparison of OCR tools. These pages contain about 20 lines per page, 7 words and 55 characters per lines on the average. Test documents, the ground truth, the experimental OCR outputs and a test program in Python for calculating character recognition accuracy using difflib are shared at osmanlica.com/test.

V. EXPERIMENTAL RESULTS

We conducted an experimental comparison of Osmanlica.com OCR using the above test set with Tesseract 4.0 [3] Arabic and Persian models pretrained with 400.000 lines and 4,500 different fonts, ABBY FineReader v15 [4], Google Docs [5] ve Miletos [6] as of the second half of 2020. We assumed that the test data weren't used in the training data of the compared models. The experimental results seem to support this assumption. We employed 3 types of character and 2 types of word recognition accuracy metric for comparisons.

A. Character recognition rates

We computed character recognition accuracy on OCR outputs using 3 different texts: raw, normalized, and joined:

Raw: This is the original OCR output without preprocessing or error correction etc. Raw recognition ratios are the lowest, since raw texts usually contain errors and need preprocessing such as text cleaning and normalization etc.

Normalized: When examined closely we see some irregularies and errors in OCR outputs requiring preprocessing and error handling before calculating accuracy. Normalizations involves 3 steps: (i) the removal of extra (white) spaces (ii) the correction of wrong parenthesis direction (Ottoman is written RTL while parenthesis are neutral according to the BIDI algorithm. In certain cases the parenthesis direction in Ottoman text are wrongly determined in the RTL text) and the incorrect use of ھ (Arabic letter heh U+0647) instead of ہ (Arabic letter ae U+06D5) or vice versa, and (iii) replacing all non-standard letters (mostly allographs) with the correct ones. Details can be found in the test.py program shared with the data set. The normalization performed here is obviously different than the standard text normalization in NLP. [7]

Joined: Some letters in Ottoman script don't join with preceding letters (non-joiners). This results in intra word spaces (zero width space) which are often confused with inter word spaces in OCR, which may cause word segmentation problems. [8, 9] In order to see the effects of word segmentation on accuracy, we joined the words in the normalized input and output texts separately into single long strings by removing all whitespace. The resulting strings are then compared with *difflib*, which computes the differences between strings in terms of insertion, deletions and substitutions as shown in the example below, where ↕, ↓, ↑ denote substitution, insert, deletion. * is used as a placeholder. The first and second lines below are the ground truth and the OCR output respectively.

قابينك اوكنده ، سرى ايله قارشى قارشو*يه
↓↕ ↑↑ ↕
قاپينك اوكنده** سرى ايله قارشى قارشى يه

TABLE 3 CHARACTER RECOGNITION RATES (%)

Tool/model	*Raw*	*Joined*	*Normalized*
Tesseract Persian	71,78	79,72	80,11
Abby Finereader	75,57	80,89	80,06
Tesseract Arabic	76,89	81,19	82,25
Miletos	75,76	86,70	86,30
Google Docs	83,24	92,75	91,93
Osmanlica.com	88,64	97,18	95,92

Overall we observe that raw character recognition rates are ~5%-10% lower than the normalized rates, since they lack normalization. In general the joined accuracy is ~1% better than the normalized for Google Docs and Osmanlica.com. On the average Tesseract Persian & Arabic and Finereader produce similar accuracy, while Miletos has ~4% higher accuracy than those. Google Docs outperforms Miletos by ~6% in normalized and joined accuracy and Osmanlica.com beats Google Docs by ~4%. In the end high OCR recognition accuracy rates of ~96%-97% is achieved on original test documents which is quite satisfactory.

The 10 most common OCR errors occored in osmanlica.com OCR output are the deletion of ا and ہ 23 times, the sustitution of '.' for '،' 19 times, ہ for ا 17 times, ك for ڭ 17 times, the insertion of ن 17 times, the deletion of '،' 16 times, the sustitution of ت for ن 14 times, the deletion of '.' 14 times, the insertion of ا 13 times. The 10 characters with the most OCR errors are ن 28 times, ہ 22 times, '،' 16 times, ڭ 17 times, ت 14 times, ا 12 times, ب 9 times, ث 8 times.

The 10 most common substitution errors are ت for ن 14 times, ہ for ا 17 times, '،'for '.' 19 times, ك for ڭ 17 times, ت for ن 7 times, ا for ')' 3 times, ب for پ 7 times, ل for ا 5 times, ث for ن 4 times. Most common deletions (omissions) are ا 23 times, ہ (Arabic letter ae) 23 times, '،' 16 times, '.' 14 times, ب 10 times, م 10 times, ن 9 times, و 9 times, ت 8 times, ل 10 times. Most frequent insertion (fictitous letters) are ن 17

times, ا 13 times, و 13 times, ‘.’ 10 times, ب 8 times, م 8 times, س 7 times, ه (ae) 7 times, ت 5 times, ل 4 times.

TABLE 4 ERROR AND LETTER FREQUENCIES

Letter	*Code*	*Error*	*Frequency*
ا	0627	0.006 (12)	0.134 (1984)
ب	0628	0.015 (9)	0.041 (610)
ت	062a	0.027 (14)	0.035 (518)
ث	062b	0.235 (8)	0.002 (34)
ج	062c	0.024 (4)	0.011 (168)
ح	062d	0.021 (4)	0.013 (189)
خ	062e	0.030 (3)	0.007 (101)
د	062f	0.006 (5)	0.061 (906)
ذ	0630	0.000 (0)	0.002 (36)
ر	0631	0.001 (2)	0.093 (1373)
ز	0632	0.022 (5)	0.015 (225)
س	0633	0.000 (0)	0.027 (403)
ش	0634	0.008 (2)	0.018 (260)
ص	0635	0.008 (1)	0.009 (130)
ض	0636	0.017 (1)	0.004 (59)
ط	0637	0.000 (0)	0.007 (103)
ظ	0638	0.000 (0)	0.003 (49)
ع	0639	0.007 (2)	0.02 (299)
غ	063a	0.018 (2)	0.007 (110)
ف	0641	0.026 (6)	0.016 (235)
ق	0642	0.013 (6)	0.03 (445)
ك	0643	0.003 (2)	0.041 (609)
ل	0644	0.007 (9)	0.081 (1203)
م	0645	0.003 (3)	0.061 (907)
ن	0646	0.025 (28)	0.074 (1102)
ه	0647/d5	0.022 (22)	0.066 (982)
و	0648	0.005 (6)	0.079 (1167)
ي	0649/A4	0.000 (0)	0.000 (0)
ء	0621	0.000 (0)	0.000 (3)
ڭ	06ad	0.258 (17)	0.004 (66)
گ	06af	0.217 (5)	0.002 (23)
چ	0686	0.048 (4)	0.006 (84)
ژ	0698	0.000 (0)	0.000 (1)
پ	067e	0.026 (1)	0.003 (38)
. -	06d4/46	0.049 (7)	(142) 0.010
،	060c	0.217(20)	(92) 0.006
؛	061b	0.400 (4)	(10) 0.001
؟	061f	0.000 (0)	(17) 0.001
:	003a	0.000 (0)	(14) 0.001
-	002d	0.000 (0)	(0) 0.000
)	0028	0.000 (0)	(0) 0.000
(	0029	0.000 (0)	(94) 0.006
٠	0660	0.400 (2)	(5) 0.000
١	0661	0.111 (1)	(9) 0.001
٢	0662	0.000 (0)	(0) 0.000
٣	0663	0.000 (0)	(5) 0.000
٤	0664	0.000 (0)	(3) 0.000
٥	0665	0.143 (1)	(7) 0.000
٦	0666	0.000 (0)	(3) 0.000
٧	0667	0.000 (0)	(0) 0.000
٨	0668	0.000 (0)	(5) 0.000
٩	0669	0.000 (0)	(0) 0.000

B. Word recognition rates

We calculated the word recognition accuracy on raw and normalized ground truth and OCR outputs. Text normalization is performed as described before. The results are shown in Table 5 and Table 6.

TABLE 5 WORD RECOGNITION ERRORS ON RAW TEXT

Tool/Model	*Raw*				
	Error	*Corrects*	*Sub.*	*Ins.*	*Del.*
Tess Arabic	85%	712	2723	184	136
Finereader	84%	665	2542	105	364
Tess. Persian	87%	555	2540	116	476
Miletos	86%	115	523	2	170
Google Docs	78%	1320	2196	538	55
Osmanlica.com	64%	1561	1896	298	114

TABLE 6 WORD RECOGNITION ERRORS ON NORMALIZED TEXT

Tool/Model	*Normalized*				
	Error	*Corrects*	*Sub.*	*Ins.*	*Del.*
Tess. Arabic	78%	979	2433	204	158
Finereader	80%	799	2396	117	375
Tess. Persian	80%	856	2207	147	507
Miletos	72%	224	412	172	4
Google Docs	62%	1892	1602	560	76
Osmanlica.com	42%	2369	1066	320	135

All raw word recognition error rates given in Table 5 are higher than 50 percent and therefore has no significance. Nevertheless Google docs and Osmanlica.com performs significantly better than the others. In the tables, substitutions (~80%-90%) represent false characters, insertions (5%-10%) represent non-existent characters and deletions (5%-10%) represent unrecognized characters in the text.

As expected, the normalized word recognition errors shown in Table 6 are roughly 4%-6% lower than the raw errors, with the exception of Osmanlica.com that has a 22% substantial improvement over the raw error. The 58% word recognition accuracy of Osmanlica.com in a way corroborates the very high characters recognition accuracy of 97%.

VI. CONCLUSIONS

We presented an OCR tool for Ottoman document in printed naksh font. The tool was developed as part of an project named “End-to-End Conversion of Ottoman Documents to Modern Turkish” This tool uses deep learning LSTM models trained with a data set containing original and synthetic documents. We carried out an experimental comparison of the tool with 5 other well known tools or models using a test set of 21 pages from different Ottoman documents. With 88,64% raw, 95,92% normalized and 97,18% joined character recognition accuracy, Osmanlica.com achieves much higher OCR performance than the others. Osmanlica.com also produced 58% word recognition accuracy which is the only rate over 50%. We shared the test data set at osmanlica.com/test for the verification of the results.

REFERENCES

[1] M. Ergin, Osmanlıca Dersleri. İstanbul, Türkiye: Boğaziçi yayınları, (2020)

[2] Y. İ. GENÇ et al, Başbakanlık Osmanlı Arşivi Rehberi, Başbakanlık Basımevi, İstanbul (2020)

[3] R. Smith, An Overview of the Tesseract OCR Engine, 9th Int. Conf. on Document Analysis and Recognition (ICDAR) (2007)

[4] Abby Finereader, pdf.abbyy.com, (2021)

[5] Google Docs, docs.google.com, (2021)

[6] Miletos OCR, www.miletos.com, (2021)

[7] Text normalization, https://en.wikipedia.org/wiki/text_ normalization, (2021)

[8] Duygulu, P., Arifoglu, D. & Kalpakli, M. Cross-document word matching for segmentation and retrieval of Ottoman divans. Pattern Anal Applic 19, 647–663 (2016)

[9] E. F. Bilgin, Ottoman-Turkish Transliteration, Masters Thesis, İstanbul Fatih University, (2012)

[10] F. T. Yarman Vural and A. Atici, A Heuristic Algorithm for Optical Character Recognition of Arabic Script, SPIE 2787, 725 (1996)

[11] Alper Atici, Fatos T. Yarman Vural, A heuristic algorithm for optical character recognition of Arabic Script, Signal Processing 62, 8799 (1997)

[12] E. Öztop, A. Y. Mülayim, V. Atalay and F.T. Yarman Vural, Repulsive Attractive Network for Baseline Extraction on Document Images, IEEE, 3184 (1998)

[13] E. Öztop, A. Y. Mülayim, V. Atalay and F.T. Yarman Vural, Repulsive Attractive Network for Baseline Extraction on Document Images, Signal Processing 75, 1 10 (1999)

[14] E. F. Can and P. Duygulu, A line based representation for matching words in historical manuscripts, Pattern Recognition Letters 32, 11261138 (2011)

[15] H. Adıgüzel, E. Sahin and P. Duygulu, A Hybrid Approach for Line Segmentation in Handwritten Documents, IEEE, 503 (2012)

[16] H. Adıgüzel, P. Duygulu, and M. Kalpaklı, Line Segmentation of Ottoman Documents, IEEE (2012)

[17] P. Sankar, C. V. Jawahar and R. Manmatha, Nearest Neighbor based Collection OCR, (DAS), (2010)

[18] Z. Yalniz and R. Manmatha, A Fast Alignment Scheme for Automatic OCR Evaluation of Books, IEEE, (2011)

[19] Z. Yalniz and R. Manmatha, An Efficient Framework for Searching Text in Noisy Document Images, IAPR, (2012)

[20] Chen, S. Kapoor and R. Bahatia, Intelligent Systems for Science and Information from the Science and Information Conf. 2013, (2014)

[21] Z. Kurt, H. I. Turkmen and M. E. Karsligil, Linear Discriminant Analysis in Ottoman Alphabet Character Recognition, (2009).

[22] Öztürk, S. Güneş, Y.Özbay, Multifont Ottoman Character Recognition, 7th IEEE Int. Conf. on Electronics Circuits and System (ICECS) (2000)

[23] N. Kilic, P. Görgel, O. N. Ucan and A. Kala, Multifont Ottoman Character Recognition Using Support Vector Machine, IEEE, 328 (2008)

[24] Onat, F. Yildiz and M. Gündüz, Ottoman Script Recognition Using Hidden Markov Model, World Academy of Science, Engineering and Technology 2, 630 632 (2008)

[25] S. Yalniz, I. S. Altıngövde, U. Güdükbay and Ö. Ulusoy, Ottoman Archives Explorer: A Retrieval System for Digital Ottoman Archives, ACM Journal on Computing and Cultural Heritage 2(3), 8 (2009)

[26] T. Bluche et al, A2iA Arabic Handwritten Text Recognition System at OpenHaRT2013 Evaluation. Int. Workshop on Document Analysis Systems (DAS) (2014)

[27] V. Pham, T. Bluche, C. Kermorvant, J. Louradour, Dropout improves recurrent neural networks for handwriting recognition. Int. Conf. on Frontiers in Handwriting Recognition (ICFHR) (2014)

[28] Moysset, R. Messina, C. Kermorvant, A Comparison of Recognition Strategies for Printed / Handwritten Composite Documents. Int. Conf. on Frontiers in Handwriting Recognition (ICFHR) (2014)