Mühendislik Bilimleri ve Tasarım Dergisi
12(2), 392–402, 2024
e-ISSN: 1308-6693

Journal of Engineering Sciences and Design
DOI: 10.21923/jesd.1383926

Araştırma Makalesi | Research Article

OTTOMAN CHARACTER RECOGNITION ON PRINTED DOCUMENTS USING DEEP LEARNING

Ali Alper DEMİR[1*], Ufuk ÖZKAYA[1]

[1] Suleyman Demirel University, Faculty of Engineering and Natural Sciences, Department of Electrical and Electronics Engineering, Isparta, Turkey

Keywords
Ottoman Documents, Character Recognition, Document Analysis, Deep Learning, Ottoman Turkish.

Abstract
In this study, a deep learning-based method is developed for character detection and recognition in printed Ottoman documents. The character detection and recognition problem are considered as an object detection problem and for this purpose, an Ottoman character recognition model is developed based on the YOLO model, which is one of the most successful methods in object detection. In addition, in this study, a dataset consisting of Ottoman document images is created in which each character in the document images is marked. Data augmentation techniques are applied to improve the accuracy of character recognition and the robustness of the method. The Ottoman character recognition network was then trained using this dataset. The trained network model was tested with the test images in the dataset. The performance evaluation of the model was performed by calculating the average precision metric, which is frequently used in the literature. The average precision value was calculated for 34 character classes in the dataset and the results were interpreted in terms of the pros and cons of the method. The results show that the proposed method can detect and recognize characters in printed Ottoman documents with great accuracy, with a weighted average precision of 98.71%.

DERİN ÖĞRENME KULLANARAK MATBU DOKÜMANLARDAKİ OSMANLICA KARAKTERLERİN TANINMASI

Anahtar Kelimeler
Osmanlıca Dokümanlar, Karakter Tanıma, Doküman Analizi, Derin Öğrenme, Osmanlı Türkçesi.

Öz
Bu çalışmada matbu Osmanlıca dokümanlardaki karakterlerin tespiti ve tanınmasına yönelik derin öğrenme tabanlı bir yöntem geliştirilmiştir. Karakter tespit ve tanıma problemi bir nesne tespit problemi olarak ele alınmış ve bu amaçla nesne tespitinde en başarılı yöntemlerden biri olan YOLO modeli temel alınarak Osmanlıca karakter tanıma modeli geliştirilmiştir. Ayrıca bu çalışmada, Osmanlıca doküman imgelerinden oluşan ve doküman imgelerindeki her bir karakterin işaretlendiği bir veri kümesi oluşturulmuştur. Karakter tanıma doğruluğunun artırılması ve yöntemin gürbüzlüğünün sağlanması için veri çoğaltma teknikleri uygulanmıştır. Daha sonra bu veri kümesi kullanılarak Osmanlıca karakter tanıma ağı eğitilmiştir. Eğitilen ağ modeli veri kümesindeki test imgeleri ile test edilmiştir. Modelin performans değerlendirmesi, literatürde sıklıkla kullanılan ortalama kesinlik metriği hesaplanarak yapılmıştır. Veri kümesindeki 34 karakter sınıfı için ortalama kesinlik değeri hesaplanmış ve sonuçlar yöntemin artı ve eksileri açısından yorumlanmıştır. Elde edilen sonuçlar değerlendirildiğinde, önerilen yöntemin matbu Osmanlıca belgelerdeki karakterleri büyük bir doğrulukla, %98,71 ağırlıklı ortalama kesinlik değeri ile, tespit edip tanıyabildiği görülmüştür.

Alıntı / Cite
Demir, A. A., Özkaya, U., (2024). Ottoman Character Recognition on Printed Documents Using Deep Learning, Mühendislik Bilimleri ve Tasarım Dergisi, 12(2), 392-402.

Yazar Kimliği / Author ID (ORCID Number)
A. A. Demir, 0000-0001-5250-0590
U. Özkaya, 0000-0002-3520-1975

Makale Süreci / Article Process	
Başvuru Tarihi / Submission Date	31.10.2023
Revizyon Tarihi / Revision Date	09.05.2024
Kabul Tarihi / Accepted Date	05.06.2024
Yayım Tarihi / Published Date	30.06.2024

[*] İlgili yazar / Corresponding author: alidemir@sdu.edu.tr, +90-246-211-1747

OTTOMAN CHARACTER RECOGNITION ON PRINTED DOCUMENTS USING DEEP LEARNING

Ali Alper DEMİR[1†], Ufuk ÖZKAYA[1]
[1]Suleyman Demirel University, Faculty of Engineering and Natural Sciences, Department of Electrical and Electronics Engineering, Isparta, Turkey

Highlights

- The proposed deep learning-based approach in this study is a powerful method for detecting and recognizing characters in printed Ottoman documents, achieving a weighted average precision score of 98.71%, underlining the potential for more effective use of historical Ottoman documents.
- In particular, the recognition accuracy for low-frequency characters added to the Ottoman alphabet from Persian and Arabic is lower than for other characters, and to solve this problem, it is suggested to retrain the model by increasing the number of characters with few samples.
- The method produces successful results on printed documents, in addition to this, in future studies the proposed method can be extended to Ottoman manuscript documents, can be converted into a word detection tool, and can be used as a character recognition stage in automatic transcription studies from Ottoman to Turkish.

Purpose and Scope

The main objective of this paper is to develop a method for the accurate detection and recognition of characters on printed Ottoman documents. This is important because Ottoman documents often contain historical and cultural information, and automating the process of character recognition can make these documents more accessible and usable.

Design/methodology/approach

The objectives were achieved by creating a labeled dataset of Ottoman document images, developing a YOLO-based deep learning model for recognizing Ottoman characters on printed documents and utilizing data augmentation to increase detection and recognition accuracy.

Findings

The paper indicates that the deep learning-based Ottoman character detection and recognition model developed achieved high accuracy in detecting and recognizing characters in printed Ottoman documents, with a weighted average precision of 98.71%. The article also considers areas for further development, practical applications, and avenues for future research, such as tackling difficulties linked to specific characters, expanding the model to manage manuscript documents, and transforming it into a word-spotting tool and an automated transcription system from Ottoman into Turkish.

Research limitations/implications

Suggestions for future research include improving the manuscript dataset, extending the model's character detection and recognition capabilities to Ottoman manuscript documents, improving the model's performance when it generates multiple predictions for certain characters, and transforming the method into a word detection tool for keyword-based searches. The recognition success of some characters with low frequency of use in the Ottoman alphabet from Arabic and Persian is low, but it is thought that this problem can be solved by retraining the model with more data for these characters.

Social Implications

This study can speed up the research on printed Ottoman documents and ensure their effective use.

Originality

In this paper, a new dataset for character detection and recognition in printed Ottoman documents is created and a deep learning-based model is developed using this dataset.

† Corresponding author: alidemir@sdu.edu.tr, +90-246-211-1747

1. Introduction

The Ottoman Empire existed for more than 6 centuries and influenced many parts of the world during its existence, leaving behind millions of archival documents from all fields that attracted the attention of researchers and historians from many countries. Since documents become worn and irritated over time, they are scanned and stored digitally. Some of the documents in the archives are handwritten and some are printed documents. To speed up the research being carried out by scientists, researchers, and historians from many parts of the world, these documents need to be processed by computers and researchers need to be able to benefit from them quickly. It is a time-consuming process for researchers to manually search for the keywords related to the subject they are interested in, in which document, and on which page, among millions of archival documents. Spotting the keywords from the documents in the archives will speed up the research considerably. Furthermore, translating Ottoman documents into other languages will make them accessible to more people who cannot read Ottoman, thus increasing the usage of the archives. Although using image processing and artificial intelligence methods have improved the understanding of Ottoman documents to some extent, there are still many unsolved issues. Overall, there has not been enough advancement to effectively exploit these documents. Especially the studies in the literature on the computer readability of manuscript Ottoman documents are insufficient. In the literature, studies have generally been carried out on computer reading of Ottoman printed documents in naskh font.

To overcome the Ottoman character recognition problem, several learning-based methods have been proposed. Since there are no common datasets prepared and published for this problem, the authors have mostly implemented methods to their hand-crafted datasets. Also, they created and used synthetic data as well as real data to train the proposed models. In the early studies presented in the literature to solve the Ottoman character recognition problem, methods such as artificial neural network (Öztürk et al., 2000; Gorgel et al., 2009), hidden Markov model (Onat et al., 2006), linear discriminant analysis (Kurt et al., 2007; Kurt et al., 2009), support vector machines (Kilic et al., 2008) and a graph-based system (Yalniz et al., 2009) were used.

Nowadays, with the increasing amount of data and computer processing power, deep learning and machine learning-based studies have begun to be applied to Ottoman documents. Bilgin Taşdemir (2023), conducted a study on deep learning-based character recognition in printed documents in Ottoman Turkish naskh font. A synthetic dataset was created, and these data were augmented with data augmentation techniques to train a hybrid model consisting of a convolutional neural network and bidirectional long short-term memory. The model was then adapted to real data by applying transfer learning. Similarly, in a different study, a web-based OCR system was introduced for Ottoman printed documents written in naskh font (Dölek and Kurt, 2023). In this system, CNN and RNN-based deep neural network models were used. Three different datasets were created for training the model: original, synthetic, and hybrid, and the trained model was tested with 21 Ottoman document images and compared with OCR tools available in the literature. Altun (2022) discussed previous attempts to use OCR technology on Ottoman documents and the challenges encountered, as well as potential solutions. The existing studies have been successful in recognizing only printed naskh font within document images. Doğru (2016), used an open-source optical character recognition platform (Tesseract, 2023) to recognize Ottoman characters. In the study by Küçükşahin (2019), two convolutional neural networks of different complexity were trained with the generated printed character dataset, and the relationship between recognition rates and network complexity was evaluated. Mondal et al., (2022), used the YOLO (You Only Look Once) v3 model, one of the object detection algorithms, to train an English word recognition model with only 1200-word images, in fact with a small number of training data. Majid and Smith (2019) used the Faster R-CNN algorithm, another object detection algorithm, for Bangla word recognition.

In most cases, a large amount of data is required to successfully train deep learning-based methods for character recognition. As far as we know, the only Ottoman character dataset available in the literature, (Uzun and Özer, 2022), contains relatively few character data. In this Ottoman character dataset, there are only 3894 characters in total, 1371 as Talik font, 411 as Rika font, 1974 characters as printed font, and 138 mixed characters. The character images in the dataset are in binary form. The lack of data quantity is a problem in the training of a deep learning-based character recognition model. In addition, this dataset cannot be used directly for character detection, it can only be used for applications such as character classification and character recognition. Therefore, a comprehensive dataset is needed for training the deep learning model to be developed for character recognition in Ottoman documents. Therefore, in this study, a dataset for character recognition and detection in Ottoman Turkish is created. In addition, an Ottoman character recognition network model is trained on this dataset and the character detection and recognition success of the model is tested.

2. Material and Method

2.1. Dataset

In order to create the data set, the book named "Osmanlı Türkçesi Kolay Okuma Metinleri – 1" (Uçar, 2021) was scanned and the raw data set was obtained as consisting of document images. Then, smaller images that have various dimensions were cropped from these document images. The characters in the cropped images were selected using the MATLAB Image Labeler and appropriate labels were assigned to the selected letters. In this data set, the number of label classes, which is the number of letters in the Ottoman alphabet, is determined as 34, including the letter لا lamelif. The plain forms of the characters in the dataset are shown in Figure 1.

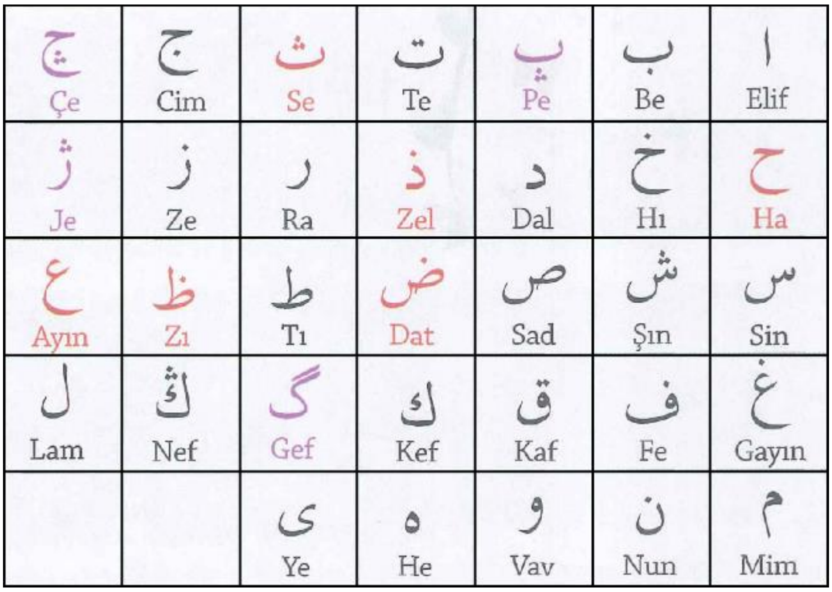

Figure 1. Images of the letters in the Ottoman character dataset (Uçar, 2021)

These writing styles of letters undergo significant changes in the text. The letterforms shown above are only preserved if they are the final sound in the word after some letters. In Ottoman Turkish, letters are written from right to left, and some letters do not combine with the next letter but are in plain form. Some letters are written together with the letter that comes before and after them, and their simple form also changes. There are differences between the way letters are written at the beginning, middle, and end. There are many rules about how letters will be written when they are combined with other letters (Tulum, 2014). In the created dataset, there are many examples of the shape changes of the letters (Figure 2).

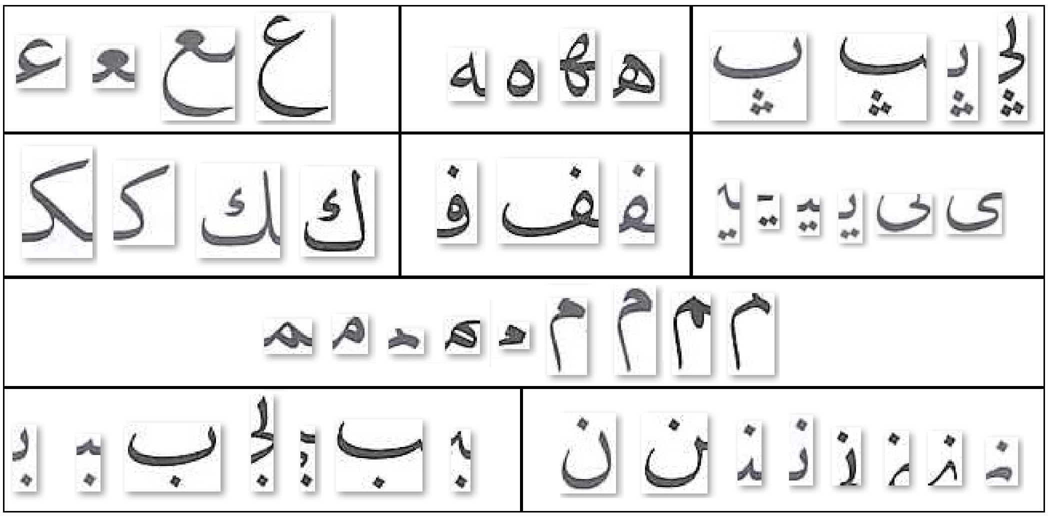
Figure 2. Examples of shape changes for some characters in the dataset

To create the dataset, 38 document pages were selected from the scanned "Kolay Okuma Metinleri-1" (Uçar, 2021) book. The selected document pages are saved as images. Then, from these 38 document images, image fragments with smaller dimensions were cut and saved. Thus, a total of 125 document images were obtained by cropping 38 document images. The original document image page is given on the left in Figure 2, and the document images that were cropped from this page are given on the right. 125 cropped images were imported to the MATLAB Image Labeler application. Label classes were created for each character class and all the characters in the document were labelled. For this, firstly, the character label is selected from the menu on the left, and then the bounding box surrounding the relevant character is drawn. The visual representation of the labeling process of the characters is given in Figure 3.

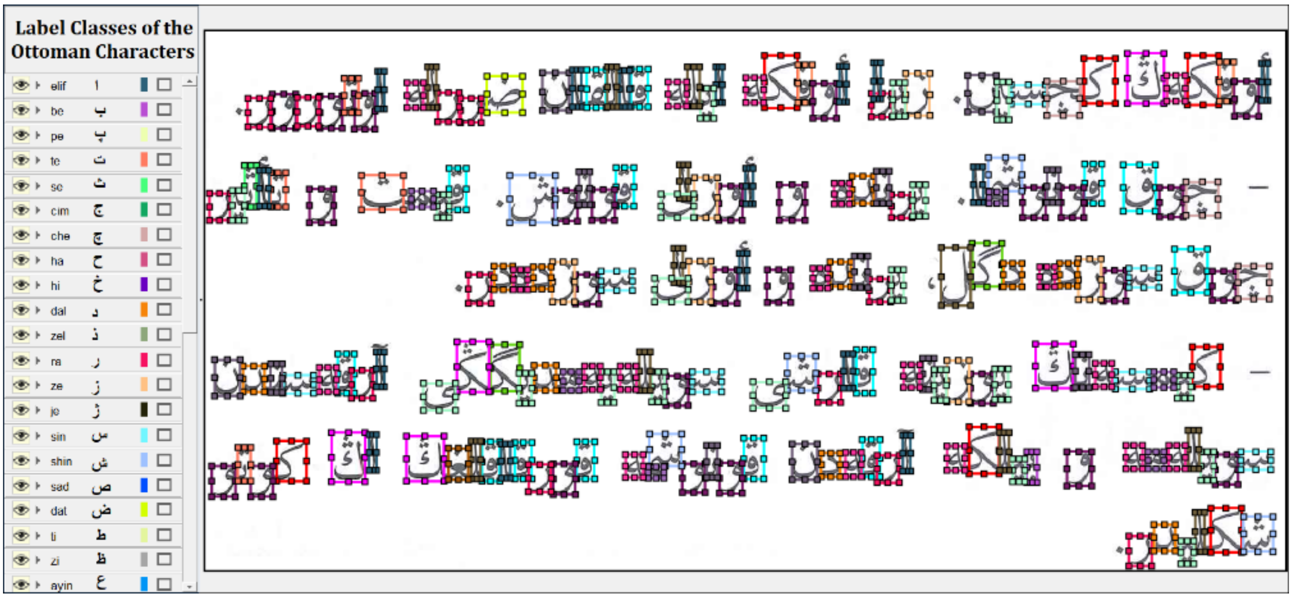

Figure 3. The process of labeling the characters

The labeling process was completed for 125 images and 34 different characters, and ground truth data including bounding boxes and label information was created. The bounding box information includes the x and y coordinates of the character's upper left corner point, as well as the lengths in the x and y directions. The total number of labeled characters in the dataset is 22,180. The number of characters in the dataset with an unbalanced distribution in terms of ratio is given in Table 1. The letters ث se, ذ zel, ض dad, and ظ zı in the Ottoman alphabet were added from Arabic, while the letters ژ je, پ pe and چ çe were added from Persian. Therefore, the frequency of use of these letters in Ottoman is not as high as others. This can be seen in the number of characters in the dataset in Table 1.

Table 1. Number of the characters in the dataset

ا **elif** 2469	ب **be** 821	پ **pe** 142	ت **te** 605	ث **se** 9	ج **cim** 295	چ **çe** 263
ح **ha** 141	خ **hı** 92	د **dal** 1248	ذ **zel** 18	ر **ra** 1753	ز **ze** 340	ژ **je** 4
س **sin** 579	ش **şın** 469	ص **sad** 171	ض **dad** 23	ط **tı** 146	ظ **zı** 20	ع **ayın** 143
غ **gayın** 192	ف **fe** 159	ق **kaf** 655	ك **kef** 790	گ **gef** 140	ڭ **nef** 415	ل **lam** 1231
م **mim** 985	ن **nun** 1540	و **vav** 1798	ه **he** 1312	ى **ye** 3032	لا **lamelif** 180	**Total** **22.180**

2.2. Deep Learning-based Ottoman Character Recognition Method

In this study, character detection and recognition problems are considered as an object detection problem. In this respect, each character is treated as an object and the location of the characters in the document images and the character class to which they belong are estimated. In order to detect and recognize the Ottoman characters, a character recognition network (CRN) is trained based on the YOLOv4 (Bochkovskiy et al., 2020) model, which produces successful results in object detection in the literature.

In the object detection task, besides determining the class of the object in the given image, the location of the object is also estimated. Gkioxari et al. (2014), the R-CNN model, which finds objects in the image and predicts their positions, has been proposed. In the R-CNN model, the image is divided into many regions, and a convolutional neural network is applied to each region. Therefore, the training process of the network and the prediction time of the trained network are quite time-consuming. Therefore, such models cannot be used in real-time object detection. Girshick (2015), accelerated the R-CNN model and increased its success rate and published the model as Fast R-CNN. Ren et al. (2015), improved the previous models and developed the Faster R-CNN model. However, these methods cannot detect real-time objects. All three models are slow to train, training takes place in multiple stages (zone recommendation, classification), and the detection of objects is slow. Afterward, methods such as YOLO (Redmon et al., 2016) and SSD (single shot detector) (Liu et al., 2016), which can also detect real-time objects, have been developed. YOLO is an algorithm for fast object detection using convolutional neural networks. With the YOLO algorithm, the class and object coordinates of all objects in the image are estimated by passing the image through the neural network at once.

In this work, a single-stage character detection algorithm is used to perform real-time character search on documents. In addition, since the YOLO algorithm is more successful in detecting small objects than other object recognition algorithms, and since there are small-sized characters in our problem, an Ottoman character recognition model has been developed in this study based on YOLOv4 architecture. Our model consists of three parts: backbone, neck, and head. The backbone of the CRN model is a feature extraction network that computes feature maps from the input images. The head and the backbone are connected by the neck. There are two main components of the neck: a path aggregation network and a spatial pyramid pooling (SPP) module. The neck sends feature maps from different network layers from the backbone as input to the head by concatenating them. In the head part of the network model, bounding boxes, objectness scores, and classification scores are estimated by processing the obtained features. The backbone of the Ottoman CRN model is CSP-DarkNet-53, which is used to extract features from the input images. The backbone part of the network model consists of five different residual block modules. The neck part of the network model combines and processes the feature map outputs of the residual blocks. On the neck, the SPP module takes the maximum pooled outputs of the low-resolution feature map and extracts the most effective features for the character recognition task. In the Ottoman CRN model, combining the feature maps increases the receptive field. Thus, the detection rate of small-sized characters also increases. High-resolution feature maps are merged with the feature maps from the SPP by using a path aggregation network (PAN). A collection of aggregated feature maps for use in character predictions is produced by the PAN module of the model. In the head part of the Ottoman CRN model, there are three modules to be used for character prediction, each module containing a Yolov3 network. The Ottoman CRN model is illustrated by a block diagram in Figure 4.

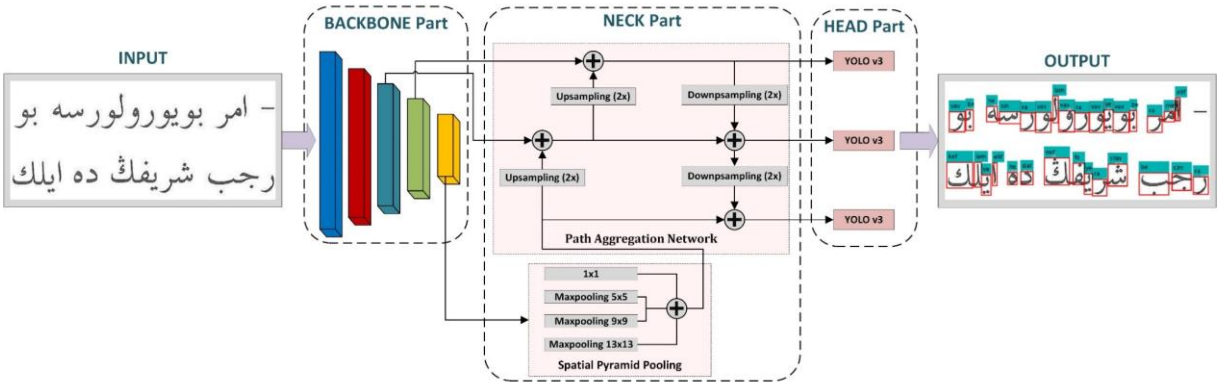

Figure 4. Block diagram of the Ottoman CRN model

To train the proposed method and evaluate its recognition performance, the dataset is randomly divided into training, validation, and test data. Out of a total of 125 images in the dataset; training, validation, and test data were allocated with ratios 80% , 10%, and 10% respectively. Therefore, 100 images were selected for training, 13 for validation, and 12 for testing, all chosen randomly from the dataset. In Figure 5, two text lines from a sample document image from the dataset are shown in the top row, and the corresponding character bounding boxes, drawn with red rectangles, are shown in the bottom row.

Figure 5. Bounding box representation of the characters in the dataset

Document images with varying sizes within the dataset were resized to $608 \times 1152 \times 3$ for CRN training. Furthermore, the input image size of the network was also adjusted to be the same size for CRN training. Depending on the size of the characters in the training data, 36 anchor boxes were set. To improve the training accuracy, data augmentation was performed. Data augmentation was applied to only the document images in the training set by applying random horizontal translation with 50% probability, random scaling by 10%, and color jittering in HSV color space. The main motivation for random scaling is to generate new expanded and contracted characters. This process allows the model to adapt to characters of different sizes. Color jittering was applied to ensure that the model correctly predicts characters for different page features. Horizontal translation helps to generate the characters in different sentence properties. Examples of document images obtained after data augmentation are given in Figure 6.

Figure 6. Examples of document images obtained after data augmentation

The Ottoman CRN architecture was trained for 100 epochs with a constant learning rate of 0.001 and using a minibatch size of 1. Since the amount of training data is relatively small, the minibatch size was chosen as 1 to train the model for each sample. The learning rate was chosen as 0.001 to prevent overfitting and to avoid getting stuck in the local minimum. The epoch number was assigned according to the success of the model in the validation dataset.

3. Experimental Results

Since the Ottoman characters recognition problem is considered as an object recognition problem, the average precision metric, which is the most widely used metric in the literature to evaluate the performance of object detection methods, was used to test the performance of the trained Ottoman CRN model. Average precision (AP) is calculated as the average of the precision values corresponding to different precision values on the precision-recall curve for each class. The calculation of precision and recall metrics are given in Equation (1) and Equation (2), respectively. Here *TP* stands for true positive, *FP* for false positive, and *FN* for false negative.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$AP = \sum_{k=0}^{k=n-1} [Recall(k) - Recall(k+1)] \times Precision(k) \tag{3}$$

$$wAP = \frac{\sum_{i=1}^{N}(AP_i \times n_i)}{\sum_{i=1}^{N} n_i} \tag{4}$$

Table 2 shows the number of characters in the test set (#C) and AP values for each character class. Accordingly, the weighted average precision value for a total of 2108 characters in the test set was calculated as 98.71%. Here, due to the imbalance in the number of characters in the dataset, the detection rate for the character ذ zel, which has 1 character in the test set, and for the characters ث se and ظ zı, which have 2 characters each, was calculated as a low value as expected. In addition, the average precision value was calculated as 0 for the character ژ je, which was never found in the test set. To evaluate the general success of the method in detecting and recognizing Ottoman characters, the weighted average precision value, which takes into account the number of characters in the character classes, was calculated. The weighted average precision (wAP) value was calculated as 98.71% by multiplying the number of characters in the character classes and their corresponding average precision values and dividing by the total number of characters in the test set. The obtained results indicate that the Ottoman CRN model performs successfully in Ottoman character detection and recognition.

Table 2. The performance of the Ottoman CRN model for each character class in the test set

Character	ا **elif**	ب **be**	پ **pe**	ت **te**	ث **se**	ج **cim**	چ **çe**
#C	250	86	14	80	2	24	20
AP	1	0.9871	1	0.9869	0.5	0.8947	0.9955
Character	ح **ha**	خ **hı**	د **dal**	ذ **zel**	ر **ra**	ز **ze**	ژ **je**
#C	26	9	113	1	177	42	0
AP	1	0.6323	1	0	0.9976	0.9507	0
Character	س **sin**	ش **şın**	ص **sad**	ض **dad**	ط **tı**	ظ **zı**	ع **ayın**
#C	42	36	21	5	16	2	13
AP	0.9956	1	0.6567	1	1	0.5	1
Character	غ **gayın**	ف **fe**	ق **kaf**	ك **kef**	گ **gef**	ڭ **nef**	ل **lam**
#C	17	14	46	65	12	35	115
AP	1	0.9286	1	0.9998	1	1	0.9913
Character	م **mim**	ن **nun**	و **vav**	ه **he**	ی **ye**	لا **lamelif**	**Total #C**
#C	103	131	178	121	272	20	**2108**
AP	0.9805	0.9923	1	1	0.9960	1	**wAP : 0.9871**

The resulting confusion matrix is given in Figure 7. As seen from the confusion matrix, the proposed method can powerfully detect and recognize the characters. Moreover, the confusion matrix also provides important information about which characters the model predicts incorrectly and which pairs of similar characters cause errors. The analysis of the confusion matrix reveals that errors in character predictions generally occur among characters that share a very similar shape.

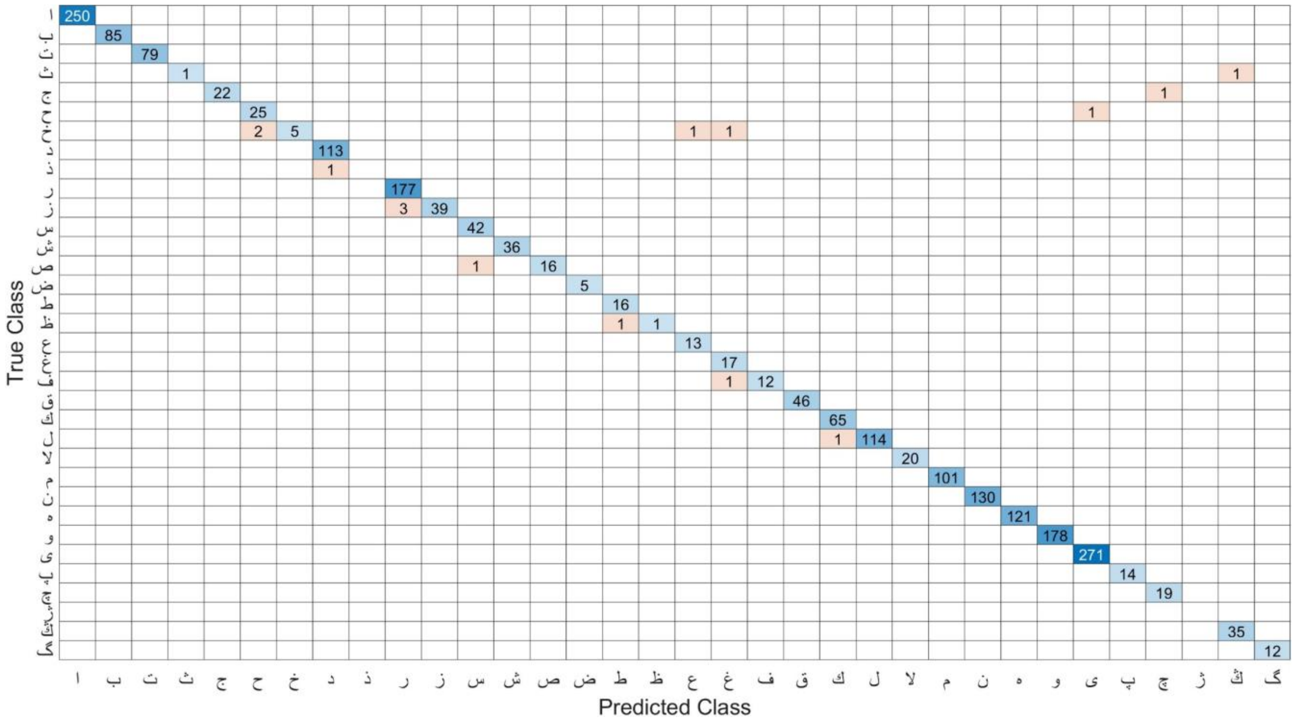

Figure 7. Confusion matrix of the proposed method for the test data

The output image containing the character predictions produced by the Ottoman CRN model against a sample test image given as input to the trained model is given in Figure 8. In this figure, the character bounding boxes in the exact reference are drawn with green dashed rectangles, and the bounding boxes of the characters predicted by the model are drawn with red rectangles. Additionally, the class of characters predicted by the model is printed above the red rectangles.

Figure 8. Output image containing character predictions of the Ottoman CRN model

Some of the characters predicted incorrectly by the Ottoman CRN model are given in Figure 9. Here, each box contains a different word image and the character predictions of the CRN model against these word images. In the Ottoman word images at the top of the boxes show only the characters for which the classification score of the CRN model is below 0.7, while the word images at the bottom show all the character predictions of the model. In other words, the model made two predictions for one character for the word images given in Figure 6. Since the highest classification score among these predictions belongs to the correct character, it is thought that false predictions can be eliminated by the multiple predictions are detected with post-processing steps and the high-scoring prediction is selected. In addition, as can be seen in Table 2, some of the characters in the dataset contain very few samples. It is thought that if the number of samples is increased for characters that are fewer in the data set than others, the model can detect and recognize these characters with high accuracy.

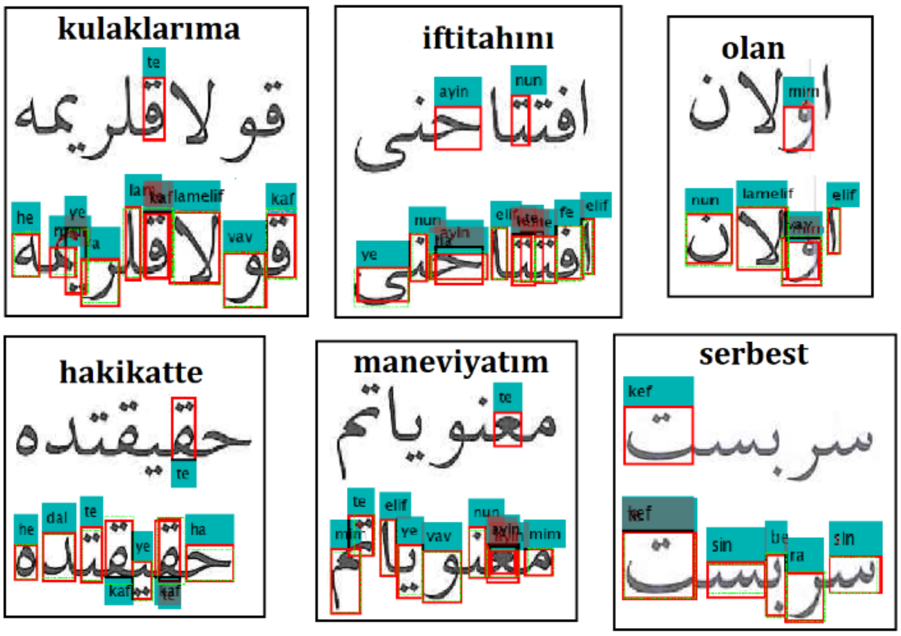

Figure 9. Images of some characters that the model predicted incorrectly

4. Result and Discussion

In this study, a deep learning-based model that can perform character detection and character recognition on Ottoman printed documents has been developed. For this purpose, a dataset containing Ottoman printed texts was created and the Ottoman CRN model was trained with this dataset. The Ottoman CRN model was used to detect and recognize the characters in the images of Ottoman documents in the test set with high accuracy. Due to the low frequency of use of some characters from Persian and Arabic in Ottoman Turkish, the recognition rate of some of these characters was low. It is thought that this problem can be solved by retraining the model by increasing the amount of data for characters with few examples. The proposed method produces successful results on printed documents, and there are other studies on Ottoman manuscript documents in the literature. However, to the best of our knowledge, computer vision and artificial intelligence applications have not been applied for the interpretation of Ottoman manuscript documents. In future studies, it is planned to create a manuscript dataset, train the model with that dataset, and perform character detection and recognition in manuscript documents. The model made more than one prediction for some characters. The success of the model will be further improved by identifying the characters in this situation and eliminating the predictions with a classification score below a certain threshold value. The proposed method can be evolved into a word spotting method by finding the characters belonging to the searched keyword in the Ottoman document images. Thus, the problem of depending on the dictionary in word spotting studies will be prevented. In addition, this method can be used as a character recognition stage at the beginning of automatic transcription studies from Ottoman to Turkish. After the characters in the text are recognized by the proposed method, words can be formed from the recognized characters and translated into modern Turkish.

Conflict of Interest

No conflict of interest was declared by the authors.

References

Altun, H. O. (2022). Osmanlı Türkçesi araştırmalarında optik karakter tanıma teknolojisinin kullanımı. Başkent 3. Uluslararası Multidisipliner Bilimsel Çalışmalar Kongresi, 23-25 Eylül 2022.

Bilgin Tasdemir, E. F. (2023). Printed Ottoman text recognition using synthetic data and data augmentation. International Journal on Document Analysis and Recognition (IJDAR), 1-15.

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Doğru, M. (2016). Ottoman-Turkish Optical Character Recognition and Latin Transcription (Master's thesis, Ankara Yıldırım Beyazıt Üniversitesi Fen Bilimleri Enstitüsü).

Dölek, İ., & Kurt, A. (2023). Derin Sinir Ağlarıyla Osmanlıca Optik Karakter Tanıma. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 38(4), 2579-2594.

Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1440-1448).

Gkioxari, G., Hariharan, B., Girshick, R., & Malik, J. (2014). R-CNNs for pose estimation and action detection. arXiv preprint arXiv:1406.5212.

Gorgel, P., Kilic, N., Ucan, B., Kala, A., & Ucan, O. N. (2009). A backpropagation neural network approach for Ottoman character recognition. Intelligent Automation & Soft Computing, 15(3), 451-462.

Onat, A., Yildiz, F., & Gündüz, M. (2006). Ottoman script recognition using hidden Markov model. IEEE Transaction on Engineering Computing Technology, 14, 71-73.

Öztürk, A., Güneş, S., & Özbay, Y. (2000, December). Multifont Ottoman character recognition. In ICECS 2000. 7th IEEE International Conference on Electronics, Circuits and Systems (Cat. No. 00EX445) (Vol. 2, pp. 945-949). IEEE.

Kilic, N., Gorgel, P., Ucan, O. N., & Kala, A. (2008, March). Multifont Ottoman character recognition using support vector machine. In 2008 3rd International Symposium on Communications, control and Signal Processing (pp. 328-333). IEEE.

Kurt, Z., Türkmen, H. I., & Karslıgil, M. E. (2007, June). Ottoman Alphabet Character Recognition by LDA. In 2007 IEEE 15th Signal Processing and Communications Applications (pp. 1-4). IEEE.

Kurt, Z., Türkmen, H. I., & Karslıgil, M. E. (2009). Linear discriminant analysis in Ottoman alphabet character recognition. In Proceedings of the European Computing Conference: Volume 2 (pp. 601-607). Springer US.

Küçükşahin, N. (2019). Design of an Offline Ottoman Character Recognition System for Translating Printed Documents to Modern Turkish (Master's thesis, Izmir Institute of Technology).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

Majid, N., & Smith, E. H. B. (2019, September). Segmentation-free bangla offline handwriting recognition using sequential detection of characters and diacritics with a faster R-CNN. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 228-233). IEEE.

Mondal, R., Malakar, S., Barney Smith, E. H., & Sarkar, R. (2022). Handwritten English word recognition using a deep learning-based object detection architecture. Multimedia Tools and Applications, 1-26.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

Tesseract, Tesseract Open Source OCR Engine, https://github.com/tesseract-ocr, Access Date: 27.09.2023.

Tulum, M. (2014). Osmanlı Türkçesine Giriş 1-7. Anadolu Üniversitesi.

Uçar, M. (2021). Osmanlı Türkçesi Kolay Okuma Metinleri 1. Hayrât Neşriyat. Isparta.

Uzun, A. B., & Özer. A., 2021. Ottoman Turkish Characters, Access Date: 09.04.2022. https://www.kaggle.com/datasets/alpbintuuzun/ottoman-turkish-characters

Yalniz, I. Z., Altingovde, I. S., Güdükbay, U., & Ulusoy, Ö. (2009). Integrated segmentation and recognition of connected Ottoman script. Optical Engineering, 48(11), 117205-117205.