Computerized Counting of Individuals in Ottoman Population Registers with Deep Learning *

Yekta Said Can and M. Erdem Kabadayı

Koc University, Rumelifeneri Yolu 34450 Sarıyer, Istanbul, Turkey

Abstract. The digitalization of historical documents continues to gain pace for further processing and extract meanings from these documents. Page segmentation and layout analysis are crucial for historical document analysis systems. Errors in these steps will create difficulties in the information retrieval processes. Degradation of documents, digitization errors and varying layout styles complicate the segmentation of historical documents. The properties of Arabic scripts such as connected letters, ligatures, diacritics and different writing styles make it even more challenging to process Arabic historical documents. In this study, we developed an automatic system for counting registered individuals and assigning them to populated places by using a CNN-based architecture. To evaluate the performance of our system, we created a labeled dataset of registers obtained from the first wave of population registers of the Ottoman Empire held between the 1840s-1860s. We achieved promising results for classifying different types of objects and counting the individuals and assigning them to populated places.

Keywords: page segmentation · historical document analysis · convolutional neural networks · Arabic layout analysis

1 Introduction

Historical documents are precious cultural resources that provide the examination of historical, social and economic aspects of the past [1]. The digitization of them also provides immediate access for researchers and the public to these archives. However, for maintenance reasons, access to them might not be possible or could be limited. Furthermore, we can analyze and infer new information from these documents after the digitalization processes. For digitalizing the historical documents, page segmentation of different areas is a critical process for further document analysis and information retrieval [2]. Page segmentation techniques analyze the document by dividing the image into different regions such as

* This work has been supported by European Research Council (ERC) Project: "Industrialisation and Urban Growth from the mid-nineteenth century Ottoman Empire to Contemporary Turkey in a Comparative Perspective, 1850-2000" under the European Union's Horizon 2020 research and innovation programme grant agreement No. 679097

backgrounds, texts, graphics, decorations [3]. Historical document segmentation is more challenging because of the degradation of document images, digitization errors and variable layout types. Therefore, it is difficult to segment them by applying projection-based or rule-based methods [3].

Page segmentation errors have a direct impact on the output of the OCR which converts handwritten or printed text into digitized characters [2]. Therefore, page segmentation techniques for the historical documents become important for the correct digitization. We can examine the literature on page segmentation under three subcategories [3]. The first category is the granular based techniques which combine the pixels and fundamental elements into large components [4], [5] and [6]. The second category is the block-based techniques that divide the pages into little regions and then combine into large homogenous areas [7] and [8]. The last one is the texture-based methods which extracts textual features classify objects with different labels [9], [10] and [11]. Except for the block-based techniques, these methods work in a bottom-up manner [3]. The bottom-up mechanisms have better performance with documents in variable layout formats [3]. However, they are expensive in terms of computational power because there are plenty of pixels or small elements to classify and connect [3]. Still, the advancement of technology of CPUs and GPUs alleviates this burden. Feature extraction and classifier algorithm design are very crucial for the performance of page segmentation methods. Although document image analysis has started with more traditional machine learning classifiers, with the emergence of Convolutional Neural Networks (CNNs), most studies use them because of their better performance.

Arabic script is used in writing different languages, e.g., Ottoman, Arabic, Urdu, Kurdish, Persian[12]. It could be written in different manners which complicate the page segmentation procedure. It is a cursive script in which connected letters create ligatures [12]. Arabic words could further include dots and diacritics which causes even more difficulties in the page segmentation [12].

In this study, we developed a software that automatically segments pages and recognizes objects for counting the Ottoman population registered in populated places. Our data comes from the first population registers of the Ottoman Empire that is realized in the 1840s. These registers are the results of an unprecedented administrative operation, which aimed to register each and every male subject of the empire, irrespective of age, ethnic or religious affiliation, military or financial status. Therefore, they aimed to have universal coverage for the male populace and thus these registers can be called (proto-) censuses. The geographical coverage of these registers is the entire Ottoman Empire in the mid-nineteenth century, which encompassed the territories of around two dozen successor states of today in Southeast Europe and the Middle East. For this paper, we are focusing on two locations: Nicaea in western Anatolia in Turkey, and Svishtov a Danubian town in Bulgaria.

In these censuses, officers prepared manuscripts without using handwritten or printed tables. Furthermore, there is not any pre-determined page structure. Page layouts can differ in different districts. There were also structural changes depending on the officer. We created a labeled dataset to give as an input to the supervised learning algorithms. In this dataset, different regions and objects are marked with different colors. We then classified all pixels and connected the regions comprising of the same type of pixels. We recognized the populated place starting points and person objects on these unstructured handwritten pages and counted the number of people in all populated places and pages. Our system successfully counts the population in different populated places.

The structure of the remaining parts of the paper is as follows. In Section 2, the related work in historical document analysis will be reviewed. We described the structure of the created database in Section 3. Our method for page segmentation and object recognition is described in Section 4. Experimental results and discussion are presented in Section 5. We present the conclusion and future works of the study in Section 6.

2 Related Works

Document image analysis studies have started in the early 1980's [13]. Laven et al. [14] developed a statistical learning based page segmentation system. They created a dataset that includes 932 page images of academic journals and labeled physical layout information manually. By using a logistic regression classifier, they achieved approximately 99% accuracy with 25 labels. The algorithm for segmentation was a variation of the XY-cut algorithm [15]. Arabic document layout analysis has also been studied with traditional algorithms in the literature. Hesham et al. [12] developed an automatic layout detection system for Arabic documents. They also added line segmentation support. After applying Sauvola binarization [16], noise filtering and skewness correction algorithms, they classified text and non-text regions with the Support Vector Machine (SVM) algorithm. They further segmented lines and words.

In some cases, the historical documents might have a tabular structure which makes it easier to analyze the layout. Zhang et al. [17] developed a system for analyzing Japanese Personnel Record 1956 (PR1956) documents which includes company information in a tabular structure. They segmented the document by using the tables and applied Japanese OCR techniques to segmented images. Richarz et al. [18] also implemented a semi-supervised OCR system on historical weather documents with printed tables. They scanned 58 pages and applied segmentation by using the printed tables. Afterward, they recognized digits and seven letters in the document.

After the emergence of neural networks, they are also tested on Arabic document analysis systems. Bukhari et al. [6] developed an automatic layout detection system. They classified the main body and the side text by using the MultiLayer Perceptron (MLP) algorithm. They created a dataset consisting of 38 histori-

cal document images from a private library in the old city of Jerusalem. They achieved 95% classification accuracy. Convolutional Neural Network is also a type of deep neural network that can be used for most of the image processing applications [19]. CNN and Long Short Term Memory (LSTM) used for document layout analysis of scientific journal papers written in English in [20] and [21]. Amer et al. proposed a CNN-based document layout analysis system for Arabic newspapers and Arabic printed texts. They achieved approximately 90% accuracy in finding text and non-text regions.

CNNs are also used for segmenting historical documents. As mentioned previously, historical document analysis has new challenges when compared to the modern printed text layout analysis, such as degraded images, variable layouts and digitization errors. The Arabic language also creates more difficulties for document segmentation due to its cursive nature where letters are connected by forming ligatures. Words may also contain dots and diacritics which could be problematic for segmentation algorithms. Although, there are studies applying CNNs to historical documents [22], [3] and [2], to the best of our knowledge, this study is the first to apply CNN-based segmentation and object recognition in historical handwritten Arabic document analysis literature.

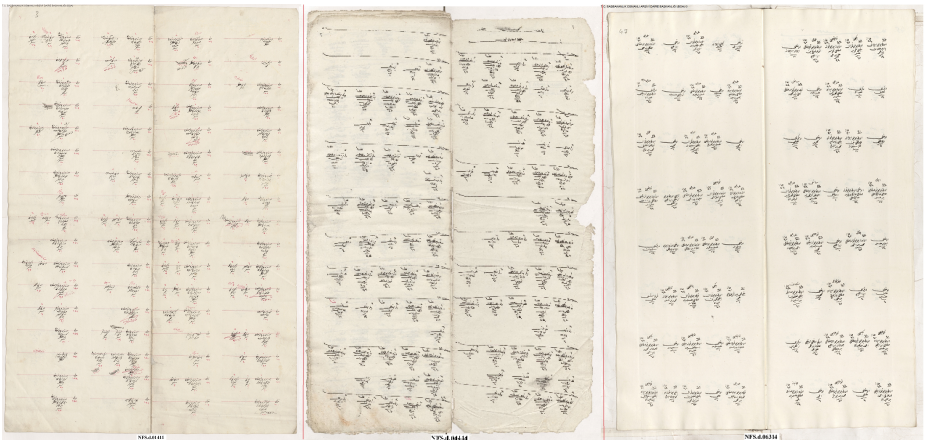


Fig. 1. Three sample pages of the registers belong to three different districts. The layout of pages can change between districts.

3 Structure of the Registers

Our case study is focusing on the registers of Nicaea and Svistov district registers, NFS.d. 1411, 1452, and NFS.d. 6314, respectively, available at the Turkish Presidency State Archives of the Republic of Turkey – Department of Ottoman

Archives in jpeg format upon request. We aim to develop a methodology to be implemented for an efficient distant reading of similar registers from various regions of the empire prepared between the 1840s and the 1860s. As mentioned above, these registers provide detailed demographic information on male members of the households, i.e., names, family relations, ages, and occupations. Females in the households were not registered. The registers became available for research at the Ottoman state archives in Turkey, as recently as 2011. Their total number is around 11,000. Until now, they have not been subject to any systematic study. Only individual registers were transliterated in a piecemeal manner. The digital images of the recordings are usually around the size of 2100x3000 pixels.

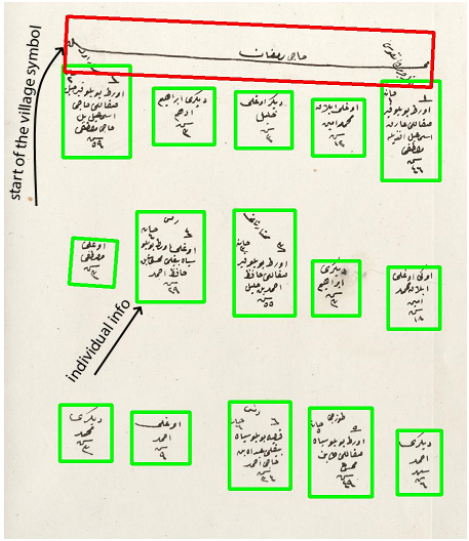


Fig. 2. Start of the populated place symbol and an individual cluster are demonstrated.

As mentioned previously, the layout of these registers can change from district to district (see Figure 1) which makes our task more complicated. In this study, we work with the generic properties of these documents. The first property is the populated place start symbol. This symbol is used in most of the districts and can mark the end of the previous populated place and start of the new one (see Figure 2). The remaining clusters are in the registers are individuals counted in the census and they include demographic information about them. There are also updates in these registers which marks the individuals when they go to the military service or decease. The officers generally draw a line on the individual and sometimes mistakenly connect the individual with an adjacent one which can cause some errors in the segmentation algorithm (see Figure 3).

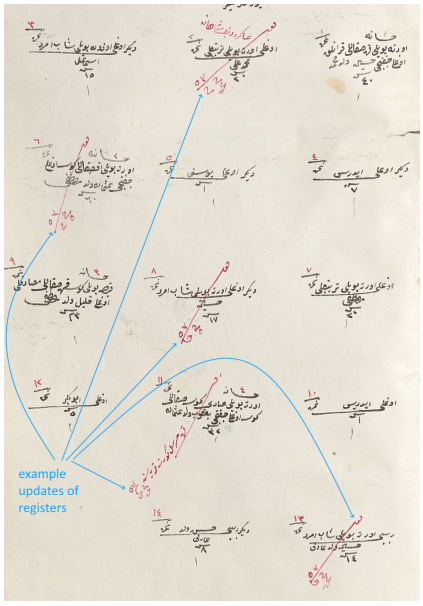


Fig. 3. Example updates of registers are shown. Some of them can connect two individuals and can cause clustering errors.

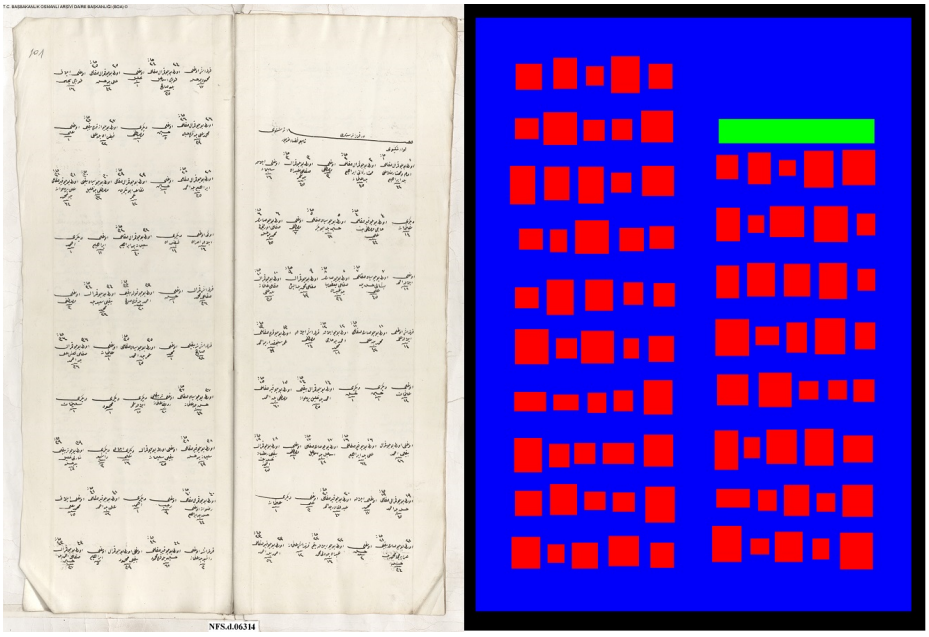


Fig. 4. A sample register page and its labeled version are demonstrated. Different colors represent different object types.

4 Automatic Page Segmentation and Object Recognition System for Counting Ottoman Population

4.1 Creating a Dataset

To be able to use the dhSegment toolbox [22], we created a dataset with labels. We created four different classes. The first one is the background which is the region between the page borders and document borders. We marked this region as black. The second class is the page region and it is marked with blue. The third one is the start of a populated place object and we colored it with green. The last one is the individual registers and we marked them with red. We marked 173 pages with the described labels. 51 of them belong to the Svistov district and 122 of them belong to the Nicaea district. An example original image and labeled version are shown in Figure 4.

4.2 Training the CNN Architecture

In order to train a CNN for our system, we used dhSegment [22] toolbox. This toolbox trained a system using the deep residual pretrained Resnet-50 architecture [23]. The toolbox has both a contracting path (follows the deep residual network in Resnet-50 [23]) and an expanding path which maps low resolution features to the original high resolution features (see terminology for expanding and contracting paths in [24]) [22]. The expanding path consists of five blocks and a convolutional layer for pixel classification and each deconvolutional step consists of upscaling of an image, concatenation of feature map to a contracting one, 3x3 convolutional and one Relu layer blocks.

In order to train the model, the toolbox used L2 regularization with 10^{-6} weight decay [22]. Xavier initialization [25] and Adam optimizer [26] are applied. Batch renormalization [27] is employed for refraining from a lack of diversity problem. The toolbox further downsized pictures and divided them into 300 300 patches for better fitting into the memory and providing support for training with batches. With the addition of margins, border effects are prevented. Because of the usage of pre-measured weights in the network, the training time is decreased substantially [22]. The training process exploits a variety of on-the-fly data augmentation techniques like rotation, scaling and mirroring. The system outputs the probabilities of each pixel belonging to one of the trained object types. Detailed metrics of one of the trained models by the integration of Tensorboard is shown in Figure 5.

4.3 Preparing the Dataset for Evaluation

We trained three different models for evaluating the performance of our system. The first two models were trained with a register of one district and tested them with a completely different district's register. For the last model, we further combined our two registers and trained a combined model. This model is tested with 10-fold cross-validation.

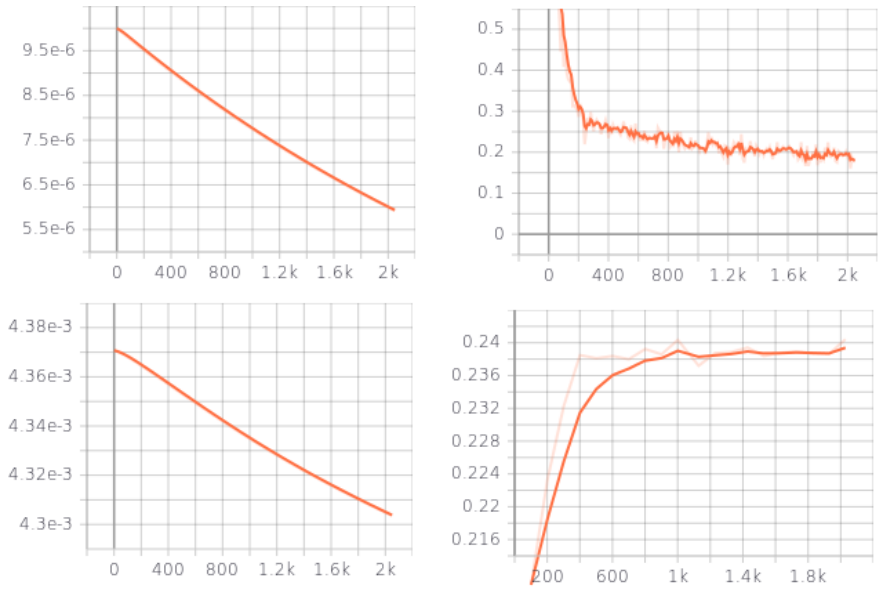


Fig. 5. Training metrics are demonstrated. In the top left, learning rate, in top right loss function, in bottom left regularized loss and in the bottom right global steps per second metrics are demonstrated. The subfigures are created with Tensorboard.

4.4 Post-Processing

In our problem, we have four different classes: a background, page, an individual and the start of a populated place, namely. Therefore, we evaluated the probabilities of pixels that belong to one of the classes. For each class, there is a binarized matrix showing the probabilities that a pixel belongs to them. By using these matrices, pixels should be connected and components should be created. Connected component analysis tool [22] is used for creating objects. After the objects are constructed for all classes, the performance of our system could be measured.

4.5 Assigning Individuals to the Populated Places

This toolbox [22] finds the objects in all pages by supporting batch processing. However, for our purposes, we need the number of people in any populated place. To this end, we designed an algorithm for counting people and assigning them to the populated places. The flowchart of our algorithm can be seen in Figure 6. Firstly, we recorded the x and y coordinates of the rectangles of the found objects. The object could be of populated place start or individual type. Furthermore, they divided each page into two blocks and we have to consider this structure also. We defined a center of gravity for each object. It is computed by averaging all four coordinates of the rectangle surrounds the object. Due to

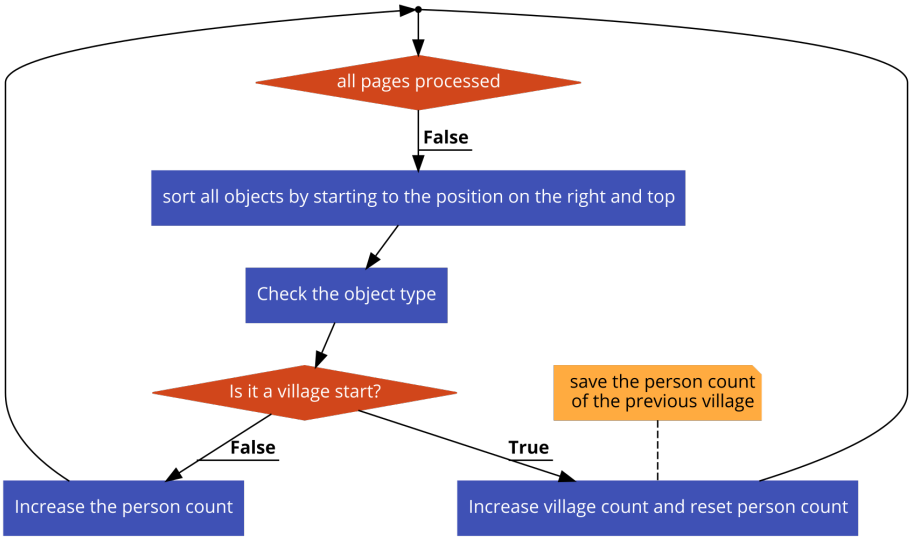


Fig. 6. Flowchart of our populated-place assigning algorithm

the structure of the Arabic language, if an object is closer to the top of the page and right of the page than any other object, it comes before. However, if the object is in the left block of a page, without looking at the distance to top, it comes after any object in the right block of the page. We first sorted populated place start objects. For all individual objects, we compared their position on page and page number with all populated place start objects. If the individual object is after a populated place start object N and before populated place start object N+1, we assigned the individual to populated place N.

5 Experimental Results and Discussion

In this section, we first define the metrics used for evaluating our system. We then present our results and discuss them.

5.1 Metrics

To evaluate our system performance, we used five different metrics. The first two metrics are low-level evaluators and they are widely used in object detection problems. We defined the third and fourth high-level metrics to evaluate the accuracy of our system.

Pixel-wise Classification Accuracy: The first metric is the pixel-wise accuracy. It can be calculated by dividing the accurately classified pixels in all documents to the number of all pixels in all documents (for all object types).

Intersection over Union The second metric is the Intersection over Union (IoU) metric. For this metric, there are the ground-truth components and the predicted components from our model. This metric can be calculated by dividing the intersection of regions of these two components to the union of regions of these two components (for all object types).

High-level Counting Errors These metrics are specific to our application for counting people in registers. For counting the individuals, the first high-level metric can be defined as the predicted count errors over the ground truth count. We can call this metric as individual counting error (ICE).

$$ICE = || \frac{PredictedIndividualCount - GroundTruthIndividualCount}{GroundTruthIndividualCount} || \quad (1)$$

We further defined a similar high-level metric for populated-place start objects which is named as the populated-place start counting error (VSCE).

$$VSCE = || \frac{PredictedPopPlaceStartCount - GroundTruthPopPlaceStartCount}{GroundTruthPopPlaceStartCount} || \quad (2)$$

5.2 Results and Discussion

We have two registers from the Nicaea district and one register from the Svistov district. In model 1, we trained with Nicaea registers and tested with the Svistov registers. In model 2, we trained a model with the Svistov district register and tested with the Nicaea registers. We further tested 10-fold cross validation on registers in the same district. In model 3, we trained and tested the model on the Svistov registers and in model 4, we trained and test on Nicaea registers with 10-fold cross-validation. In model 5, we combined the whole dataset and evaluated the model with 10-fold cross-validation. The pixel-wise accuracy, IoU and counting error results are provided in Table 1. Note that the first three metrics are provided for a different number of object classification. The last two metrics are the error of finding the number of individuals and the populated-place start objects. We further provided correctly predicted and mistakenly predicted raw binarized images in Figures 7 and 8, respectively. The best ICE results are obtained when the Svistov registers are used for the training. The worst accuracy is obtained when the system is trained with Nicaea registers and tested with the Svistov register. Furthermore, the populated-place start counting error is 0% for all models which means that our system can recognize populated-place start objects perfectly.

As mentioned before, the layout of registers depends on the districts and the officer. For our registers, individuals in Nicaea are widely separated, whereas the distance between registers is less in Svistov registers. The average number of registered individuals in a Nicaea register page is approximately 40 and 80 in a Svistov register which confirms the above statement. Therefore, when the

system is trained with loosely put Nicaea registers and tested in closely written clusters in Svistov, the counting error increases and the number of mistakes for counting multiple registers as one start to occur (see Figure 8). Whereas, if we change training and test parts, the system error for counting objects approaches to 100% as we expected. If we mix the dataset and apply 10-fold cross-validation, we achieved counting errors in between. For our purposes, although high-level metrics are more crucial, low-level metrics showed the general performance of our system. They are also beneficial for comparing the performances of different models. Furthermore, even though IoU metric results are low, our classification errors are close to 0%. It could be inferred that the structure of registers is suitable for automatic object classification systems. The documents do not have printed tables, but their tabular-like structures make it easier to cluster and classify them.

Table 1. Results with different metrics are presented for five different models.

Trained on	**Tested with**	**Pixel-wise Acc. (%)**	**IoU (%)**	**ICE (%)**	**VSCE** (%)
Nicaea	Nicaea	94.25	80.91	1.65	0
Nicaea	Svistov	92.72	73.6	11.57	0
Svistov	Svistov	93.04	72.29	0.76	0
Svistov	Nicaea	85.36	47.95	0.27	0
Mixed	Mixed	87.26	48.54	2.26	0

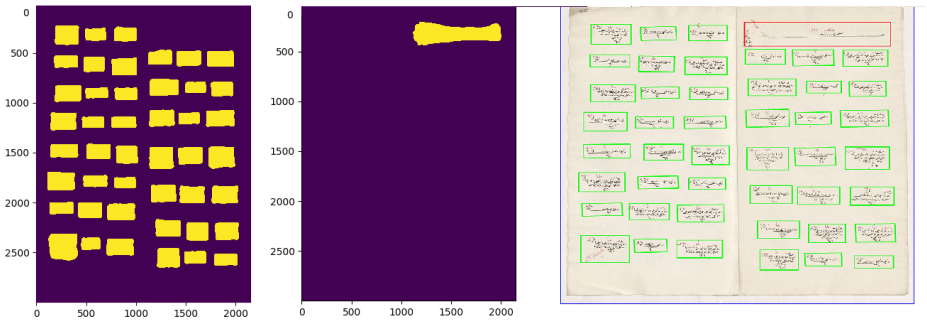


Fig. 7. A sample prediction made by our system. In the left, a binarized prediction image for counting individuals, in the middle a binarized image for counting populated-place start and in the right, the objects are enclosed with rectangular boxes. Green boxes for individual register counting and the red box for counting the populated-place start object.

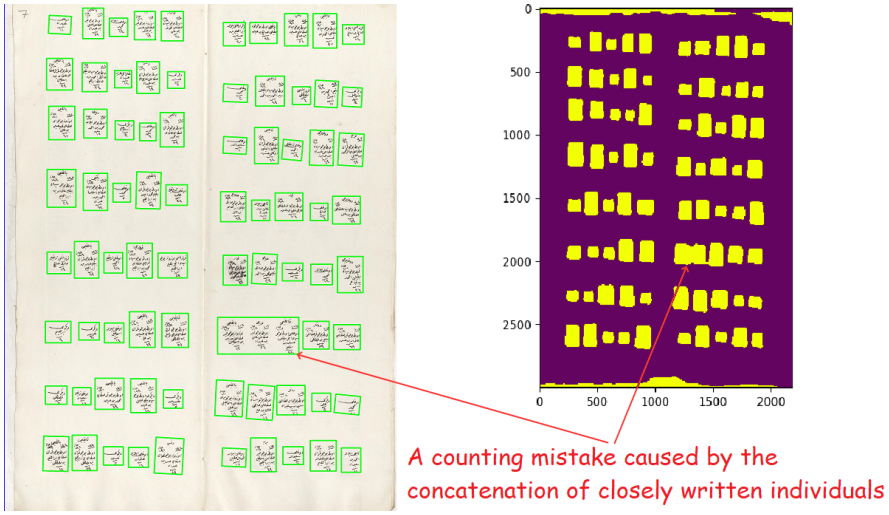


Fig. 8. A sample counting mistake. All three individual registers are counted as 1. This results in two missing records in our automatic counting system.

6 Conclusion and Future Works

In this study, we developed an automatic individual counting system for the registers recorded in the first censuses of the Ottoman Empire which are held between 1840-1860. The registers are written in Arabic script and their layouts highly depend on the district and the officer in charge. We created a labeled dataset for three registers and evaluated our system on this dataset. We further developed an algorithm for assigning people to populated-places after detecting individual people and populated-place start symbols. For counting the populated-place start symbols, we achieved 0% error. Furthermore, we achieved the maximum individual counting error of 0.27%. We inferred from these results that the models should be trained with closely placed and noisy registers (Svistov register in our case study). When these models are tested with a clean and a loosely placed one (Nicaea register in this case study), the system counts individuals accurately. However, if a model is trained with a loosely placed register and tested with closely placed one, the number of counting errors is increasing. Our aim is to develop a generic system that can be implemented for efficient counting and distant reading of all registers prepared between the 1840s and the 1860s. Since it is impossible to label all registers, we will strategically label the closely placed and noisy ones to develop such a system. As future works, we plan to develop an automatic handwriting recognition system for the segmented individual register objects.

References

1. M. S. Kim, K. T. Cho, H. K. Kwag, and J. H. Kim, "Segmentation of handwritten characters for digitalizing korean historical documents," in *Document Analysis Systems VI*, S. Marinai and A. R. Dengel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 114–124.
2. C. Wick and F. Puppe, "Fully convolutional neural networks for page segmentation of historical document images," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 287–292.
3. Y. Xu, W. He, F. Yin, and C.-L. Liu, "Page segmentation for historical handwritten documents using fully convolutional networks," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 541–546.
4. M. Baechler and R. Ingold, "Multi resolution layout analysis of medieval manuscripts using dynamic mlp," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 1185–1189.
5. A. Garz, R. Sablatnig, and M. Diem, "Layout analysis for historical manuscripts using sift features," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 508–512.
6. S. S. Bukhari, T. M. Breuel, A. Asi, and J. El-Sana, "Layout analysis for arabic historical document images using machine learning," in *2012 International Conference on Frontiers in Handwriting Recognition*. IEEE, 2012, pp. 639–644.
7. S. Uttama, J.-M. Ogier, and P. Loonis, "Top-down segmentation of ancient graphical drop caps: lettrines," in *Proceedings of 6th IAPR International Workshop on Graphics Recognition, Hong Kong*, 2005, pp. 87–96.
8. N. Ouwayed and A. Belaïd, "Multi-oriented text line extraction from handwritten arabic documents," 2008.
9. R. Cohen, A. Asi, K. Kedem, J. El-Sana, and I. Dinstein, "Robust text and drawing segmentation algorithm for historical documents," in *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*. ACM, 2013, pp. 110–117.
10. A. Asi, R. Cohen, K. Kedem, J. El-Sana, and I. Dinstein, "A coarse-to-fine approach for layout analysis of ancient manuscripts," in *2014 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 140–145.
11. K. Chen, H. Wei, J. Hennebert, R. Ingold, and M. Liwicki, "Page segmentation for historical handwritten document images using color and texture features," in *2014 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 488–493.
12. A. M. Hesham, M. A. Rashwan, H. M. Al-Barhamtoshy, S. M. Abdou, A. A. Badr, and I. Farag, "Arabic document layout analysis," *Pattern Anal. Appl.*, vol. 20, no. 4, pp. 1275–1287, Nov. 2017.
13. G. Nagy, "Twenty years of document image analysis in pami," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 1, pp. 38–62, 2000.
14. K. Laven, S. Leishman, and S. Roweis, "A statistical learning approach to document image analysis," in *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, 2005, pp. 357–361.
15. Jaekyu Ha, R. M. Haralick, and I. T. Phillips, "Recursive x-y cut using bounding boxes of connected components," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 2, Aug 1995, pp. 952–955 vol.2.

16. J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen, "Adaptive document binarization," in *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, vol. 1. IEEE, 1997, pp. 147–152.
17. K. Zhang, Z. Shen, J. Zhou, and M. Dell, "Information extraction from text regions with complex tabular structure," 2019.
18. J. Richarz, G. A. Fink *et al.*, "Towards semi-supervised transcription of handwritten historical weather reports," in *2012 10th IAPR International Workshop on Document Analysis Systems*. IEEE, 2012, pp. 180–184.
19. T. Matsumoto, T. Yokohama, H. Suzuki, R. Furukawa, A. Oshimoto, T. Shimmi, Y. Matsushita, T. Seo, and L. O. Chua, "Several image processing examples by cnn," in *IEEE International Workshop on Cellular Neural Networks and their Applications*, Dec 1990, pp. 100–111.
20. T. M. Breuel, "Robust, simple page segmentation using hybrid convolutional mdlstm networks," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov 2017, pp. 733–740.
21. D. Augusto Borges Oliveira and M. Palhares Viana, "Fast cnn-based document layout analysis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1173–1180.
22. S. Ares Oliveira, B. Seguin, and F. Kaplan, "dhsegment: A generic deep-learning approach for document segmentation," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Aug 2018, pp. 7–12.
23. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
24. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
25. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.
26. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
27. S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Advances in neural information processing systems*, 2017, pp. 1945–1953.