A METHOD FOR DIGITAL RECOGNITION OF GEORGIAN TEXT FROM IMAGES

Julieta Tabeshadze
PhD Student in Informatics,
Samtskhe-Javakheti State University,
Rustaveli St. 124, Akhaltsikhe, Georgia
+995 598 09 98 46, julietatabeshadze@gmail.com
https://orcid.org/0009-0008-7463-7801

Abstract. This article discusses the implementation of an optical character recognition (OCR) algorithm for the Georgian alphabet using the MATLAB programming environment. The aim of the study is to develop an effective system for the digital recognition of Georgian text that overcomes challenges related to low-resolution images, font variations, uneven background structure, and noise.

The proposed algorithm is based on several stages of digital image processing: initial image filtering, conversion to grayscale and binary modes, text segmentation, and comparison with binary reference matrices. For character recognition, a correlation analysis method is used, which identifies the characters extracted from a new image by comparing them with pre-formed templates.

The algorithm was tested on images with different qualities and structures. The results showed that under appropriate pre-processing conditions, high-accuracy recognition of Georgian text is achievable.

This study emphasizes the potential of digital image processing technologies in the digitization of the Georgian language and cultural heritage. The proposed method can also be adapted to other writing systems, giving the research both theoretical and practical significance.

Keywords: Optical character recognition (OCR); Georgian alphabet; text digitization; text recognition from images; binarization; segmentation; correlation analysis; MATLAB; reference templates.

Introduction

The Georgian script is one of the oldest writing systems in the world and is included in UNESCO's list of Intangible Cultural Heritage. In the contemporary era, where the digitization of linguistic resources is a critical task for the preservation and accessibility of national heritage, the development of efficient technologies for the digital recognition of the Georgian alphabet has become especially important.

Optical character recognition (OCR) systems are regarded as a leading approach in the automated processing of textual data. Their main function is to recognize text contained in images and convert it into an editable digital format. Although many such systems exist at the international level, their application to the Georgian language often encounters challenges—commonly showing low accuracy, lack of dedicated support, or limited functionality. Moreover, the performance of these systems is significantly influenced by image quality, font type, text layout, and background noise.

The present study aims to develop and evaluate an algorithm for recognizing Georgian script using the MATLAB programming environment. The algorithm follows classical stages of digital image processing: pre-filtering, segmentation, pattern matching, and exporting the recognized text into a file. The methodology is modular, making it adaptable for other writing systems as well, thereby granting the model a degree of universality.

Through this research, we seek to contribute to the digitization of the Georgian language and cultural heritage by improving recognition accuracy and usability, especially for users without professional technical expertise. Therefore, this article presents the algorithm's structure, its technical underpinnings, and the results achieved during system testing.

Methods

The implementation of the text recognition algorithm was carried out using the MATLAB programming environment, which is known for its high efficiency in matrix operations and its extensive set of tools for digital signal and image processing. The developed system is structured into several stages, allowing each phase to be evaluated and improved independently.

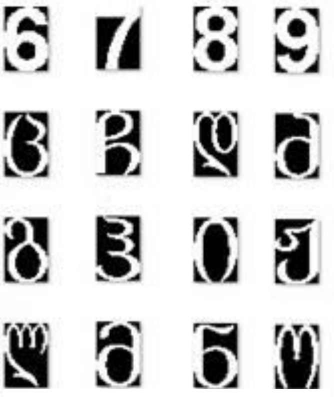
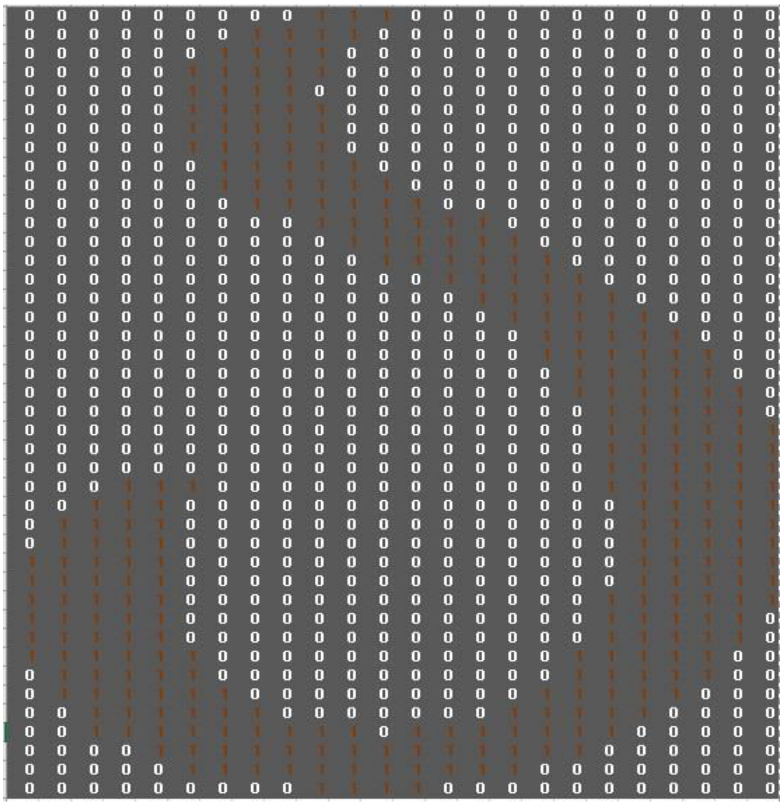


Fig. 1. Reference matrix

1. Formation of Reference Character Matrix

In the first stage, a reference binary matrix is created, containing digital representations of all characters in the Georgian alphabet. Each character is displayed in a black-and-white grid format, where black pixels are represented by the value "0" and white pixels by "1". This matrix is used in the subsequent stage for character comparison.



0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Fig. 2. Sample of the letter “ა”: black pixels in the binary matrix are represented by “0” and white pixels by “1”.

2. Preprocessing of the Image

The imported image is converted to grayscale using the rgb2gray function, and then further transformed into a binary image. To reduce noise and enhance character clarity, digital filtering is applied, which significantly improves the distinguishability of the symbols.

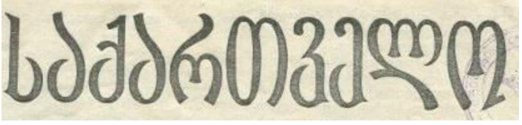


Fig. 2. Image to be processed

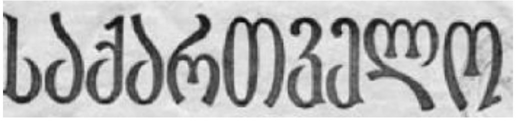


Fig. 3. Image in grayscale mode

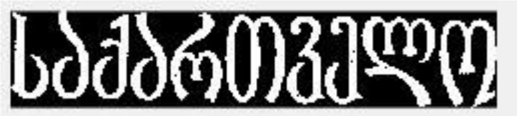


Fig. 4. Binary image

3. Segmentation

During the segmentation process, individual lines, words, and characters are separated from one another. This stage ensures accurate comparison between each extracted character and its corresponding reference template.

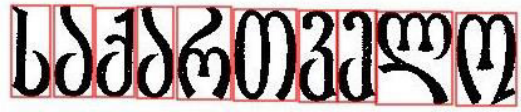


Fig. 4. Segmentation

4. Correlation-Based Comparison

The extracted characters from the binary image are compared to reference matrices using correlation analysis. Each character is identified as the one that shows the highest degree of similarity to a reference template at the pixel level.

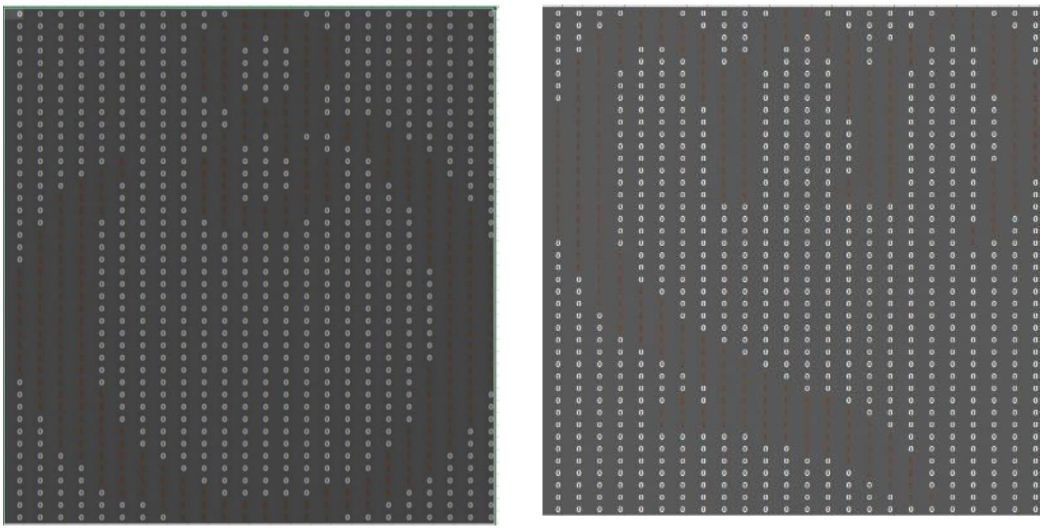
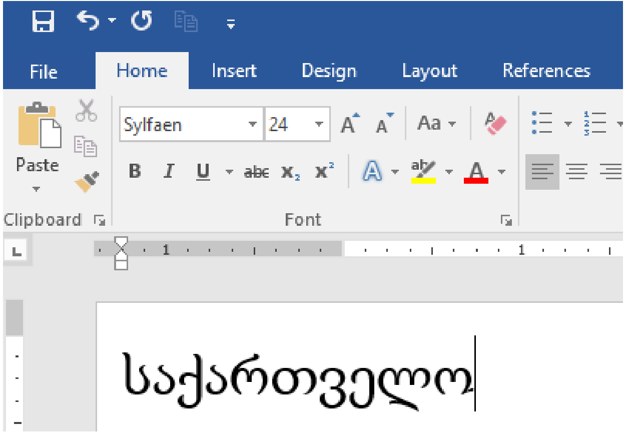


Fig. 5. Reference matrix samples for the letters “ტ” and “ლ”

5. Text Generation and Export

The recognized characters are assembled into a text string and saved as a .txt file using the fprintf function. The resulting file is opened in Microsoft Word via the winopen command, providing the user with easy access to the digitized text.



საქართველო

Fig. 6. Opening the recognized text – final stage of the digitization process

Results

The functional evaluation of the developed algorithm was conducted on various types of images containing Georgian script in different fonts, resolutions, and background structures. Each image featured either the complete alphabet or textual fragments, and recognition was performed using MATLAB.

As a result of preprocessing, noise was significantly reduced, and characters were visually enhanced. The segmentation module effectively extracted individual lines and characters from the image, and correlation-based comparison enabled high-accuracy identification of characters.

The average recognition accuracy of the algorithm reached 75–80% on images meeting the following criteria: at least 10 pixels per character, uniform background, and low noise. In more challenging conditions—such as low resolution, uneven backgrounds, and non-standard font styles—accuracy ranged between 60–65%.

The results are illustrated above, showing the original image, the filtered version, and the final recognized text. The algorithm successfully recognized most letters, and the output was exported in .txt format, which can be accessed and edited using a text editor.

It is worth noting that the results were particularly stable for standard fonts and structured page layouts, indicating the system's practical potential for use in the digitization of Georgian-language documents.

Discussion

The results obtained from the study confirm that the OCR algorithm implemented in MATLAB performs effectively for Georgian text recognition. The achieved 75–80% accuracy on high-quality images highlights the importance of proper preprocessing and filtering in enhancing system performance.

While standard OCR systems are widely used across various languages, the unique graphic features of the Georgian alphabet—such as the coexistence of angular and rounded elements—often result in recognition errors. The proposed algorithm offers two main advantages: (1) it relies on predefined binary reference matrices, which ensures transparency in the identification process, and (2) the MATLAB environment allows for fast experimental modifications and customization of the algorithm for specific needs.

Nevertheless, the system has some limitations: recognition accuracy drops significantly in cases where the background is highly uneven or when the text is of very low resolution. Additionally, correction for rotated characters is not currently implemented, which may be addressed in future improvements.

The study underscores the need for continuous adaptation of image-based text recognition algorithms to the graphic characteristics of each specific writing system. The successful implementation of an algorithm tailored to the Georgian script not only demonstrates the viability of the technical model but also presents an opportunity to enhance automation in the digitization of the Georgian language.

Conclusion

This article presented a model of an optical character recognition algorithm designed to convert Georgian text from images into a digital format. The algorithm, implemented in the MATLAB environment, demonstrated high effectiveness, particularly under conditions with standard fonts, high resolution, and minimal noise.

The system is based on comparison with pre-defined binary reference templates, which provides a transparent and customizable recognition process. Accurate use of filtering and segmentation significantly improves recognition accuracy, while direct export of the recognized text offers users a simple and efficient interface.

In conclusion, the proposed approach is effective and promising both for research purposes and practical applications—especially in the context of digitizing and preserving Georgian cultural heritage. Moreover, the modular nature of the algorithm provides opportunities for further development and adaptation to other writing systems.

REFERENCES

beridze, l., gogiberidze, r., & k'ach'akhidze, n. (2014). *MATLAB-i st'udent'ebisatvis [MATLAB].* Retrieved from https://gtu.ge/book/matlabi_L_beridze.pdf

devdariani, z. (2013). *signalis damushaveba [Signal Processing].* Retrieved from https://www.emc.ge/docs/14_Signal_Processing.pdf

qubaneishvili, e. (2018). *biosignalebis tsipruli damushaveba [Digital Processing of Biosignals].* Retrieved from https://gtu.ge/book/ims/cifruli_damushaveba.pdf

labadze, o. (2012). *signalebis teoria [Theory of Signals].* Retrieved from https://tsu.ge/data/file_db/library/signalebis%20teoria.pdf

Adalı, T., & Haykin, S. (2010). *Adaptive Signal Processing: Next Generation Solutions*. 1st ed. Wiley-IEEE Press.

Hayes, M. H. (n.d.). *Statistical Digital Signal Processing and Modeling*.

Ifeachor, E. C., & Jervis, B. W. (1993). *Digital Signal Processing: A Practical Approach*. Addison-Wesley.

Ingle, V. K., & Proakis, J. G. (2016). *Digital Signal Processing Using MATLAB: A Problem Solving Companion.* 4th ed. Cengage Learning.

Lacoste, R. (2010). Chapter 7 - No Fear with FIR: Put a Finite Impulse Response Filter to Work. In *Robert Lacoste's The Darker Side* (pp. 93–110). Newnes. https://doi.org/10.1016/B978-1-85617-762-7.00007-1

Lyons, R. G. (2010). Understanding Digital Signal Processing. 3rd ed., Prentice Hall.

MathWorks. (2025). *Signal Processing Toolbox User's Guide*. MATLAB R2025a Documentation.

Oppenheim, A. V., & Schafer, R. W. (2009). *Discrete-Time Signal Processing* (3rd ed.). Prentice Hall.

Parker, M. (2017). Chapter 5 - Finite Impulse Response (FIR) Filters. In *Digital Signal Processing 101* (2nd ed., pp. 41–57). Newnes. https://doi.org/10.1016/B978-0-12-811453-7.00005-6

Papoulis, A. (1984). *Signal Analysis*. McGraw-Hill.

Pickus, K. (2006). Book-review. *AJS Review*, 30(1), 206–208. https://doi.org/10.1017/S0364009406290093

Proakis, J. G., & Manolakis, D. G. (2006). *Digital Signal Processing: Principles, Algorithms, and Applications* (4th ed.). Pearson.

Raffoul, Y. N. (2025). Chapter 3 - Z-transform. In *Difference Equations and Applications* (pp. 83–136). Academic Press. https://doi.org/10.1016/B978-0-44-331492-6.00009-1

Smith, S. W. (1997). *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing.