

THE ALGORITHMS FOR INITIAL PROCESSING AND RECOGNITION OF HANDWRITTEN TEXT IMAGES

¹Iskandarova Sayyora Nurmamatovna

²Kuchkarov Zoir Turdiboy ogli

¹Associate Professor of Tashkent University of Information Technologies named after Muhammad al-Khorazmi, Ph.D.

²Graduate student of Samarkand branch of Tashkent University of Information Technologies named after Muhammad al-Khorazmi

<https://doi.org/10.5281/zenodo.7603570>

Abstract. *This article presents the steps of binary morphological processing of handwritten texts, the steps of segmentation for handwritten texts and their classification procedures.*

A comparative analysis of the results obtained in CNN+LSTM+CTC and CNN+LSTM+fuzzy CTC neural network architectures for handwritten text image recognition and processing is provided.

Keywords: *CNN(Convolutional Neural Network), lstm(Long Short Term Memory), CTC(Connection-ist Temporal Classification), binary morphological operation, neural network.*

Introduction

The world is paying special attention to the development of computer vision systems. One of the urgent issues in this field is the development, improvement and implementation of methods and algorithms for the analysis and recognition of objects in the image. In foreign countries, including the USA, the Russian Federation, France, Switzerland, Greece, Iran, India, and other countries, great attention is paid to solving the theoretical and practical problems of analyzing and recognizing handwritten text images.

In the world, large-scale scientific researches aimed at improving existing methods and algorithms for creating automated systems for processing and analyzing handwritten text images, as well as developing new computational algorithms, is being conducted. The accuracy of the system depends on the quality of the image of the given handwritten text, and it is one of the important issues to improve the image quality, including the development of binarization, line and word segmentation algorithms.

Currently, handwritten text image analysis systems are developing rapidly, as these systems are widely used in the recognition of handwritten text in banking, postal communication, forensics and other fields. The analysis of the research conducted in this direction showed that the images of the given handwritten text are often insufficient quality, which causes various problems in the automation of their processing. Despite the fact that a number of methods and algorithms for improving image quality have been developed to date, the problem of improving and developing new methods and algorithms that increase the quality of handwritten text images has not been sufficiently studied.

Main part

A binarization algorithm based on morphological operations. The proposed algorithm consists of the following steps [1].

Stage 1. Initial processing stage. At this stage, the initially given image is smoothed. It uses an adaptive Wiener filter to smooth the background texture and increase the brightness between the text and the background. The general appearance of the Wiener filter is as follows:

$$I(x, y) = \mu + \frac{(\delta^2 - v^2)(I_s(x, y) - \mu)}{\delta^2},$$

Here μ – 3×3 size mean value, δ^2 – 3×3 size window dispersion, v^2 – average value of dispersion.

Stage 2. The stage of separation of points with high and low intensity. In this step, matching and splitting operations are performed on the smoothed handwritten text image to separate points with high and low intensity. An overview of separation and connection operations is as follows:

Separation action:

$$I_o(x, y) = I(x, y) \odot S_{m \times n} = (I(x, y) \oplus S_{m \times n}) \ominus S_{m \times n}$$

Connection action:

$$I_c(x, y) = I(x, y) \otimes S_{m \times n} = (I(x, y) \ominus S_{m \times n}) \oplus S_{m \times n},$$

here $I(x, y) \oplus S_{m \times n}$ – structuring element $S_{m \times n}$ intensification action; $I(x, y) \ominus S_{m \times n}$ – structuring element $S_{m \times n}$ wrecking action. In this stage $S_{m \times n}$ as a structuring element $S_{1 \times 7}$ element is obtained in the following form:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Stage 3. Formation of image differences. The image with the difference of the images obtained in the previous step is formed based on the idea of separating the vertical boundary points and they are determined by the following formula.

$$D(x, y) = \text{abs}(I_o(x, y) - I_c(x, y)).$$

Stage 4. Tutashuv. At this stage, the $D(x, y)$ image is affected with the $S_{5 \times 5}$ structuring element. In this case, resulting from the connection $D_c(x, y)$ is formed by the following formula:

$$D_c(x, y) = D(x, y) \otimes S_{5 \times 5}.$$

Step 5. Binary. In this step, the image obtained in step 4 is converted to a binary image based on the following formula.

$$B(x, y) = \begin{cases} 255, & \text{if } D_c(x, y) > t_o \text{ is so;} \\ 0, & \text{otherwise,} \end{cases}$$

here t_o is the binarization threshold, which is determined by Otsu's method [8].

Algorithms for segmentation of handwritten text lines are divided into three main groups: algorithms based on analysis of horizontal projections, coloring and {Xaf} substitutions. In addition, there are algorithms that cannot belong to any of these groups and cannot be combined into one group because they do not have any generalizing properties.

Algorithms based on analysis of horizontal projections segment lines by finding minimal "islands" in the calculated horizontal projection of a given image. These algorithms are used to segment typewritten strings, but they can be adapted to segment handwritten strings [2]. In this case, the image is divided into vertical slices and horizontal projections are analyzed for each of them.

In binarization algorithms [3], consecutive black pixels are painted in the horizontal direction. If the distance between the white space is within the specified threshold, it will be filled with black pixels. In painted images, the area of connected components is treated as lines.

It is known that straight lines in images can be determined using {Xaf} substitution [4]. The slope of handwriting lines in an image can be determined by applying {Xaf} displacement to the center of gravity of each connected component of the analyzed image [5]. Strings of handwritten text are segmented using proximity criteria and continuity direction of connected components.

Intersecting components are a problem for algorithms based on horizontal projection analysis (because they increase the value of the projection profile where it should be minimal) and painting algorithms (because they use connected components of text pixels to create lines), but from the third group does not affect some algorithms, in particular for the algorithm presented in [5].

To find intersecting elements from different lines, you can use properties such as the size of text binding components, whether a single component is attached to several lines, or vice versa, is not attached to any line. After identifying such suspicious components, it should be determined whether they belong to a certain line or whether it is necessary to divide them into elements related to different lines. Such a vertical decomposition of components is a complex issue.

A simple solution to this problem is to partition the component with horizontal lines [6], but more sophisticated approaches are possible, such as selecting individual barcode [7].

Usually word segmentation is done after string segmentation. However, there are algorithms that allow first segmentation of words, and then segmentation of strings, based on the proximity criterion and the evaluation of the continuity direction of the connected components.[8].

The problem of segmentation of words in handwritten text is more complicated than in typewritten text. In typewritten text, if the distance between words is more or less constant, and the distance between letters in a word is much less than the distance between words, then its complexity is is that the distance between words in the written text varies greatly.

This issue is resolved as follows. Words are formed from the connected components of the text of the line in question based on the analysis of the distance between these components. The problem of orienting this distance to the distance between word components or the distance between words is solved by classification methods.

Taking into account the sensitivity of the distance between connected components to the shape of these components, many methods of distance calculation have been proposed. For example, in [7] it was proposed to calculate this distance as the Euclidean distance between the nearest points of neighboring components or components drawn inside rectangles. When crossing such rectangles, it is recommended to use the minimum distance between the points of adjacent connected components located on the same horizontal line.

Another approach to distance determination was proposed in [8], where each connected component is drawn inside a convex polygon and the center of gravity of this polygon is determined. Then, the points of the centers of gravity of the adjacent connected components are connected by a straight line, and the intersection points of the convex polygon and the straight line are determined. The distance between the intersection points is taken as the distance between adjacent connected components.

The proposed algorithm is based on the idea of projection analysis, and unlike the algorithm proposed by the authors of [8], the image is painted to obtain the boundary lines more accurately. The algorithm for separating lines and words of handwritten text consists of the following steps [1].

Step 1. Morphological closure with structuring S element size 1×13 is applied to given B binary image:

$$B' = B \bullet S .$$

Step 2. B' is equal to k (or approximately equal) B_t ($t = \overline{1, k}$) is divided into parts.

Step 3. A horizontal projection is calculated for each B_t block based on the following formula:

$$P_t(i) = \sum_{j=1}^N B_t(i, j), \quad i = 1, \dots, M .$$

Horizontal axes with the highest peak in the image portion projection are considered as line candidates.

Step 4. The thresholds for each part are calculated according to the following formulas in order to distinguish the lines dividing the lines in the projection:

$$\theta_t = \frac{1}{M} \sum_{j=1}^M P_t(i)$$

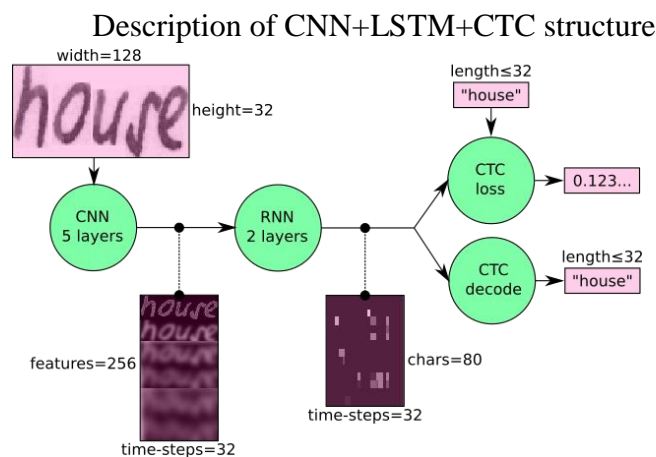
and lines are separated:

$$L_t(i) = \begin{cases} 1, & \text{if } P_t(i) > \theta_t; \\ 0, & \text{otherwise} \end{cases}$$

Step 5. A vertical projection is calculated for word segmentation on each resulting line as in step 3, and words are extracted based on the analysis of this projection.

The development of neural network models in the recognition of handwritten texts did not remain without its influence. Analysis of the recognition results was carried out based on the hybrid model built on the basis of the following structure.

Picture 1.



Based on this structure, it performs segmentation by repeated reading in recognition and has 29 output classes.

This model is trained on the trimmed dataset of words and trained on word matching. CTC training includes its alphabet characters.

These word paths are created by training in CTC. It also helps to get the correct result in the sequence of letters[8].

A dataset of Uzbek writing was formed for the CNN+LSTM Fuzzy CTC hybrid model. Usually, the segmentation procedure in such models is performed during the LSTM model

operation. Each process is trained as a time-dependent process, and the recognition process also uses the time-distributed in turn. It has a limited time step. Each input image is distributed over time. Typically, hybrid neural networks of this structure require word or string segmentation, respectively, in the form of printed text.

Table 1.

A training dataset with row segmentation

№	The title of the read book	Number of rows trained
1	Letters of Latin graphics model_1	2015

are saved to the tmp folder in order. Line segmentation is performed familiarly and sequence recognition is performed. Text will be placed in the memo line.

Based on the CNN+LSTM+Fuzzy CTC hybrid neural model, the training and recognition models were created based on the following data (Table 1) in the software.

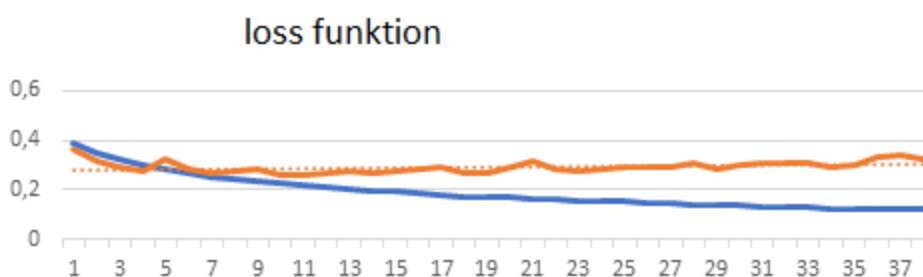
The model is stored in a json file. HDF5 files allow us to store data in a structured way regardless of its type. It serves to extend a limited capability in other programming languages.

We can observe its structural appearance as follows.

Based on the loss values from our trained model file, the result (Picture 2) was obtained.

Picture 2.

A graph of the loss function in the trained model



A model generation was obtained from every 100 steps. This means that each letter is modeled for recognition. If the value is variable and the exact uniformity is not repeated and the value of the loss function does not approach 0.001, we observe that it leads to a large error in recognition as soon as we create its model. [8].

We get a comparative analysis of CNN +RNN +CTC and CNN+LSTM+Fuzzy CTC hybrid neural models of trained data.

Table 2.

A comparative analysis of CNN+RNN+CTC handwritten text recognition

№	Name of the book	CNN+LSTM+CTC	CNN+LSTM+FuzzyCTC
1	Letters of Latin graphics model_1	8	96

Conclusion

We observed the complexity of achieving the full result by using CNN, 45-72 percent recognition results were achieved in the language grammar in the CNN+RNN+CTC model. It was observed that the order of words leads to many deficiencies in recognition. We have studied whether it is possible to increase the percentage of recognition by CNN+LSTM+Fuzzy CTC by using the relevance function in extracting the gradient of the order of points. We got the results

(Table 2). It should be emphasized that the grammar of the language, the order of words, and the recommendation and evaluation in the upper class when the letters are recognized by the image are brought to the efficiency of the result. These results are for text written in Latin graphics.

REFERENCES

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2012. – 1104 с.
2. Местецкий Л.М. Непрерывная морфология бинарных изображений. Фигуры. Скелеты. Циркуляры. – М.: ФИЗМАТ-ЛИТ, 2009. – 287 с. Likforman-Sulem L., Zahour A., Taconet B. Text line segmentation of historical documents: a survey. *Int. Journal of Document Analysis and Recognition*, 9, 2-4, 2007, pp. 123-138.
3. Li Y., Zheng Y., Doermann D., Jaeger S. A new algorithm for detecting text line in handwritten documents. *Int. Workshop on Frontiers in Handwriting Recognition*, 2006, pp. 35-40.
4. Kim G., Govindaraju V., Srihari S.N. An architecture for handwritten text recognition systems. *Int. Journal on Document Analysis and Recognition*, 1999, 2, 1, pp. 37-44.
5. Marti U.V., Bunke H. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In *Proc. of the 6th Int. Conf. on Document Analysis and Pattern Recognition*, 2001, pp. 159-163.
6. Nakajima Y., Mori S., Takegami S., Sato S. Global methods for stroke segmentation. *Int. Journal on Document Analysis and Recognition*, 1999, 2, 1, pp. 19-23
7. Shah M., Jethava G. A literature review on hand written character recognition // *Indian Streams Research Journal*, Vol. 3, Issue 2, 2013, pp. 1-19.
8. Seni G., Cohen E. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27(1), 1994, pp. 41-52.
9. Kudratov G., Eshmuradov D., Yadgarova M. GENERAL ISSUES OF PROTECTION OF THE BACKLINE COMPUTER NETWORKS // *Science and Innovation*. – 2022. – Т. 1. – №. 8. – С. 684-688.