Armenian Texts Recognition via Neural Networks

Anna Hovakimyan
Yerevan State University
Yerevan, Armenia
e-mail: ahovakimyan@ysu.am

Narine Ispiryan
Yerevan State University
Yerevan, Armenia
e-mail: nispiryan@ysu.am

Gevorg Narimanyan
Yerevan State University
Yerevan, Armenia
e-mail:
gevorgnarimanyan@yahoo.com

ABSTRACT
Nowadays the text recognition on different languages by computer is an urgent task. Need to solve this problem is due to the fact that in many areas there is a vast amount of archival information contained in the paper, and it is necessary to keep this information on the web and make it more accessible to broad audience for editing and using. Gathering of this kind of information is enough extensive work, and its implementation is often hindered by the lack of time and resources. That is the reason to have automated processes for recognition of scanned pictures of texts.

There are a number of free text recognition systems for English and other several languages, while for Armenian does not exist such an alternative.

In this paper a solution of this problem for scanned printed texts in Armenian language and its implementation with neural networks are presented. On the base of this approach the software was developed that recognized the printed Armenian texts successfully enough. In future it is foreseen to refine the characteristics of the developed neural networks system to provide the more effective and adequate text recognition.

Keywords
Text recognition, text dividing, neural networks.

1. INTRODUCTION
The problem of text recognition in different languages always has been actual for different purposes in various areas. Need to solve this problem is due to the fact that in many areas there is a vast amount of archival documents, which are contained in the paper. It is necessary to keep the scanned documents on the web that should be more accessible for broad audience to edit and use them in various aims.

Gathering information repeatedly is enough extensive work and its implementation is often hindered by the lack of time and resources. That is the reason to have automated processes for recognition of scanned documents.

There are a number of free text recognition systems for English and several other languages, and it will be desired to have such tools for Armenian language [1, 2] . As a rule the existing systems are private and their source codes are not accessible for adaptation and localization.

In the current paper a solution of the problem for Armenian printed texts recognition and its implementation with neural networks are presented. On the base of this approach the software was developed.

This software system consists of two parts. The first one provides input data preliminary processing that are given as images. The problems of text separating into lines, and lines separating into words, as well as words separating into letters are solved. The second part provides text recognition via neural network. This part involves tasks such as neural network design, creation of samples for neural network training, and neural network training. The recognized text is presented through any text editor, where it is possible to correct errors mechanically. It is obvious that the time taken for such kind of correction is much less than the time that is needed for keying in all the text. The developed software provides enough high percent of text recognition so the words after recognition are understandable and text editing will be very fast.

In the future it is assumed to prepare more flexible samples for neural network training, as well as to increase the network dimension that may lead to more effectiveness and adequacy of text recognition.

2. DATA PRELIMINARY PROCESSING
The text being under consideration is scanned and presented as image. Before its recognition data preliminary processing is being performed. This process consists of the following steps: text dividing into lines, lines dividing into words, words dividing into letters. Then neural network is designed and trained for text recognition letter by letter.

We assume, that text is horizontal oriented (Fig.1).

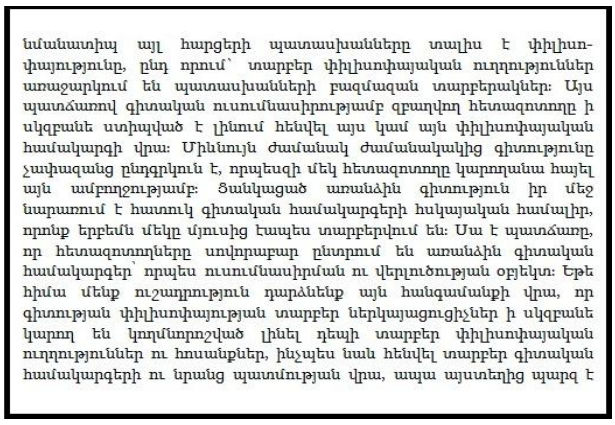
նմանատիպ այլ հարցերի պատասխանները տալիս է փիլիսո-
փայությունը, ընդ որում՝ տարբեր փիլիսոփայական ուղղություններ
առաջարկում են պատասխանների բազմազան տարբերակներ: Այս
պատճառով գիտական ուսումնասիրությամբ զբաղվող հետազոտողը ի
սկզբանե ստիպված է լինում հենվել այս կամ այն փիլիսոփայական
համակարգի վրա: Միևնույն ժամանակ ժամանակակից գիտությունը
չափազանց ընդգրկուն է, որպեսզի մեկ հետազոտողը կարողանա հայել
այն ամբողջությամբ: Ցանկացած առանձին գիտություն իր մեջ
ներառում է հատուկ գիտական համակարգերի հսկայական համալիր,
որոնք երբեմն մեկը մյուսից էապես տարբերվում են: Սա է պատճառը,
որ հետազոտողները սովորաբար ընտրում են առանձին գիտական
համակարգեր՝ որպես ուսումնասիրման ու վերլուծության օբյեկտ: Եթե
հիմա մենք ուշադրություն դարձնենք այն հանգամանքի վրա, որ
գիտության փիլիսոփայության տարբեր ներկայացուցիչներ ի սկզբանե
կարող են կողմնորոշված լինել դեպի տարբեր փիլիսոփայական
ուղղություններ ու հոսանքներ, ինչպես նաև հենվել տարբեր գիտական
համակարգերի ու նրանց պատմության վրա, ապա այստեղից պարզ է

Figure 1. Image of text

The image of text is considered as a matrix $B=\{b_{ij}\}$ ($0 \le b_{ij} \le 1$, i=1... n, j=1... m), where n is the width of the image in pixels, m is the height, b_{ij} are shades of gray of the pixel in

the position (i,j) of the image. 0 is assigned to black color, and 1 is assigned to white color.

2.1. Text dividing into lines

The solution of this problem adds up to finding the upper and lower bounds for each row of the text (Fig.2). The dividing approach is based on that fact that the pixels illumination between lines is much less than this one within the lines.

At first the average illumination s_j of all pixels, and then the average illumination s(B) of the whole image are calculated.

$$s_j = s_j(B) = \frac{1}{n} \cdot \sum_{i=1}^{n} b_{ij}$$

$$s(B) = \frac{1}{m} \cdot \sum_{j=1}^{m} s_j(B)$$

Because the pixels' illumination between lines is less than within lines, the bounds of the line can be determined by the whole picture average lighting number.

$$s^t = k^t * s(B),\ 0<k^t<1,$$

$$s^b = k^b * s(B),\ 0<k^b<1.$$

The upper bound of a line is fixed if

- the average illumination of the current pixel-line is greater than s^t,
- the average illumination of two previous pixels-lines is less than s^t, and
- the average illumination of the next some pixel-lines is greater than s^t.

The lower bound of a line is fixed if

- the upper bound of a line has already been fixed,
- the average illumination of the current pixel-line is greater than s^b,
- the average illumination of the next some pixel-lines is less than s^b.

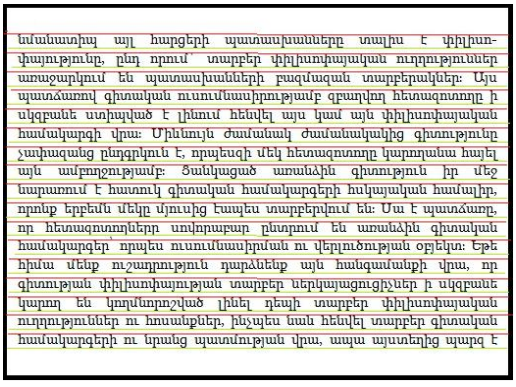

Figure 2. Text dividing into lines

2.2. Lines dividing into words

Lines partitioning into words is similar to text dividing into lines. The idea is the same: the pixels illumination between words is much less than this one within the words. As an input for the lines partitioning algorithm is a line obtained as a result of text dividing (Fig. 3).

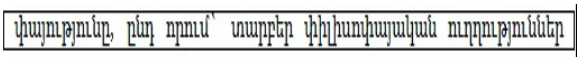

Figure 3.The image of a separated line of text

The algorithm is implemented in two steps:

- blacken all pixels around black pixels by two pixels to result in vague line in order to obtain explicit parts between words (Fig. 4),

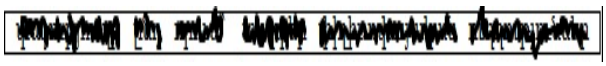

Figure 4.The vague line

- divide words as such as divide line but in this case it must identify the beginning and the end of words (Fig. 5). For this purpose it must be calculated the average illumination c_i of all pixels of the vague line, and then the average illumination c(B) of the vague line

$$c_i = c_i(B) = \frac{1}{m} \cdot \sum_{j=1}^{m} b_{ij}$$

$$c(B) = \frac{1}{n} \cdot \sum_{i=1}^{n} c_i(B)$$

Because the pixels illumination between words is less than within words, the bounds of the word can be determined by the vague line's average lighting number.

$$c^l = k^l * c(B),\ 0<k^l< 1,$$

$$c^r = k^r * c(B),\ 0<k^r < 1.$$

The algorithm considers columns of pixels to determine the bounds of the word in the following manner. The initial bound of a word is fixed if

- the average illumination of the current pixel-column is greater than c^i,
- the average illumination of the previous pixel-columns is less than c^i, and
- the average illumination of the next some pixel-columns is greater than c^i.

The final bound of a word is fixed if

- the initial bound of the word has already been fixed,
- the average illumination of the current pixel-column is greater than c^r,
- the average illumination of the next some pixel-columns is less than c^r, and
- the average illumination of the two previous pixel-columns is greater than c^r.

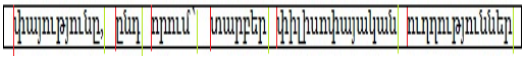

Figure 5. Line dividing into words

2.3. Word dividing into letters

As an input for the algorithm dividing a word into letters is a word obtained as a result of line dividing. Words recognizing algorithm is presented in three steps. It aims to determine the boundary columns between the letters in a word.

In the first step an average lighting value of pixel-columns that form the word image is calculated. Then the word is separated by vertical lines according to pixel columns, whose pixel lighting is less than the average one (Fig. 6).

Figure 6.Letters potential boundaries identifying (step 1)

In the second step from the boundaries obtained after the first step only those are kept in mind that have a pixel lighting less than a given threshold (Fig. 7).

Figure 7. Letters potential boundaries identifying (step 2)

In the third step the main three lines that are typical for Armenian letters are found (Fig. 8).

Figure 8. Identifying of the most typical three boundary lines of Armenian letters

With those pixel-lines and bound columns obtained after the first step the letters real bounds are determined. If a considered column is surrounded by black pixels from the two sides in any of three identified lines then that bound is false and must be rejected (Fig.9).

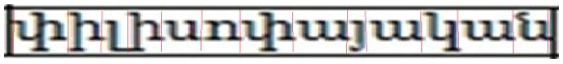

Figure 9. Letters real boundaries identifying (phase 3)

So as a result of the described process a two-dimensional arrays of letters images are obtained.

3. NEURAL NETWORK FOR ARMENIAN TEXT RECOGNITION

In the paper for Armenian text recognition a mechanism of neural networks is suggested. It must be mentioned that networks with a single layer of adaptive weights have a number of important limitations in terms of the range of functions which they can represent [3]. To allow for more general mappings it might be considered successive transformations corresponding to networks having several layers of adaptive weights. In fact many examples show that networks with just two layers of weights are capable of approximating any continuous functional mapping [4].

It is important to note that the experiments conducted with single-layer neural network developed as draft version did not give enough satisfactory results in Armenian text recognition. The situation was essentially changed while a two-layer neural network with back-propagation training model was used.

A multi-layer neural network has several layers that are parallel to one another, and where each layer's output signals are input to the next one (Fig. 10).

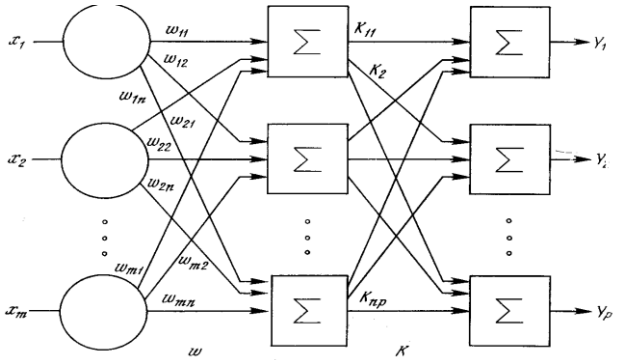

Figure 10. Multi-layer neural network

The neural network training with teacher is used when a training sample is a combination of input and desired output vectors. During the training phase the network compares the resulted output with the desired output, and depending on the size of the error the back-propagation algorithm changes weight matrix until the error rate for all the input vectors reaches an acceptable level [3,4].

In our case the input vectors are Armenian letters presented as sets of pixels. Each black pixel is coded by 1, and each white pixel is coded by 0 (Fig.11).

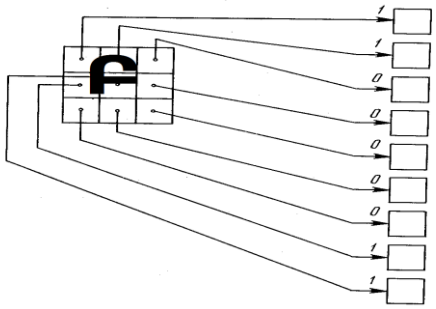

Figure 11. Letter coding

Calculations during network training are performed in turn layer by layer via back-propagation algorithm. The output vector of the previous layer serves as an input vector of the next one (Fig. 12). The experiments show that sigmoidal function of activation is the most convenient for this problem solving.

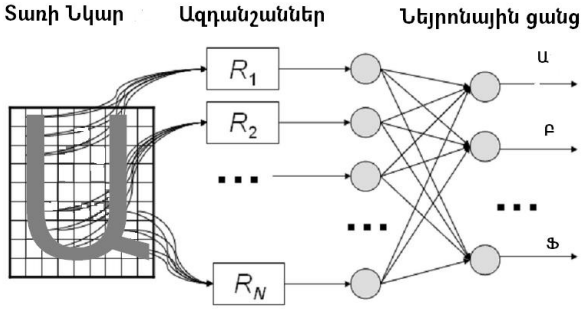

Figure 12. Neural network for Armenian letters recognition

Software was developed that provides enough high percent of text recognition (Fig. 13).

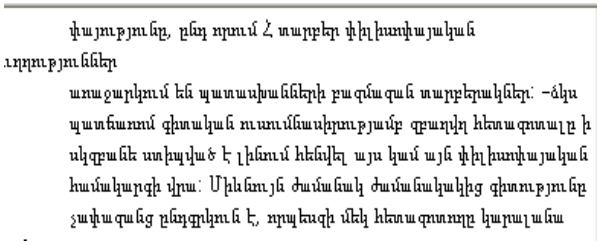

Figure 13. Recognized text

The recognized text is presented in any text editor environment, where it is possible to correct errors mechanically. Note that the text in Fig. 13 is a recognized part of the text presented in Fig.2. It appears that the recognition is successfull enough.

The developed software can be applied in any area, where the convertation of Armenian scanned texts is needed.

REFERENCES

[1] Shi H., Pavlidis T., "Font Recognition and Contextual Processing for More Accurate Text Recognition*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 17, № 9,* pp. 39-44,1997.

[2] V.L. Arlazarov, P.A.Kuratov, O.A.Slavin, " Recognition of lines of printed texts", *Information technologies and computing systems № 1*, pp. 48-54,1996 (in Russian).

[3] Ch.M.Bishop,"Neural Networks for Pattern Recognition", *Birmingham, Clarendon Press, Oxford,UK,* 1995.

[4] Wang J., Jean J. Segmentation of merged characters by neural networks shortest path. *Pattern Recognition 27,Vol.5* , pp. 649-658, 1994.