

Recognition of traditional Mongolian script using primitives and template matching methods

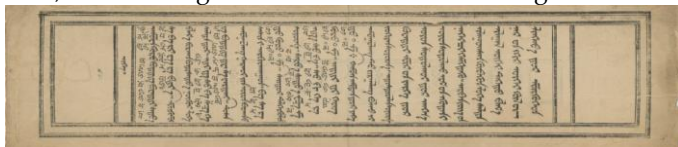
Byambasuren Ivanov, Uuganbaatar Dulamragchaa, Markhamet Musa, Otgonnaran Ochirbat

Abstract – Designing an appropriate algorithm and methods in case of speed and accuracy for character recognition has become a necessity in regard to importance of character recognition for various purposes such as: keeping, maintaining, and promoting the one's cultural and historical heritages, scriptures etc. Traditional Mongolian script, which has a unique writing style and multi-font variations, brings challenges to character recognition. In this paper we primarily studied an Optical Character Recognition (OCR) of a typewritten and woodcut printed Mongolian Script by using primitives and template matching methods. Template matching method has two phases which are separating letters and then recognizing them each of which are processed separately, whereas in the Primitive method separation and recognition are done simultaneously. We developed a software and tested the template matching (TM) method. This method worked well with typewritten documents only with certain fonts but couldn't do so well on woodcut print recognition. So further, we have developed an algorithm for recognition of Mongolian script by decomposing them into containing primitives. We assume that all Mongolian script letters contain 7-primitive elements. At first primitive elements are extracted by using the modified Hough Transform method and make the primitive arrays. And then these elements from first to end of the array are compared with Character Identification Vector (CIV) and recognizes the characters. The primitive method is able to recognize any type of printed document with higher accuracy and more efficient than the method.

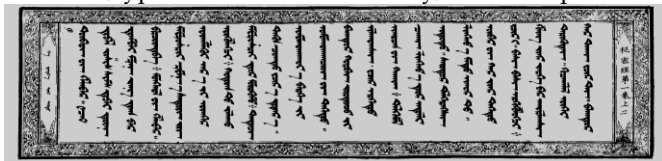
Keywords – Traditional Mongolian Script, Optical character recognition, Template matching, CIV, Hough Transform, Mongolian language, Primitive;

1. INTRODUCTION

Till today lots of books, scriptures, and historical writings have been kept in the public and national libraries, museums, temples and individuals separately. Digitizing all these historical and ancient books and scriptures has emerged to facilitate an access and preservation. The Traditional Mongolian script is unique in writing. Because of its unique characteristics the Mongolian script recognition needs special research and studies. Unlike English or Cyrillic Mongolian writing, the Mongolian traditional script is written from up to down, and from right to left as shown in the images below.



Year/type: Written in 15th Century/ Woodcut print



Title: Ganjuur / Woodcut print

All the Mongolian scripts are written in the fixed place left or right side of the straight line which we call it baseline. Mongolian script characters in regard to their positions in words are divided into three groups: *first*, *middle*, and *end letters*. Before many papers have been published regarding the Mongolian script recognition i.e. [1], [2], [3]. But according to those works, it is essential to mention that in those methods that use training, there are a lot of challenges such as correct training, time consuming, and dependence of size, type and font. In this regard we have proposed primitive method which use thinning method to detect noisy lines and stroke, and through which we can produce a primitive elements. We also have studied and tested traditional template matching method for the purpose of further comparison. In Section 2 we write about recognizing Mongolian script using the template matching method. In Section 3 we write about recognizing Mongolian script letter using Primitive method and in the section 4 we will write about conclusion. In Section 5, References are listed.

2. RECOGNITION OF MONGOLIAN SCRIPT USING TEMPLATE MATCHING METHOD

We produced a software to recognize a typewritten documents using a Template Matching method. However the result of the process heavily depends on the type and format of the input and template characters. This method is suitable only for recognizing typewritten printed documents. The main recognition flowchart of the algorithm is shown in the following Figure-1.

- Byambasuren Ivanov is currently pursuing a PhD in CS and a research scientist at Institute of Physics and Technology, Mongolian Academy of Science, Ulaanbaatar, Mongolia, PH+976 458090. E-mail: byambasureni@mas.ac.mn
- Uuganbaatar Dulamragchaa, PhD, is a director of Informatics department of Institute of Physics and Technology, Mongolian Academy of Science, Ulaanbaatar, Mongolia, PH+976 458090. E-mail: uuganbaatard@mas.ac.mn
- Markhamet Musa is a research scientist at Institute of Physics and Technology, Mongolian Academy of Science, Ulaanbaatar, Mongolia, PH+976 458090. E-mail: markhametm@gmail.com
- Otgonnaran Ochirbat is a Computer science faculty at School of Engineering and Applied Science, National University of Mongolia, Ulaanbaatar, Mongoli E-mail: otgonnaran@seas.num.edu.mn

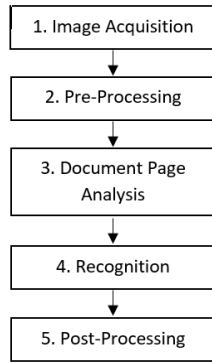


Figure 1. Steps of Template matching recognition

- 1. Image acquisition:** Input image. Image could be acquired through multiple sources or devices.
- 2. Pre-Processing:** In this step we would prepare and analyze the input image for further use.
 - Binarization: Any image or colors will be converted into black and white colors through local or global binarization. In this step we use threshold method to analyze and process the image into black and white colors. Thus:

$$\text{Value}(x, y) = 255 \text{ if } \text{Value}(x, y) > T^*$$

$$\text{Value}(x, y) = 0 \text{ if } \text{Value}(x, y) \leq T^*$$

**T is the value of Threshold*
 - Noise Removal: Noise is a common problem in most of the image understanding problems. We used a halftones method.
 - Skew Correction: We used the analysis of projection profile method for the skew correction step.

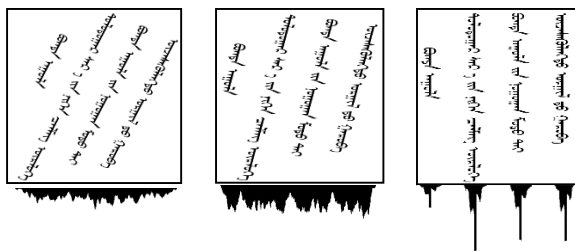


Figure 2. Projection Profile Analysis method

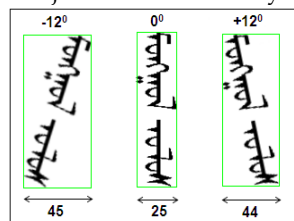


Figure 3. Skew correction

The algorithm based on projection assumes an axis-aligned scan.

```

for theta = -angle to + angle by resolution do {
  for r = 0 to NROWS do
    for c = 0 to NCOLS do
      if (image[r][c] == BLACK) {
        rotate(r, c, theta, &new_row);
        ++proj_prof[new_row];
      }
    }
  angle_measure[theta] = criterion_function(proj_prof[]);
}
skew_angle = choose_skew(angle_measure[]);
    
```

Figure 4. Skew correction algorithm

- 3. Document Page Analysis:** The process of document page analysis aims to decompose a document image into a hierarchy of homogenous regions, such as figures, background, text blocks, text lines, words, characters, graphs, etc.

-Line Segmentation using projection profile analysis method.

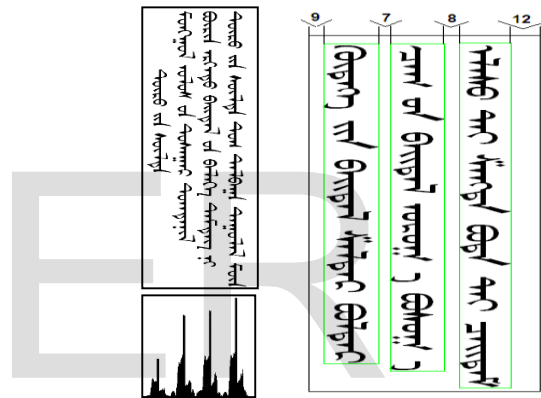


Figure 5. Line segmentation

-Word Segmentation using projection profile analysis method.

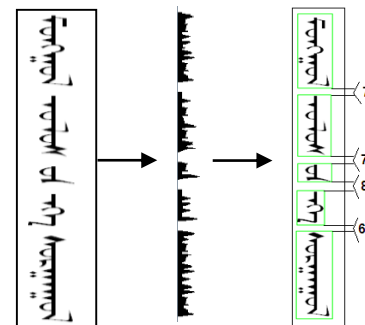


Figure 6. Word segmentation

- 4. Letter Recognition:** This step analyzes and recognizes the character accordingly with the given template. The result depends on the type and fonts of the input image. We used Template matching (Matrix Matching) method which is the way of recognizing the character by comparing and matching the input image with the template (pre-prepared) character as shown in fig.7 below. In our software we have

used CMs Urgo, and Ulaanbaatar fonts only. For other fonts, it has to be predefined as a template.

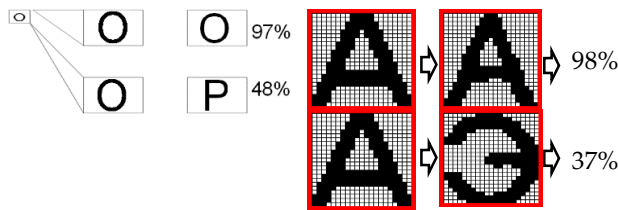


Figure 7. Template matching method

To do so we first remove the base-line (spine-line) from the word as shown in the Fig.8 below.

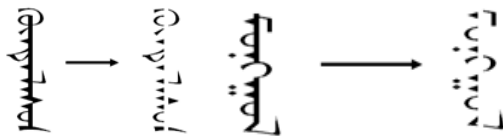


Fig 8. Removing baseline.

Here we can see that letters are separated in between, and also divided in to two parts by baseline. We can divide Mongolian script letters into two categories such as letters which are written on both sides of the base-line and the letters which are written only in the left side of the base-line as shown in the Figure-9 below.

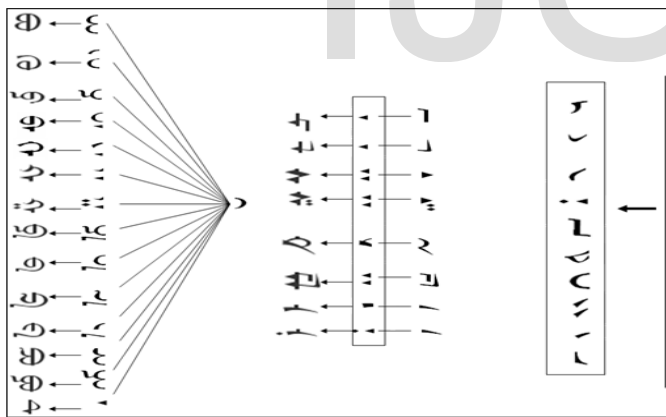


Figure 9. Reconstructing letters after removing the baseline for CMs Urgo font.

So this Fig.9 illustrates only the CMs Urga Font type. After removing the base line we use projection analysis method to segment the letters from one another just like a word segmentation which we have done previously. After separating the letters we do recognize the letters by matching with the template letters which is pre-prepared. The recognition process flows from up to down and right to left. First we do recognize the right side characters of the base-line and then we combine them with the left side characters which can only then be a complete letter. By doing

so we can recognize all the both side written letters. After that we do recognize the only left side written letters. In simply, the word recognition sequence contains the following steps.

- a) Identifying and removing the base-line of the word.
 - b) Identifying the locations of the characters which remained after removing the base-line.
 - c) Identifying the right side characters and finding and joining it with the respective left side characters.
 - d) Identifying all the left side written letters only.
- Please see Fig.10.

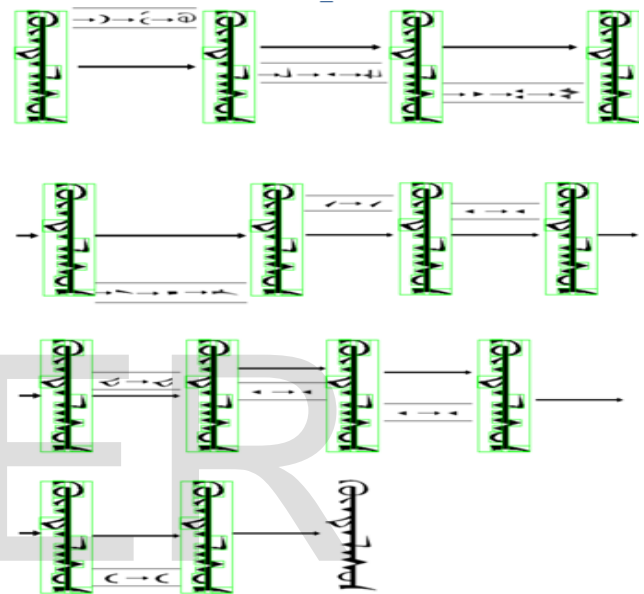


Figure 10. Mongolian script recognition character.

5. Post-Processing: Post processing does analysis, validation and spell checking etc. on the result (output) of the word recognition processes. Generally the post-processing step is a separate research topic, and we aim to continue this study in our further research works.

3. RECOGNITION OF MONGLIAN SCRIPT USING PRIMITIVES

Unlike the Template Matching method, the Primitive method is more accurate, and capable of recognizing any type of documents such as typewritten, woodcut prints, even the handwritten documents if well developed. We created an algorithm for optical character recognition in Mongolian script by decomposing them in to respective primitive elements. So far we suggest 7-primitive elements [4] for all the Mongolian script letters. This method performs separation and recognition simultaneously.

The main recognition flowchart of the algorithm [4] is shown in the following Fig.11.

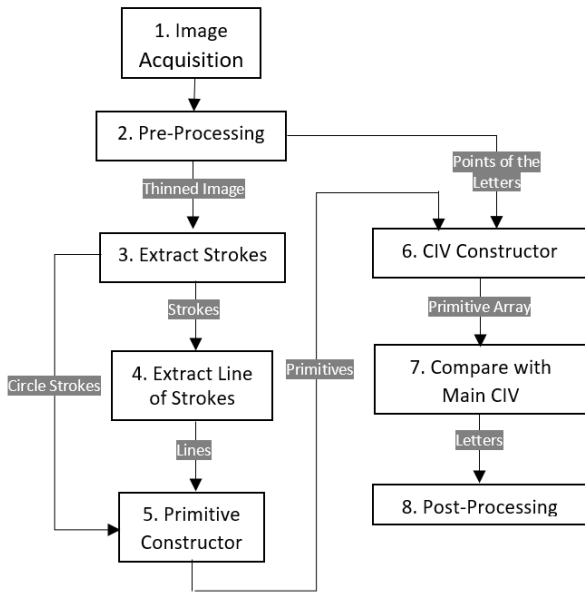
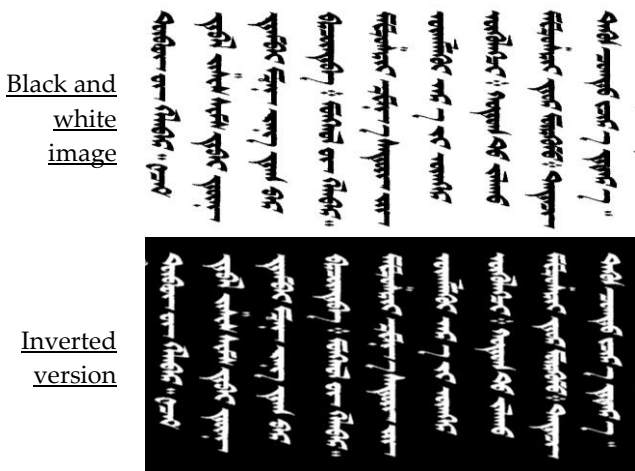


Figure 11. The main flowchart of the algorithm.

- 1. Image acquisition:** Input image. Image could be acquired through multiple sources or devices.
- 2. Pre-Processing:** After binarization, filtering, and smoothing the image, each lines and words are separated, using horizontal and vertical projections analysis respectively. As we have identified the baseline of the word through projection analysis method, we can identify the positions of the dots whether it is left or right of the line. Separated words go to thinning process as shown in Fig.12. Thinning algorithm is taken from [6]. After this level, the positions and the number of dots are saved to an array and eliminated from the main image. These processes have a great impact on correct extraction of strokes, primitives, positions and number of dots.



After thinning process

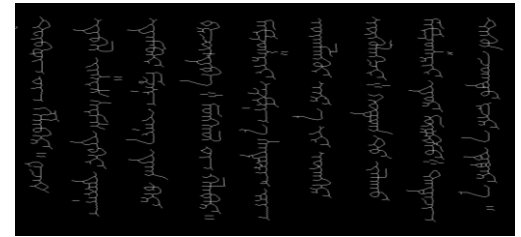


Figure 12. Thinning process on Ganjuur.

- 3. Extract Strokes:** The image without dots goes to strokes extraction procedures. In this step, we should find the starting and junction pixels. A starting pixel is the one which does not have more than one neighborhood and the junction pixel is the one which has more than two neighborhoods. We start from the most right and up pixel of the word and go through the black pixels or chain code. If we reach the first junction pixel, then these pixels are for one stroke and omitted from the main image and saved in the first stroke image. The start pixel and junction pixel of that stroke are omitted from the list of start and junction pixels. This procedure continues for all the pixels in the main image. If the starting and the junction pixels of the stroke were the same, it is a circle and we call it O primitive. After this process the strokes go to the extracting the lines of strokes level and extract the primitives, using the modified Hough Transform [5].

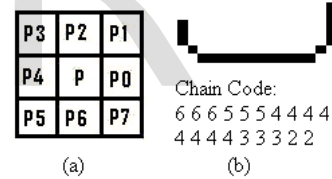


Fig 13. (a) The octagonal neighborhoods of a pixel, and (b) the skeleton of the letter and its chain code.

- 4. Extract Lines of Strokes:** Hough Transform is a suitable method to extract the lines of a stroke [5]. Each stroke in a thinned image is a collection of joined pixels in which there are no more than two neighborhoods. These collections are called Single Stroke and the others are called Multi Stroke. By this method we get the variety of lines in a simple single stroke which is not trusty. So we choose those lines that places on the chain code [7]. According to the Octagonal neighborhood, chain code is an array of these neighborhoods that the skeleton of the letter is laid on it as shown in the Fig.13 above. And the Fig.16 below illustrated the skeleton of the letter and its chain code. For calculating the chain code we start from the beginning of the stroke and by considering the table which is shown in Fig.13 (a), go across the main body and give a number of direction for each pixel. We use Hough Transform by choosing the lines laid on the chain.

5. **Primitive Constructor:** In this process all the primitives are constructed. We propose that each letter of Mongolian script can be constructed by utmost 7-primitives as shown below table.1.

No.	Shape of Primitives	Name of Primitives	Code of Primitives
1	/	Slash	S
2	\	Backslash	B
3		Vertical	V
4	—	Horizontal	H
5	⌋	C shape	C
6	○	Circle	O
7	└	Corner	L

Table 1. The sets of Primitives in Proposed OCR

Extracting Primitives: All the primitives that exist in the words are extracted. If the main image is noisy, Hough Transform with 45-degree may have more problems. For recovering this problem the 22.5-degree angle is selected. This means that the extracted lines are in 8-angles of 0, 22.5, 45, 67.5, 90, 112.5, 135, and 157.5-degrees. By considering the 3x3 neighborhood of each pixel in main angles of 0, 45, 90, and 135 degrees are calculated and nominated D0, D1, D2, and D3 respectively. According to these values for each line, the value of W is calculated and the values of parameter space are added by them. This algorithm of Hough Transform is shown below in Fig 15.

By using the rules below, the primitives are stored in Stroke Identification Vector (SIV) and are ready to be recognized. SIV, as shown in Fig.14, is a Vector that defines the primitives, positions and number of dots in a stroke accordingly.

Primitive 1	Primitive 2	Primitive 3	Primitive 4	Points (left)	Points (right)
-------------	-------------	-------------	-------------	---------------	----------------

Figure 14. CIV and SIV

- Rule-1: "B", "H", "S" and "V" primitives are specified by lines angles' ± 22.5 degrees.
- Rule-2: The "C -Shape" primitive is the rotation of S collection by 180 degrees.

$$S = \{ [0 \pm 22.5, 45 \pm 22.5, 90 \pm 22.5, 135 \pm 22.5, 0 \pm 22.5] \\ [0 \pm 22.5, 90 \pm 22.5, 0 \pm 22.5] \\ [0 \pm 22.5, 45 \pm 22.5, 135 \pm 22.5, 0 \pm 22.5] \\ [45 \pm 22.5, 90 \pm 22.5, 135 \pm 22.5, 0 \pm 22.5] \}$$

If the sequence of lines is $\{90^\circ, 0^\circ, 90^\circ\}$ and the number of pixels in 0 degree line (NP_2) is more than the total number of pixels in other lines, then the primitive is "H", (1).

$$\text{If } S = \{90^\circ, 0^\circ, 90^\circ\} \text{ and } NP_2 \geq (NP_1 + NP_3) \Rightarrow 'H' (1)$$

- Rule-3: If the sequence of collection S is $\{90^\circ, 0^\circ, 90^\circ\}$, it is "S" or "L-Shape" primitive (2)


$$\text{If } S = \{90^\circ, 45^\circ, 0^\circ\} \text{ and } (NP_2 < NP_1) \text{ and } (NP_2 < NP_3) \Rightarrow 'L' (2)$$

$$\text{If } S = \{90^\circ, 45^\circ, 0^\circ\} \text{ and } (NP_2 \geq NP_1) \text{ and } (NP_2 \geq NP_3) \Rightarrow 'S' (2)$$

```

For each pixel at (xi, yi) coordinate do
    D0=P0+P4    --for 0 degree
    D2=P1+P5    --for 45 degree
    D4=P2+P6    --for 90 degree
    D6=P7+P3    --for 135 degree
    D8=D0
    For i=0:7
        θ=i* π/8
        ρ=xi cosθ+yi sinθ
        W(i) = { Di if i is even
                max(Di-1 mod 2, Di+1 mod 2) if i is odd }
        H(ρ,θ)=H(ρ,θ)+W(i);
    End For
End For
    
```

Figure 15. The algorithm of modified Hough Transform

As illustrated in Fig.16, for letter  there are two parts. One part is a line 90 degree and the other is a backslash with 157.5 degree. The SIV of this letter is "VB00". SIV has four positions of primitives and has two positions for storing the number of dots.

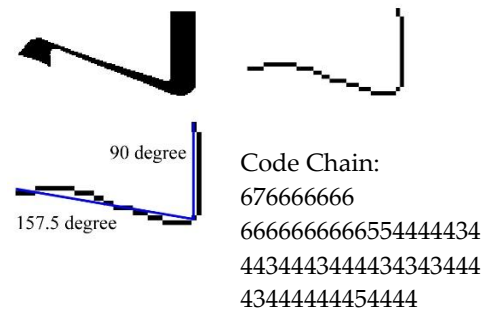


Figure 16. The Mongolian script with its thinned image. The lines are extracted by using the chain codes.

6. **CIV Constructor:** For each Mongolian script letter we have Character Identification Vector (CIV), which is similar to SIV. The CIV of Mongolian script letters are shown in Table 2 which shows containing primitive elements for respective letters letter.

Separate Letters	Primitives Set				Points	
	P1	P2	P3	P4	Points (left)	Points (right)
ᠠ	V	H			0	0
ᠡ	V	B			0	0
ᠢ	V	H	B		0	0
ᠣ	S				0	0
ᠤ	S	C			0	0
ᠤ	O				0	0
ᠤ	O	C			0	0
ᠤ	V	H			1	0
ᠤ	V	H	B		1	0
ᠤ	V	H	H	C	0	0
ᠤ	S	C	H		0	0
ᠤ	H	C			0	0
ᠤ	S	C	H		2	0
ᠤ	V	H	V	H	2	0
ᠤ	V	B	S	C	0	0
ᠤ	B	S			0	0
ᠤ	V	H	B		0	0
ᠤ	B	S			0	2
ᠤ	V	H	C		0	0
ᠤ	S	B	S	V	0	0
ᠤ	V	H	L		0	0
ᠤ	V	H	B	L	0	0
ᠤ	V	H	H	V	0	0
ᠤ	V	S	B	C	0	0
ᠤ	V	B	V		0	0
ᠤ	V	S	S		0	0
ᠤ	V	B	B		0	0
ᠤ	V	S	C		0	0
ᠤ	V	S	C	C	0	0
ᠤ	V	S	H		0	0
ᠤ	S	B	S		0	0

Table 2. The Primitives of Mongolian script letters

7. Compare with Main CIV: This step constructs the letters by matching the primitive array with CIV. The steps are shown in the Fig. below.

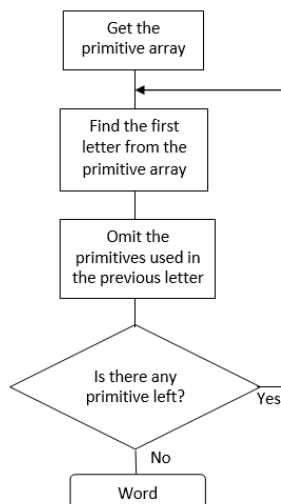


Figure 16. Character constructing flowchart.

EXAMPLE:

The example, as shown in Fig.17, contains two words and five letters. The first column of words are taken from the woodcut print script which is processed through thinning and then divided into the letters. So the following word primitives contain VH10, VH00, VSCC00, O00, and VHB00.

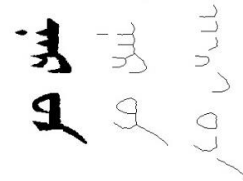


Figure 17. Example from Ganjuur

4. CONCLUSION

In this paper we studied two methods for the recognition of traditional Mongolian script. The algorithm of primitive method, which we proposed in this paper, recognizes the Mongolian script words without any predefined information about types, size, and their font style. On the other hand, Template matching method fully depends on the predefined information and template. Template matching method performs recognition through two phases, which are separation and recognition of the letters separately. This method is time and resource consuming. On the other hand, we have primitives which performs separation and recognition simultaneously. In this paper mainly we tried to recognize two types of printed documents which are typewritten and woodcut print. According to software which we have developed for the experiment the template matching method was 91% accurate with typewritten document as it has standard fonts and sizes. But, this method was quite weak in recognizing the woodcut print as it has no standard fonts, sizes, or types. As the primitive method doesn't depend on any kind of predefined fonts, sizes or information, thus we consider that according to the [3] result, it can increase the accuracy and speed efficiency.

REFERENCES

1. Liangrui Peng, Changsong Liu, Xiaoqing Ding, Janming Jin, Youshou Wu, Huawe Wang, Yanhua Bao, "Multi-font printed Mongolian document recognition system", International Journal on Document Analysis and Recognition, June 2010, Volume 13, Issue 2, PP 93-106.
2. S. Batsuuri, L. Choimaa, B. Unursaikhan and J.Ko, "Line profile based fast approach for recognizing traditional Mongolian Script", 2017 7th International Conference on Ubi-Media Computing and Workshops (UMEDIA), Ulaanbaatar, Mongolia, 2014, pp.182-185.
3. S.Batsuuri, B.Unursaikhan, L.Choimaa, "Traditional Mongolian Script feature extraction based on Black pixels in

- Bounding box", International Journal of Software Engineering and its Applications 2017 11(6) 53-60.
4. S. Ensafi, M. Eshghi, and M. Naseri "Recognition of Separate and Adjoint Persian Letters Using Primitives" IEE Symposium on Industrial Electronics and Application (ISIEA 2009), Oct 4-6, 2009, Kuala Lumpur, Malaysia.
 5. S. Touj, N. Amara and H. Amiri, "Generalized Hough Transform for Arabic optical character recognition", in Proc.Document Analysis and Recognition, ICAR 2003, August 3-6. 2003, pp.1242-1246.
 6. B.K. Jang and R.T. Chin, "One-pass parallel thinning: analysis, properties, and quantitative evaluation", IEEE Transactions on pattern Analysis and Machine Intelligence, vol 14, no. 11, pp.1129-1140, Nov. 1992.
 7. S. Ensafi, M. Eshghi, M. Miremadi, M. Naseri and A. Keipour, "Recognition of Separate and Adjoint Persian Letters in Less than Three Letter Subwords Using Primitives," Proceeding of Iran 17rd Electrical Engineering conference, Tehran,Iran, 2009.

IJSER