I.J. Mathematical Sciences and Computing, **2015, 2, 22-29**
Published Online August 2015 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijmsc.2015.02.03

Available online at http://www.mecs-press.net/ijmsc

Clustering of Multi Scripts Isolated Characters Using *k*-Means Algorithm

Neeru Garg[a], Munish Kumar[b]

[a]*Computer Science & Technology, Lovely Professional University, Phagwara, Punjab*
[b]*Department of Computer Science, Punjab University Rural Centre, Kauni, Muktsar, Punjab*

Abstract

The aim of this paper is script identification problem of handwritten text which facilitates the clustering of data according to their type of script. In this paper, collection of different types of handwritten text document i.e. Devanagari, Gurumukhi and Roman is taken as input and then cluster of all these documents according to script type whether i.e. Devanagari, Gurumukhi, or Roman was prepared. Clustering of handwritten multi-script document scheme proposed in this paper is divided into two phases. First phase used to extract the features of given text images. In the second phase, features extracted in the previous phase were used for clustering with k-Means algorithm. In feature extraction phase, we have extracted four types of features, namely, circular curvature feature, horizontal stroke density feature, pixel density feature value and zoning based feature. In this study, we have considered 4,850 samples of isolated characters of Devanagari, Gurumukhi and Roman script.

Index Terms: Clustering, Script identification, Stroke density, Zoning, *k*-Means.

1. Introduction

Character recognition is an important area in image processing and pattern recognition fields. The main aim of character recognition is to translate human readable characters into machine processable format. Handwritten character recognition has received extensive attention in academic and production field. The handwritten character recognition system can be of two types online and offline. In online handwriting, characters are written generally on pressure sensitive surface and in case of off-line handwriting, characters are presented in digital image format. Character recognition and clustering can solve many complex problems and makes human job easier. Most of the states in India have more than one language of communication. Thus, many official documents are containing multi-script texts in nature. Script identification makes the task of

* Corresponding author. Tel.:
E-mail address:

analysis and recognition of the text easier by suitably selecting the modalities of optical character recognition system. There are different techniques that can be used for clustering multi-script handwritten documents. A few attempts have been made to isolate and identify the scripts of the texts in the case of multi-script Indian documents. Most of these attempts consider mono-script texts. In this paper, we have considered tri-lingual (Devanagari, Gurumukhi and Roman) documents which require script recognition at isolated character level. For clustering, we have considered *k*-means algorithm. Before inputting the data to be processed, the image has to be resized to 100×100. Multi-script text recognition is becoming popular in offices, library, banks, insurance companies and post offices. Devanagari is most popular script in India and used for writing the Hindi, Marathi and Nepali, and is the most common script used to write Sanskrit. Several other languages have scripts which are related to Devanagari, such as Bengali and Gujarati. The Devanagari script represents the sounds of the Hindi language with remarkable consistency. Whereas many letters of the English alphabet can be pronounced in many different ways, the letters of the Devanagari script are pronounced consistently (with a few minor exceptions). Thus, it is relatively easy to learn. It consists of 11 vowels and 33 consonants. It is written from left to right. Gurumukhi script is used primarily for Punjabi language, which is the world's 10th most widely spoken language. Some of the properties of Gurumukhi script are: Gurumukhi script is cursive and the character set consist of 41 consonants, 9 vowels, 3 sound modifiers (semi-vowels) and 3 half characters. The Roman alphabet was derived largely from the Greek and was almost the same as the one we use today. Roman alphabets wrote only in uppercase or capital letters with beautifully proportioned straight lines, curves, and angles until quite late in their history. Some of the relevant properties are: The Roman script has 26 each of upper and lower case characters. While the capital letters of the Roman script occupy the middle and the upper zones, most of the lower case characters have a spatial spread that covers only the middle zone or the middle and the upper zones. The structure of the Roman alphabet contains more vertical and slant strokes. We have divided this paper into seven sections. In section 2, we have presented a brief overview of data collection. Digitization and preprocessing of data has been discussed in section 3. Section 4, presents feature extraction phase and in section 5, we have presented *k*-Means algorithm. Section 6, includes experimental results carried out in this study. Conclusion has been presented in section 7.

2. Data Collection

We have collected 4,850 samples of isolated handwritten characters of Devanagari, Gurumukhi and Roman scripts from 50 different writers. All the writers were requested to write each character of Devanagari, Gurumukhi and Roman. Few samples of this data set are shown in Fig. (1-3).

Fig. 1. Character set of Devanagari script

Fig. 2. Character set of Gurumukhi script

Fig. 3. Character set of Roman script

3. Digitization and Pre-Processing

All these documents were scanned at 300 dpi resolution and stored in gray scale format. The isolated character image was normalized into 100×100 using NNI algorithm and then converted into bitmap.

4. Feature Extraction

Feature extraction is the phase which is used to measure the relevant shape contained in the character. We have presented four feature extraction techniques, namely, circular curvature, pixel density, horizontal stroke density and zoning based features as discussed in following sub-sections:

4.1. Curvature Features

Here (x, y) represents the coordinates of *x*-axis and *y*-axis at the particular pixel and (xc, yc) are coordinates of the center of the zone.

- Firstly every character image is divided into 100 zones each of equal sized.

- Then 100 values for *r* per zone are calculated by using

$$r^2 = (x - xc)^2 + (y - yc)^2$$

- Then average is calculated from these 100 values to get one feature value per zone.

As such, 100 feature elements have been extracted for each image/sample.

4.2. Pixel Density

Pixel density is defined after filling holes if any in the image. It is calculated by finding sum of all the foreground pixels zone wise in the pattern and then dividing it with the size of the pattern and denoted as follows

$$\text{Pixel Density (Pattern)} = \frac{\textbf{Sum of foreground pixels in the pattern}}{\textbf{Size of the pattern}}$$

We have achieved 100 feature elements per sample with pixel density based feature extraction technique.

4.3. Horizontal Stroke Density

Stroke density is calculated after calculating stroke length which is defined as the number of pixels in a stroke. In horizontal stroke density, we have considered, the sum of horizontally foreground that pixels of the pattern were calculated and divided by the size of the pattern. To extract these features the image is divided into 25×25 pixel zone and we have extracted 16 feature elements per sample to recognize the character.

4.4. Zoning

Zoning based feature extraction technique is a well known technique in the area of pattern recognition. We have divided the image into 16 different zones each of equal sized and calculated the number of foreground pixels in each zone.

5. K-Means

K-Means clustering performs cluster analysis using an algorithm that can handle large number of cases, but that requires you to specify the number of clusters that is *k* here in the algorithm. The *k*-mean algorithm takes the input parameter, *k*, and partitions a set of *n* objects into *k* clusters so that the resulting intra-cluster similarity is high but the inter-cluster similarity is low. Cluster similarity is measured with regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid. The main goal of clustering is to identify relatively homogeneous groups of objects based on selected characteristics. It works as follows:

1. Firstly, it randomly selects *k* of objects as the cluster mean or center.
2. Each of the remaining objects is assigned to the clusters based on the distance between the objects.
3. It then computes the new mean for each cluster. This process is repeated until no change is there in clusters.

In this paper, we have used SPSS tool for *k*-Means clustering to make clusters from the given set of data.

6. Experimental Results and Discussion

In this section, we have presented experimental results carried out in this study. For experimentation work, we have used 1,800 samples of Devanagari script, 1,750 samples of Gurumukhi script and 1,300 samples of Roman script. Here, cluster 1, denotes Devanagari script, cluster 2 denotes Roman script and cluster 3 denotes Gurumukhi script.

Feature-wise experimental results of testing are presented in the following sub-sections.

6.1. Pixel Density based Features

Here, 100 feature values were used per image. By using *k*-Means clustering algorithm total 17 iterations were performed to get the clusters. After 17 iterations the maximum absolute coordinate change for any center was found to be .000. The minimum distance between initial centers was found to be .016. After performing 17 iterations and have been 100 feature values by using pixel density feature, the clusters are given below graphically in Fig. 4.

1st cluster contains 1,580 characters
2nd cluster contains 1,546 characters
3rd cluster contains 1,714 characters

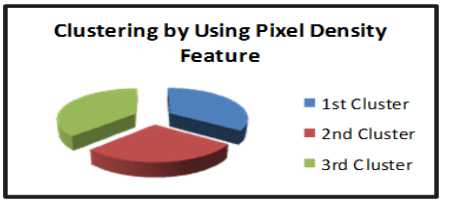

Fig. 4. *K*-Means clustering using pixel density based feature

6.2. Horizontal Stroke Density based Features

Using this technique, 16 feature values per text image were used to make clusters. Total 24 iterations were performed to get the results by using *k*-Mean clustering. The maximum absolute coordinate change for any center was found to be .000. The minimum distance between initial centers was found to be .544. After performing 24 iterations on 16 feature values by using Horizontal stroke density feature the clusters have been depicted in Fig. 5.

1st cluster contains 2,047 characters
2nd cluster contains 1,066 characters
3rd cluster contains 1.737 characters

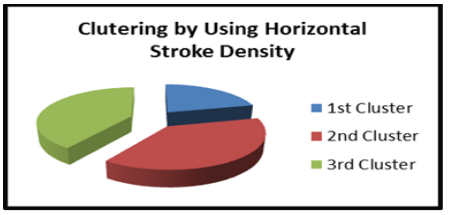

Fig. 5. *K*-Means clustering using horizontally stroke density

6.3. Curvature Features

In this technique 100 feature values per text image were used to make clusters. Total 16 iterations are performed to get the results by using *k*-Mean clustering. The maximum absolute coordinate change for any centre was found to be .000. The minimum distance between initial centers was found to be 131.655. After performing 16 iterations on 100 feature values by using circular curvature feature the clusters have been presented below:

1st cluster contains 1,795 characters
2nd cluster contains 1,505 characters
3nd cluster contains 1,550 characters

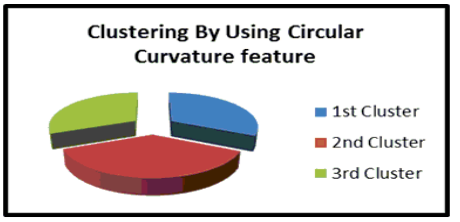

Fig. 6. *K*-Means clustering using circular curvature

6.4. Zoning based Features

In this technique 16 feature values per text image were used to make clusters. Total 15 iterations are performed to get the results by using *k*-Mean clustering. The maximum absolute coordinate change for any center was found to be .001. The minimum distance between initial centers was found to be .523. After performing 15 iterations on 16 feature values by using zoning based the clusters have been shown in Fig. 7.

1st cluster contains 1,837 characters
2nd cluster contains 1,300 characters
3nd cluster contains 1,713 characters

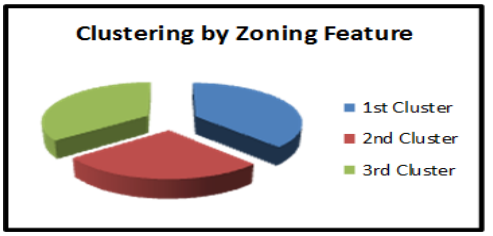

Fig. 7. *K*-Means clustering using zoning based features

7. Conclusions

In this paper, we have presented a technique for clustering of multi-script text. Here, 1,800 samples of Devanagari script, 1,750 samples of Gurumukhi script and 1,300 samples of Roman were used for experimentation work. The clusters include number of samples obtained after using various feature extraction techniques as mentioned below.

Pixel density feature:

Devanagari	Roman	Gurumukhi
1,580	1,546	1,714

Zoning based feature:

Devanagari	Roman	Gurumukhi
1,837	1,300	1,713

Horizontal stroke density based feature:

Devanagari	Roman	Gurumukhi
2,047	1,066	1,737

Curvature feature:

Devanagari	Roman	Gurumukhi
1,795	1,505	1,550

References

[1] P. E. Ajmire and S. E. Warkhede, "Handwritten Marathi character (vowel) recognition", *Advances in Information Mining* (0975–3265), Vol. 2, pp.11-13, 2010.

[2] C. Sureshkumar and T. Ravichandran, "Handwritten Tamil Character Recognition using RCS Algorithm", *International Journal of Computer Applications* (0975 – 8887), Vol. 8(8), pp. 21-25, 2010

[3] B. V. Dhandra and H. Mallikarjun, "Global and Local Features Based Handwritten Text Words and Numerals Script Identification", *International Conference on Computational Intelligence and Multimedia Applications*, Vol. 2, pp. 471-475, 2007.

[4] H. A. Kumar and T. Ravinder, "Comparative Study of Different Classifiers for Devanagari Handwritten Character Recognition", *International Journal of Engineering Science and Technology*, Vol. 2 (7), pp. 2681-2689, 2010.

[5] S. Kumar, G. Shrivastava and S. Sanjay, "Support Vector Machine for Handwritten Devanagari Numeral Recognition", *International Journal of Computer Applications (0975 – 8887)*, Vol. 7 (11), pp. 9-14, 2010.

[6] U. Pal, T. Wakabayashi and F. Kimura, "Handwritten Bangla Compound Character Recognition using Gradient Feature", *International Conference on Information Technology*, pp. 208-213, 2007.

[7] S. V. Rajashekararadhya and V. P. VanajaRanjan, "Efficient Zone Based Feature Extraction Algorithm for Handwritten Numeral of Four Popular South Indian Scripts", *Journal of Theoretical and Applied Information Technology*, pp. 1171-1181, 2008.

[8] G. G. Rajput and H. B. Anita, "Handwritten Script Recognition using DCT and Wavelet Features at Block Level", *IJCA Special issue on Recent Trends in Image Processing and Pattern Recognition(RTIPPR)*, pp. 158-163, 2010.

[9] D. V. Sharma and U. Jain, "Recognition of Isolated Handwritten Characters of Gurumukhi Script using Neocognitron", *International Journal of Computer Applications (0975–8887)*, Vol. 10 (8), pp. 10-16, 2010.

[10] D. V. Sharma and P. Jhajj, "Recognition of Isolated Handwritten Characters in Gurumukhi Script", *International Journal of Computer Applications*, Vol.4 (8), pp. 9-17, 2010.

[11] M. Kumar, R. K. Sharma and M. K. Jindal, "Segmentation of Lines and Words in Handwritten Gurmukhi Script Documents", *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia, Allahabad*, pp. 28-30, 2010.
[12] M. Kumar, M. K. Jindal and R. K. Sharma, "Segmentation of Isolated and Touching Characters in Offline Handwritten Gurmukhi Script Recognition", *International Journal of Information Technology and Computer Science*, Vol. 6(2), pp. 58-63, 2014.

Authors' profiles

Neeru Garg received his Post Graduate degree in Computer Science & Technology from Lovely Professional University, Phagwara, Punjab, India in 2011.

Munish Kumar received his Bachelors degree in Information Technology from Punjab Technical University, Jalandhar, India in 2006 and Post Graduate degree in Computer Science & Engineering from Thapar University, Patiala, India in 2008. He received his Ph.D. degree in Computer Science from Thapar University, Patiala, Punjab, India. He started his carrier as Assistant Professor in computer application at Jaito centre of Punjabi university, Patiala. He is working as Assistant Professor in Panjab University Rural Centre, Kauni, Muktsar, Punjab, INDIA. His research interests include Character Recognition.

How to cite this paper: Neeru Garg, Munish Kumar,"Clustering of Multi Scripts Isolated Characters Using k-Means Algorithm", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.1, No.2, pp.22-29, 2015.DOI: 10.5815/ijmsc.2015.02.03