IJCSN International Journal of Computer Science and Network, Volume 3, Issue 5, October 2014
ISSN (Online) : 2277-5420 **www.IJCSN.org**
Impact Factor: 0.274

Image Normalization and Preprocessing for Gujarati Character Recognition

Jayashree Rajesh Prasad

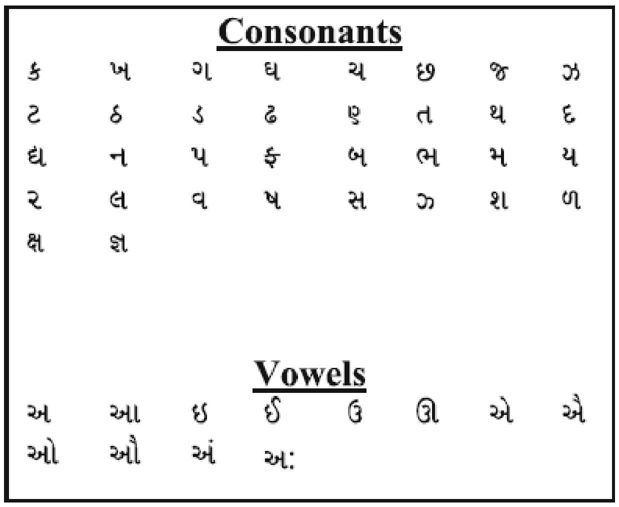
Department of Computer Engineering, Sinhgad College of Engineering, University of Pune,
Pune, Mahaashtra 411048, India

Abstract - Pattern recognition has been an important area in computer vision applications. In the case of a planar image, there are four basic forms of geometric distortion caused by the change in camera location: translation, rotation, scaling and skew. So far, a number of methods have been developed to solve these distortions, such as moment invariants', Fourier descriptor, Hough transformation, shape matrix' and the principle axis method. All of the above methods can be made invariant to translation, rotation and scaling. However, they become useless when pattern is skewed: when the direction of the camera or scanner is not vertical to the planar image or the sampling intervals in the x and y directions are not equal, the image is skewed. Authors present a moment based normalization process for character images for the purpose of enhancing the performance of character recognize for isolated characters of Gujarati. This paper focuses on character pre-processing and normalization stage of handwritten character recognition for Gujarati

Keywords - *Affine transformation, distortion, moments, rotation, scaling, translation.*

1. Introduction

Gujarati is also the name of the script used to write the Gujarati language spoken by about 50 million people in western India. Gujarati has 12 vowels and 34 consonants. Apart from the basic symbols, other symbols called vowel modifiers are used to denote the attachment of vowels with the core consonants. Gujarati belongs to the genre of languages that use variants of the Devanagri script. No significant work is found in the literature that addresses the recognition of Gujarati language [2]. The Gujarati script is derived from the Devanagri script. Other languages like Sanskrit, Hindi, and Marathi use a similar script. Some of the Gujarati characters are very similar in appearance. With sufficient noise these characters can easily be misclassified. Often, these characters are misclassified even by humans who then need to use context knowledge to correct the error. Unlike Devanagri, the Gujarati characters within a word are separated by white space. The intra-word characters in the Devanagri are connected with an over stroke. This eases the problem of character separation in a word. This section addresses the problem of discriminating between subset of such characters. Fig. 1 shows the set of printed characters in Gujarati.



Consonants

ક	ખ	ગ	ઘ	ચ	છ	જ	ઝ
ટ	ઠ	ડ	ઢ	ણ	ત	થ	દ
ધ	ન	પ	ફ	બ	ભ	મ	ય
ર	લ	વ	ષ	સ	હ	શ	ળ
ક્ષ	જ્ઞ						

Vowels

અ	આ	ઇ	ઈ	ઉ	ઊ	એ	ઐ
ઓ	ઔ	અં	અ:				

Figure 1: Gujarati character set with consonants and vowels

Author finds some issues which characterize the Gujarati script. Although authors discuss the issues with reference to Gujarati, they are largely applicable to most Indian languages. As mentioned before, the characters within a word in Gujarati are separated by white-space. Each of the consonants in Gujarati can have one of 11 strokes adjacent to, over, or under it. These represent the vowels. Ligatures also are very common and can themselves have the vowel strokes around them. An OCR system would thus need to handle a large variety of fonts along with these characteristics. The association of the vowel stroke with the character is in itself an interesting problem for recognition.

2. Data Set Description

The availability of data set that captures variations encountered in real world is a critical part of any experimental research. Due to significant advances in the OCR research in recent years, several data sets are available for English. To the best of our knowledge, no handwritten Gujarati data sets exist. Therefore 360 samples from different writers are collected for each character in Gujarati alphabet i.e. 34 consonants and 12 vowels. Thus this data set consists of 16560 samples altogether. The characters are scanned at 300 dots per inch resolution. Our experiments are executed on unconstrained handwritten characters. These characters have skew in them and also may have noise pixels. Some characters are also observed to be broken at locations which have fine links.

2.1 Zone Boundary Structure of Gujarati Script

The shapes of many Gujarati characters are similar to those of the phonetically corresponding characters of Devanagari script. As in the case of other Indic scripts, Gujarati also does not have the distinction of lower and upper cases. In spite of its similarity with the Devanagari script, Gujarati script has many distinct characteristics such as the absence of the so called shirorekha i. e. header line in the script and differences in the shapes of many of the consonants and vowels etc. Consonant-vowel combinations occur very often in most of the Indic languages including Gujarati. This is denoted by attaching a symbol, unique for each vowel, to the consonant, called a dependent vowel modifier or matra. The matra can appear before, after, above or below the core consonant. Similar to the text written in Devanagari or Bangla scripts, text in Gujarati script can also be divided into three logical zones –upper, middle and lower [3].

2.2 Preprocessing

As the first step, image and data preprocessing serve the purpose of extracting regions of interest, enhancing and cleaning up the images, so that they can be directly and efficiently processed by the feature extraction stage. Digital scanners are default image acquisition devices; they are fast, versatile, mobile, and are relatively cheap. In OCR applications, however, digital scanners suffer from a number of limitations e.g. geometrical distortions. Due to absence of standard image acquisition procedures for OCR data sets, efficient preprocessing is required. Initially, the scanned images undergo normalization operation.

3. Normalization

Character normalization is considered to be the most important pre-processing operation for character recognition. General approach to image normalization includes mapping an image onto a standard plane of a predefined size, so as to give a representation of fixed dimensionality for classification. The goal for character normalization is to reduce the within class variation of the shapes of the characters in order to facilitate feature extraction process and also improve their classification accuracy. Basically there are different approaches for character normalization: linear normalization, moment-based normalization and nonlinear normalization [4]. In order to increase both the accuracy and the interpretability of the digital data during the image processing phase, authors use normalization in pre-processing stage. For this moment-based normalization, a well-known technique in computer vision and pattern recognition applications as proposed by [5] is used. The key idea of using moment-based normalization is to obtain a normalized image from a geometric transformation procedure that is invariant to any affine distortion of the image. This enhances the recognition rate of character even when the character samples from different writers exhibit affine geometric variations [4].

Here, the phrase, ‘affine transformation’ refers to a transformation that is a combination of single transformations such as translation or rotation or reflection on an axis. Initially, original image is resized and converted to grey scale format that subsequently undergo normalization operation. Features are then extracted from normalized image having a standard size and orientation. The normalized image is obtained from a geometric transformation procedure that is invariant to any affine distortions of the image. Normalized image is converted back to its original size and orientation during recognition phase. This normalization process is described in details further.

3.1 Image Moments and Affine Transforms

Let $f(x,y)$ denote a digital image of size $M \times N$. Its geometric moments M_{pq} and central moments μ_{pq}, with $p,q = 0,1,2 \ldots$ are defined respectively as,

$$M_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x,y)$$

and

$$\mu_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(\bar{x}-x)^p(\bar{y}-y)^q f(x,y)$$

where

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}.$$

An image $g(x,y)$ is said to be an affine transform of $f(x,y)$ if there is a matrix $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ and vector $d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$ such that $g(x,y) = f(x_a, y_a)$ where,

$$\begin{pmatrix} x_a \\ y_a \end{pmatrix} = A.\begin{pmatrix} x \\ y \end{pmatrix} - d.$$

Affine transformation includes shearing in x direction denoted as $A_x = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}$; shearing in the y direction which is denoted by $A_y = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}$; and scaling in both x and y directions which corresponds to $A_s = \begin{pmatrix} \alpha & 0 \\ 0 & \delta \end{pmatrix}$.

Moreover, it is straightforward to show that any affine transform A can be decomposed as a composition of the aforementioned three transforms e.g., $A = A_s \cdot A_y \cdot A_x$, provided that $a_{11} \neq 0$ and det $(A) \neq 0$.

3.2 Applying image normalization

The normalization procedure consists of the following steps for a given image $f(x,y)$.

i. Center the image $f(x,y)$; this is achieved by setting in (2.4) the matrix $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and the vector $d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$ with

$$d_1 = \frac{m_{10}}{m_{00}}, d_2 = \frac{m_{01}}{m_{00}},$$

where m_{10}, m_{01}, and m_{00} are the moments of $f(x,y)$ as used in (2.3). This step is aimed to achieve translation invariance. Let $f_1(x,y)$ denote the resulting centered image as shown in Fig. 2 – 6 (a).

ii. Apply a shearing transform to $f_1(x,y)$ in the x (2.2) direction with matrix $A_x = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}$ so that the resulting image as shown in Fig. 2.2- 2.6 (b), denoted by $f_2(x,y) = [A_x[f]_1(x,y)]$. is achieved.

(2.3)

iii. Apply a shearing transform to $f_1(x,y)$ in the y direction with matrix $A_y = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}$ so that the resulting image as shown in Fig. 2.2- 2.6 (c), denoted by $f_3(x,y) = [A_y[f]_2(x,y)]$. is achieved.

iv. Scale $f_3(x,y)$ in both x and y directions (2.4) with $A_s = \begin{pmatrix} \alpha & 0 \\ 0 & \delta \end{pmatrix}$ so that the resulting image as shown in Fig. 2.2- 2.6 (d), denoted by $f_4(x,y) = [A_s[f]_3(x,y)]$. is achieved.

Experiments show that, values α, β, γ and δ are advisable to be 10 % of image dimensions. If these values are exceeded to 50 % of image dimensions, it results into deformation.

The above normalization procedure can be also explained as follows:

Since we are using handwritten samples of 360 writers, there is a lot of variety in writing styles, stroke directions and character shapes violation by these writers. Technically, lets treat these style variations as a general affine transformation viewed as a composition of translation, shearing in both x and y directions, and scaling in both x and y directions. The four steps in the normalization procedure are designed to eliminate effects of each of these distortion or variation components.

Step (i) specified above, eliminates the translation of the character image by adjusting the center of the image; steps (ii) and (iii) eliminate shearing in the x and y directions; step (iv) eliminates scaling distortion by forcing the normalized image to a standard size. Fig. 2-6, show results of subsequent operations on few characters such as k, Ka, ca, j and T during image normalization. The final image (e) shown all these figures is the normalized image, based on which subsequent feature extraction is performed. It is important to note that each step in the normalization procedure is readily invertible.

This helps to convert the normalized image back to its original size and orientation.

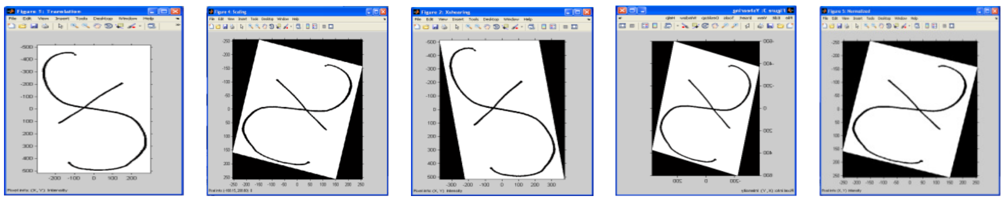


Fig. 2 (a) Image of k after translation; (b) Image after X shearing; (c) Image after Y shearing; (d) Image after scaling; (e) Normalized image.

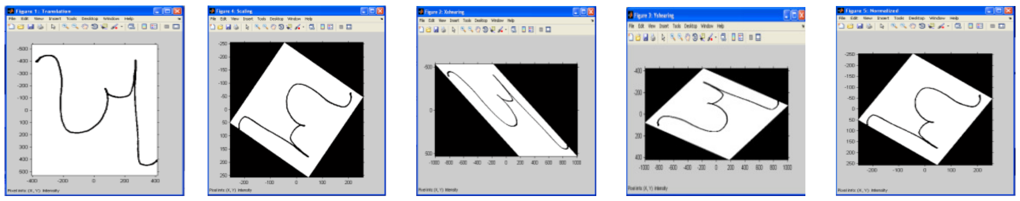


Fig. 3 (a) Image of Ka after translation; (b) Image after X shearing; (c) Image after Y shearing; (d) Image after scaling; (e) Normalized image.

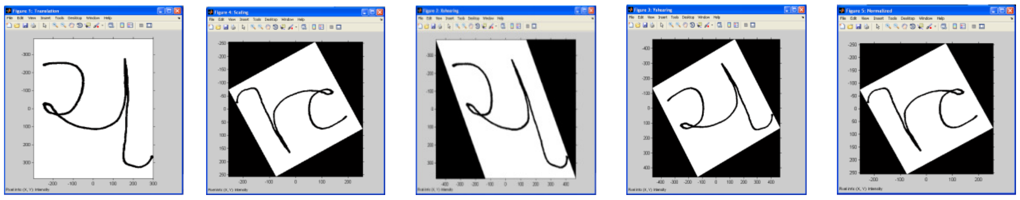


Fig..4 (a) Image of ca after translation; (b) Image after X shearing; (c) Image after Y shearing; (d) Image after scaling; (e) Normalized image.

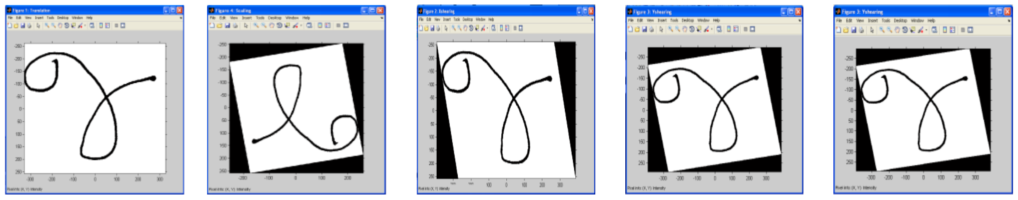


Fig. 5 (a) Image of j after translation; (b) Image after X shearing; (c) Image after Y shearing; (d) Image after scaling; (e) Normalized image.

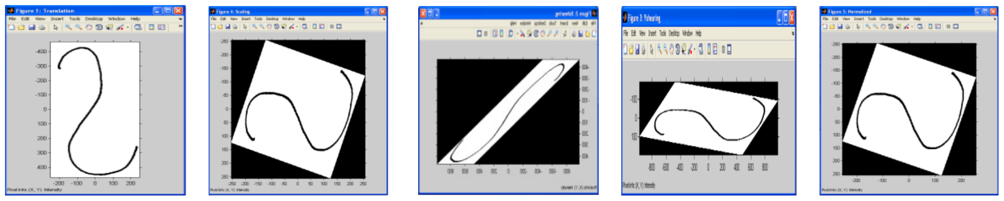


Fig. 6 (a) Image of T after translation; (b) Image after X shearing; (c) Image after Y shearing; (d) Image after scaling; (e) Normalized image.

4. Skeletonization

The aim of the skeletonization is to extract a region-based shape feature representing the general form of an object. It is a common pre-processing operation in pattern recognition. The major skeletonization techniques are:

i. detecting ridges in distance map of the boundary points,
ii. calculating the Voronoi diagram generated by the boundary points, and
iii. the layer by layer erosion called thinning.

In digital spaces, only an approximation to the "true skeleton" can be extracted. There are two requirements to be complied with:

i. topological i.e. to retain the topology of the original object,
ii. geometrical i.e. forcing the "skeleton" being in the middle of the object and invariance under the most important geometrical transformation including translation, rotation, and scaling

Skeletonization removes pixels on the boundaries of objects but does not allow objects to break apart. The pixels remaining make up the image skeleton. This operation preserves the Euler number. Authors use a combined skeletonization and thinning approach using a image processing toolbox in Matlab 2011a.

5. Thinning

This morphological operation is used to remove selected foreground pixels from binary images, somewhat like erosion or opening. It is particularly used along with skeletonization. These two operations are performed for the purpose of extraction of pattern descriptor feature that is not in the scope of this paper. Author uses the following algorithm [6] for thinning:

1. Divide the image into two distinct subfields in a checkerboard pattern.
2. In the first sub-iteration, delete pixel p from the first subfield if and only if the conditions G_1, G_2 and G_3 are all satisfied.
3. In the second sub-iteration, delete pixel p from the second subfield if and only if the conditions G_1, G_2 and G_3' are all satisfied.

Condition G_1 **:** $X_H(p) = 1,$

Where $X_H(p)$ represents of subfield of image in a checkerboard pattern of size (3×3) and p denotes the center pixel.

$$X_H(p) = \sum_{i=1}^{4} b_i$$

$$b_i = \begin{cases} 1, & if\ x_{2i-1} = 0\ and\ (x_{2i} = 1\ or\ x_{2i+1} = 1) \\ 0, & otherwise \end{cases}$$

$x_1, x_2, \dots, x_8$ are the values of the eight neighbors of p, starting with the east neighbor and numbered in counter-clockwise order.

Condition G_2 **:** $2 \le \min[\{n_1](p), n_2(p)\} \le 3$

where $N(p)$ represent 4 connected components,

$$n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k}$$

$$n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1}$$

Condition G_3 **:** $(x_2 \vee x_3 \vee \overline{x}_8) \wedge x_1$

Condition G_3' **:** $(x_6 \vee x_7 \vee \overline{x}_4) \wedge x_5 = 0$

The two sub-iterations together make up one iteration of the thinning algorithm. When the user specifies an infinite number of iterations, the iterations are repeated until the image stops changing. The conditions are all tested using `applylut` function with pre-computed lookup tables in Matlab.

6. Conclusion

Author describes a normalization process that achieves invariance under affine geometric distortions. In this application authors apply a normalization procedure to the image so that it meets a set of predefined moment criteria. A recognition rate of 86.6 % [1] for the isolated characters of Gujarati is achieved. This paper does not focus on the classification details of this application. Author has focused on the moment based image normalization applied in the pre-prepossessing stage of a character recognition task.

References

[1] Jayashree Prasad, Uday Kulkarni, 'Gujarati Character Recognition using weighted k-NN with mean chi square Distance Measure', DOI 10.1007/s13042-013-0187-z International Journal of Machine Learning and Cybernetics, Springer, ISSN 1868-8071.

[2] M. Maloo and Kale, "Gujarati Script Recognition: A Review," In: International Journal on Computer Science and Engineering (IJCSE), 2011. [3] A. Name, "Dissertation Title", M.S.(or Ph.D.) thesis, Department, University, City, Country, Year.

[3] S. Antani and Lalitha Agnihotri, "Gujarati Character Recognition," In: ICDAR, pp. 418-421, 1999.

[4] Mohamed Cheriet, Nawwaf Harma, Cheng-Lin Liu, Ching Y. Sen, "Character Recognition Systems A Guide for Students and Practitioners," A John Wiley & Sons, Inc., Publication, 2007.

[5] Marius Bulacu, Lambert Schomaker and Axel Brink, "Text-Independent Writer Identification and Verification on Offline Arabic Handwriting," In: ICDAR, IEEE Computer Society, Volume. II, pp. 769-773, 2007.

[6] Lam, L., Seong-Whan Lee, and Ching Y. Suen, Thinning Methodologies- A Comprehensive Survey, In: IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 14, No. 9, page 879, September 1992.

First Author Jayashree Rajesh Prasad graduated in Computer Science and Engineering from North Maharashtra University in 1996 and completed M.E. in Computer Engineering from Pune University in 2004. She pursued Ph.D. in Computer Science and engineering from Swami Ramananda Tirtha University, Nanded in 2014. She has a research project "Conversion of Gujarati Script to Speech", funded by BCUD (University of Pune) to her credit. She works with Sinhgad College of Engineering, Pune. Her research interests are in the field of Soft Computing, pattern recognition and image processing. She is Life member of Computer Society of India, Life Member of Indian Society for Technical Education, Member of IAENG (International Association of Engineers) and Member of IACSIT (International Association of Computer Science and Information Technology).