REAL-TIME CHARACTER READING SYSTEM FOR MARATHI SCRIPT USING RASPBERRY PI

Sushama Shelke*, Shaila Apte †

**Collegeof Engineering, Pune, India, shelke.sushma@gmail.com, †Rajarshi Shahu College of Engineering, Pune, India,*

Keywords: Raspberry Pi, Python, Marathi character recognition, template matching.

Abstract

Development of real-time recognition systems for character recognition for Indian scripts is a challenging task. This paper presents a novel real time character reading system for Marathi script. The system is developed using Raspberry Pi and OpenCV Python. The reading system is composed of two modules, namely, image acquisition module and character recognition module. The image acquisition module consists of the camera and the character recognition module does the pre-processing, recognizes the characters and converts them into speech. In this paper, Marathi printed characters without any modifiers are acquired using camera and applied to the character recognition module. The recognition rate achieved is 92% using template matching.

1 Introduction

Real-time systems developed using embedded systems are gaining popularity in various fields due to their portability, low cost and lower power requirements. The character recognition task finds its application in various fields like Postal Automation for address recognition, in Banking for cheque recognition, Offices for Data entry and form filling etc. The post-processing of character recognition systems can also include speech conversion as a hearing aid to physically handicapped and elderly people or also can be used as a teaching and learning aid.

Marathi language is derived from Devanagari script. It has a number of characters in the set with 36 consonants and 16 vowels [1]. The script has rich structural features which makes this pattern recognition task complex. The characters have a horizontal line below which the characters are written. This line also joins the words in a sentence. The vowels take different shapes and are connected to the consonants in different styles, which are termed as modifiers. The modifiers stretch above the header line, in line with the character or are joined at the bottom of the character. The character set also includes conjunct characters that are formed by joining two or more characters. Based upon the writing style of the characters, the character recognition systems can be classified into two types, namely, printed character recognition systems and handwritten character recognition systems. While, based upon the nature of character acquisition process, the systems are classified into two types, namely offline recognition systems and online recognition systems. Lot of research is carried out in offline character recognition for various Indian scripts including printed and handwritten characters, where the character images are stored in the memory prior to applying to the recognition engine.

Initially, the research in character recognition started with printed Devanagari characters in 1970s. Lot of research was carried out for printed Devanagari character recognition too [2-4]. This was followed by the development of systems for hand-written numeral recognition for Devanagari [5]. The success in handwritten numeral recognition encouraged the researchers to develop systems for off-line handwritten character recognition [6].

Further advances led to the use of soft computing techniques like Artificial Neural Network and Fuzzy logic for Devanagari character recognition [7-12] in order to improve the recognition accuracy, especially in handwritten character recognition systems. Researchers also applied multiple features [13], multiple classifiers and classifiers with multiple stages [14-15] to improve the recognition performance of handwritten Devanagari characters. The offline systems are generally implemented using general purpose processors or personal computers with programming languages like C or MATLAB.

Research in online character recognition system is generally based on deployment of recognition algorithm using hardware like FPGA [16-17] or a DSP processor [18] to improve the recognition speed. Both such systems are expensive due to the hardware involved.

The real-time numeral or character recognition systems are developed by acquiring the images using camera, graphic tablet [19], finger movements [20] etc. In this paper, a real-time system for Marathi script is proposed, that reads the characters using the camera. The system is developed using Raspberry Pi and OpenCV with Python programming language. The Raspberry Pi preprocesses the character image, recognizes the characters using template matching and then finally converts the characters into speech. The system developed using Raspberry Pi and OpenCV is portable and cheap, moreover it recognizes the characters and converts into speech in real time. The characters used in this system are isolated without any modifiers. The font size of the characters can be varied, while the font used is Kiran.

The next part of the paper is organized as follows: Section II describes the system design. Section III presents the image acquisition module. In Section IV, the character recognition module is discussed. Section V puts forth the results obtained and Section VI finally presents the conclusion.

2 System design

This system is designed to read Marathi characters. Figure 1 shows the block diagram of the proposed system. The system is designed using Raspberry Pi. It consists of Image Acquisition Module, Character Recognition Module followed by Post-processing of the recognized characters. At first, the image of the page containing the characters is acquired. The test character is recognized and then converted to speech and applied to the earphones or the speaker. The next section explains the Image Acquisition Module in detail.

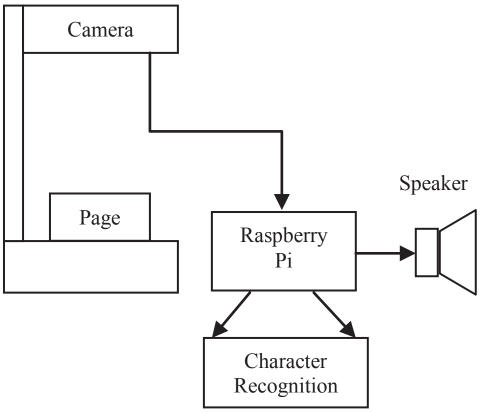


Figure 1: Proposed system design.

3 Image acquisition module

The image acquisition module consists Raspberry Pi and the camera interfaced to it. The details of these components are given further.

3.1 Raspberry Pi

Raspberry Pi is also termed as a mini computer. It is a credit card size computer. It has 900 MHz quad-core ARM processor with integrated OpenCV (Open Source Computer Vision) library on Raspberry Pi. The OpenCV library supports real time image processing. The programming language used in this system to program the Raspberry Pi is Python. It is a high level programming language used to write programs for Raspberry Pi. It has 1 GB RAM, quad core ARM cortex A7 CPU, 4 USB ports, Ethernet, HDMI and Audio-Video port. The camera is interfaced to Raspberry Pi using Camera Serial Interface (CSI). The cost of Raspberry Pi is less and is affordable for standalone system development.

3.2 Image Acquisition

The Pi camera module is connected to the Raspberry Pi via camera serial interface. Pi camera captures the image of the page with characters. It is 5 megapixel fixed focus camera and can be controlled with the help of the software. The camera has a 15 pin cable which can be easily inserted into the CSI port.

This system is designed to recognize 36 printed characters as indicated in Figure 2. Each page consists of about 5-6 lines with 15-20 characters on each line. The pages with printed characters are created using Kiran font, which is used to type Marathi characters. The font size can be varied. The system is tested for the font sizes 12, 24 and 48.

क ख ग घ
च छ ज झ
ट ठ ड ढ
ण त थ द
ध न प फ
ब भ म य
र ल व श
ष स ह ळ
अ इ उ ए

Figure 2: Characters used in the proposed system.

The next section discusses the character recognition module and the steps involved in it in detail.

4 Character recognition module

This module includes pre-processing techniques [21] such as skew detection and correction, RGB to grav scale conversion. It also includes image segmentation, template matching and conversion to speech. These techniques are further explained in detail.

4.1 Skew detection and correction

The image captured using camera is prone to skew. The image skew is corrected using Affine transformation. The Affine transformation is two-dimensional that transforms input co-ordinates (x_1, y_1) into output co-ordinates (x_2, y_2) by applying various linear operations like translation, rotation, scaling etc. The rotation of the output image is done with respect to image center. The transformation for image rotation is given by,

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}. \qquad (1)$$

4.2 Color to gray image conversion

The image captured using the camera is a color image in RGB space. It is converted to gray scale for further processing. The gray scale image is stored using 8 bits which has gray scales from 0 to 255.

4.3 Image segmentation

The lines and characters in the image are segmented with the help of horizontal and vertical projection profiles. The horizontal and vertical projection profiles are

$$P(x) = \sum_{x} f(x, y) \tag{2}$$

and

$$P(y) = \sum_{y} f(x, y) \tag{3}$$

respectively.

The projection profile can also be used to separate the words in the characters. In this system, we are considering isolated characters without modifiers.

4.4 Template matching

The recognition of the characters is done using template matching. This is the simplest method of recognition, where every character template is matched with the characters in the image. Here, template or filter mask *h(x, y)* is generated for each character. The template is then matched with the image *f(x, y)*, which is of larger size than the template. This image consists of number of characters to be recognized. The matching is done using finding the correlation coefficient between the character template and the image. The correlation result contains peaks at the location of the matches between the template and the underlying character image. This matching is indicated by R (x,y) which is the value between -1 and 1, where higher the value, higher the match.

$$\mathrm{R(x,y)} = \frac{\sum_{x}\sum_{y}(f-\bar{f})(h-\bar{h})}{\sqrt{\left(\sum_{x}\sum_{y}(f-\bar{f})^2\right)\left(\sum_{x}\sum_{y}(h-\bar{h})^2\right)}} \tag{4}$$

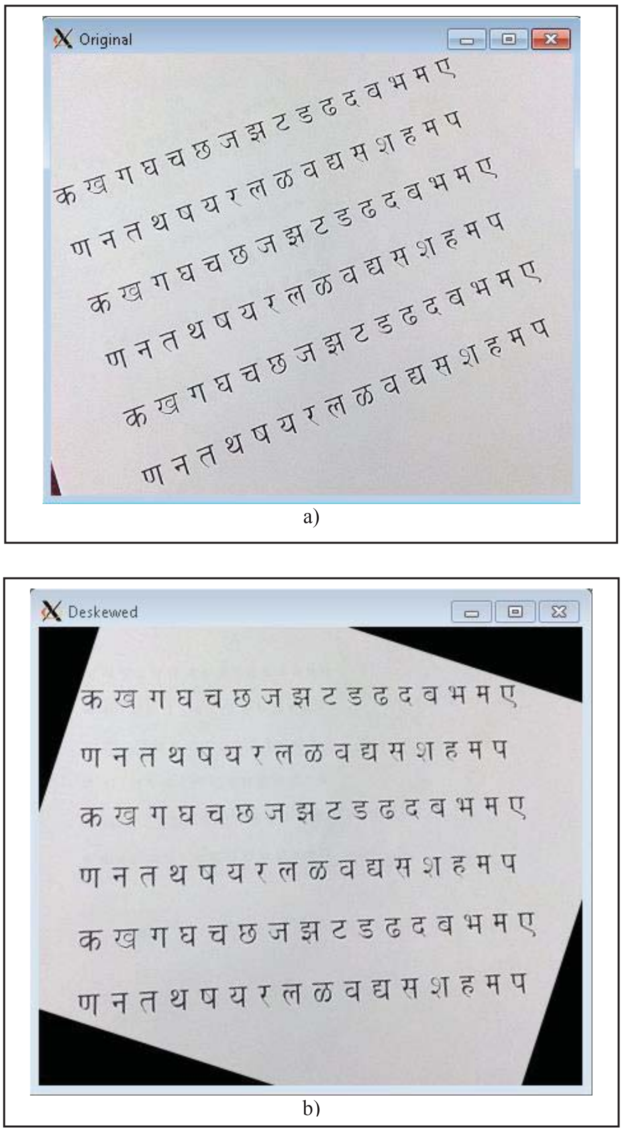
Here, $\bar{h}$ and $\bar{f}$ are the mean of the template and the original image respectively.

4.5 Speech conversion

The speech signal for each character is recorded in .wav format and stored in the memory. The .wav file for the recognized character is played for converting the text into speech using earphones or a speaker. The earphones are connected to 3.5 mm audio/video jack of the Raspberry Pi.

5 Results and discussion

In this system, a Marathi character reading system is developed with the help of Raspberry Pi. The font used for Marathi characters is Kiran font. The characters in the images are in various font sizes. The samples used are in 12, 24 and 48 sizes. The character images are first acquired using the camera. The images are then pre-processed and applied to template matching algorithm. The matched characters are further played using earphones or speaker. The pages are scanned using Pi camera. The file format used for scanning the images is png. The images might contain the skew which is corrected using Affine transformation. The dimensions of the image are obtained and the image is rotated through the center to obtain the deskewed image. The original image with skew and deskewed image is shown in Figures 3a) and 3b) respectively. Here the skew is approximately 18 degree in anticlock-wise direction.

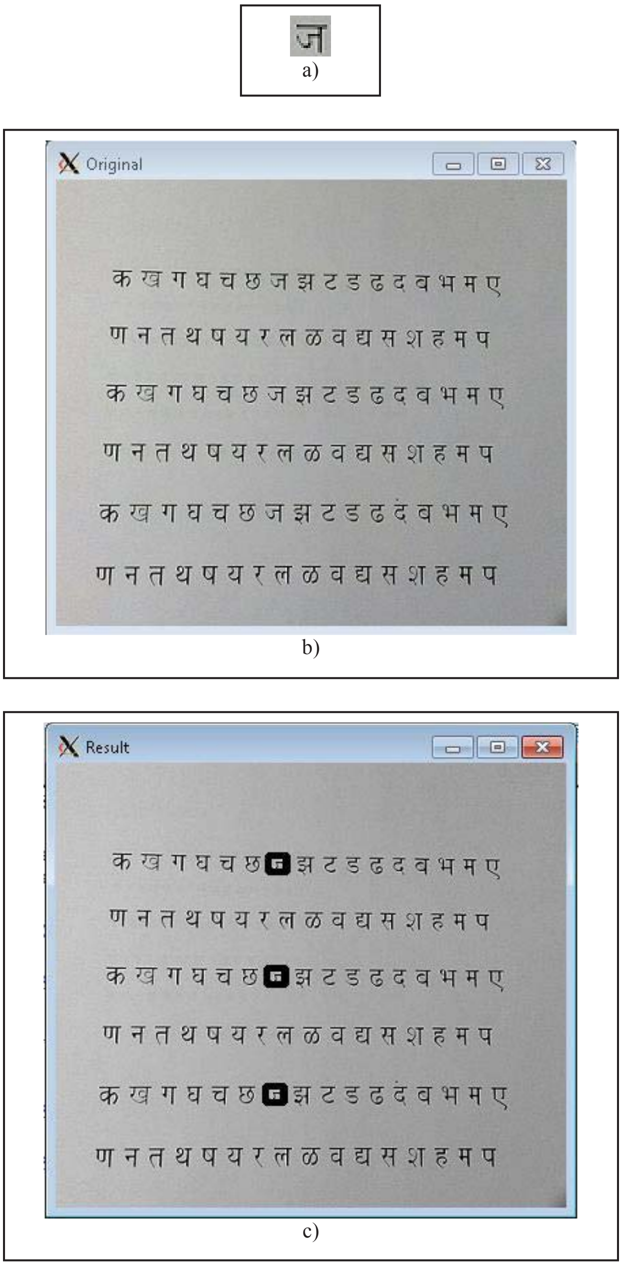

Original
क ख ग घ च छ ज झ ट ड ढ द ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प
क ख ग घ च छ ज झ ट ड ढ द ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प
क ख ग घ च छ ज झ ट ड ढ द ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प
Deskewed
क ख ग घ च छ ज झ ट ड ढ द ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प
क ख ग घ च छ ज झ ट ड ढ द ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प
क ख ग घ च छ ज झ ट ड ढ द ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प

a)

b)

Figure 3: Skew Correction a) Original image with skew, b) Deskewed image.

The skew corrected image is further converted into gray scale. The gray scale image is matched with the template using correlation to obtain the result that ranges between -1 to +1. The template matching results in positive output +1 if there is exact matching between the image and the template. While in case of exact mismatch between the test image and the template, the result of template matching is -1. The sample character template, the original image and the template matching result is shown in Figures 4a), 4b) and 4c) respectively.



a)

क ख ग घ च छ ज झ ट ड ढ द ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प
क ख ग घ च छ ज झ ट ड ढ द ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प
क ख ग घ च छ ज झ ट ड ढ दं ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प

b)

क ख ग घ च छ ■ झ ट ड ढ द ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प
क ख ग घ च छ ■ झ ट ड ढ द ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प
क ख ग घ च छ ■ झ ट ड ढ दं ब भ म ए
ण न त थ ष य र ल ळ व द्य स श ह म प

c)

Figure 4: Template matching result: a) Template, b) Original image and c) Matched result.

The template matching fails in case if the original image is of smaller size than the template. The template matching fails in case of translation and rotation. Hence, deskewing and resizing the characters to a fixed size is essential. In case of deskewing, sometimes, the shape of the character changes which results in misclassification of the character. The recognition rate obtained by template matching is 92%. The system is also developed for isolated handwritten Marathi characters as shown in Figure 5. In this figure, the highlighted image template was applied which resulted in correct classification. The recognition rate for handwritten characters is 87% for single writer with a font size that is approximately of same size. The future work is dedicated towards development of algorithm which recognizes the character with modifiers and developing a hardware module for reading a book. Also development of other feature extraction and classification techniques that are rotation and translation invariant need to be developed in order to improve the recognition rate.

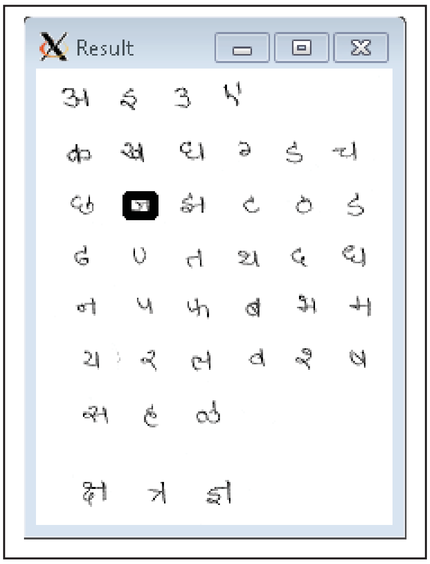


Figure 5: Handwritten character set.

5 Conclusion

This module includes pre-processing Development of real-time portable and low cost embedded systems using open source is the need of the current era. This paper presents an embedded system developed for Marathi character recognition using Raspberry Pi and Python programming language. The system captures the images in real-time and recognizes the characters using template matching. The proposed system also corrects the skew occurring in the image. Finally, the recognized character is read out by converting the text into speech output.

6 References

[1] U. Pal, B. B. Chaudhari. "Indian script character recogntion: a survey", *Pattern Recognition*, **volume** 37, pp. 1887-1899, (2004).

[2] V. Bansal, R.M.K. Sinha. "Partitioning and searching dictionary for correction of optically read Devanagari

character strings", 5^{th} *International Conference on Document Analysis and Recognition*, pp. 653-656, (1999).

[3] V. Bansal, R. M. K. Sinha. "On how to describe shapes of Devanagari characters and use them for recognition", 5^{th} *International Conference on Document Analysis and Recognition*, pp. 410-413, (1999).

[4] V. Bansal. "Integrating knowledge sources in Devanagari text recognition," *Ph.D. Thesis*, IIT Kharagpur, (1999).

[5] R. Bajaj, L. Dey, S. Chaudhary. "Devanagari numeral recognition by combining decision of multiple connectionist classifiers", *Sadhana*, **volume** 27, pp. 59-72, (2002).

[6] U. Pal, N. Sharma, T. Wakabayashi, F. Kimura. "Off-line handwritten character recognition of Devanagari script", 9^{th} *International Conference on Document Analysis and Recognition*, pp. 496-500, (2007).

[7] M. Hanmandlu, A. V. Nath, A. C. Mishra, V. K. Madasu. "Fuzzy model based recognition of handwritten hindi numerals using bacterial foraging", 6^{th} *International Conference on Computer and Information Science*, pp. 309–314, (2007).

[8] M. Hanmandlu, J. Grover, V. K. Madasu, S. Vasikarla. "Input fuzzy modeling for the recognition of handwritten Hindi numerals", *International Conference on Information Technology,* pp.208–213, (2007).

[9] M. Hanmandlu, O. V. R. Murthy. "Fuzzy model based recognition of handwritten numerals", *Pattern Recognition*, **volume** 40, pp. 1840–1854, (2007).

[10] M. Hanmandlu, K. R. M. Mohan, S. Chakraborty, S. Goyal, D. R. Choudhury. "Unconstrained handwritten character recognition based on fuzzy logic", *Pattern Recognition*, **volume** 36, pp. 603–623, (2003).

[11] S. Shelke, S. Apte. "*Handwritten Marathi Character Recognition using Neural Networks - Single-stage, Multi-stage and Hybrid Classification Techniques*", Lambert Academic Publishing, Germany, 2012.

[12] S. Shelke, S. Apte. "A Fuzzy based Classification Scheme for Unconstrained Handwritten Devanagari Character Recognition", *International Conference on Communication, Information and Computing Technology*, pp. 1-5, (2015).

[13] S. Arora, D. Bhattacharjee, M. Nasipuri, D. Basu, M. Kunda. "Combining multiple feature extraction techniques for handwritten Devanagari character recognition", *IEEE Region Colloquium and the Third ICIIS*, pp. 1-6, (2008).

[14] S. Arora, D. Bhatcharjee, M. Nasipuri, L. Malik. "A two stage classification approach for handwritten Devanagari characters", *International Conference on Computational Intelligence and Multimedia Application,* **volume** 2, pp. 399–403, (2007).

[15] S. Shelke, S. Apte. "A novel multi-feature multi-classifier scheme for unconstrained handwritten Devanagari character recognition," 12^{th} *International Conference on Frontiers in Handwriting Recognition*, pp. 215-219, (2010).

[16] K. L. Rice, M. A. Bhuiyan, T. M. Taha, C. N. Vutsinas, M. C. Smith. "FPGA Implementation of Izhikevich Spiking Neural Networks for Character Recognition", *International Conference on Reconfigurable Computing and FPGAs*, pp. 451 – 456, (2009).

[17] N. Toosizadeh, M. Eshghi. "Design and implementation of a new Persian digits OCR algorithm on FPGA chips", 13^{th} *European Signal Processing Conference*, pp. 1 – 4, (2005).

[18] W. Ahmad, S. A. Hassan, M. A. Irfan, A. Bais, G. M. Hassan, S. A. Ali Shah, M. Khawaja. "Design and implementation of real time LPR system on a fixed point DSP", 6^{th} *International Conference on Emerging Technologies*, pp. 159-163, (2010).

[19] M. M. Farhad. "Real time numeric character recognition system based on finger movements," *International Conference on Electrical Information and Communication Technology*, pp. 1–5, (2013).

[20] A. Beltran, S. Mendoza. "Efficient algorithm for real-time handwritten character recognition in mobile devices", 8^{th} *International Conference on Electrical Engineering Computing Science and Automatic Control*, pp. 1–6, (2011).

[21] R. C. Gonzalez, R. E. Woods S.L Eddins. *Digital Image Processing using MATLAB*, 2^{nd} ed., McGraw Hill: New Delhi, (2010).