Natl. Acad. Sci. Lett. (July–August 2014) 37(4):381–391
DOI 10.1007/s40009-014-0253-4

SHORT COMMUNICATION

Efficient Feature Extraction Techniques for Offline Handwritten Gurmukhi Character Recognition

Munish Kumar · R. K. Sharma · M. K. Jindal

Received: 15 April 2013 / Revised: 6 November 2013 / Accepted: 11 December 2013 / Published online: 18 July 2014

Abstract As a result of advances in optical character recognition research, several techniques for handwritten character recognition have surfaced. These techniques require good quality features as their input for the recognition process. In this paper, we have proposed two different feature extraction techniques, namely, parabola curve fitting based features and power curve fitting based features for offline handwritten *Gurmukhi* character recognition. In order to assess the quality of features in offline handwritten *Gurmukhi* character recognition, we have also analyzed the performance of other recently proposed feature extraction techniques, namely, zoning, diagonal, directional, transition, intersection and open end points, gradient and chain code features. Each technique has been tested by using 3,500 images of offline handwritten *Gurmukhi* characters. Support vector machine (SVM) and *k*-NN classifiers have been used to recognize the characters in this work. The proposed system achieves a recognition accuracy of 98.10 and 97.14 % using *k*-NN and SVM classifiers, respectively, when power curve fitting based features are used as input to the classification process.

M. Kumar (✉)
Department of Computer Science, Panjab University Rural Centre, Kauni, Muktsar, Punjab, India
e-mail: munishcse@gmail.com

R. K. Sharma
School of Mathematics & Computer Applications, Thapar University, Patiala, Punjab, India
e-mail: rksharma@thapar.edu

M. K. Jindal
Department of Computer Science & Applications, Panjab University Regional Centre, Muktsar, Punjab, India
e-mail: manishphd@rediffmail.com

Keywords Feature extraction · Parabola curve fitting based features · Power curve fitting based features · *k*-NN · SVM

Introduction

Nowadays, we are being influenced a lot by computers and almost all the important processing is being done electronically. Keeping in mind today's demand, it becomes important that the transfer of data between human beings and computers should be simple and fast. Document analysis and recognition plays a major role in data transfer between human beings and computers. Optical character recognition (OCR) is the most essential part of a document analysis system. OCR has two streams: offline OCR and online OCR. In online handwriting recognition, data is captured during the writing process with the help of a special pen and an electronic surface whereas offline documents are scanned images of prewritten text, generally, on a sheet of paper. Offline handwritten character recognition, usually abbreviated as Offline HCR, is the process of converting handwritten text into machine processable format. Research work has been continued in this field since the late 1960s, throughout the world. It is still an active area of research as the problem involved is complex in nature. Till now, no solution has been offered that solves the problem correctly and efficiently as far as Indian scripts are concerned. Though many researchers have worked to recognize the characters of Indian scripts but the problem of interchanging data between human beings and computing machines is still a challenging one. Most of the published work on Indian scripts recognition deals with printed documents and a few articles deal with handwritten character recognition problem. Handwriting recognition

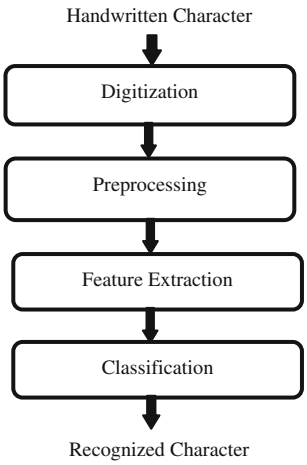

Fig. 1 Block diagram of offline handwritten character recognition system

provides a methodology for improving the interface between the user and the computer as it enables computers to read and process handwritten documents. Offline handwriting recognition is significantly different from online handwriting recognition, because here, stroke information is not available [1, 2]. Research in offline handwritten character recognition is popular owing to its practical usage such as recognition of text in bank cheques, recognition of prescription by doctors, automatic pin code reading of postal mails etc. There are four major stages in handwritten character recognition problem: digitization, pre-processing, feature extraction and classification. The block diagram of handwritten character recognition system is given in Fig. 1. Before feature extraction phase, we have performed digitization and pre-processing activities on character image. Digitization is the process of converting paper based handwritten *Gurmukhi* character into electronic form. The electronic conversion is accomplished by using a procedure whereby a character image is scanned and an electronic representation of the original image of character, in the form of a TIFF image, is produced. Digitization produces the digital image which is fed to the pre-processing phase. In this phase, the gray level character image is normalized into a window of size 100 × 100 using nearest neighborhood interpolation (NNI) algorithm. After normalization, we produce bitmap image of the normalized image. Now, the bitmap image is transformed into a thinned image using parallel thinning algorithm proposed by Zhang and Suen [3]. Feature extraction is an important component of a character recognition engine. The main aim of feature extraction phase is to detect various features of digitized character image, which maximized the recognition accuracy. The extracted features should be able to assign a unique classification score to a character. Classification is the decision making stage of an Offline HCR system. This stage makes use of the features extracted in previous stage for deciding the class membership. Two classifiers, namely, *k*-NN and SVM have been considered for recognition purpose. As such, a well-defined feature extraction algorithm makes the classification process more effective and efficient. For recognition of patterns appearing in each such image, we have proposed two feature extraction

S. No.	Character	Character name	S. No.	Character	Character name	S. No.	Character	Character name	S. No.	Character	Character name
1	ੳ	*urha*	2	ਅ	*aara*	3	ੲ	*eeri*	4	ਸ	*sassa*
5	ਹ	*hahha*	6	ਕ	*kakka*	7	ਖ	*khkha*	8	ਗ	*gagga*
9	ਘ	*ghaga*	10	ਙ	*naiaa*	11	ਚ	*chcha*	12	ਛ	*chachcha*
13	ਜ	*jajja*	14	ਝ	*jhaja*	15	ਞ	*nana*	16	ਟ	*tainka*
17	ਠ	*thaththa*	18	ਡ	*dadda*	19	ਢ	*dhada*	20	ਣ	*naana*
21	ਤ	*tata*	22	ਥ	*thatha*	23	ਦ	*dada*	24	ਧ	*dhadha*
25	ਨ	*nanna*	26	ਪ	*pappa*	27	ਫ	*phpha*	28	ਬ	*babba*
29	ਭ	*bhaba*	30	ਮ	*mamma*	31	ਯ	*jaiyaa*	32	ਰ	*rarra*
33	ਲ	*lalla*	34	ਵ	*vava*	35	ੜ	*rarha*			

Fig. 2 *Gurmukhi* script character set

Fig. 3 **a** Digitized image of *Gurmukhi* character (ਕ) **b** thinned image of *Gurmukhi* character (ਕ)

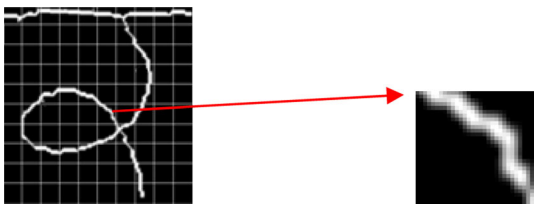

Fig. 4 Parabola curve fitting based feature extraction

techniques, namely, parabola curve fitting based features and power curve fitting based features.

Researchers have used different types of features for character recognition and consequently different feature extraction methods have surfaced for representation of characters, such as zoning, diagonal, directional, transition, intersection and open end points, gradient and chain code features etc. Hanmandlu et al. [4] have reported zoning features for handwritten Hindi numerals. They have divided the input image into 24 zones in their work and compute the vector distance for each pixel position in the grid from the bottom left corner and normalize these distances to [0, 1] in order to obtain the features. Kumar et al. [5] have achieved 94.29 % accuracy for offline handwritten *Gurmukhi* character recognition with intersection and open end points as features and SVM with polynomial kernel as the classifier. Sharma and Jhajj [6] have proposed the zoning feature extraction technique for extracting features of the characters in *Gurmukhi* script. Sethi and Chatterjee [7] have presented a Devanagari hand-printed numeral recognition system. Bansal and Sinha [8] have proposed a technique for complete Devanagri script recognition. Sharma et al. [9] have proposed a directional chain code features based quadratic classifier and obtained 80.36 % accuracy for handwritten Devanagari characters. Pardeep et al. [10] have used diagonal feature extraction technique for handwritten character recognition system. Rajput and Mali [11] have reported chain code features for Marathi handwritten numeral recognition and they achieved 98.15 % recognition accuracy. Rodriguez and Perronnin [12] have reported gradient histogram features for word spotting in handwritten documents. Basu et al. [13] have proposed a hierarchical approach for handwritten Bangla characters recognition. Garain et al. [14] have achieved an accuracy of 96.30 % for Bangla handwritten character recognition. Roy et al. [15] have proposed a system for automatically sorting of the postal documents. They have employed a two-stage Multi-Layer Perceptron (MLP) based classifier to recognize Bangla and Arabic numerals. Sharma et al. [16] have developed an online handwritten Gurmukhi script recognition system. Kumar et al. [17] have presented a good review on handwritten Indian scripts recognition systems. Ashok and Rajan [18] have presented a handwritten character recognition system using RBF network. Pal et al. [19] dealt with recognition of offline handwritten Bangla compound characters using MQDF. Roy et al. [20] have proposed a scheme for lexicon-driven bi-lingual (Bangla and English) city names recognition for Indian postal automation. Kumar [21] has brought in an artificial intelligence based technique for machine recognition of handwritten Devanagari script. Pal et al. [22] have put forth an offline handwritten Oriya script recognition system. Pal et al. [23] have assimilated a comparative study of handwritten Devanagari character recognition. Bhattacharya et al. [24] have proposed a scheme for Bangla character recognition. They have achieved a recognition accuracy of 94.7 % and 92.1 % for training and testing, respectively. Lehal and Singh [25] have developed a complete recognition system for printed *Gurmukhi* script, where connected components are initially segmented using thinning based approach. Jindal et al. [26] have provided a complete recognition system for recognition of degraded printed *Gurmukhi* script documents. These works have motivated authors to propose two efficient feature extraction techniques, namely, parabola curve fitting based features and power curve fitting based features for offline handwritten *Gurmukhi* character recognition.

Gurmukhi Script and Data Collection

Gurmukhi script is the script used for writing the *Punjabi* language. The word *Gurmukhi* has been derived from the *Punjabi* term "*Guramukhi*", which means "from the mouth of the Guru". *Gurmukhi* script is the tenth most widely used script in the world [27]. *Gurmukhi* script has three vowel bearers, thirty-two consonants, six additional consonants, nine vowel modifiers, three auxiliary signs and three half characters. Writing style of *Gurmukhi* script is from top to bottom and left to right. In *Gurmukhi* script, there is no case sensitivity. The character set of the basic 35 *akhars* of the *Gurmukhi* script is given in Fig. 2. In *Gurmukhi* script, most of the characters have a horizontal line at the upper part called headline and characters are connected with each other through this line.

Table 1 Parabola curve fitting based feature values for the *Gurmukhi* character (ਕ) shown in Fig. 4

Zone	*a*	*b*	*c*	Zone	*a*	*b*	*c*
Z_1	0.311	0.3723	0.0165	Z_2	0.6719	0.5657	0.0217
Z_3	0.5542	0.4834	0.0098	Z_4	0.526	0.4578	0.006
Z_5	0.4744	0.4299	0.0039	Z_6	0.3417	0.3707	0.0047
Z_7	0.3015	0.345	0.0056	Z_8	0.3021	0.3465	0.0045
Z_9	0.3257	0.3618	0.0063	Z_{10}	0.3182	0.3563	0.0054
Z_{11}	0	0	0	Z_{12}	0	0	0
Z_{13}	0	0	0	Z_{14}	0	0	0
Z_{15}	0	0	0	Z_{16}	0	0	0
Z_{17}	0.2732	0.3415	0.0078	Z_{18}	0.2804	0.3438	0.0069
Z_{19}	0	0	0	Z_{20}	0	0	0
Z_{21}	0	0	0	Z_{22}	0	0	0
Z_{23}	0	0	0	Z_{24}	0	0	0
Z_{25}	0	0	0	Z_{26}	0	0	0
Z_{27}	0.3055	0.3631	0.0106	Z_{28}	0.2869	0.3485	0.0107
Z_{29}	0	0	0	Z_{30}	0	0	0
Z_{31}	0	0	0	Z_{32}	0	0	0
Z_{33}	0.275	0.3424	0.0101	Z_{34}	0.2645	0.3377	0.0103
Z_{35}	0.2704	0.3402	0.01	Z_{36}	0	0	0
Z_{37}	0.2756	0.3444	0.0105	Z_{38}	0.2716	0.3428	0.0118
Z_{39}	0	0	0	Z_{40}	0	0	0
Z_{41}	0	0	0	Z_{42}	0.2506	0.335	0.0103
Z_{43}	0.2794	0.3493	0.012	Z_{44}	0.2926	0.3604	0.0137
Z_{45}	0.2682	0.3456	0.0126	Z_{46}	0.2808	0.3496	0.0114
Z_{47}	0.2787	0.3486	0.0112	Z_{48}	0.2905	0.3538	0.0127
Z_{49}	0	0	0	Z_{50}	0	0	0
Z_{51}	0	0	0	Z_{52}	0.2968	0.3557	0.014
Z_{53}	0.2967	0.3554	0.0135	Z_{54}	0	0	0
Z_{55}	0.2995	0.358	0.0144	Z_{56}	0.2923	0.357	0.0149
Z_{57}	0.3001	0.3632	0.0139	Z_{58}	0.2942	0.3578	0.0132
Z_{59}	0	0	0	Z_{60}	0	0	0
Z_{61}	0	0	0	Z_{62}	0.287	0.3509	0.0134
Z_{63}	0	0	0	Z_{64}	0.284	0.3489	0.0128
Z_{65}	0.2783	0.3467	0.0128	Z_{66}	0.2931	0.357	0.0139
Z_{67}	0.2874	0.3522	0.014	Z_{68}	0	0	0
Z_{69}	0	0	0	Z_{70}	0	0	0
Z_{71}	0	0	0	Z_{72}	0.295	0.3586	0.0145
Z_{73}	0.3023	0.3648	0.0147	Z_{74}	0.3065	0.3683	0.0153
Z_{75}	0.3072	0.3688	0.0154	Z_{76}	0	0	0
Z_{77}	0.306	0.371	0.0161	Z_{78}	0.3079	0.3711	0.0156
Z_{79}	0	0	0	Z_{80}	0	0	0
Z_{81}	0	0	0	Z_{82}	0	0	0
Z_{83}	0	0	0	Z_{84}	0	0	0
Z_{85}	0	0	0	Z_{86}	0	0	0
Z_{87}	0.3135	0.3777	0.0165	Z_{88}	1	1	1
Z_{89}	0	0	0	Z_{90}	0	0	0
Z_{91}	0	0	0	Z_{92}	0	0	0
Z_{93}	0	0	0	Z_{94}	0	0	0
Z_{95}	0	0	0	Z_{96}	0	0	0

Table 1 continued

Zone	*a*	*b*	*c*	Zone	*a*	*b*	*c*
Z_{97}	0	0	0	Z_{98}	0.3078	0.3692	0.0166
Z_{99}	0	0	0	Z_{100}	0	0	0

Table 2 Power curve fitting based feature values for a *Gurmukhi* character (ਕ੍ਰ) as shown in Fig. 4

Zone	*a*	*b*	Zone	*a*	*b*	Zone	*a*	*b*
Z_1	1	0.8632	Z_2	0.7502	1	Z_3	0.8746	0.9747
Z_4	0.9341	0.9504	Z_5	0.967	0.9408	Z_6	0.9847	0.9077
Z_7	0.9678	0.8468	Z_8	0.9814	0.845	Z_9	0.9665	0.8613
Z_{10}	0.973	0.8649	Z_{11}	0	0	Z_{12}	0	0
Z_{13}	0	0	Z_{14}	0	0	Z_{15}	0	0
Z_{16}	0	0	Z_{17}	0.9419	0.891	Z_{18}	0.9495	0.8798
Z_{19}	0	0	Z_{20}	0	0	Z_{21}	0	0
Z_{22}	0	0	Z_{23}	0	0	Z_{24}	0	0
Z_{25}	0	0	Z_{26}	0	0	Z_{27}	0.9145	0.9087
Z_{28}	0.9156	0.8869	Z_{29}	0	0	Z_{30}	0	0
Z_{31}	0	0	Z_{32}	0	0	Z_{33}	0.9271	0.8806
Z_{34}	0.9236	0.8756	Z_{35}	0.9214	0.8703	Z_{36}	0	0
Z_{37}	0.9176	0.8755	Z_{38}	0.8883	0.8891	Z_{39}	0	0
Z_{40}	0	0	Z_{41}	0	0	Z_{42}	0.9066	0.8792
Z_{43}	0.8791	0.8653	Z_{44}	0.8599	0.8893	Z_{45}	0.8798	0.8752
Z_{46}	0.8828	0.8505	Z_{47}	0.8867	0.8487	Z_{48}	0.8597	0.85
Z_{49}	0	0	Z_{50}	0	0	Z_{51}	0	0
Z_{52}	0.8437	0.8453	Z_{53}	0.8465	0.8438	Z_{54}	0	0
Z_{55}	0.8388	0.8481	Z_{56}	0.8269	0.8631	Z_{57}	0.8298	0.8558
Z_{57}	0.8298	0.8558	Z_{57}	0.8298	0.8558	Z_{60}	0	0
Z_{61}	0	0	Z_{62}	0.8346	0.8395	Z_{63}	0	0
Z_{64}	0.8451	0.834	Z_{65}	0.8442	0.8321	Z_{66}	0.831	0.8351
Z_{67}	0.8289	0.8314	Z_{68}	0	0	Z_{69}	0	0
Z_{70}	0	0	Z_{71}	0	0	Z_{72}	0.8266	0.8368
Z_{73}	0.8186	0.8508	Z_{74}	0.8115	0.8613	Z_{75}	0.8078	0.8644
Z_{76}	0	0	Z_{77}	0.8017	0.8741	Z_{78}	0.8073	0.865
Z_{79}	0	0	Z_{80}	0	0	Z_{81}	0	0
Z_{82}	0	0	Z_{83}	0	0	Z_{84}	0	0
Z_{85}	0	0	Z_{86}	0	0	Z_{87}	0.7963	0.8775
Z_{88}	0.7968	0.8578	Z_{89}	0	0	Z_{90}	0	0
Z_{91}	0	0	Z_{92}	0	0	Z_{93}	0	0
Z_{94}	0	0	Z_{95}	0	0	Z_{96}	0	0
Z_{97}	0	0	Z_{98}	0.7981	0.8558	Z_{99}	0	0
Z_{100}	0	0						

Table 3 Partitioning strategies of training and testing data

Strategy	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*
Training data (%)	50	55	60	65	70	75	80	85	90	95	99
Testing data (%)	50	45	40	35	30	25	20	15	10	5	1

Table 4 Recognition accuracy based on *k*-NN classifier for various feature extraction techniques

Strategy	Zoning (%)	Diagonal (%)	Directional (%)	Intersection (%)	Transition (%)	Gradient (%)	Chain code (%)	Parabola curve fitting (%)	Power curve fitting (%)
a	82.86	81.94	79.83	81.31	79.83	77.43	66.51	92.97	96.74
b	78.16	80.00	76.38	80.25	78.16	76.13	65.65	87.56	96.63
c	82.21	82.50	81.29	82.21	81.29	84.43	68.07	93.14	97.86
d	86.04	85.63	86.04	81.63	86.04	84.65	70.69	91.84	92.00
e	79.62	84.10	76.48	82.38	84.10	84.86	70.19	94.48	97.90
f	84.00	88.57	85.60	86.74	87.09	86.74	72.22	94.40	97.37
g	81.86	84.00	81.43	86.71	84.00	86.71	68.42	94.14	97.00
h	83.05	91.43	85.90	85.14	89.33	85.14	74.47	95.62	98.10
i	89.71	93.14	86.57	83.71	83.14	93.14	76.28	95.43	96.29
j	85.71	83.43	77.71	89.71	74.86	89.04	64.57	88.00	95.43
k	88.57	68.57	68.57	91.43	57.14	91.43	82.86	80.00	88.57

Table 5 Recognition accuracy based on SVM with linear kernel classifier for various feature extraction techniques

Strategy	Zoning (%)	Diagonal (%)	Directional (%)	Intersection (%)	Transition (%)	Gradient (%)	Chain code (%)	Parabola curve fitting (%)	Power curve fitting (%)
a	64.86	81.31	77.43	72.00	58.11	71.94	84.68	81.37	82.86
b	63.62	80.25	76.13	73.33	57.59	70.99	82.98	80.19	81.84
c	63.93	82.21	79.50	77.14	60.21	73.01	84.35	81.50	84.43
d	64.90	83.59	80.16	77.39	60.24	74.92	85.55	81.63	84.65
e	66.19	84.38	79.62	77.71	62.10	75.66	85.04	82.38	84.86
f	69.37	88.34	84.00	81.26	67.31	80.54	90.97	86.74	89.03
g	69.86	87.29	81.86	82.43	69.14	81.22	90.57	86.71	88.14
h	71.24	88.38	85.14	83.05	69.52	83.39	92.38	88.76	90.48
i	73.71	93.14	89.71	83.71	72.00	87.09	94.86	89.43	94.57
j	77.71	92.00	89.71	84.00	70.86	89.04	93.14	85.71	94.29
k	85.71	94.29	91.43	88.57	77.14	91.42	94.28	94.29	97.14

Table 6 Recognition accuracy based on SVM with polynomial kernel classifier for various feature extraction techniques

Strategy	Zoning (%)	Diagonal (%)	Directional (%)	Intersection (%)	Transition (%)	Gradient (%)	Chain code (%)	Parabola curve fitting (%)	Power curve fitting (%)
a	58.97	76.06	74.23	84.23	57.89	66.17	64.74	76.40	80.00
b	57.59	74.79	76.32	82.92	57.02	65.71	64.31	74.67	81.40
c	59.14	77.43	78.07	84.64	59.29	69.82	65.64	77.43	81.64
d	61.47	78.69	78.12	85.22	61.14	71.50	64.81	78.04	81.88
e	61.05	79.71	78.10	85.90	62.57	71.70	64.85	79.05	84.00
f	66.40	84.34	82.29	91.20	67.66	79.78	70.40	84.91	84.69
g	68.14	83.71	83.43	90.86	69.71	79.31	71.28	83.71	86.14
h	68.19	87.05	83.62	92.00	68.95	82.85	69.52	86.10	87.24
i	72.00	90.00	84.00	94.00	73.14	89.09	75.71	89.43	88.29
j	73.71	89.71	84.00	91.43	72.00	89.04	78.28	89.71	94.29
k	82.86	91.43	85.71	94.29	80.00	94.28	80.00	94.29	94.29

Table 7 Recognition accuracy based on SVM with RBF kernel classifier for various feature extraction techniques

Strategy	Zoning (%)	Diagonal (%)	Directional (%)	Intersection (%)	Transition (%)	Gradient (%)	Chain code (%)	Parabola curve fitting (%)	Power curve fitting (%)
a	58.97	76.06	74.23	84.23	57.89	66.17	64.74	76.40	80.00
b	57.59	74.79	76.32	82.92	57.02	65.71	64.31	74.67	81.40
c	59.14	77.43	78.07	84.64	59.29	69.82	65.64	77.43	81.64
d	61.47	78.69	78.12	85.22	61.14	71.50	64.81	78.04	81.88
e	61.05	79.71	78.10	85.90	62.57	71.70	64.85	79.05	84.00
f	66.40	84.34	82.29	91.20	67.66	79.78	70.40	84.91	84.69
g	68.14	83.71	83.43	90.86	69.71	79.31	71.28	83.71	86.14
h	68.19	87.05	83.62	92.00	68.95	82.85	69.52	86.10	87.24
i	72.00	90.00	84.00	94.00	73.14	89.09	75.71	89.43	88.29
j	73.71	89.71	84.00	91.43	72.00	89.04	78.28	89.71	94.29
k	82.86	91.43	85.71	94.29	80.00	94.28	80.00	94.29	94.29

Table 8 Recognition accuracy based on 5-fold cross validation technique for various features and classifiers

Feature extraction technique	Classification technique			
	k-NN (%)	Linear-SVM (%)	Polynomial SVM (%)	RBF SVM (%)
Zoning features	78.79	65.10	61.32	65.07
Diagonal features	78.93	81.83	77.99	67.74
Directional features	74.52	78.15	75.71	57.83
Intersection and open end point features	79.65	75.05	78.06	73.74
Transition features	75.45	60.83	61.30	60.80
Gradient features	80.42	80.33	73.11	76.76
Chain code features	65.92	74.92	64.95	72.47
Parabola curve fitting based features	84.17	83.39	81.80	81.46
Power curve fitting based features	90.06	83.98	83.79	81.53

For the present work, we have collected samples of isolated handwritten *Gurmukhi* characters from one hundred different writers. These writers were requested to write each *Gurmukhi* character.

Proposed Feature Extraction Techniques

We propose two feature extraction techniques, namely, parabola curve fitting based features and power curve fitting based features, in this work. First time, we have used these features for character recognition task. These techniques have been compared with other recently proposed feature extraction techniques, namely, zoning, diagonal, directional, transition, intersection and open end points, gradient and chain code features. Two classifiers, namely, *k*-NN and SVM have been considered in order to perform these comparisons. The system first prepares a skeleton of the character as shown in Fig. 3a and b so that meaningful feature information about the character can be extracted.

Parabola Curve Fitting Based Features

The thinned image of a character is divided into n (=100) zones. A parabola is then fitted to the series of ON pixels (foreground pixels) in each zone using the least square method. A parabola $y = a + bx + cx^2$ is uniquely defined by three parameters: *a, b* and *c*. For each zone, a parabola is fitted using least square method. Values of a, b and c are calculated using Least Square Method. Parabola curve fitting based feature set elements corresponding to *Gurmukhi* character (ਕ) as shown in Fig. 4, are given in Table 1.

$$y = a + bx + cx^2$$

$$\sum y = na + b\sum x + \sum x^2$$

$$\sum xy = \mathrm{a}\sum x + b\sum x^2 + c\sum x^3$$

$$\sum x^2 y = a\sum x^2 + b\sum x^3 + c\sum x^4$$

As such, this will give $3n$ features for a given bitmap.

The steps that have been used to extract these features are given below.

Table 9 Confusion matrix based upon parabola curve fitting features and *k*-NN classifier

Character	Confused with characters							
ਉ	ਉ 97%	ਧ 3%						
ਅ	ਅ 96%	ਣ 4%						
ਏ	ਏ 86%	ਹ 3%	ਢ 2%	ਠ 1%	ਧ 3%	ਨ 2%	ਭ 2%	ਵ 1%
ਸ	ਸ 84%	ਘ 2%	ਣ 2%	ਥ 4%	ਨ 4%	ਢ 4%		
ਹ	ਹ 79%	ੜ 2%	ਚ 4%	ਫ 4%	ਰ 1%	ਵ 10%		
ੜ	ੜ 91%	ਝ 3%	ਰ 2%	ਢ 3%	ੜ 1%			
ਖ	ਖ 92%	ਙ 2%	ਠ 3%	ਸ਼ 3%				
ਗ	ਗ 90%	ਸ 6%	ਲ 4%					
ਘ	ਘ 96%	ਖ 2%	ਚ 2%					
ਙ	ਙ 81%	ਸ 2%	ਟ 2%	ਠ 4%	ਣ 4%	ਯ 3%	ਢ 4%	
ਚ	ਚ 82%	ਹ 2%	ਛ 4%	ਫ 2%	ਣ 5%	ਥ 2%	ਦ 3%	
ਛ	ਛ 59%	ਚ 6%	ਟ 4%	ਠ 5%	ਡ 9%	ਫ 1%	ਥ 4%	ਵ 12%
ਜ	ਜ 82%	ਚ 1%	ਛ 1%	ਧ 3%	ਰ 4%	ਵ 6%	ੜ 3%	
ਝ	ਝ 89%	ਸ 1%	ਚ 1%	ਛ 2%	ਣ 1%	ਨ 4%	ਪ 1%	ੜ 1%
ਞ	ਞ 85%	ਏ 1%	ਛ 3%	ਡ 2%	ਨ 4%	ਲ 5%		
ਟ	ਟ 77%	ਏ 2%	ਸ 4%	ਠ 4%	ਡ 3%	ਣ 2%	ਦ 2%	ਵ 6%
ਠ	ਠ 88%	ਚ 4%	ਫ 3%	ਣ 4%	ਲ 1%			
ਡ	ਡ 92%	ਝ 3%	ਪ 4%	ੜ 1%				
ਢ	ਢ 67%	ਹ 4%	ੜ 5%	ਚ 8%	ਛ 4%	ਠ 5%	ਨ 2%	ਪ 5%
ਣ	ਣ 77%	ਅ 2%	ਹ 2%	ਙ 5%	ਚ 4%	ਞ 2%	ਲ 4%	ਵ 4%
ਤ	ਤ 86%	ਸ 2%	ਡ 2%	ਫ 2%	ਫ 3%	ਭ 2%	ਰ 1%	ਵ 2%
ਥ	ਥ 95%	ਣ 2%	ਰ 3%					
ਦ	ਦ 75%	ਡ 5%	ਝ 5%	ਫ 5%	ਫ 6%	ਣ 1%	ਰ 1%	ਵ 2%
ਧ	ਧ 95%	ਥ 3%	ਏ 2%					
ਨ	ਨ 86%	ਹ 3%	ੜ 2%	ਠ 4%	ਫ 4%	ਰ 1%		
ਪ	ਪ 92%	ਸ 2%	ਥ 1%	ਸ਼ 3%	ਧ 2%			
ਫ	ਫ 74%	ਖ 3%	ਚ 7%	ਠ 4%	ਨ 4%	ਵ 8%		
ਬ	ਬ 92%	ਸ 1%	ਫ 1%	ਧ 2%	ਨ 4%			
ਭ	ਭ 80%	ਸ 3%	ਠ 1%	ਡ 6%	ਫ 2%	ਨ 4%	ਵ 4%	
ਮ	ਮ 86%	ਸ 4%	ਙ 1%	ਚ 1%	ਣ 1%	ਰ 5%	ਵ 2%	
ਯ	ਯ 86%	ਚ 3%	ਛ 3%	ਠ 2%	ਡ 4%	ਫ 1%	ਣ 1%	
ਰ	ਰ 71%	ੜ 5%	ਠ 4%	ਡ 6%	ਣ 4%	ਮ 4%	ਲ 2%	ਵ 4%
ਲ	ਲ 76%	ਸ 2%	ੜ 4%	ਛ 6%	ਠ 3%	ਡ 2%	ਫ 3%	ਣ 4%
ਵ	ਵ 85%	ੜ 1%	ਟ 4%	ਡ 3%	ਨ 6%	ਲ 1%		
ੜ	ੜ 77%	ਲ 1%	ਹ 3%	ਡ 4%	ਨ 4%	ਰ 3%	ਵ 8%	

Step I: Divide the thinned image into n (=100) number of equal sized zones.

Step II: For each zone, fit a parabola using the least square method and calculate the values of *a, b* and *c* (Fig. 4).

Step III: Corresponding to the zones that do not have a foreground pixel, set the values of *a, b* and *c* to zero.

Step IV: Normalize the feature values in the scale [0, 1].

Power Curve Fitting Based Features

The thinned image of a character is again divided into n (= 100) zones. A power curve is fitted to the series of ON pixels (foreground pixels) in every zone using the least square method. A power curve of the form $y = ax^b$ is uniquely defined by two parameters: a and b. For each zone, a power curve is fitted using least square method. Parabola curve fitting based feature set elements corresponding to *Gurmukhi* character (ਕ) as shown in Fig. 4, are given in Table 2. Thus the values of a and b are calculated as follows:

$$y = ax^b$$

$$\log y = \log\ a + b\ \log x$$

$$\text{Put } \log\ y = Y, \log\ a = a \,\&\, \log\ x = X$$

$$\text{So, } Y = a + bX$$

$$\sum Y = na + b\sum X$$

$$\sum XY = A\sum X + b\sum X^2$$

These parameters will give $2n$ features for a given bitmap.

The steps that have been used to extract these features are given below.

Table 10 Confusion matrix based upon power curve fitting features and *k*-NN classifier

Character	Confused with characters							
ੳ	ੳ 100%							
ਅ	ਅ 100%							
ੲ	ੲ 93%	ਖ 2%	ਨ 3%	ਵ 2%				
ਸ	ਸ 88%	ਗ 4%	ਡ 2%	ਨ 1%	ਮ 2%	ਲ 3%		
ਹ	ਹ 92%	ਗ 2%	ਡ 4%	ੜ 2%				
ਕ	ਕ 96%	ਢ 2%	ੜ 2%					
ਖ	ਖ 87%	ਨ 12%	ਲ 1%					
ਗ	ਗ 88%	ਸ 3%	ਛ 2%	ਡ 2%	ਰ 5%			
ਘ	ਘ 100%							
ਙ	ਙ 98%	ਨ 2%						
ਚ	ਚ 94%	ਡ 4%	ਢ 2%					
ਛ	ਛ 80%	ਅ 2%	ਖ 3%	ਡ 4%	ਢ 2%	ਨ 2%	ਲ 3%	ੜ 4%
ਜ	ਜ 83%	ਹ 2%	ਝ 4%	ਨ 8%	ੜ 3%			
ਝ	ਝ 92%	ੲ 2%	ਖ 2%	ਡ 3%	ਵ 1%			
ਞ	ਞ 85%	ਛ 3%	ਨ 8%	ਵ 4%				
ਟ	ਟ 82%	ਛ 4%	ਡ 2%	ਨ 2%	ਢ 2%	ਰ 4%	ਲ 2%	ਵ 2%
ਠ	ਠ 93%	ਫ 3%	ਰ 4%					
ਡ	ਡ 84%	ਛ 7%	ਨ 4%	ਤ 4%	ਫ 1%			
ਢ	ਢ 94%	ਨ 2%	ਫ 2%	ਵ 2%				
ਨ	ਨ 98%	ਲ 2%						
ਤ	ਤ 88%	ਨ 3%	ਭ 4%	ਰ 5%				
ਥ	ਥ 85%	ਸ 3%	ੜ 4%	ਨ 4%	ਧ 3%	ਲ 1%		
ਦ	ਦ 98%	ਛ 2%						
ਧ	ਧ 91%	ਭ 4%	ਬ 1%	ਲ 4%				
ਨ	ਨ 70%	ਛ 10%	ਟ 3%	ਡ 2%	ਢ 3%	ਨ 4%	ਫ 4%	ਰ 4%
ਪ	ਪ 100%							
ਫ	ਫ 75%	ਛ 9%	ਠ 4%	ਟ 2%	ਨ 4%	ਰ 6%		
ਬ	ਬ 96%	ਰ 4%						
ਭ	ਭ 88%	ਡ 4%	ਨ 8%					
ਮ	ਮ 95%	ਸ 2%	ਨ 3%					
ਯ	ਯ 96%	ਡ 3%	ਲ 1%					
ਰ	ਰ 85%	ਹ 2%	ਗ 3%	ਛ 2%	ਨ 4%	ੜ 4%		
ਲ	ਲ 90%	ਖ 1%	ਝ 1%	ਧ 2%	ਨ 6%			
ਵ	ਵ 85%	ਛ 5%	ਨ 8%	ੜ 2%				
ੜ	ੜ 93%	ਢ 4%	ਵ 3%					

Step I: Divide the thinned image into n (= 100) number of equal sized zones.

Step II: In each zone, fit a power curve using the least square method and calculate the values of a and b.

Step III: Corresponding to the zones that do not have a foreground pixel, set the value of a and b as zero.

Step IV: Normalize the feature values in the scale [0, 1].

Results and Discussion

Two classifiers, namely, *k*-NN and SVM have been used in this work in order to compare the proposed feature extraction techniques with other recently proposed feature extraction techniques. Each technique has been tested with 3,500 images of handwritten *Gurmukhi* characters. We have also analyzed the performance of other recently proposed feature extraction techniques, namely, zoning, diagonal, directional, transition, intersection and open end points, gradient and chain code features. A performance analysis has also been carried out in order to find the best feature set for a given offline handwritten *Gurmukhi* character. We have divided the data set using partitioning strategies as depicted in Table 3. These strategies are considered to establish how the classifiers behave for different training data.

Performance Analysis Based on *k*-NN Classifier

We have done experiments using *k*-NN classifier for the value of $k = 1, 3, 5, 7$. From these experiments we have achieved best accuracy for the value of $k = 5$. In this sub-

section, results of performance analysis of partitioning strategies (*a, b,…, k*) based on the 5-NN classifier are presented (Table 4). It has been noted that the power curve fitting based on features with 5-NN classifier achieved maximum recognition accuracy of 98.10 % when we use strategy *h*.

Performance Analysis Based on SVM with Linear Kernel Classifier

In this sub-section, results of performance analysis of partitioning strategies (*a, b,…, k*) based on SVM with linear kernel classifier are presented (Table 5). One can see that power curve fitting based features enable us to achieve a recognition accuracy of 97.14 % when we use strategy *k* and SVM with linear kernel.

Performance Analysis Based on SVM with Polynomial Kernel Classifier

In this sub-section, results of performance analysis of partitioning strategies (*a, b,…, k*) based on SVM with polynomial kernel classifier are presented (Table 6). It has been observed that the power curve fitting based features make this possible to achieve a recognition accuracy of 94.29 % when we use strategy *k* and SVM with polynomial kernel.

Performance Analysis Based on SVM with RBF Kernel Classifier

In this sub-section, results of performance analysis of partitioning strategies (*a, b,…, k*) based on SVM with RBF kernel classifier are illustrated (Table 7). It has been seen that power curve fitting based features and SVM with RBF kernel achieved maximum recognition accuracy of 94.29 % when we use strategy *k*.

Performance Analysis Based on 5-Fold Cross Validation Technique for Various Features and Classifiers

In this sub-section, we have presented experimental results based on 5-fold cross validation technique. In general, *k*-fold cross validation divides, complete data set of each category into *k* equal subsets. Then one subset is taken as testing data and remaining $k-1$ subsets are taken as training data. By cross validation each sample of training data is also predicted and it gives the percentage of correctly recognized testing dataset. The system achieves a recognition accuracy of 90.06, 83.98, 83.79 and 81.53 % using *k*-NN, SVM with linear kernel, SVM with polynomial kernel and SVM with RBF kernel classifiers, respectively, when power curve fitting based features are used with 5-fold cross validation technique in the classification process (Table 8).

As given in Table 8, a maximum recognition accuracy of 84.17 % with parabola curve fitting based features and *k*-NN classifier could be achieved. For this case, confusion matrix for the *Gurmukhi* characters is given in Table 9. Also, power curve fitting based features and *k*-NN classifier, could achieve a maximum recognition accuracy of 90.06 %. The confusion matrix for this case is given in Table 10.

Conclusion

In this work, we have proposed efficient feature extraction techniques for offline handwritten *Gurmukhi* character recognition. The classifiers that have been employed in this work are *k*-NN and SVM with three flavors, *i.e.*, SVM with linear kernel, SVM with polynomial kernel and SVM with RBF kernel. The system achieves a recognition accuracy of 97.14, 94.29 and 94.29 % using SVM with linear kernel, SVM with polynomial kernel and SVM with RBF kernel classifiers, respectively, when power curve fitting based features are used as input to the classification process. In this case, 99 % data was taken in training and 1 % data was considered in testing (Strategy *k*). It has also been seen that the results achieved using parabola curve fitting based features are also better than recently proposed feature extraction techniques. Maximum recognition accuracy of 95.62 % could be achieved when parabola curve fitting based features were used with *k*-NN classifier and Strategy *h*. When we use SVM classifier with parabola curve fitting based features, maximum recognition accuracy of 94.29 % could be achieved. In this case, 99 % data was taken in training and 1 % data was considered in testing (Strategy *k*). In this work, the highest recognition accuracy of 98.09 % could be achieved when power curve fitting based features were used with *k*-NN classifier. In this case, 85 % data was taken in training and 15 % data was considered in testing (strategy *h*). Using 5-fold cross validation technique and power curve fitting based features, we have achieved a recognition accuracy of 90.06, 83.98, 83.79 and 81.53 % using *k*-NN, SVM with linear kernel, SVM with polynomial kernel and SVM with RBF kernel classifiers, respectively. As such, the results obtained using the power curve fitting based features are promising. This technique can further be explored by combining with other techniques for achieving higher recognition accuracy.

References

1. Lorigo LM, Govindaraju V (2006) Offline arabic handwriting recognition: a survey. IEEE Trans PAMI 28(5):712–724

2. Plamondon R, Srihari SN (2000) On-line and off- line handwritten character recognition: a comprehensive survey. IEEE Trans PAMI 22(1):63–84
3. Zhang TY, Suen CY (1984) A fast parallel algorithm for thinning digital patterns. Commun ACM 27(3):236–239
4. Hanmandlu M, Grover J, Madasu VK, Vasikarla S (2007) Input fuzzy for the recognition of handwritten Hindi numeral. In: Proceedings of the ITNG, pp 208–213
5. Kumar M, Sharma RK, Jindal MK (2011) SVM based offline handwritten *Gurmukhi* character recognition. In: Proceedings of the SCAKD, Vol 758, pp 51–62
6. Sharma DV, Jhajj P (2010) Recognition of isolated handwritten characters in *Gurmukhi* script. Int J Comput Appl 4(8):9–17
7. Sethi K, Chatterjee B (1976) Machine recognition of constrained hand-printed Devanagari numerals. J Inst Electr Telecom Eng 22:532–535
8. Bansal V, Sinha RMK (2000) Integrating knowledge sources in Devanagari text recognition. IEEE Trans Syst, Man Cybern 30(4):500–505
9. Sharma N, Pal U, Kimura F, Pal S (2006) Recognition of off-line handwritten Devanagari characters using quadratic classifier. In: Proceedings of the ICVGIP, pp 805–816
10. Pardeep J, Srinivasan E, Himavathi S (2010) Diagonal feature extraction based handwritten character system using neural network. Int J Comput Appl 8(9):17–22
11. Rajput GG, Mali SM (2010) Marathi handwritten numeral recognition using Fourier descriptors and normalized chain code. In: Proceedings of the recent trends in image processing and pattern recognition (RTIPPR), pp 141–145
12. Rodriguez JA, Perronnin F (2008) Local gradient histogram features for word spotting in unconstrained handwritten documents. In: Proceedings of the international conference on frontiers in handwriting recognition, pp 7–12
13. Basu S, Das N, Sarkar R, Kundu M, Nasipuri M, Basu DK (2009) A hierarchical approach to recognition of handwritten Bangla characters. Pattern Recognit 42(7):1467–1484
14. Garain U, Chaudhuri BB, Pal TT (2002) Online handwritten Indian script recognition: a human motor function based framework, In: Proceedings of the 16th international conference on pattern recognition (ICPR), Vol 3, pp 164–167
15. Roy K, Vajda S, Pal U, Chaudhuri BB (2004) A system towards Indian postal automation, In: Proceedings of the 9th international workshop on frontiers in handwriting recognition (IWFHR), pp 361–367
16. Sharma A, Kumar R, Sharma RK (2008) Online handwritten *Gurmukhi* character recognition using elastic matching. Int J Congr Image Signal Process 2:391–396
17. Kumar M, Jindal MK, Sharma RK (2011) Review on OCR for handwritten indian scripts character recognition. In: Proceeding of international conference on DPPR 2011, pp 268–276
18. Ashok J, Rajan EG (2011) Offline handwritten character recognition using radial basis function. Int J Adv Netw Appl 2(4):792–795
19. Pal U, Wakabayashi T, Kimura F (2007) Handwritten Bangla compound character recognition using gradient feature. In: Proceedings of the 10th international conference on information technology (ICIT), pp 208–213
20. Roy K, Alaei A, Pal U (2010) Word-wise handwritten Persian and Roman script identification. In: Proceedings of the international conference on frontiers in handwriting recognition (ICFHR), pp 628–633
21. Kumar D (2008) AI approach to hand written Devanagri script recognition. In: Proceedings of the IEEE region 10th international conference on EC3-energy,computer,communication and control systems, Vol 2, pp 229–237
22. Pal U, Wakabayashi T, Kimura F (2007) A system for off-line Oriya handwritten character recognition using curvature feature. In: Proceedings of 10th International Conference on Information Technology (ICIT), pp 227–229
23. Pal U, Wakabayashi T, Kimura F (2009) Comparative study of Devanagari handwritten character recognition using different feature and classifiers. In: Proceedings of 10th international conference document analysis and recognition (ICDAR '09), pp 1111–1115
24. Bhattacharya U, Shridhar M, Parui SK (2006) On recognition of handwritten Bangla characters. In: Proceedings of International Conference on Computer Vision, Graphics and Image Processing (ICVGIP), pp 817–828
25. Lehal GS, Singh C (2000) A Gurmukhi script recognition system. In: Proceedings of the 15th ICPR, pp 557–560
26. Jindal MK, Lehal GS, Sharma RK (2009) On segmentation of touching characters and overlapping lines in degraded printed *Gurmukhi* script. Int J Image Graph 9(3):321–353
27. http://en.wikipedia.org/wiki/Punjabi_language