3. Development of OHWR System for Gurmukhi

R.K. Sharma, Palki Sharma, Narpinder Sharma, Harjeet Singh, Karun Verma,
Ravinder Kumar, Rajesh Kumar,
School of Mathematics & Computer Applications,
Thapar University, Patiala, Punjab (INDIA)

Abstract

Handwritten character recognition is a complex task owing to various writing styles of different individuals. A number of authors have worked on the problem of handwritten character recognition. They have, in general, used structural and statistical features in their work. Handwritten character recognition systems have also been proposed for Gurmukhi script by some authors. This work presents a system to recognize online handwritten Gurmukhi characters, Gurmukhi numerals, Roman numerals and special characters. A sufficiently large annotated database has been created in this work for the strokes used in writing these Gurmukhi symbols. The recognition engines developed in this work are based on 100 samples (Engine 1) and 22 samples (Engines 2, 3 and 4) of each class collected from different writers. These samples have been preprocessed and annotated. LibSVM classifier has been used in this work for classification purpose. The system presented in this work could attain the highest recognition accuracy of 96.8% at 5-fold cross validation for Gurmukhi characters, of 90.0% at 4-fold cross validation for Gurmukhi numerals, of 90.2% at 5-fold cross validation for Gurmukhi numerals and Roman numerals, of 84.0% at 4-fold cross validation for special characters, Gurmukhi and Roman numerals. Postprocessing of classes has also been performed in this work in order to refine the recognition process results. In this postprocessing, the sequence of classes is analyzed; overwritten strokes are identified and resolved. A heuristic based algorithm has been developed to form Gurmukhi aksharas from the identified classes.

1. Introduction

This is a well-established fact that handwritten character recognition is a complex task. This is due to different handwriting styles of individuals, and also the cursiveness in their handwriting. Handwritten character recognition is divided into two categories: offline and online. In offline handwritten character recognition, data are scanned images taken from a prewritten text, usually on a sheet of paper. In online handwriting recognition systems, data are captured while writing with the help of a special pen and an electronic surface. These systems generally include the phases of preprocessing, features extraction and classification. Feature extraction is a very significant phase in a character recognition system, which is used to decide for the relevant shape contained in the character. The performance of a character recognition system largely depends on the features, which have been extracted.

A good amount of work has been reported in literature on handwritten character recognition problem since last couple of years. If we compare the work reported for non-Indic scripts with Indic scripts, a good number of systems are available for handwritten character recognition for non-Indic scripts. However, there is comparatively less amount of work that has been reported for handwritten character recognition systems in Indic scripts. Some of the works that have been carried out for *Devanagari, Bangla, Tamil* and *Oriya* for both offline and online handwritten character recognition systems are discussed in next section.

2. Related work

This section presents a brief survey of related work carried out for both non-Indic and Indic scripts.

In 1987, Almuallim and Yamaguchi proposed a technique for cursive handwritten *Arabic script* recognition. They segmented the words into strokes and these strokes were further classified using geometrical and topological features. Roy *et al.* (2004) presented a system towards Indian postal automation for sorting of postal documents written in *Arabic* script. They extracted features based on the density of black pixels and number of components inside a block. A recognition accuracy of 92.1% was achieved by them for handwritten numeral recognition with two stage MLP classifier. Alaei *et al.* (2010) proposed a two-stage methodology for isolated handwritten Persian character recognition. They extracted features based on modified chain code directional frequencies. They obtained a recognition accuracy of 98.1% and 96.6% for 8-class and 32-class problems, respectively using support vector machine (SVM) classifier.

In 1993, Dutta and Chaudhury reported a system for recognition of isolated *Bangla* alphabets and numerals. They used curvature features in their work and also used feed forward neural network for classification. Their system achieved a recognition accuracy of 90.0% for numeral recognition and a recognition accuracy of 85.0% for alphabets recognition. Bhattacharya *et al.* (2002) presented a hybrid scheme for hand printed numeral recognition based on a self-organizing network and MLP classification techniques. Bhattacharya *et al.* (2006) proposed a scheme for *Bangla* character recognition. Here, they extracted features by computing local chain code histograms of input character shape. The recognition accuracy achieved in their work is 92.1%. Sharma *et al.* (2006) proposed a scheme for offline handwritten *Devanagari* character recognition based on quadratic classifier. They achieved an accuracy of 98.9% and 80.4% for *Devanagari* numerals and characters, respectively using 5-fold cross validation technique. The work carried out by Pal *et al.* (2007a) deals with recognition of off-line handwritten *Bangla* compound characters using gradient features. They achieved a recognition accuracy of 89.9% with 5-fold cross validation technique. Pal *et al.* (2007b) have also proposed a system for offline handwritten *Devanagari* character recognition. In this work, they have extracted directional information obtained from the arc tangent of the gradient. Using this technique, they achieved a recognition accuracy of 94.2% with 5-fold cross validation. Pal *et al.* (2007c) proposed an offline handwritten *Oriya* script recognition system. Here, they

Gurmukhi OHWR

extracted curvature features and obtained an accuracy of 94.6% from few offline handwritten *Oriya* samples. Rajashekararadhya and Vanajaranjan (2008) proposed a technique using zoning features for offline handwritten numeral recognition of four widely used Indian scripts. The recognition accuracy achieved in their work is 98.6% for Kannada numerals with SVM classifier. Desai (2010) proposed a technique for *Gujarati* handwritten numeral recognition. In this work, he has used features abstracted from four different profiles of digits with multilayered feed forward neural network. In this work, he achieved a recognition accuracy of 82.0% for *Gujarati* handwritten digit identification. Rampalli and Ramakrishnan (2011) discussed online and offline strategies for recognition of handwritten *Kannada* characters. Bhattacharya *et al.* (2012) presented an efficient two stage approach for handwritten *Bangla* characters recognition. Some work has been carried out by researchers for the recognition of handwritten *Gurmukhi* script characters. One such work is proposed by Sharma *et al.* (2008) for online handwritten *Gurmukhi* script recognition. Sharma and Jhajj (2010) used zoning based features for handwritten *Gurmukhi* character recognition. They employed two classifiers, namely; *k*-NN and SVM for recognition. They reported a maximum recognition accuracy of 72.5% and 72.0%, respectively with *k*-NN and SVM classifiers. Kumar *et al.* (2011a) extracted intersection and open end points features for offline handwritten *Gurmukhi* character recognition. They used SVM for classification and reported a recognition accuracy of 94.3% in their work. They have also extended their work in order to extract curvature features for offline handwritten *Gurmukhi* character recognition (2011b).

3. Introduction to Gurmukhi script and data collection

Gurmukhi is the script for writing *Punjabi* language and is derived from the old *Punjabi* term *"Guramukhi"*, which means "from the mouth of the Guru". *Gurmukhi* is the 12th most widely used script in the world. Writing style of *Gurmukhi* script is from top to bottom and left to right. In *Gurmukhi* script, there is no case sensitivity. Most of the characters in *Gurmukhi* script have a horizontal line at the upper part called headline and characters are connected with each other through this line. Modern *Gurmukhi* has forty-one consonants (vianjan), nine vowel symbols (lāga mātrā), two symbols for nasal sounds (bindī and ippī), and one symbol which duplicates the sound of any consonant (addak). In addition, four conjuncts are used: three subjoined forms of the consonants Rara, Haha and Vava, and one half-form of Yayya. *Gurmukhi* characters are written in three zones, namely, upper zone, lower zone and middle zone.

3.1 Gurmukhi Writing Zones

Upper zone: It contains headline, part of strokes and matras *(tippi, adak, lawan etc.)*.

Middle zone: It contains consonants and sub part of vowels. Middle zone is the busiest zone in *Gurmukhi* script.

Lower zone: It contains some vowels and pairian characters *(Raa,Vaa, Halant, Haa etc.)*.

Figure 3.1 contains a Punjabi word explaining three zones of *Gurmukhi* script.

Figure 3.1: Three zones in Gurmukhi script

3.2 Gurmukhi consonants

Characters of *Gurmukhi* script alphabet (35 akhars) can be grouped to form a grid. The top row has three vowel holders followed by two consonants. Table 3.1 depicts these consonants.

3.3 Gurmukhi consonants with dot subscript

Gurmukhi consonants with a dot subscript are also called consonant with *pairi bindi.* These consonants are given in Table 3.2.

3.4 Gurmukhi vowel modifiers

Gurmukhi has ten vowels, or *laga matras.* These are given in Table 3.3.

Table 3.1: Gurmukhi consonants

ੳ	ਅ	ੲ	ਸ	ਹ
ਕ	ਖ	ਗ	ਘ	ਙ
ਚ	ਛ	ਜ	ਝ	ਞ
ਟ	ਠ	ਡ	ਢ	ਣ
ਤ	ਥ	ਦ	ਧ	ਨ
ਪ	ਫ	ਬ	ਭ	ਮ
ਯ	ਰ	ਲ	ਵ	ੜ

Table 3.2: Gurmukhi consonants with pairi bindi

ਸ਼	ਖ਼	ਗ਼	ਜ਼	ਫ਼	ਲ਼

Table 3.3: Gurmukhi vowels

ਅ()	ਆ(◌◌ਾ)	ਇ(ਿ◌◌)	ਈ(◌◌ੀ)	ਉ(◌◌ੁ)
ਊ(◌◌ੂ)	ਏ(◌◌ੇ)	ਐ(◌◌ੈ)	ਓ(◌◌ੋ)	ਔ(◌◌ੌ)

3.5 Gurmukhi auxiliary symbols and half characters

Superscripts: These auxiliary symbols are presented in upper zone. These include *tippi, bindi,* and *adhak.*

Subscript: In the second row subscript symbols i.e. paireen consonants are Halant, Haa, Raa and Vaa.

Table 3.4: Gurmukhi Superscript and Subscript

◌◌ੰ	◌◌ਂ	◌◌ੱ	
◌◌੍	◌◌੍ਹ	◌◌੍ ਰ	◌◌੍ ਵ

3.6 Gurmukhi Numerals

Following table contains the characters of the *Gurmukhi* script.

Table 3.5: Gurmukhi numerals

੦	੧	੨	੩	੪	੫
੬	੭	੮	੯		

3.7 Gurmukhi Punctuations

Punctuation symbols represent separation of heading and text, or line break *etc.* These are *visarg, dandi* and *dodandi.* These are given in Table 3.6.

Table 3.6: Gurmukhi Punctuations

ਃ	।	॥

Data Collection: In this work, data has been collected with the help of Dell Latitude XT3 Tablet PC. A typical pen movement with any capturing device includes two actions, namely, PEN DOWN and PEN UP. The connected parts of the pen trace between PEN DOWN and PEN UP is called a Stroke. We have collected data for *Gurmukhi* characters

Gurmukhi OHWR

taking stroke as the smallest unit. Information on the data that has been collected in this work is given in Table 3.7.

Table 3.7: Details of collected data

Device Used	Number of Writers	Number of Words	Number of Aksharas	Number of Strokes	Number of Annotated Strokes
Dell Latitude XT3 Tablet PC	78	33,493	1,45,760	2,22,092	2,22,092

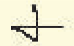
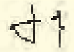
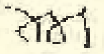
Collected data are now assigned unique Ids at stroke level, indicating that ID is considered as a class. Ids are distributed among three zones, i.e., upper, middle and lower zones. Details of these classes for Gurmukhi Characters, Gurmukhi numerals, Roman numerals and special characters are presented in Table 3.8.

As mentioned, strokes are assigned unique IDs. In order to save space, 20 example strokes have been given in Table 3.9. These strokes constitute the dependent strokes for *Gurmukhi* characters.

Table 3.8: Details of Classes

***Gurmukhi* Characters**			***Gurmukhi* Numerals**		
Strokes in Lower Zone	Strokes in Middle Zone	Strokes in Upper Zone	Strokes in Lower Zone	Strokes in Middle Zone	Strokes in Upper Zone
7	82	12	-	15	-
***Gurmukhi* and Roman Numerals**			**Special Characters, *Gurmukhi* and Roman Numerals**		
Strokes in Lower Zone	Strokes in Middle Zone	Strokes in Upper Zone	Strokes in Lower Zone	Strokes in Middle Zone	Strokes in Upper Zone
-	39	-	-	56	-

Table 3.9: Example strokes and their Ids

ID	STROKE	ID	STROKE
141		151	
142		152	
143		153	
144		154	
145		155	
146		156	
147		157	
148		158	
149		159	
150		160	

4. Preprocessing and Feature Extraction

During preprocessing, a series of operations are performed on the stroke points. While writing on Tablet-PC, there are chances of noise or distortions due to hardware or software limitations, so one needs to preprocess the stroke points. The algorithms that have been applied on raw points and their sequence are given in Figure 4.1.

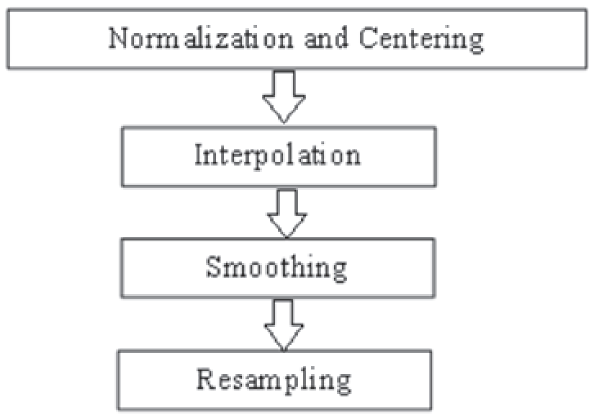


Figure 4.1: Preprocessing steps

- **Normalization and Centering:** In this step, input stroke is fitted into a fixed size window, and is moved to assumed center location.
- **Interpolation:** Higher speed of handwriting results into missing points. These missing points have been interpolated using Bezier interpolation technique in this work.
- **Smoothing:** Jitters are removed by modifying each point of the stroke with mean value of *k*-neighbors and angle subtended at k^{th} position at each end. It usually averages a point with its neighbors.
- **Resampling:** In this process, the stroke points are put equidistant from neighboring points.

These steps have been applied on the strokes collected for different *Gurmukhi* symbols and considered for next stage of feature extraction.

Performance of an online handwriting character recognition system largely depends upon the features that we select for inputting to a classifier. It is important to identify correct features for better recognition results. Also, there is no standard method for computing features of scripts, different scripts have different features. In this present study, stroke points, i.e., *X-Y* coordinates, after preprocessing, have been considered as the features to be inputted to classifier.

Classification Results

The *X-Y* coordinate features, as mentioned above, are used for recognition process. SVMs are group of supervised learning methods that can be applied for classification. A SVM classifier takes the set of input data and predicts to classify. SVM classifier is trained by a given set of training data and a model file is prepared to classify test data based upon this model file. SVM classifier has been considered with four different kernels, namely, linear kernel, polynomial kernel, RBF kernel and sigmoid kernel in this work. It has been seen that effectiveness of SVM depends on the kernel being used.

We have experimented with *k*-fold cross validation on all the above mentioned kernels with example values of learning rate (γ) and tolerance (ε). *LibSVM* has been used by us in all the experimentation. Following table contains the values of parameters that have been experimented in this work.

Table 5.1: SVM Parameters and their values

Number of folds in *k*-fold cross validation	Types of kernel	Values of learning rate (γ)	Values of tolerance (ε)
3, 4 and 5	Linear, polynomial, RBF and sigmoid	0.001 (0.001) 0.020	0.1 (0.1) 0.5

SVM parameters and their values mentioned in Table 5.1 are experimented on four Engines developed in this work and results are illustrated in following sections.

5.1 Experimentation for Engine 1: In this experiment, a recognition engine has been developed for recognition of *Gurmukhi* akhars. For this engine, we achieved the highest recognition accuracy when SVM with RBF kernel at 5-fold cross validation was trained. These results for example values of γ and ε Table 5.2. These results are also graphically displayed in Figure 5.1.

Table 5.2: Recognition Accuracy based on 5-fold cross validation

Kernel: RBF					
γ	ε=0.1	ε=0.2	ε=0.3	ε=0.4	ε=0.5
0.001	91.628	91.5248	91.6925	91.7828	91.6151
0.002	93.7822	93.7564	93.7049	93.692	93.6791
0.003	94.6207	94.5949	94.5433	94.5175	94.3369
0.004	95.227	95.1367	95.0851	94.9432	94.9174
0.005	95.5753	95.485	95.485	95.2657	95.2012
0.006	95.7559	95.743	95.6914	95.743	95.6269
0.007	95.9623	95.9494	95.9623	95.8462	95.8204
0.008	96.1042	96.0268	96.0139	96.0655	96.001
0.009	96.2332	96.2074	96.1816	96.1558	96.0397
0.01	96.2977	96.3106	96.259	96.2332	96.2203
0.011	96.4267	96.4267	96.3364	96.388	96.3622
0.012	96.4654	96.4912	96.4138	96.4396	96.4009
0.013	96.6331	96.5299	96.5299	96.5428	96.5557
0.014	96.6718	96.5686	96.6073	96.4783	96.4267
0.015	96.6073	96.5815	96.5944	96.4912	96.4912
0.016	96.6589	96.646	96.6202	96.6073	96.5686
0.017	96.6718	96.6331	96.6589	96.646	96.5944
0.018	96.7234	96.6718	96.6718	96.5944	96.6331
0.019	96.6847	96.6331	96.6847	96.7234	96.646
0.02	96.7105	96.7363	96.6331	96.7105	96.8395

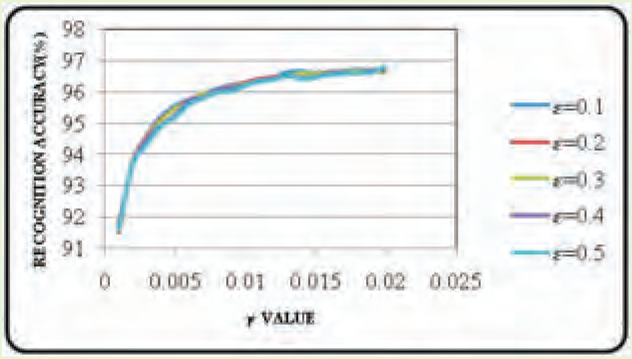


Figure 5.1: Recognition accuracy as a function of learning rate for Gurmukhi characters using 5-fold cross validation

5.2 Experimentation for Engine 2: In the second experiment, a recognition engine for Gurmukhi numerals has been developed. The parameters for training have again been considered as given in Table 5.1. The best recognition accuracy for this engine has again

Table 5.3: Performance analysis based on 4-fold cross validation

Kernel: RBF					
γ	ε=0.1	ε=0.2	ε=0.3	ε=0.4	ε=0.5
0.001	86.0606	86.9697	86.9697	86.6667	86.3636
0.002	84.2424	85.1515	85.4545	85.4545	86.0606
0.003	83.9394	84.8485	85.4545	85.7576	86.6667
0.004	84.2424	85.1515	85.4545	85.7576	86.0606
0.005	84.2424	84.8485	85.4545	85.1515	85.7576
0.006	83.9394	84.8485	85.4545	85.7576	86.6667
0.007	84.2424	85.1515	85.4545	85.7576	86.0606
0.008	84.2424	85.1515	85.4545	85.7576	86.0606
0.009	84.2424	85.1515	85.7576	85.4545	86.0606
0.01	84.2424	84.8485	85.4545	85.1515	85.7576
0.011	84.5455	85.1515	85.4545	86.0606	86.3636
0.012	83.9394	84.8485	85.4545	85.7576	86.6667
0.013	84.2424	84.8485	85.4545	86.0606	86.3636
0.014	84.2424	85.1515	85.4545	85.7576	86.0606
0.015	84.5455	85.1515	85.4545	86.0606	86.3636
0.016	84.2424	85.1515	85.4545	85.7576	86.0606
0.017	84.5455	85.1515	85.4545	85.7576	86.0606
0.018	84.2424	85.1515	85.7576	85.4545	86.0606
0.019	84.2424	84.5455	85.1515	85.1515	86.0606
0.02	84.2424	84.8485	85.4545	85.1515	85.7576

been achieved when RBF kernel was used. This accuracy was achieved when 4-fold cross validation was employed. Table 5.3 contains the results for different values of γ and ε. These results are also depicted in Figure 5.2 for this engine.

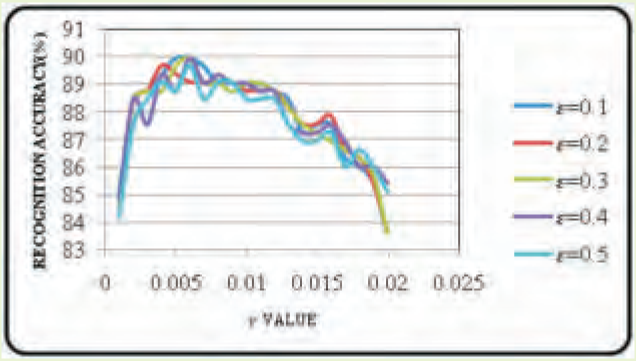


Figure 5.2: Recognition accuracy as a function of learning rate for Gurmukhi numerals using 4-fold cross validation

5.3 Experimentation for Engine 3: In this experiment, a recognition engine for *Gurmukhi* and Roman characters has been developed. Experimentation suggests that for these characters, the best accuracy is obtained when SVM with polynomial kernel and 5-fold cross validation is employed. Table 5.4 contains the results on recognition accuracy for different learning rates and tolerance values. These results are also depicted graphically in Figure 5.3.

Table 5.4: Performance analysis based on 4-fold cross validation

Kernel: Polynomial					
γ	ε=0.1	ε=0.2	ε=0.3	ε=0.4	ε=0.5
0.001	89.5317	90.2204	89.9449	88.9807	88.9807
0.002	89.3939	89.6694	89.6694	88.843	88.9807
0.003	89.2562	89.2562	89.5317	88.843	88.843
0.004	88.9807	89.3939	89.6694	88.7052	88.843
0.005	89.5317	89.3939	89.2562	88.843	88.9807
0.006	89.2562	89.2562	89.5317	88.843	88.9807
0.007	89.1185	88.9807	89.6694	88.5675	88.9807
0.008	88.9807	89.3939	89.6694	88.7052	88.843
0.009	88.9807	89.1185	89.2562	88.5675	88.843
0.01	89.5317	89.3939	89.2562	88.843	88.9807
0.011	89.1185	88.9807	89.6694	88.843	89.2562
0.012	89.2562	89.2562	89.5317	88.843	88.9807
0.013	89.2562	89.1185	89.6694	89.3939	89.3939
0.014	89.1185	88.9807	89.6694	88.5675	88.9807
0.015	89.2562	89.1185	89.6694	88.843	88.843
0.016	88.9807	89.3939	89.6694	88.7052	88.843
0.017	88.9807	89.1185	89.6694	89.3939	89.3939
0.018	88.9807	89.1185	89.2562	88.5675	88.843
0.019	88.9807	89.2562	89.5317	88.843	88.9807
0.02	89.5317	89.3939	89.2562	88.843	88.9807

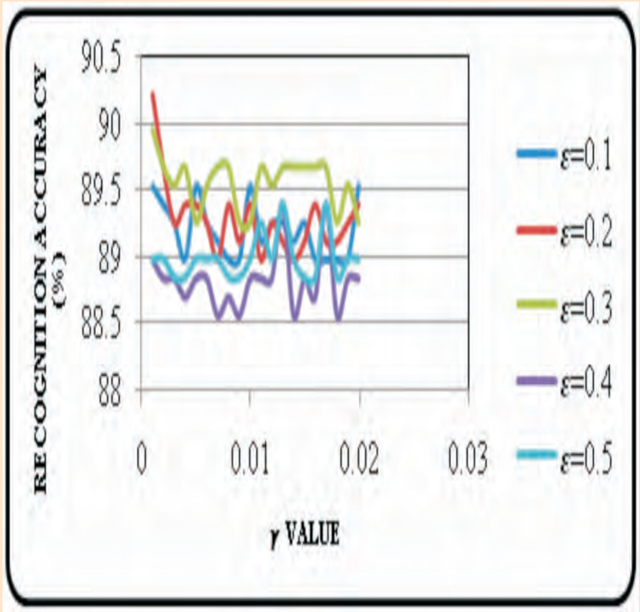


Figure 5.3: Recognition accuracy as a function of learning rate for Gurmukhi and Roman characters using 5-fold cross validation

5.4 Experimentation for Engine 4: In this experiment a recognition engine has been developed for recognizing special, *Gurmukhi* and Roman characters. Experimentation reveals that the best accuracy can be achieved when we consider SVM with linear kernel and 4-fold cross validation. Table 5.5 contains the recognition accuracy for different learning rates and tolerance values when linear kernel with 4-fold cross validation is used. Figure 5.4 depicts these results graphically.

Table 5.5: Performance analysis based on 4-fold cross validation

Kernel: Polynomial					
γ	ε=0.1	ε=0.2	ε=0.3	ε=0.4	ε=0.5
0.001	83.7029	84.0355	83.7029	83.1486	83.2594
0.002	83.0377	83.2594	83.2594	82.816	83.0377
0.003	83.0377	83.2594	83.1486	83.0377	82.5942
0.004	82.7051	83.0377	83.1486	82.7051	82.9268
0.005	82.816	83.0377	83.3703	82.816	82.9268
0.006	83.0377	83.2594	83.1486	83.0377	82.5942
0.007	83.1486	82.9268	82.816	83.0377	83.0377
0.008	82.7051	83.0377	83.1486	82.7051	82.9268
0.009	82.9268	83.2594	83.2594	83.2594	82.816
0.01	82.816	83.0377	83.3703	82.816	82.9268
0.011	82.9268	83.0377	83.2594	83.0377	82.7051
0.012	83.0377	83.2594	83.1486	83.0377	82.5942
0.013	82.9268	83.1486	83.592	82.816	82.816
0.014	83.1486	82.9268	82.816	83.0377	83.0377
0.015	83.0377	82.9268	82.9268	82.816	82.4834
0.016	82.7051	83.0377	83.1486	82.7051	82.9268
0.017	82.816	83.1486	83.1486	82.9268	82.3725
0.018	82.9268	83.2594	83.2594	83.2594	82.816
0.019	83.1486	83.0377	83.0377	83.0377	82.2616
0.02	82.816	83.0377	83.3703	82.816	82.9268

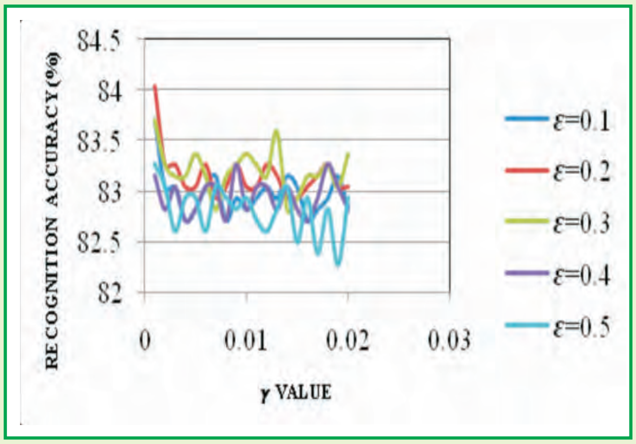


Figure 5.4: Recognition accuracy as a function of learning rate for Special, Gurmukhi and Roman characters using 4-fold cross validation

6. Conclusion

The work presented in this report proposes online handwritten *Gurmukhi* character recognition systems for *Gurmukhi* characters, numerals and other special characters and their combination. The classifier that has been employed in this work is *LibSVM* with four kernels, namely, *linear, polynomial, RBF* and *sigmoid;* with different *k*-fold strategies. Following table summarizes the recognition accuracy achieved for the four engines developed in this work.

Table 6.1: Recognition accuracy achieved for the engines proposed in this work

Engines	No. of samples (per class)	Learning rate (γ)	Tolerance (ε)	*LibSVM* kernel	*k*-fold	Accuracy (%)
Engine 1 (*Gurmukhi* characters)	100	0.02	0.5	RBF	5	96.8
Engine 2 (*Gurmukhi* Numerals)	22	0.005	0.1	RBF	4	90.0
Engine 3 (*Gurmukhi* and Roman Numerals)	22	0.001	0.2	Polynomial	5	90.2
Engine 4 (Special Characters, *Gurmukhi* and Roman Numerals)	22	0.001	0.2	Polynomial	4	84.0

This accuracy can possibly further be increased by considering a combination of classifiers and by considering a larger dataset while training the classifier. One can also experiment with other features that can be extracted from the stroke points.

References

Almuallim, H., & Yamaguchi, S. (1987). A method of recognition of Arabic cursive handwriting, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 9(5), 715-722

Alaei, A., Nagabhushan, P., & Pal, U. (2010). A new two-stage scheme for the recognition of Persian handwritten characters, *In the Proceedings of 12th ICFHR,* 130-135

Bhattacharya, U., Das, T. K., Datta, A., Parui, S. K., & Chaudhuri, B. B. (2002). A hybrid scheme for handprinted numeral recognition based on a self-organizing network and MLP classifiers, *International Journal for Pattern Recognition and Artificial Intelligence,* 16(7), 845-864

Bhattacharya, U., Shridhar, M., & Parui, S. K. (2006). On recognition of handwritten Bangla characters, *In the Proceedings of ICVGIP,* 817-828

Bhattacharya, U., Shridhar, M., Parui, S. K., Sen, P. K., & Chaudhuri, B. B. (2012). Offline recognition of handwritten Bangla characters: an efficient two-stage approach, *Pattern Analysis and Applications,* 15(4), 445-458

Dutta, A., & Chaudhury, S. (1993). Bengali alpha-numeric character recognition using curvature features, *Pattern Recognition,* 26(12), 1757-1770

Kumar, M., Sharma, R. K., & Jindal, M. K. (2011a). SVM based offline handwritten Gurmukhi Character Recognition, *In the Proceedings of SCAKD*-2011, 52-63

Kumar, M., Jindal, M. K., & Sharma, R. K. (2011b). Offline handwritten Gurmukhi character recognition using curvature feature, *In the Proceedings of AMOC,* 981-989

Pal, U., Wakabayashi, T., & Kimura, F. (2007a). Handwritten Bangla compound character recognition using gradient feature, *In the Proceedings of 10th ICIT,* 208-213

Pal, U., Sharma, N., Wakabayashi, T., & Kimura, F. (2007b). Off-line handwritten character recognition of Devanagari script, *In the Proceedings of 9th ICDAR,* 446-450

Pal, U., Wakabayashi, T., & Kimura, F. (2007c). A system for off-line Oriya handwritten character recognition using curvature feature, *In the Proceedings of 10th ICIT,* 227-229

Rajashekararadhya, S. V., & Ranjan, P. V. (2008). Efficient zone based feature extraction algorithm for handwritten numeral recognition of four popular south Indian scripts, *Journal of Theoretical and Applied Information Technology, 4 (12),* 1171-1181

Rampalli, R., & Ramakrishnan, A. G. (2011). Fusion of complementary online and offline strategies for recognition of handwritten Kannada characters, *Journal of Universal Computer Science,* 17, 81-93

Roy, K., Vajda, S., Pal, U., & Chaudhuri, B. B. (2004). A system towards Indian postal automation, *In the Proceedings of 9th IWFHR,* 361-367

Sharma, N., Pal, U., Kimura, F., & Pal, S. (2006). Recognition of off-line handwritten Devanagari characters using quadratic classifier, *In the Proceedings of ICVGIP,* 805-816

Sharma, A., Kumar, R., & Sharma, R. K. (2008). Online handwritten Gurmukhi character recognition using elastic matching, *In the Proceedings of Congress on Image and Signal Processing,* 391-396

Sharma, D. V., & Jhajj, P. (2010). Recognition of isolated handwritten characters in *Gurmukhi script, International Journal of Computer Applications,* 4(8), 9-17

Desai, Apurva A,(2010). Gujarati handwritten numeral optical character reorganization through neural network, *Journal of Pattern Recognition Society,* 2582-2589.

❋❋❋