7-2020

A Self Controlled RDP Approach for Feature Extraction in Online Handwriting Recognition Using Deep Learning

Sukhdeep Singh
Boise State University

Vinod Kumar Chauhan
University of Cambridge

Elisa H. Barney Smith
Boise State University

This is an author-produced, peer-reviewed version of this article. The final, definitive version of this document can be found online at *Applied Intelligence*, published by Springer. Copyright restrictions may apply. https://doi.org/10.1007/s10489-020-01632-4

A Self Controlled RDP Approach for Feature Extraction in Online Handwriting Recognition using Deep Learning

Sukhdeep Singh

Department of Electrical and Computer Engineering, Boise State University, Boise, Idaho, USA

Home-page: https://sites.google.com/site/sransingh13/

E-mail: sukha13@ymail.com

Vinod Kumar Chauhan

Department of Engineering, University of Cambridge, Cambridge, United Kingdom

Home-page: http://www.eng.cam.ac.uk/profiles/vk359/

E-mail: vk359@eng.cam.ac.uk

Elisa H. Barney Smith

Department of Electrical and Computer Engineering, Boise State University, Boise, Idaho, USA

Home-page: https://coen.boisestate.edu/EBarneySmith/

E-mail: ebarneysmith@boisestate.edu

Submitted to: Applied Intelligence 2020

Abstract

The identification of accurate features is the initial task for benchmarked handwriting recognition. For handwriting recognition, the objective of feature computation is to find those characteristics of a handwritten stroke that depict the class of a stroke and make it separable from the rest of the stroke classes. The present study proposes a feature extraction technique for online handwritten strokes based on a self controlled Ramer-Douglas-Peucker (RDP) algorithm. This novel approach prepares a smaller length feature vector for different shaped online handwritten strokes without preprocessing and without any control parameter to RDP. Thus, it also overcomes the shortcomings of the traditional chain code based feature extraction approach that requires preprocessing of data, and the original RDP algorithm that requires a control parameter as an input to RDP. We further propose a deep learning network of 1-dimensional convolutional neural networks (Conv1Ds) for recognition, which trains in few minutes due to the smaller dimension of the convolution combined with smaller length feature vectors. The proposed approach can be applied to different scripts and different writing styles. The key aim of the present study is to provide a script independent feature extraction technique that is well suited for smaller devices. It improves the recognition over the best reported accuracy in the literature which was achieved using hidden Markov models with directional features, from 87.67% to 95.61% on a Gurmukhi dataset. For Unipen online handwriting datasets the results are at par with the literature.

Keywords: Online Handwriting Recognition · Feature Extraction · Ramer- Douglas-Peucker · Gurmukhi · Unipen · Deep Learning

1 Introduction

The rapid advancement in technology has resulted in increased use of devices like Tablet PDAs, PCs, mobile phones and pen tablets. The support of these devices for free flow natural handwriting has also made it possible to get the attention of worldwide researchers for online handwriting recognition (OHWR). OHWR is the process where the time sequential pattern of pen movement in the trajectory of writing is decoded in the same way as speech recognition. Great results have been achieved by OHWR researchers with different feature extraction [1–12], classification and recognition techniques [13–20]. Recently, much research has been carried out on OHWR field thanks to the evolution of data capture technologies [21–25]. Though a lot has been done for OHWR, methods for script independent and varied writing style studies needs more attention from pattern recognition researchers.

Whatever recognition technique is used, the computation of correct features is the key aspect for a successful OHWR system. Because the feature extraction in OHWR is the representation of strokes or characters individually, it is a vital stage in OHWR that affects the recognition accuracy greatly. In the present study, we solve the feature extraction problem by modifying the traditional chain code directional feature vector by detecting key points of the writing trajectory for OHWR, where key points act as representative points of strokes for OHWR. The proposed straight line directional feature extraction based on key points in a handwriting trajectory is motivated by the original Ramer-Douglas-Peucker (RDP) [26] algorithm and it is a modification of the traditional chain code based feature extraction approach. The RDP technique plots a new curve similar in shape to the original curve using fewer points than the original curve. RDP has been successfully used in key point detection in digital images and it can also be used for OHWR. The RDP algorithm is controlled by a parameter epsilon (E) and choosing the epsilon control parameter value can be a difficult task for different writing styles and different length strokes. The proposed technique calculates the control parameter value dynamically.

As consistent and reliable handwriting recognition is highly significant for real life applications, the present study focuses on proposing a script independent and highly reliable feature extraction technique for online handwriting. The proposed scheme finds the self controlled RDP based key points in an online handwriting trajectory and builds the smaller length feature vector by calculating the straight line directions between the key points. Most of the existing OHWR techniques use all the coordinate points of a trajectory for feature extraction. The key novelty of the proposed scheme is that it recognizes individual units using a smaller length feature vector without preprocessing. As deep learning is being used extensively in pattern recognition and image processing these days, we have also validated the proposed scheme using deep learning. Further, two different benchmarked datasets Unipen with Latin dig- its [27] and dataset of Gurmukhi strokes [28] have been used for validation of the proposed algorithm. Since the smaller length feature vector produces less time-space complexity, this work will be very useful for OHWR in small and low performance devices.

2 Literature Review

In previous years pattern recognition researchers have tried to express the OHWR trajectory in a variety of ways. They have expressed the pen tip movement using key points of the trajectory and calculating the straight/curve directions between these points. In this section, we review the related work for finding representative points in digital shapes. The methods available for finding key points are divided based on the approach used or a control parameter.

In addition to the original RDP, the two other primary techniques for key point detection in digital shapes are provided by Masood [30] and Carmona- Poyato et al. [31]. All these methods need an explicit value of the control parameter. Other methods for digital curve key point detection have also been developed. Wu [32] considered the key point detection approach vital for object measurement and recognition, and proposed an adaptive method for polygonal approximation of a digitized curve. A simple and iterative method for key point detection has been proposed by Kolesnikov and Franti [33] which works in three phases and uses reduced-search dynamic programming. A key point detection algorithm without a control parameter has been proposed by Marji and Siy [34], where they applied it to digital shapes of different sizes. The work to find the optimal approximation for closed contours has been done by Kolesnikov and Franti [35] where they proposed a method based on dynamic programming. Masood and Haq [36] evaluated the 21st century algorithms for key points detection and concluded that high quality approximation is ensured by stabilization algorithms, and they proposed an iterative algorithm for the same. A technique for digital shape polynomial approximation was proposed by Bhowmick and Bhattacharya [37]. Two different sequential techniques based on discrete geometry have been proposed by Ngyuen and Debled-Rennesson [38], and these techniques are also applicable to disconnected and noisy curves. Prasad et al. [29] proposed a framework to make existing key point detection approaches non-parametric. In their work, they tried to modify the existing RDP [26] and other techniques proposed by Masood [30] and Carmona-Poyato et al. [31] for key point detection.

As OHWR has many commonalities to digital shape and curve identification, their key point detection schemes are also suitable for OHWR. Studies that use a key point detection scheme for OHWR are very limited. Only a few pattern recognition researchers have contributed in this direction. A method using chain code directions based on key points in the online handwriting trajectory was used by Li and Yeung [39]. In their work, they recognized alphanumeric characters. Chinese character OHWR based on extending the work of four directional features to eight directional features was done by Bai and Huo [40]. Liu and Zhou [41] used directional features and trajectory based normalization for Japanese character recognition in online handwriting. Handwritten symbols were recognized based on dominant points by LaViola and Zeleznik [42], and Delaye and Anquetil [43]. In the available literature, no work for OHWR using RDP algorithm key points has been found. The proposed self controlled RDP scheme not only finds a reduced number of key points in an online handwriting trajectory, it does so without an explicit control parameter value and without preprocessing. The smaller length feature vector makes OHWR require less resources when using deep learning.

Nowadays, deep learning based models have achieved great success in online handwriting recognition, specifically convolutional neural networks (CNN) [44], recurrent neural networks (RNN) [45] and long short term memory (LSTM)

[46] which is a special case of RNN. Recently, their hybrid architectures have also been studied for handwriting recognition [47]. CNNs are useful for capturing the spatial relationship in the data and LSTM/RNN uses memory units to keep the memory of the previous state that makes it suitable to work with time series data and sequential data. [46] describes that the Google's online handwriting

recognition system deployed in different Google products is based on LSTM architectures. [48] studies CNN-RNN based hybrid architecture for unconstrained line and word recognition of offline data. [47] studies CNN-LSTM based hybrid architecture for character recognition and OCR using teacher- student learning approach for offline data. Both the previous architectures are computationally expensive due to the use of 2-dimensional CNNs (2D-CNN), as discussed in [49] which proposed a novel Conv1D based architecture for online handwriting recognition of Chinese characters. The key novelties of the proposed scheme are:

- The proposed scheme is easily applicable to different character sets and different writing styles.
- The proposed scheme does not require an explicit control parameter value.
- The RDP based key points detection scheme is very fast, so it is well suited for real time devices.
- The proposed scheme works without preprocessing of data and the recognition task can be performed with less time complexity.
- The smaller length feature vector prepared with the proposed approach will be very useful for low performance devices such as smart phones.
- The proposed approach is script independent.
- The proposed work can be enhanced for OHWR of larger units such as words.
- This work can also be extended for segmentation.
- The proposed deep learning model for recognition is computationally efficient due to the use of a Conv1D based network, which results in faster training and can be trained on a single machine without GPUs.
- Smaller feature length vectors combined with a computationally efficient network lead to faster training of the model in a few minutes.

3 Proposed Scheme

The objective of this work is to propose a scheme that finds key points for online handwritten strokes/characters and then uses these key points as a feature vector in OHWR using deep learning. The proposed key point detection scheme uses the same fundamental philosophy of the original RDP [26] algorithm, but it does not require an explicit control parameter value. The self controlled RDP scheme calculates the new curve from the existing curve and the shape of the new curve resembles the original curve, but it has fewer points than the original curve. The proposed algorithm is given in Algorithm 1.

Algorithm 1 ReducedPointList (P_{xy})

Input: P_{xy} is the point list that contains all points of the original input stroke. **Output:** R_{xy} is the reduced point list. R_{xy} represent the same curve that the P_{xy} points represent, but R_{xy} has fewer points than P_{xy}.

1: **if** ControlV *alue* does not exist **then**
2: Difference$_x$ ← $Maximum_x$ − $Minimum_x$
3: Difference$_y$ ← $Maximum_y$ − $Minimum_y$
4: Global $ControlV\ alue \leftarrow (Difference_x + Difference_y)/20$
5: **end if**
6: N ← Number of points in P_{xy}
7: **for** $i \leftarrow 2$ to $N-1$ **do**
8: d ← Calculate perpendicular distance between the point $P_{xy}[i]$ and straight line joining $P_{xy}[1]$ and $P_{xy}[N]$ points
9: **if** d > max **then**
10: ind ← i

    dmax ← d
    end if
end for
if dmax > *ControlValue* **then**
    $SubList1_{xy} \leftarrow P_{xy}[1\ to\ ind]$
    $SubList2_{xy} \leftarrow P_{xy}[ind\ to\ N]$
  $R1_{xy} \leftarrow$ ReducedPointList($SubList1_{xy}$)
  $R2_{xy} \leftarrow$ ReducedPointList($SubList2_{xy}$)
  $end1 \leftarrow$ Number of points in $R1_{xy}$
  $end2 \leftarrow$ Number of points in $R2_{xy}$
  $R_{xy} \leftarrow (R1_{xy}[1\ to\ end1 - 1],\ R2_{xy}[1\ to\ end2])$
else
    $P_{xy} \leftarrow (P_{xy}[1],\ P_{xy}[N])$
end if
return R_{xy}

In Algorithm 1, *VariableName*$_{xy}$ denotes the stroke coordinate (x, y), and *VariableName*$_x$ and *VariableName*$_y$ represent the *x* and *y* coordinates of the specific variable, respectively. The Global variable *ControlValue* is initialized only once and its value is used in all recursive calls. Initially, all points of the original stroke S are given as an input to the reduced point list in Algorithm 1. In the beginning, the first and last points $P_{xy}[1]$ and $P_{xy}[N]$ are kept in the reduced point list. At every step, Algorithm 1 estimates a point sequence by a line segment from the first to the last point. The farthest point from the line segment is selected. If the distance is greater than or equal to the *ControlValue*, then the selected point is kept in the reduced point list and Algorithm 1 is called recursively for the two sub sequences before and after the selected point, otherwise the estimate is accepted. After all recursive steps of Algorithm 1, the list R_{xy} consists of all points of the reduced point list and $R_{xy} \subset P_{xy}$. A new stroke SRDP consisting of R_{xy} points is generated. SRDP has fewer points than P_{xy}, but its shape is similar. Fig. 1 shows the 107 original points and 12 RDP based key points of a sample Gurmukhi stroke. In this scenario, the number of points to be used for classification is reduced by a factor of 9.

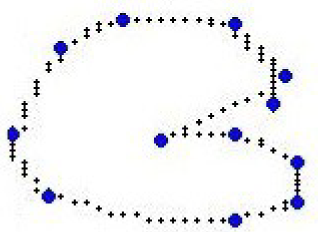

Fig. 1 A Gurmukhi stroke showing original points (black) and RDP points (blue)

4 Conv1D based Network

Convolutional neural networks (CNN), also known as ConvNet, are one of the most popular deep learning techniques, which specializes in capturing spatial and temporal relationship in the input data. Generally, CNN refers to 2-dimensional CNN (Conv2D) but it exists in two more variants, namely 1-dimensional CNN (Conv1D) and 3-dimensional CNN (Conv3D). Conv2D, which is suitable for image classification, has shown very good results for handwriting recognition [44] but they are heavily parameterized which are computationally expensive and need huge amount of memory, and Conv1D uses lesser number of parameters, requiring lesser memory and lesser computational power [49]. Unlike Conv2D, which can take hours and days for training of the model [47] [49], Conv1D can be trained in few minutes so can be trained on a single machine without GPUs. Thus Conv1D, which is suitable for problems like time series data, is a good choice for online handwriting recognition.

To recognize online handwriting we used a network, for which details about the architecture are displayed in Fig. 2. The network consists of three layers of Conv1Ds, two fully connected dense layers, including one output layer, two 1-dimensional max pooling (MaxPooling1D), three dropout layers and one flatten layer, as discussed:

Conv1D Layers: We have used Conv1Ds with kernel size of 3, padding 'same' and ReLU (Rectified Linear Unit) as the activation function which is the most commonly used activation function in deep networks, especially with CNNs. It helps to capture the spatial information in the sequence of coordinates of each character. These layers are computationally very efficient since they contain fewer parameters in comparison with the 2-dimensional CNN (Conv2Ds).

Maxpooling1D Layers: This layer helps to reduce the size of the model by half, where it subsamples by taking the maximum of two adjacent values in the CNN.

Dropout Layers: This layer helps to control the overfitting by dropping some of the inputs.

Dense Layers: These are fully connected layers, used as hidden layers and as the output layer with 64 neurons and neurons equal to the number of classes, respectively. The output layer uses softmax as the activation function for classification.

Layer (type)	Output Shape
cnn1 (Conv1D)	(None, 20, 32)
cnn2 (Conv1D)	(None, 20, 64)
drop1 (Dropout)	(None, 20, 64)
max_pool1 (MaxPooling1D)	(None, 10, 64)
cnn3 (Conv1D)	(None, 10, 128)
drop2 (Dropout)	(None, 10, 128)
max_pool2 (MaxPooling1D)	(None, 5, 128)
flat1 (Flatten)	(None, 640)
dense1 (Dense)	(None, 64)
drop3 (Dropout)	(None, 64)
dense2 (Dense)	(None, #classes)

Fig. 2 The Conv1D based architecture uses three Conv1Ds with Dropout and Maxpooling1D and two dense layers including the output layer. This diagram is generated using the Keras library.

One character represented by a padded sequence of length 20, containing *(x_i, y_i)* coordinates obtained from Algorithm 1, after z-score normalization acts as the input layer of dimension (20, 2) to the first Conv1D layer (cnn1) with 32 filters. The output of cnn1 is passed to the second Conv1D layer (cnn2) with 64 levels and then we apply the MaxPooling1D layer (max pooling1) which reduces the parameters by half. This is followed by a dropout layer (drop1) with 10% drop rate, because CNNs learn quickly and can lead to overfitting. The resulting output is then fed to a third Conv1D layer (cnn3) with 128 filters and then we again apply a MaxPooling1D layer (max pooling2) which again reduces the parameters by half, followed by a dropout layer (drop2) with 20% drop rate to control the training of models. Then the output is flattened to pass through the first fully connected dense layer (sense1) with 64 neurons. Then we apply another dropout layer (drop3) of 25% to avoid the overfitting and the output is passed to the second fully connected dense layer (dense2) which acts as the output and produces output as a one hot vector of length equal to the number of classes in the dataset. This architecture is computationally very efficient due to the use of Conv1Ds as compared with Conv2Ds, because the later takes a huge number of parameters in the training process, which is reflected by the empirical results.

5 Experiments

This section discusses the experimental setup and reports the results of the empirical study.

5.1 Experimental Setup

To validate the efficiency of the proposed scheme, the algorithm was tested on different datasets. In these experiments, we have used two benchmarked online handwritten datasets: Unipen for digits [50] and Gurmukhi stroke classes [28]. The Unipen digits dataset consists of different writers handwriting samples for all numeral classes 0-9. The online handwritten Gurmukhi strokes dataset was written in an online handwriting environment with contributions from 100 writers. The dataset of online handwritten Gurmukhi strokes used in our experiments include 34k samples of data for 62 classes. Each character/stroke is represented as a padded input of 20 sequences of (x_i, y_i) pairs and the data point uses padding of zero if there are less than 20 sequences of (x_i, y_i) pairs. This data is normalized using z-score normalization:

$$x_i = (x_i - \bar{x}) \ / \sigma_y, \quad y_i = (y_i - \bar{y}) / \sigma_y,$$

where $\bar{x}$, $\bar{y}$, and σ_y are means and standard deviation of x and y coordinates, respectively. This converts the data into data of mean 0 and standard deviation of 1. The experiments are performed using the Keras library with TensorFlow as a back-end, on a MacBook Air (a single machine without a GPU), with Core i5 processor and 8 GB RAM. We have used mini-batch size of 64 data points with Adam optimizer to solve the optimization problem due to its popularity and performance. This study focused on OHWR that maintains the consistency and reliability for different scripts and character sets. This has been validated with train and test data of different sizes, respectively. While performing experiments for each dataset, datasets were divided randomly into 90:10, 80:20, 70:30, 60:40 and 50:50 ratios for training and test, respectively, and results are reported as the average of five random runs in each train- test data size. Overall, fifty experiments are conducted in the present work as shown in Table 1, where 25 experiments were carried out for each dataset used in the present study.

This is an author-produced, peer-reviewed version of this article. The final, definitive version of this document can be found online at *Applied Intelligence*, published by Springer. Copyright restrictions may apply. https://doi.org/10.1007/s10489-020-01632-4

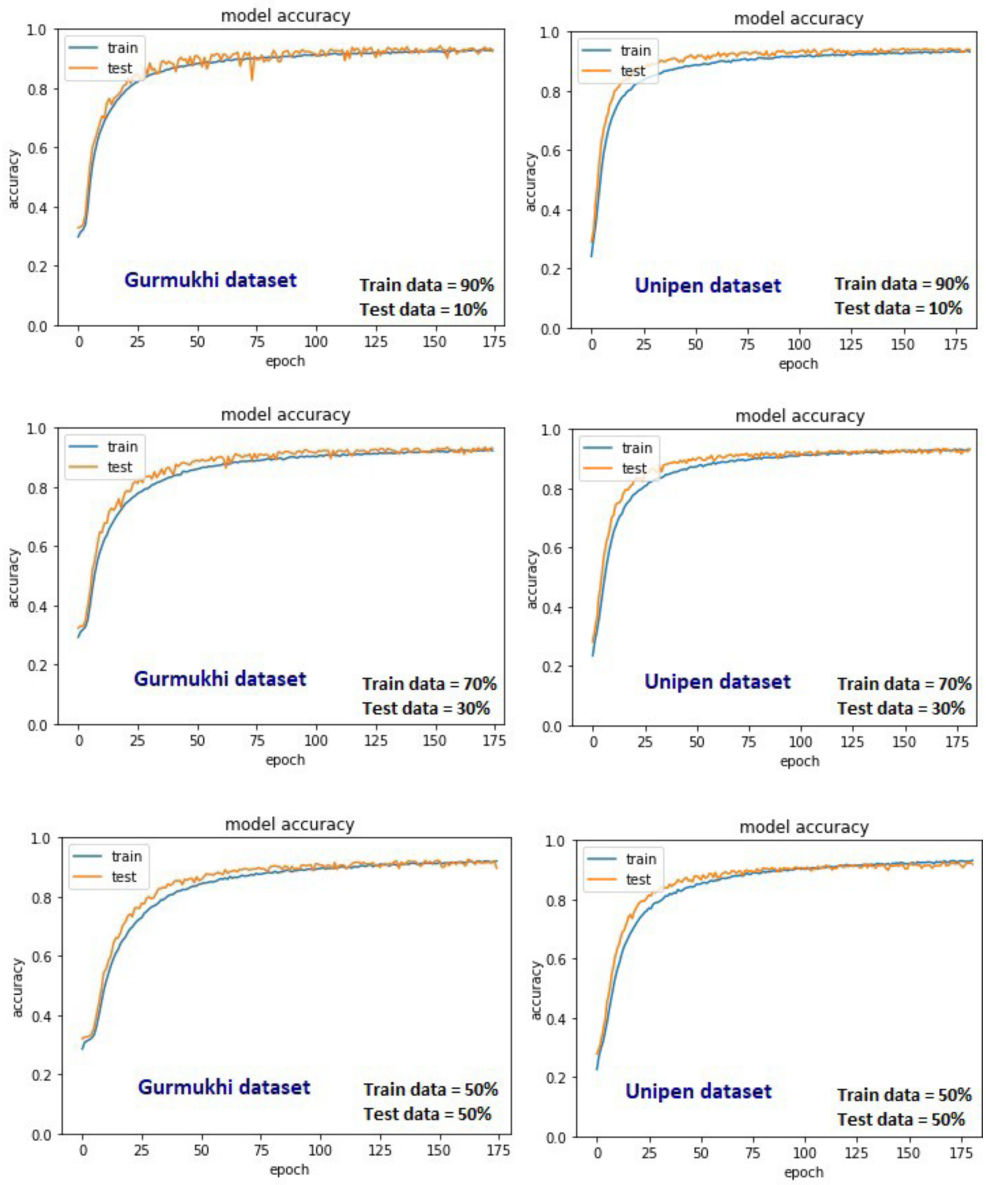

Fig. 3 Accuracy of the model is plotted for different train and test sizes of Gurmukhi and Unipen datasets, where the accuracy of each model is plotted as the average of five random runs. The consistent and close performance of the model on different train and test datasets show nice training of the model without any overfitting.

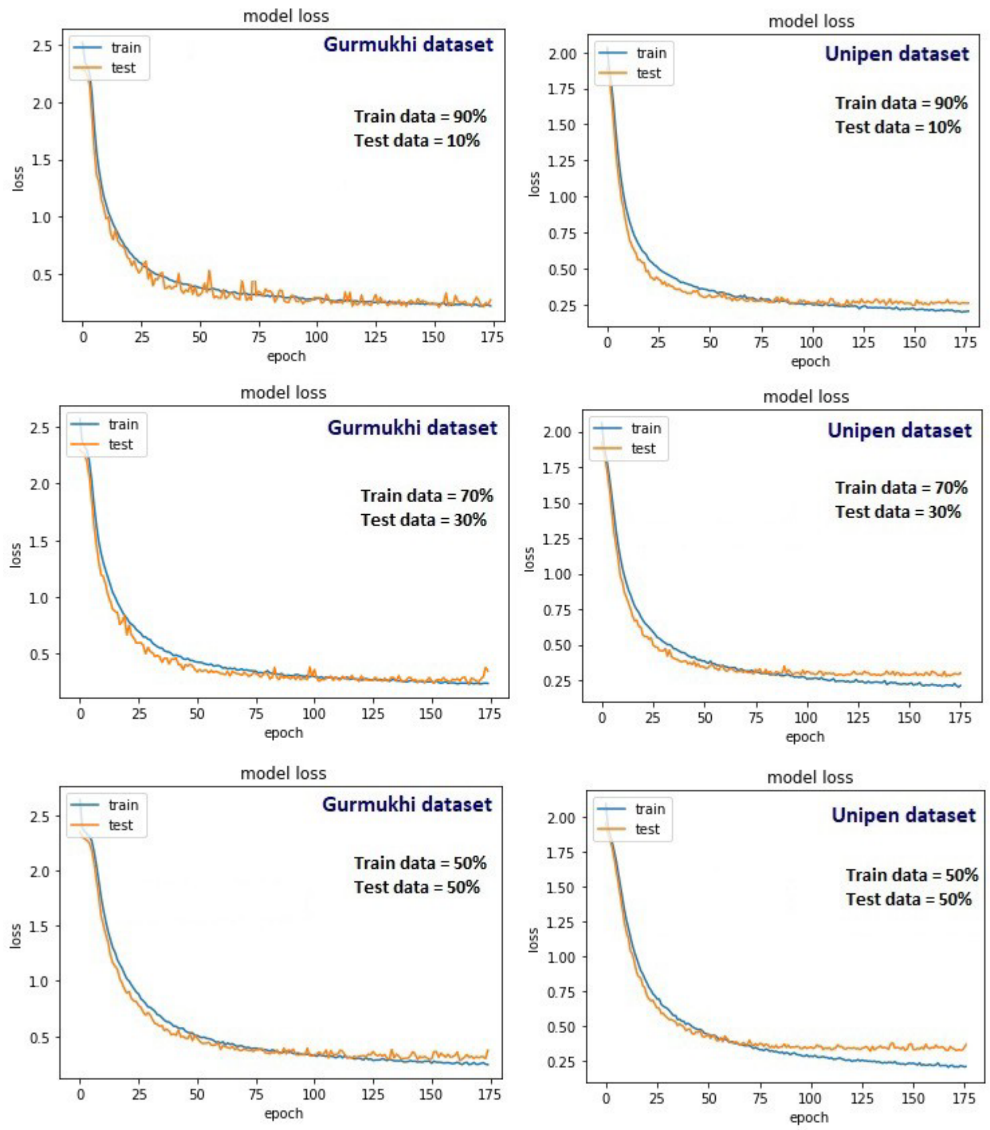

Fig. 4 Loss of the model is plotted for different train and test sizes of Gurmukhi and Unipen datasets, where the loss of each model is plotted as the average of five random runs. Similar to accuracy results in Figure 3, it shows nice training of the model without any overfitting.

5.2 Experimental Results

Fig. 3 and 4 depict analysis of the training of the model on the Gurmukhi dataset, and plots the accuracy and loss against the number of epochs. These figures, clearly indicate the model training has little overfitting, because the train and the test accuracies and losses follow each other closely. Moreover, it is also interesting to observe that initially the test accuracies are better than the train accuracies, i.e., the model has better generalization than its performance on the training data because of proper regularization of the model using dropout layers. As it is clear from these figures, the Adam optimizer hits the maximum results around epoch 100, so we can stop training at that point.

This is an author-produced, peer-reviewed version of this article. The final, definitive version of this document can be found online at *Applied Intelligence*, published by Springer. Copyright restrictions may apply. https://doi.org/10.1007/s10489-020-01632-4

Table 1 Test Accuracy using Conv1D Network

			#Experiment Results (5 Runs for each size)		
Train (%)	Test(%)	Dataset	Average accuracy (%)	Best accuracy (%)	Standard deviation
90	10				
		Gurmukhi	95.03	**95.61**	0.357
		Unipen	94.89	**95.39**	0.353
80	20				
		Gurmukhi	94.60	95.02	0.276
		Unipen	94.37	94.55	0.192
70	30				
		Gurmukhi	94.19	94.74	0.458
		Unipen	93.89	94.34	0.280
60	40				
		Gurmukhi	93.62	94.07	0.242
		Unipen	93.49	93.86	0.203
50	50				
		Gurmukhi	93.19	93.35	0.104
		Unipen	92.90	93.23	0.227

Moreover, the training of the model does not require GPUs and it is computationally very efficient due to the use of Conv1D, and can be trained on a single machine in a few minutes. For the Unipen dataset, each epoch takes only 3-4 seconds and the model hits its peak accuracy in around 100 epochs. That means, training of the deep learning model takes only 5-7 minutes for the Unipen dataset. Similarly, for the Gurmukhi dataset, each epoch takes only 8-9 seconds and the model hits the peak accuracy in around 100 epochs, i.e., the model training takes only 13-15 minutes, without using GPUs. This faster training of the model is attributed to two factors: First, smaller length feature vector, obtained from the proposed feature extraction technique which uses only key points, and second, computationally efficient network architecture based on the lower dimensionality of the Conv1D which takes significantly fewer training parameters as compared with Conv2D and as we know training will be faster if there are fewer parameters to learn.

Table 1 presents the test accuracy of the model with the proposed feature extraction scheme on the Unipen and Gurmukhi datasets. The proposed work achieved 95.61% as maximum recognition rate for the Gurmukhi dataset [28] [50] and for the Unipen digits dataset the best results attained are 95.39%. Our observation found that the benchmarked Gurmukhi dataset has a large number of stroke classes and some classes also have high degree of isomorphic nature. But this study achieved around 95% consistent recognition accuracy for different set of experiments in the Gurmukhi dataset, and these results have been obtained using a computationally less expensive system with very small length feature vector without preprocessing and postprocessing of data other than RDP. The evaluation of the proposed stroke and digit recognition scheme has taken the data independence criteria into consideration, which means the data samples used for training the recognizer are not included in the test data.

This is an author-produced, peer-reviewed version of this article. The final, definitive version of this document can be found online at *Applied Intelligence*, published by Springer. Copyright restrictions may apply. https://doi.org/10.1007/s10489-020-01632-4

Table 2 Comparison of Test Results for Conv1D Network with Different Features

Dataset	Train - Test Size (%)	Resampled data	Recognition Accuracy (%) RDP with fixed control value E=0.75	E=1	E=1.75	E=3.5	Algorithm 1
Gurmukhi							
	50-50	93.46	90.67	91.22	91.38	91.35	93.19
	60-40	92.98	91.20	91.79	92.01	91.96	93.62
	70-30	94.17	91.88	92.44	92.77	92.12	94.19
	80-20	94.39	92.08	92.90	93.01	92.62	94.60
	90-10	94.71	92.05	92.95	92.93	92.75	95.03
	Overall	93.94	91.58	92.26	92.42	92.16	94.13
Unipen							
	50-50	92.21	82.24	84.97	89.12	90.94	92.90
	60-40	92.58	83.10	85.98	90.16	91.53	93.49
	70-30	93.00	84.11	87.05	90.88	92.55	93.89
	80-20	93.46	84.50	87.58	91.70	93.07	94.37
	90-10	93.64	85.37	88.31	92.11	93.55	94.89
	Overall	92.98	83.86	86.78	90.79	92.33	93.91

Further, the effectiveness of using Algorithm 1 for feature extraction to recognize online handwriting with Conv1D is also validated versus the experiments performed for the same deep learning architecture with all points data and using the RDP scheme [26] with several fixed control values. While performing experiments with all data points with the Conv1D architecture, we have not used the raw/original data, because there is high degree of variability for number of points in original strokes/characters that requires significant zero padding for the feature data to be used with the same deep learning architecture, and it results in a lower recognition rate. To overcome the non-even spacing of the original data units as strokes and digits, we re-sampled the original data units to a fixed number of points, and then used it directly with the Conv1D based network for recognition. The results in Table 2 demonstrate that using Algorithm 1 outperforms both these two scenarios. Table 2 also depicts that the different static values of the control parameter E for the RDP algorithm give different results for Unipen and Gurmukhi datasets, it is not possible to decide the optimum static value of E that is applicable for different datasets. Thus the dynamically decided control parameter value E in Algorithm 1 avoids this problem. Further the results obtained with resampled data are close to the Algorithm 1 results, but these results are obtained after preprocessing of the Gurmukhi and Unipen data, and the feature vector length is also 200 which is 10 times the Algorithm 1 feature vector length.

5.3 Comparative study

The major objective of the present study is not to recognize online handwriting with a best accuracy. Instead, the present study focuses on OHWR that maintains the consistency and reliability for different scripts. We are proposing a novel self controlled RDP based scheme that produces a smaller length feature vector for OHWR, which makes the CNN computationally less expensive. For validation of the proposed feature extraction and Conv1D based network, we compared our recognition results for the Gurmukhi and Unipen datasets with the literature. We noted that the results for the Gurmukhi dataset (95.61%) are 6% to 7% higher than the best past results [28] [50] and the best accuracy attained for the more widely studied Unipen numerals dataset in the present study is also at par with the literature, but the present study achieved this recognition rate using the smaller length

Table 3 Recognition rates for Gurmukhi and numerals dataset of Unipen

Dataset	Train Size (%)	Accuracy Rate (%)	Methods/Comments/References
Gurmukhi			
	80	84.33	HMM; Chain code features [28]
	66	85.47	HMM; Chain code features [28]
	50	87.10	HMM; Chain code features [28]
	90	86.78	HMM; DPFE features [50]
	90	87.71	SVM; DPFE features [50]
	80	86.80	HMM; DPFE features [50]
	80	87.67	SVM; DPFE features [50]
	70	85.18	HMM; DPFE features [50]
	70	86.80	SVM; DPFE features [50]
	90	95.61	Conv1D; Self controlled RDP; this paper
	80	95.02	Conv1D; Self controlled RDP; this paper
	70	94.76	Conv1D; Self controlled RDP; this paper
	60	94.07	Conv1D; Self controlled RDP; this paper
	50	93.35	Conv1D; Self controlled RDP; this paper
Unipen			
	21	91.8	HMM; R01/V05 [51]
	50	98.8	KNN [52]
	63	96.8	HMM; R01/V06 cleaned [53]
	75	94.8	MLP; Regional-Fuzzy Representation (RFR) [1]
	75	97	MLP; RFR + DI [1]
	66	87.8	Genetic programming [54]
	40	96.8	HMM-SDTW [55] [56]
	40	96.2	DAG-SVM-GTDW [56]
	20	95.5	HMM-SDTW [55] [56]
	20	96	DAG-SVM-GTDW [56]
	33	98.8	OnSNT [57]
	20	98.7	OnSNT [57]
	50	98.8	OnSNT [57]
	66	98.9	OnSNT [57]
	90	98.9	OnSNT [57]
	67	97.10	CSDTW [58]
	66	97.95	Elastic matching; Quadratic Discrimination [59]
	66	94.78	SVM; Beta–elliptic approach [60]
	33	98.8	SVM; K_{KL} [61]
	33	94.8	Generative Systems; K_{KL} [61]
	33	94.8	Generative Systems; K_{φ} [61]
	33	94.9	Generative Systems; K_{FS} [61]
	67	97.1	Cluster generative Statistical Dynamic Time Warping [62]
		97	Multi CNN [63]
	90	97.3	AdaBoost; Global features [64]
	90	95.39	Conv1D; Self controlled RDP; this paper
	80	94.55	Conv1D; Self controlled RDP; this paper
	70	94.34	Conv1D; Self controlled RDP; this paper
	60	93.86	Conv1D; Self controlled RDP; this paper
	50	93.23	Conv1D; Self controlled RDP; this paper

feature vector and is computationally much less expensive a 2D-CNN. This accuracy rate can be further improved using post-processing techniques as we have not done any preprocessing or post-processing in the present study. As the Unipen dataset has been extensively used in the previous studies, and it is hard to present all available results here, so a selective part of the available studies for recognition of numerals in Unipen dataset is given here for comparison. Table 3 presents the comparison of results for the Gurmukhi and Unipen datasets in writer-independent circumstances. Our recognition results achieved on all set of experiments for Gurmukhi and Unipen datasets suit well for comparison with past results. Further, the self controlled RDP technique results are attained without huge preprocessing and post-processing of data, and good recognition accuracy is achieved with a smaller length feature vector. Recently, in 2017 Verma and Sharma [65] recognized online handwritten Gurmukhi characters based on zone and stroke identification, where they used SVM classifier for recognition and attained 74.8% overall character recognition accuracy. The Gurmukhi dataset used in this study was different from [28] dataset.

6 Conclusion

The present study proposes a novel approach for OHWR to build a RDP point based smaller size feature vector which is suitable to be used with deep learning. The study employs a Conv1D based network which trains in a few minutes on a single machine without any GPUs due to the use of Conv1Ds, and shows that the proposed approach is consistent and reliable, and we are able to attain 94.51% and 94.55% recognition accuracy for the Gurmukhi and Unipen datasets, respectively. The proposed approach is script independent and does not require any preprocessing of data, and it is very useful for low performance devices such as smart phones having memory constraints. This work can be further extended for other applications where tracing of a trajectory is needed. Overall, the present study shows the use of employing key points of the trajectory for OHWR with deep learning.

References

1. M. Parizeau, A. Lemieux, and C. Gagne. Character recognition experiments using unipen data. In Proceedings of Sixth International Conference on Document Analysis and Recognition, pages 481–485, Sep. 2001.

2. S. Jaeger, S. Manke, J. Reichert, and A. Waibel. Online handwriting recognition: the npen++ recognizer. Intl. Journal Document Analysis and Recognition 3:169-180. IJDAR, 3:169–180, 03 2001.

3. B. Chakraborty and G. Chakraborty. A new feature extraction technique for on-line recognition of handwritten alphanumeric characters. Information Sciences, 148(1):55 – 70, 2002.

4. M. Hanmandlu, K. M. Mohan, S. Chakraborty, S. Goyal, and D. Choudhury. Unconstrained handwritten character recognition based on fuzzy logic. Pattern Recognition, 36(3):603 – 623, 2003.

5. G. Abandah and T. M Malas. Feature selection for recognizing handwritten Arabic letters. Dirasat Engineering Sciences Journal, 37, 01 2010.

6. D. A. Al-Taani and S. Al-Haj. Recognition of on-line Arabic handwritten characters using structural features. Journal of Pattern Recognition Research, 5:23–37, 01 2010.

7. V. Ghods and E. Kabir. Feature extraction for online Farsi characters. In 2010 12th International Conference on Frontiers in Handwriting Recognition, pages 477–482, Nov 2010.

8. B. Verma, J. Lu, M. Ghosh, and R. Ghosh. A feature extraction technique for online handwriting recognition. volume 2, pages 1337 – 1341 vol.2, 08 2004.

9. R. Ghosh and P. P. Roy. A novel feature extraction approach for online Bengali and Devanagari character recognition. In 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), pages 483–488, Feb 2015.

10. M. Rani and Y. Meena. An efficient feature extraction method for handwritten character recognition. pages 302–309, 12 2011.

11. G. A. Fink, S. Vajda, U. Bhattacharya, S. K. Parui, and B. B. Chaudhuri. Online Bangla word recognition using sub-stroke level features and hidden markov models. In 2010 12th International Conference on Frontiers in Handwriting Recognition, pages 393–398, Nov 2010.

12. T. Bluche, H. Ney, and C. Kermorvant. Feature extraction with convolutional neural networks for handwritten word recognition. Proceedings of the twelfth International Conference on Document Analysis and Recognition, pages 285–289, 2013.

13. C. Zanchettin, B. L. D. Bezerra, and W. W. Azevedo. A KNN-SVM hybrid model for cursive handwriting recognition. Proceedings of the International Joint Conference on Neural Networks, pages 1–8, 2012.

14. V. K. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear SVM: a review',' Artificial Intelligence Review, 2018. [Online]. Available: https://doi.org/10.1007/s10462-018-9614-6

15. S. Jaeger, S. Manke, J. Reichert, and A. Waibel. Online handwriting recognition: the npen++ recognizer. International Journal on Document Analysis and Recognition, 3:169–180, 2001.

16. D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Convolutional neural network committees for handwritten character classification. Proceedings of the International Conference on Document Analysis and Recognition, pages 1135 – 1139, 2011.

17. T. Artieres, S. Marukatat, and P. Gallinari. Online handwritten shape recognition using segmental hidden Markov models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(2):205–217, 2007.

18. S. Nebti and A. Boukerram. Handwritten characters recognition based on nature- inspired computing and neuro-evolution. Applied Intelligence, 38(2):146–159, Jun 2012.

19. S. A. Mahmoud and W. G. Al-Khatib. Recognition of Arabic (Indian) bank check digits using log-gabor filters. Applied Intelligence, 35(3):445–456, May 2010.

20. A. Bharath and S. Madhvanath. HMM-based lexicon-driven and lexicon-free word recognition for online handwritten Indic scripts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34:670–682, 2012.

21. G. Ren and V. Ganapathy. Recognition of online handwriting with variability on smart devices. In ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7605–7609, May 2019.

22. S. Mandal, S. R. Mahadeva Prasanna, and S. Sundaram. An improved discriminative region selection methodology for online handwriting recognition. International Journal on Document Analysis and Recognition (IJDAR), 22(1):1–14, 2019.

23. E. Bilgin-Tasdemir and B. Yanikoglu. A comparative study of delayed stroke handling approaches in online handwriting. International Journal on Document Analysis and Recognition (IJDAR), 22(1):15–28, 2018.

24. R. Zouari, H. Boubaker, and M. Kherallah. Multi-language online handwriting recognition based on beta-elliptic model and hybrid TDNN-SVM classifier. Multimedia Tools Appl., 78(9):12103–12123, May 2019.

25. V. Carbune, P. Gonnet, T. Deselaers, H. Rowley, A. Daryin, M. Calvo, L.-L. Wang, D. Keysers, S. Feuz, and P. Gervais. Fast multi-language LSTM-based online handwriting recognition. ArXiv, 02 2019.

26. D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The International Journal for Geographic Information and Geovisualization, 10(2):112–122, 1973.

27. I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. Unipen project of on-line data exchange and recognizer benchmarks. Proceedings of International Conference on Pattern Recognition, pages 29–33, 1994.

28. S. Singh, A. Sharma, and I. Chhabra. Online handwritten Gurmukhi strokes dataset based on minimal set of words. ACM Transactions on Asian and Low-Resource Language Information Processing, 16(1):1–20, 2016.

29. D. K. Prasad, M. K. H. Leung, C. Quek, and S. Y. Cho. A novel framework for making dominant point detection methods non-parametric. Image and Vision Computing, 30:843–859, 2012.

30. A. Masood. Dominant point detection by reverse polygonization of digital curves. Image and Vision Computing, 26(5):702–715, 2008.

31. A. Carmona-Poyato, F. J. Madrid-Cuevas, R. Medina-Carnicer, and R. Munoz-Salinas. Polygonal approximation of digital planar curves through break point suppression. Pattern Recognition, 43(1):14–25, 2010.

32. W. Y. Wu. An adaptive method for detecting dominant points. Patten Recognition, 36(10):2231–2237, 2003.

33. A. Kolesnikov and P. Franti. Reduced-search dynamic programming for approximation of polygonal curves. Patten Recognition, 24(14):2243–2254, 2003.

34. M. Marji and P. Siy. Polygonal representation of digital planar curves through dominant point detection a nonparametric algorithm. Pattern Recognition, 37(11):2113–2130, 2004.

35. A. Kolesnikov and P. Franti. Polygonal approximation of closed discrete curves. Patten Recognition, 40(4):1282–1293, 2007.

36. A. Masood and S. A. Haq. A novel approach to polygonal approximation of digital curves. Image and Vision Computing, 18(3):264–274, 2007.

37. P. Bhowmick and B. B. Bhattacharya. Fast polygonal approximation of digital curves using relaxed straightness properties. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(9):1590–1602, 2007.

38. T. P. Nguyen and I. Debled-Rennesson. A discrete geometry approach for dominant point detection. Patten Recognition, 44(1):32–44, 2011.

39. X. Li and D. Y. Yeung. Online handwritten alphanumeric character recognition using dominant points in strokes. Pattern Recognition, 30(1):31–44, 1997.

40. Z. Bai and Q. Huo. A study on the use of 8 directional features for online handwritten Chinese character recognition. Proceedings of the International Conference on Document Analysis and Recognition, pages 262–266, 2005.

41. C. L. Liu and X. D. Zhou. Online Japanese character recognition using trajectory based normalization and direction feature extraction. Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pages 217–222, 2006.

42. J. LaViola and R. Zeleznik. A practical approach for writer-dependent symbol recognition using a writer-independent symbol recognizer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29:1917–1926, 2007.

43. A. Delaye and E. Anquetil. Hbf49 feature set: A first unified baseline for online symbol recognition. Pattern Recognition, 46(1):117–130, 2013.

44. Z. Li, L. Jin, and S. Lai, "Rotation-free online handwritten Chinese character recognition using two-stage convolutional neural network," in 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Aug 2018, pp. 205–210.

45. X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing Chinese characters with recurrent neural network," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 849–862, 2018.

46. D. Keysers, T. Deselaers, H. A. Rowley, L. Wang, and V. Carbune, "Multi-language online handwriting recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1180–1194, June 2017.

47. H. Ding, K. Chen, W. Hu, M. Cai, and Q. Huo, "Building compact CNN-DBLSTM based character models for handwriting recognition and OCR by teacher-student learning," in 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Aug 2018, pp. 139–144.

48. K. Dutta, P. Krishnan, M. Mathew, and C. V. Jawahar, "Improving CNN-RNN hybrid networks for handwriting recognition," in 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Aug 2018, pp. 80–85.

49. J. Gan, W. Wang, and K. Lu, "A new perspective: Recognizing online handwritten Chinese characters via 1-dimensional CNN," Information Sciences, vol. 478, pp. 375 – 390, 2019.

50. S. Singh, A. Sharma, and I. Chhabra, "A dominant points-based feature extraction approach to recognize online handwritten strokes," International Journal on Document Analysis and Recognition, vol. 20, no. 1, pp. 37–58, 2017.

51. X. Li, M. Parizeau, and R. Plamondon. Model-based on-line handwritten digit recognition. Proceedings of the Fourteenth International Conference on Pattern Recognition, 1998.

52. L. Prevost and M. Milgram. Modelizing character allographs in omni-scriptor frame: a new non-supervised clustering algorithm. Pattern Recognition Letters, pages 295–302, 2000.

53. J. Hu, S. G. Lim, and M. K. Brown. Writer independent on-line handwriting recognition using an HMM approach. Pattern Recognition, (33).

54. A. Teredesai, E. H. Ratzlaff, J. Subrahmonia, and V. Govindaraju. On-line digit recognition using off-line features. Indian Conference on Computer Vision, Graphics & Image Processing, 01 2002.

55. C. Bahlmann and H. Burkhardt. Measuring HMM similarity with the bayes probability of error and its application to online handwriting recognition. Proceedings of the Sixth International Conference on Document Analysis and Recognition, 2001.

56. C. Bahlmann, B. Haasdonk, and H. Burkhardt. On-line handwriting recognition with support vector machines a kernel approach. Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition, pages 49–54, 2002.

57. E. H. Ratzlaff. Methods, report and survey for the comparison of diverse isolated character recognition results on the unipen database. Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR), 1:623–628, 2003.

58. C. Bahlmann and H. Burkhardt. The writer independent online handwriting recognition system flog on hand and cluster generative statistical dynamic time warping. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(3):299–310, 2004.

59. H. Mitoma, S. Uchida, and H. Sakoe. Online character recognition based on elastic matching and quadratic discrimination. Proceedings of the Eighth International Conference on Document Analysis and Recognition, 2005.

60. M. Kherallah, L. Haddad, A. M. Alimi, and A. Mitiche. Online handwritten digit recognition based on trajectory and velocity modeling. Pattern Recognition Letters, 29(5):580 – 594, 2008.

61. T. M. T. Do and T. Artieres. Learning mixture models with support vector machines for sequence classification and segmentation. Pattern Recognition, 42(12):3224–3230, 2009.

62. H. Mirvaziri, M. Javidi, and N. Mansouri. Handwriting Recognition Algorithm in Different Languages: Survey. IVIC 2009: International Visual Informatics Conference, 487–497, 11 2009.

63. D. V. Pham. Online Handwriting Recognition Using Multi Convolution Neural Networks. SEAL 2012: Simulated Evolution and Learning, 310–319, 2012.

64. M. Mori, S. Uchida, and H. Sakano. Global feature for online character recognition. Pattern Recognition Letters, 35:142–148, 2014.

65. K. Verma and R. Kumar Sharma. Recognition of online handwritten Gurmukhi characters based on zone and stroke identification. Sadhana, 42, 05 2017.