Vietnam J Comput Sci (2015) 2:133–142
DOI 10.1007/s40595-014-0038-1

REGULAR PAPER

A combined static and dynamic feature extraction technique to recognize handwritten digits

Anuj Sharma

Received: 21 August 2014 / Accepted: 30 December 2014 / Published online: 21 January 2015

Abstract The recent advances in the feature extraction techniques in recognition of handwritten digits attract researchers to work in this area. The present study includes recognition of handwritten digits using hybrid feature extraction technique including static and dynamic properties of handwritten digit images. In this paper, static properties include number of non-zero (white) pixels in square, horizontal, vertical and diagonal styles as sub regions of a binary image. The dynamic properties include features from recovery of drawing order of original image. The extraction of dynamic features include two stages: first stage recover the drawing order of an image and second stage compute the chain code directions from recovered drawing order. The algorithm for recovery of drawing order uses properties of writing behavior. The support vector machine has been used as recognition method for the proposed feature extraction scheme. We have achieved an overall error rate of 0.73 % for mnist data set including 60,000 training images and 10,000 test images. Our feature extraction technique results in feature vector length of an image equals to 356. The achieved results strengthen our proposed technique usability as error rate achieved is at par with literature (<1 %) and the length of feature vector per image is small in comparison to input feature vector length of 784 which has been commonly used in previous work. The developed system is stable and useful in real life applications.

Keywords Offline and online handwriting · Recovery of drawing order · Chain codes · Support vector machine

A. Sharma (✉)
Department of Computer Science and Applications, Panjab University, Chandigarh, India
e-mail: anujs@pu.ac.in

1 Introduction

The recognition of scanned handwritten text is referred as offline handwriting recognition and the recognition of handwriting while writing is called online handwriting recognition. The present study has been experimented for scanned handwritten digit images. The recent advances in recognition of handwritten digits proves importance of this research topic and explain that alone recognition rate is not only important but also the stability and smoothness of developed system [2,25]. The present study is an effort to recognize digit images using their static and dynamic properties. The one example of static property is the number of non-zero (white) pixels in binary images. The dynamic property includes features represent dynamic information and we have extracted these features from offline images after applying technique as recovery of drawing order. Most of the literature is available for recognition of scanned handwritten digits using static properties. Some of the selected literatures in context of handwritten digits using static properties are discussed here. A recent study includes novel prototype generation technique to recognize handwritten digits in two stages. An adaptive resonance theory-based algorithm used in first stage for an initial solution and second stage solve optimization problem associated to classification [7]. A hybrid convolution neural network and support vector machine model that automatically retrieves features based on convolution neural network is a recent study to recognize scanned digits [25]. In this work, convolution neural networks works as a feature extractor and support vector machine as a recognizer. This way the designed hybrid model extracts features from raw images and performs classification. A study related to recognition of digit images include multicolumn deep convolution neural network that able to extract features from an image [2]. Similar convolution neural network used to extract features and train network with different number of input and hidden layer neurons [16]. A deep neural net and hidden markov model-based recognizer developed where deep convolution network learning is batch-mode based. The deep convolution network observed to be an impressive technique in context of cpu computation time and classification [4]. An adaboost learning algorithm used to recognize digit images achieved significant error rate less than 1 % [9]. A feature extractor was applied to digit images where number of coefficient of images is computed on learned basis and a local maximum operation is performed [10]. This includes support vector machine-based training with proposed feature vector and designed system was compared against techniques as sparse coding, gabor wavelets and principal component analysis.

In present study of features from dynamic properties, these features are extracted after recovery of drawing order of digits in first step and computation of its chain codes in second step. The selected literature for recovery of drawing order has been discussed here. An instance-based stroke recovery from multistroke characters after selecting sufficient number of instances to cover various characters [8]. A recovery scheme of drawing order of handwritten letters after generating several paths and the best path is selected [23]. A survey to recover pen trajectories multiple approaches to recover drawing order as skeleton, contours, ambiguous zones detections and ambiguous zones analysis are discussed [17]. A study based on matching cost function that is the summation of positional distortions cost and directional difference cost between the template stroke and its matching path. A bidirectional search algorithm based on dynamic programming has been used to find best path [22]. A three phase approach used to restore writing order where first phase include neural network to obtain possible edge continuity relations, second phase include double traced lines identification and last third phase find candidates of single stroked paths by depth first search [21]. In other study, a method proposed to extract strokes from handwritten characters and graphemes. This method includes three stages as: vectorisation, merging of skeletal branches with loop recovery and adjustment of near junctions [19]. A recent study includes recovery of drawing order of handwritten digits where chain code directions have been used to traverse pixels of a digit image [24]. In this paper, an iterative method has been presented to traverse all pixels of an image where a pixel is visited one time and directions of trajectory is based on chain code computations. The chain code is a linear structure that results from quantization of the centers of adjacent boundary elements in an image array [6,15]. The combined feature set of offline and online features have been used with recognition technique as support vector machine. The support vector machine is one of the popular classifier and has been used for recognition of digit images [3,12].

The present study main objective is to use combined feature vector using static and dynamic features for a stable system. This paper includes six sections including this section. The system overview is presented in section two. The section three discusses feature computation of preprocessed images and the support vector machine for two or more classes recognition problem have been discussed in Sect. 4. The Sect. 5 presents outcome of experiments and last Sect. 6 conclude this paper with findings.

2 System overview

The present system development include major components as preprocessing, feature extraction, SVM training and recognition. The developed model include two stages as stage I and stage II. These stages are:

stage I: train set → preprocessing → feature extraction → SVM training → trained model.

stage II: test set → preprocessing → feature extraction → Recognition using trained model → output classes of test set.

The stage I focus on classifier training using train set and stage II recognizes the test set. In stage I, train set images are preprocessed first where images are resized and transformed to binary form. The features are computed and each image results in a feature vector. The train feature set is now trained with SVM and results in trained model as output of stage I for all classes in train set. The stage II preprocess and compute features vectors of test set images as happened in stage I. The computed feature vectors of test set images are recognized with trained model obtained through stage I and results in identification of classes for each image of test set.

The preprocessing part of stages I and II are not challenging in view to see the complexities of two tasks as it includes size normalization and binarization of images. The feature extraction technique results in feature vector of reasonable small length that makes our system lighter and help in fast execution as compared to the case where feature extraction is not applied. The details of preprocessing and feature extraction including suitable examples are discussed in next section. The SVM training is the step which needs maximum time for execution among all components of system as it trains large benchmarked data set mnist with 60,000 images. Our feature extraction scheme reduces length of feature vector to less than half of input feature vector length and save training time. The SVM training is batch mode in nature and small length of feature vector allow space to large amount of train data. The recognition using trained model in step II find all results for test image with respect to trained classes and the class with maximum number of counts is the recognized class. There may be instances where more than one class could have same number of counts; we select class with minimum index as the recognized class.

3 Preprocessing and computation of features

As discussed in Sect. 1, we have computed two types of features, namely, static and dynamic features. This section presents the procedure of preprocessing of image and their feature computations. The preprocessing of an image and their computation of static and dynamic features are explained in respective Sects. 3.1, 3.2 and 3.3.

3.1 Preprocessing

The preprocessing of images includes two stages as size normalization and binarization. An input image is normalized to fixed size and converted to binary format where pixel value one presents white color and pixels with value zero are black in color. The size normalization includes rescaling the original image to the new desired size image. The original image pixel index is changed to the new image pixel index using rescaling the image. The image conversion to binary form include popular iterative threshold algorithm by [18]. The common steps of iterative threshold algorithm [18] include:

1. Find corners of image with image background and other are image pixels.
2. Compute means background and object gray-level.
3. Update background objects distinction.
4. Stop if updation of background object not needed, otherwise repeat step 2.

The Figs. 1 and 2 present original image and its conversion to size normalized binary image.

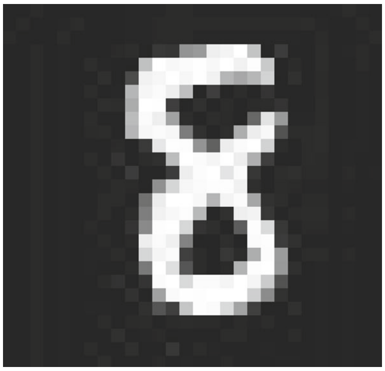


Fig. 1 Original image in gray scale with dimensions 28 × 28 pixels

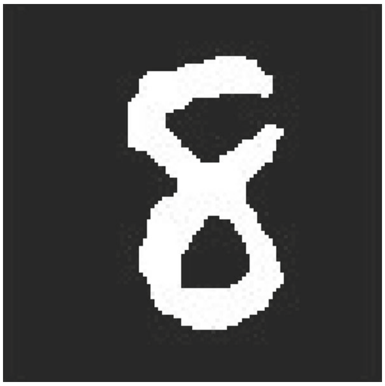


Fig. 2 Size normalized binary image with dimensions 100 × 100 pixels

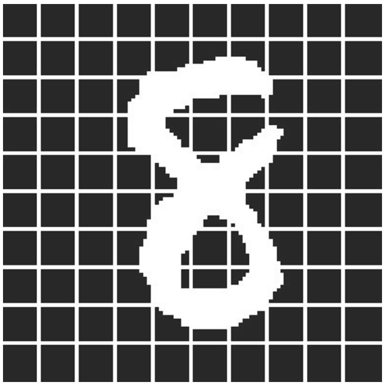


Fig. 3 Computing non-zero pixels from square regions in an image

3.2 Static features

The static features are referred in this paper to static information of an image. There are three set of features computed in this category that include computation of number of white pixels in different sub regions of an image. These three sets of static features are:

1. Image divided into equal sizes of squares as shown in Fig. 3.
2. Image divided into equal number of horizontal and vertical sizes as shown in Figs. 4 and 5.
3. Image divided into diagonal style left and right directions as shown in Figs. 6 and 7, respectively.

The numbers of white pixels are computed from three static features sets from above steps as 1, 2 and 3.

3.3 Dynamic features

The dynamic features includes chain code features from two set of images as skeleton and boundary pixel images, respectively. The size normalized binary image is converted to

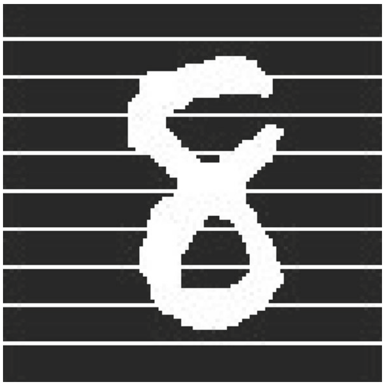


Fig. 4 Computing non-zero pixels from horizontal regions in an image

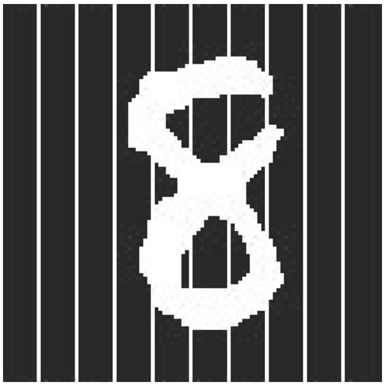


Fig. 5 Computing non-zero pixels from vertical regions in an image

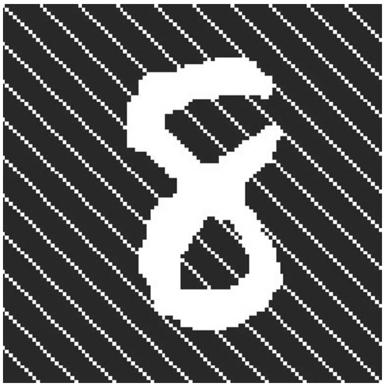


Fig. 6 Computing non-zero pixels from right-aligned diagonal regions in an image

skeleton as one image and boundary pixel image as other image. The Figs. 8 and 9 represents skeleton and boundary pixel images. The skeleton image refers to thinning of image and boundary pixel image store the boundary pixels of binary image [11]. To calculate chain code features, the drawing order of an image is computed in first step and second step include computation of chain codes. The assumptions play important role in drawing order recovery as mentioned in literature [24]. The assumptions in present study are:

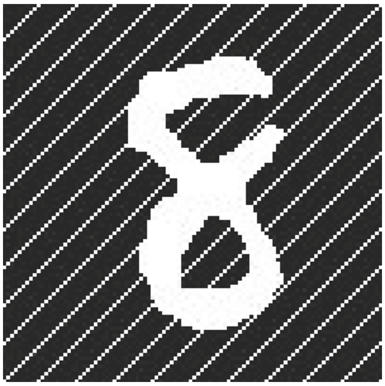


Fig. 7 Computing non-zero pixels from left-aligned diagonal regions in an image

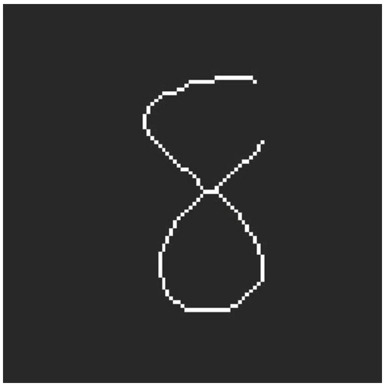


Fig. 8 Skeleton image from size normalized binary image

1. The sub regions close to origin of image are selected first.
2. The point in sub region with one neighbor and close to origin is selected first to traverse.
3. If no pixel is available with one neighbor as happen in case of loop, the pixel at left top position is first to traverse.
4. In case of more than two neighbor pixels to traverse, the direction close to previous pixel path is followed and respective pixel is selected iteratively.

These assumptions used to traverse pixels of image and applied recursively in an image till all non-zero pixels are not visited and each non-zero pixels should be visited once only. The Figs. 10 and 11 present traversal of pixels of images in Figs. 8 and 9, respectively. The chain code computation includes movement of next pixel and we have used eight directions as presented in Table 1. The points from recovery of drawing order of skeleton and boundary pixel images are further resampled so as to place them at equal distances from neighboring pixels. The details of recovery of drawing order in context of handwritten digits where chain code-based direction movement applied has been discussed in recent study [24]. The

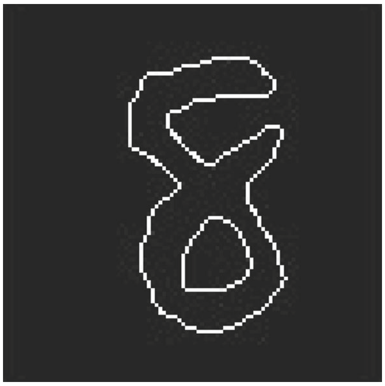


Fig. 9 Boundary pixel image from size normalized binary image

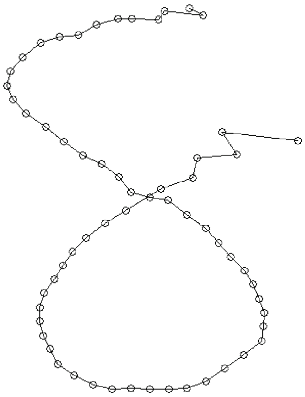


Fig. 10 Recovery of drawing order of skeleton image

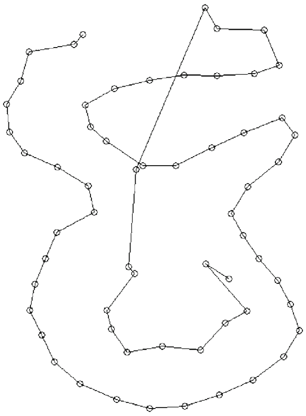


Fig. 11 Recovery of drawing order of boundary pixel image

following steps (1) to (5) explain the computation of static and dynamic features for an input image presented in Fig. 1.

1. The width and height of images after size normalization are fixed to 100 by 100 pixels. The selection of 100 by 100 pixels for size normalization achieve lowest error rate in our experiments as compared to selections of 28 by 28 pixels, 56 by 56 pixels, 84 by 84 pixels and 112 by 112 pixels. The Figs. 1 and 2 presents original image and its conversion to size normalized binary image.
2. There are three sets of static features computed as discussed in Sect. 3.2. The three sets are: (i) The pixels in squares area of an image are shown in Fig. 3. Here, the squares are of uniform size of ten by ten pixels and the number of white pixels is calculated. (ii) The white pixels in horizontal and vertical sub regions of an image are calculated as presented in Figs. 4 and 5. The horizontal and vertical sub regions are of equal area in our experiments. (iii) The white pixels in sub regions of an image of equal width and diagonally aligned as shown in Figs. 6 and 7. This makes offline feature vector of length 156 where first hundred are from squares sub regions, next eighteen are horizontal and vertical sub regions and rest thirty-eight are from diagonal sub regions.
3. The points are resampled to 61 and 131 for skeleton and boundary pixel images drawing order, respectively. We have experimented to resample from 41 to 91 for skeleton and 91 to 181 for boundary pixels images drawing order. The selection of 61 and 131 points for skeleton and boundary pixels images drawing order are made on the basis of lowest error rate achieved.
4. To compute chain codes for skeleton image drawing order with 61 points, a total of 60 chain codes are computed as two consecutive pixels form one chain code direction. We have selected occurrences of chain code directions ‘1’ to ‘8’ for consecutive twelve chain codes from length 60 that results in feature vector of length 48. The same procedure has been applied for boundary pixel image drawing order.
5. This chain code feature has been applied twice for skeleton and boundary pixel image drawing orders as: first find chain codes occurrences for all 61 and 131 points of skeleton and boundary pixel image drawing order; second, find chain code occurrences for chain codes derived from every odd pixel position from 61 and 131 points of skeleton and boundary pixel image drawing order.

Table 1 Chain code directions and their scope

Chain code	Scope in degrees
1	>337.5 or ≤ 22.5
2	>22.5 or ≤ 67.5
3	>67.5 or ≤ 112.5
4	>112.5 or ≤ 157.5
5	>157.5 or ≤ 202.5
6	>202.5 or ≤ 247.5
7	>247.5 or ≤ 292.5
8	>292.5 or ≤ 337.5

This makes combined feature vector of static and dynamic features for steps (2) and (5) as:
{5, 38, 40, 60, 73, 94, 100, 96, 57, 9, 62, 100, 100, 86, 53, 49, 40, 40, 45, 11, 100, 100, 61, 1, 0, 0, 0, 0, 27, 43, 41, 72, 100, 79, 27, 14, 50, 86, 98, 38, 0, 0, 21, 89, 100, 100, 100, 81, 15, 0, 0, 28, 86, 100, 100, 86, 100, 81, 10, 0, 2, 86, 100, 98, 40, 0, 9, 79, 77, 11, 28, 100, 100, 49, 0, 0, 0, 26, 100, 66, 0, 59, 100, 71, 50, 50, 59, 91, 100, 58, 0, 5, 40, 77, 99, 100, 93, 61, 31, 0, 572, 238, 586, 588, 332, 748, 605, 710, 506, 542, 591, 493, 502, 551, 469, 641, 638, 560, 588, 575, 351, 223, 219, 224, 203, 149, 27, 0, 590, 589, 559, 292, 224, 218, 225, 193, 131, 527, 506, 359, 249, 276, 306, 279, 195, 30, 0, 583, 514, 498, 316, 251, 279, 312, 265, 171, 0, 1, 1, 4, 6, 0, 0, 0, 2, 9, 1, 0, 0, 0, 0, 0, 0, 4, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0, 5, 2, 5, 0, 2, 0, 0, 0, 0, 1, 1, 8, 1, 5, 0, 3, 3, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 0, 1, 0, 0, 0, 0, 0, 0, 5, 2, 2, 2, 0, 5, 0, 1, 0, 0, 3, 2, 3, 4, 0, 0, 0, 4, 2, 0, 1, 0, 0, 0, 5, 0, 4, 5, 3, 0, 0, 0, 0, 0, 0, 4, 8, 0, 0, 0, 0, 0, 0, 0, 0, 5, 4, 3, 0, 1, 0, 0, 0, 0, 1, 7, 3, 0, 0, 0, 0, 0, 5, 7, 0, 5, 2, 1, 0, 0, 0, 1, 3, 1, 3, 3, 2, 3, 0, 0, 0, 2, 0, 0, 0, 2, 2, 4, 1, 1, 4, 0, 3, 4, 0, 0, 0, 2, 2, 3, 2, 0, 0, 0, 3, 0, 0, 2, 4, 2, 3, 1, 0, 0, 0, 0, 0, 0, 3, 7, 2, 3, 2, 2, 1, 2, 0, 0, 2, 1, 0, 0, 0, 1, 0, 2, 1}.

This feature vector is of length 356 where first 156 are offline feature vector and rest are online feature vector. In offline feature vector of length 156, first 100, next 18 and next 38 are derived from squares, horizontal and vertical, and diagonal sub regions of a binary image. In online features of length 200, The chain code occurrences for first 40, next 24, next 88 and next 48 are derived from skeleton image drawing order from 60 chain codes, 30 chain codes (odd position pixels), and from boundary pixels drawing order, 130 chain codes and 65 chain codes (odd pixels positions), respectively.

4 Recognition

The feature vector computed in previous section becomes input to recognition phase. The recognizer used in our work is SVM [26]. The SVM is one of the useful methods to classify linearly and non-linearly separable data. The technique of computation of support vectors has been explained in this section. For a given training data: The SVM solves labeled training data (x_i, y_i), $i = 1, 2, 3, \ldots n$ for the decision function $h(x) = sign(w.z{+}b)$, where $z = \phi_z(x) \in Z$ is a feature map of x [26]. The w and b are the solutions of:

$$\text{minimize } \frac{1}{2}(w.w) + C\sum_{i=1}^{n}\xi_i$$
$$s.t.\ \ y_i((w.z_i)+b) \geq 1-\xi_i,\ \ \xi_i \geq 0, i = 1, 2, \ldots n \qquad (1)$$

C is penalty parameter and ξ_i is slack variable. The SVM training is a large quadratic programming problem with computational complexity grows to $O(n^3)$. The computational complexity of its dual form is less as compared to primal formulation. The dual form of (1) derived through lagranges multipliers reduce the complexity of problem and dual form becomes:

$$\text{minimize } f(\alpha) = -\sum_{i=1}^{n}\alpha_i + \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j (z_i.z_j)$$
$$s.t.\ 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{n}\alpha_i y_i = 0 \text{ for } i = 1, 2, \ldots n \qquad (2)$$

Here, α_i is positive langrange multiplier. The decision function for dual form is: $h(x) = \text{sign}(\sum_{i=1}^{N}\alpha_i y_i z_i.z + b)$.

To classify non-linear separable data, a kernel trick is applied that takes data to classify in higher dimensional feature space $\phi_z(x)$. The decision function including kernel becomes: $h(x) = \text{sign}(\sum_{i=1}^{N}\alpha_i y_i K(z_i.z) + b)$ and $K(z_i.z_j) = \phi_z(x_i).\phi_z(x_j)$ is a valid kernel function that satisfy mercer condition. Some of the popular kernels are: (1) Linear, $K(z_i.z_j) = z_i.z_j$, (2) Gaussian, $K(z_i.z_j) = exp(-\gamma\|z_i - z_j\|^2)$, (3) Polynomial, $K(z_i.z_j) = (p + z_i.z_j)^q$, (4) Sigmoid, $K(z_i.z_j) = \tanh(k.z_i.z_j - \delta)$.

A large quadratic programming problem breaks into small quadratic problems with two variables at a time in sequential minimal optimization (SMO) algorithm proposed by [20]. The SMO algorithm common stages as: computation of working set of two variables and updating respective lagranges multipliers of current working set variables. These stages are repeatedly used until convergence level is achieved. We have used second order information to compute working set as proposed in literature by [5]. This second order information helps in faster execution of SMO algorithm and achieves minimum number of working sets. These working set pairs are referred as maximal violating pair where violation is subject to KKT conditions [1]. For a maximal violating pair $B(i, j)$:

$$\begin{aligned} i &\in \arg\max_t \left\{-y_t f(\alpha^k)_t | t \in I_{\text{up}}(\alpha^k)\right\} \\ j &\in \arg\min_t \left\{\text{Sub}(\{i,t\} | t \in I_{\text{low}}(\alpha^k),\right. \\ &\quad \left. -y_t \nabla f(\alpha^k)_t < -y_i \nabla f(\alpha^k)_i\right\} \end{aligned} \qquad (3)$$

$$\text{Sub}(\{i,t\} \equiv \nabla f(\alpha)^T \triangle\alpha + \tfrac{1}{2}\triangle\alpha^T \nabla^2 f(\alpha)\triangle\alpha,$$
$$\triangle\alpha_l = 0 \text{ for all } l \neq i, j.$$

This allows working set to checks only $O(n)$ possible pairs for selecting j. Let $a_{ij} = (z_i, z_i) + (z_j, z_j) - 2(z_i, z_j)$ and $b_{ij} = -y_i\nabla f(\alpha^k)_i + y_j\nabla f(\alpha^k)_j$, this makes working set selection as:

$$\begin{aligned} i &\in \arg\max_t \{-y_t f(\alpha^k)_t | t \in I_{\text{up}}(\alpha^k)\} \\ j &\in \arg\min_t \frac{-b_{ij}^2}{a_{ij}} | t \in I_{\text{low}}(\alpha^k), \\ &\quad -y_t\nabla f(\alpha^k)_t < -y_i\nabla f(\alpha^k)_i\} \end{aligned} \qquad (4)$$

Table 2 Class ids and their respective recognition results

Class	Images	Images recognized correctly for respective degrees						Avg rec per class (%)
		3	5	7	9	11	13	
0	980	975	977	977	975	975	975	99.58
1	1,135	1,126	1,128	1,129	1,128	1,128	1,128	99.37
2	1,032	1,020	1,026	1,026	1,026	1,026	1,025	99.30
3	1,010	999	1,002	1,003	1,004	1,004	1,004	99.27
4	982	969	974	974	974	972	972	99.03
5	892	885	887	888	888	887	886	99.42
6	958	945	954	951	951	951	950	99.20
7	1,028	1,014	1,019	1,020	1,020	1,019	1,020	99.09
8	974	951	963	964	964	960	959	98.58
9	1,009	987	994	995	996	991	990	98.33
	Average rec rate (%)	98.71	99.24	**99.27**	99.26	99.13	99.09	

Table 3 Confusion matrix for class id's recognized incorrectly for lowest error rate with poly degree as seven

Truth	Prediction									
	0	1	2	3	4	5	6	7	8	9
0							1		2	
1			2				1	2	1	
2	2				1			2	1	
3			2			1		2	2	
4							3		1	4
5	1			2			1			
6	2					1			4	
7		1	5		1					1
8	1	1	2	2		1		1		2
9					7	2		4	1	

The updated lagranges multiplier for i, j is as follows:

$$\alpha_i^{\text{new}} = \begin{cases} \alpha_i + \frac{y_i b_{ij}}{a_{ij}} & \text{if } a_{ij} > 0 \\ \alpha_i + \frac{y_i b_{ij}}{\tau} & \text{if } a_{ij} \leq 0 \end{cases}$$
$$\alpha_j^{\text{new}} = \begin{cases} \alpha_j - \frac{y_j b_{ij}}{a_{ij}} & \text{if } a_{ij} > 0 \\ \alpha_j - \frac{y_j b_{ij}}{\tau} & \text{if } a_{ij} \leq 0 \end{cases} \qquad (5)$$

The τ is a constant. Then, α_i^{new} and α_j^{new} need to be clipped in range [0,C] as discussed in SMO algorithm.

$$\alpha_i^{\text{new,clipped}} = \begin{cases} 0 & \text{if } \alpha_i^{\text{new}} < 0 \\ C & \text{if } \alpha_i^{\text{new}} < C \\ \alpha_i^{\text{new}} & \text{otherwise} \end{cases} \qquad (6)$$

before $\alpha_j^{\text{new,clipped}}$, the updated α_i^{new} need to use for α_j^{new} as:

$$\alpha_j^{\text{new}} = y_i \left(y_i \alpha_i + y_j \alpha_j - y_i \alpha_i^{\text{new,clipped}} \right) \qquad (7)$$

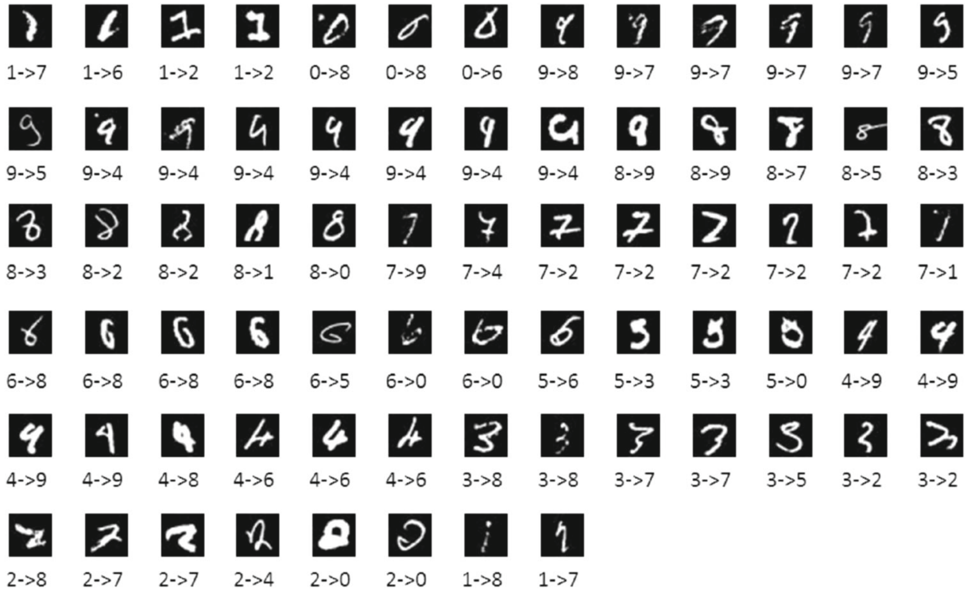
$$\alpha_j^{\text{new,clipped}} = \begin{cases} 0 & \text{if } \alpha_j^{\text{new}} < 0 \\ C & \text{if } \alpha_j^{\text{new}} < C \\ \alpha_j^{\text{new}} & \text{otherwise} \end{cases} \qquad (8)$$

The updated lagranges multipliers are multiplied with respective targets as '−1' and '+1' to either side of classification boundaries and final values of α's are referred as support vectors. To solve multiclass problem, one-against-one approach has been opted. In this approach for m class problem, it constructs $m(m-1)/2$ classifiers, where each classifier uses the training data from two classes chosen out of m classes. The present experiments uses 'max win' approach where a class is recognized that wins maximum votes. In any case, classes with same number of votes, the class with smallest index is accepted.

5 Experiments and results

The computed feature vector has been used with SVM recognition technique and experiments are conducted using benchmark dataset mnist [13]. The mnist dataset include 60,000 training images and 10,000 test images. The experiments are

Fig. 12 Seventy three test set images recognized incorrectly with their corresponding labels as x(true) → y(predicted)



conducted using scientific tools as matlab and libsvm [5]. Each image of the train and test set transformed to feature vector of length 356. The transformed train set undergoes SVM training and results in trained set with computed support vectors for individual classes. This trained set is used to recognize classes of transformed test set. The polynomial kernel used in SVM recognition method achieved better results as compared to other kernels as linear, rbf and sigmoid. The Table 2 presents results with polynomial kernel with degrees 3, 5, 7, 9, 11 and 13. The results show that degree 7 and 9 achieve better results as compared to degrees 3, 5, 11 and 13. The best recognition accuracy result achieved as 99.27 % (0.73 % as error rate) using polynomial degree 7. The column average recognition per class in Table 2 presents recognition accuracy for classes from 0 to 9 using kernel degrees 3, 5, 7, 9, 11 and 13. The average result for each class shows that the present system capability under different kernel degrees to behave stable. In literature, we have observed very low error rate for used data set mnist as 0.19 % [25]. Our prime focus is not to achieve lowest error rate for used data set mnist but to propose a technique with hybrid features where static and dynamic features could be used together. The proposed feature extraction technique is challenging in nature as it includes drawing order of an image where original drawing order is not known. Our reported error rate as 0.73 % give scope for future work to extend present study as reported error rate justify that proposed technique is promising. The confusion matrix for reported error rate as 0.73 % has been presented in Table 3 where class ‘9’ shows maximum number of incorrect predictions as ‘14’ and class ‘0’ shows minimum incorrect prediction as ‘3’. The images for result of confusion matrix have been presented in Fig. 12 with labels $x \rightarrow y$ for each result in image. The x presents true label and y presents predicted label. The incorrect recognized images vision to human recognition has been explained for mnist data set in literature [12]. We also find some of our images same for which incorrect recognition happened in literature.

Table 4 Offline, online and combined (offline and online) feature vector results

Feature	Error rate (%)
Offline	1.46
Online	1.78
Combined (Offline and Online)	0.73

As present study includes implementation of hybrid features comprising static and dynamic in nature, we have separately recognized mnist data set for static and dynamic features. These separate test results help in understanding of effectiveness of static and dynamic features exclusively. The Table 4 present results for static and dynamic features as 1.46 and 1.78 % error rates, respectively. These two error rates are higher as compare to result reported using static and dynamic features together. The error rate for handwritten digit recognition with mnist data set has been reported below one percent in recent study as referred in literature. We find that our scheme do not achieve minimum error rate as compared to literature but we are able to achieve error rate below 1 % that is at par with the work done in this area. Some of the selected previous results are presented in Table 5 where we find that our approach report error rate 0.73 % with feature vector length per image as 356. In Table 5, we have presented selective results of literature where mnist data set have been used. The mnist data set have been used extensively in literature. Therefore, it is not easy to present all results from literature. The progress made in recognition of mnist data set using different approaches is available in [14].

Table 5 Selected test result reported in literature using mnist dataset

References	Error rate (%)	Feature vector length
[7]	1.46	448
[25]	0.19	784
[2]	0.23	784
[16]	0.39	784
[4]	0.83	784
[9]	0.87	784
Our approach	0.73	356

Our results indicate that proposed hybrid feature technique results in smaller feature vector length as 356. The feature vector length as 356 is small as compared to 784 which has been commonly used in literature as presented in Table 5. The selection of small feature vector length makes our system lighter and capable to work better with machines having memory constraints. Also, our approach include technique of feature extraction of an image that presents uniqueness and novelty of our approach to understands images through both static and dynamic behavior.

6 Conclusion

As discussed in section one that objective of present study to provide a novel feature scheme that merges both static and dynamic handwriting features. We observed that our system is stable from study observed in experiments section and able to achieve error rate less than 1 % which is at par with the work done in this area. We have observed following important outcomes of study:

- The proposed technique include recovery of drawing order for dynamic features on the basis numerically proved assumptions and merge static features together for final feature vector as input to SVM classifier.
- The computed feature vector is smaller in length as compared to reported work previously and it also achieves error rate less than 1 % as happened in good selected results in literature and desired for real-life use applications.
- The classifier as SVM is batch mode in nature and input feature vector must satisfy memory constraints. We find the length of feature vector as 356 is less than half of 784 (commonly used previously).
- We have presented intra comparisons for static and dynamic features separately and the proposed method also compared to literature work in Table 5 under Sect. 5.

The work with combined feature scheme is new and need to be implemented for more complex systems where recovery of drawing order is a challenging task. The recovery of drawing order becomes more challenging in the cases where original drawing order is not known. The recovery of drawing order has attracted researchers in recent past as it has important applications in the field of biometric systems. The applications of this field could be applied to financial systems, signature verifications and documents recognition where static and dynamic information can be used together.

Acknowledgments Author is thankful to anonymous reviewers for their valuable suggestions to improve the manuscript.

References

1. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**, 121–167 (1998)
2. Ciresan, D.C., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: Computer Vision and Pattern Recognition, pp. 3642–3649 (2012)
3. Decoste, D., Schoelkopf, B.: Training invariants support vector machines. Mach. Learn. J. **46**, 1–3 (2002)
4. Deng, L., Yu, D.: Deep convex network: a scalable architecture for speech pattern classification. Interspeech, pp. 2285–2288 (2011)
5. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using second order information for training support vector machines. J. Mach. Learn. Res. **6**, 1889–1918 (2005)
6. Freeman, H.: Computer processing of line-drawing images. Comput. Surveys **6**(1), 57–97 (1974)
7. Impedovo, S., Mangini, F.M., Barbuzzi, D.: A novel prototype generation technique for handwriting digit recognition. Pattern Recogn. **47**, 1002–1010 (2014)
8. Iwakiri, Y., Shiraishi, S., Feng, Y., Uchida, S.: On the possibility of instance-based stroke recovery. In: Proc. ICFHR, pp. 29–34 (2012)
9. Kegl, B., Fekete, R.B.: Boosting products of base classifiers. In: Proceedings ICML (2009)
10. Labusch, K., Barth, E., Martinetz, T.: Simple method for high-performance digit recognition based on sparse coding. IEEE Trans. Neural Netw. **19**(11), 1985–1989 (2008)
11. Lam, L., Lee, S.W., Suen, C.Y.: Thinning methodologies—a comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intel. **14**(9), 869–885 (1992)
12. Lauer, F., Suen, C.Y., Bloch, G.: A trainable feature extractor for handwritten digit recognition. Pattern Recogn. **40**(6), 1816–1824 (2007)
13. Lecun, Y., Bootou, L., Bengio, Y., Haffner, P.: Gradient based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
14. Lecun, Y., Cortes, C., Burges, C.J.C.: The MNIST database of handwritten digits (2014). http://yann.lecun.com/exdb/mnist/
15. Lu, C.C., Dunham, J.G.: Highly efficient coding schemes for contour lines based on chain code representations. IEEE Trans. Commun. **39**(10), 1511–1514 (1991)
16. Meier, U., Ciresan, D.C., Gambardella, L.M., Schmidhuber, J.: Better digit recognition with a committee of simple neural nets. In: Proceedings of ICDAR (2011)
17. Nguyen, V., Blumenstein, M.: Techniques for static handwriting trajectory recovery: a survey. In: ACM Proceedings of 9th IAPR workshop on Document and Analysis Systems, pp. 463–470 (2010)

18. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
19. Pervouchine, V., Leedham, G., Melikhov, K.: Three-stage handwriting stroke extraction method with hidden loop recovery. In: Proc. ICDAR (2005)
20. Platt, J.: Equential minimal optimization: a fast algorithm for training support vector machines. Microsoft Research MSRTR9814 (1998)
21. Qaio, Y., Nishiara, M., Yasuhara, M.: A framework toward restoration of writing order from single-stroked handwriting image. IEEE Trans. Pattern Anal. Mach. Intel. **28**(11), 1724–1737 (2006)
22. Qiao, Y., Yasuhara, M.: Recover writing trajectory from multiple stroked image using bidirectional dynamic search. Proc. ICPR **2**, 970–973 (2006)
23. Rousseau, L., Camillerapp, J., Anquetil, E.: What knowledge about handwritten letters can be used to recover their drawing order ? In: Proc. ICDAR (2006)
24. Sharma, A.: Recovery of drawing order in handwritten digit images. IEEE Proc. ICIIP **2013**(1), 437–441 (2013)
25. Suen, C.Y., Niu, X.X.: A novel hybrid cnnsvm classifier for recognizing handwritten digits. Pattern Recogn. **45**, 1318–1325 (2012)
26. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)