

# ONLINE HANDWRITING RECOGNITION USING SUPPORT VECTOR MACHINE

Abdul Rahim Ahmad<sup>1</sup>  
Marzuki Khalid<sup>2</sup>

Christian Viard-Gaudin<sup>3</sup>  
Emilie Poisson<sup>3</sup>

<sup>1</sup>Universiti Tenaga Nasional,  
Jalan Kajang-Puchong, 43009 Kajang, Selangor, Malaysia.  
[abdrahim@uniten.edu.my](mailto:abdrahim@uniten.edu.my)

<sup>2</sup>Centre for Artificial Intelligence and Robotics (CAIRO)  
Universiti Teknologi Malaysia, Jalan Semarak, 54100 Kuala Lumpur, Malaysia.  
[marzuki.rubiyah@utmkl.utm.my](mailto:marzuki.rubiyah@utmkl.utm.my)

<sup>3</sup>IRCCyN - Division ISA - UMR 6597  
IVC, Ecole polytechnique de l'université de Nantes, France.  
[christian.viard-gaudin@polytech.univ-nantes.fr](mailto:christian.viard-gaudin@polytech.univ-nantes.fr)

## ABSTRACT

Discrete Hidden Markov Model (HMM) and hybrid of Neural Network (NN) and HMM are popular methods in handwritten word recognition system. The hybrid system gives better recognition result due to better discrimination capability of the NN [3]. Support Vector Machine (SVM) is an alternative to NN. In speech recognition (SR), SVM has been successfully used in the context of a hybrid SVM/HMM system. It gives a better recognition result compared to the system based on hybrid NN/HMM[4]. This paper describes the work in developing a hybrid SVM/HMM OHR system. Some preliminary experimental results of using SVM with RBF kernel on IRONOFF, UNIPEN and IRONOFF-UNIPEN character database are provided.

## 1. INTRODUCTION

The research investigates new methods in handwriting recognition. There have been many handwriting recognition systems available in the market. There are two distinct handwriting recognition domains; online and offline, which are differentiated by the nature of their input signals. In offline system, static representation of a digitized document is used in applications such as check, form, mail or document processing. On the other hand, online handwriting recognition (OHR) systems rely on information acquired during the production of the handwriting. They require specific equipment that allows the capture of the trajectory of the writing tool. Mobile communication systems such as Personal Digital Assistant (PDA), electronic pad and

smart-phone have online handwriting recognition interface integrated in them. Therefore, it is important to further improve on the recognition performances for these applications while trying to constrain space for parameter storage and improving processing speed.

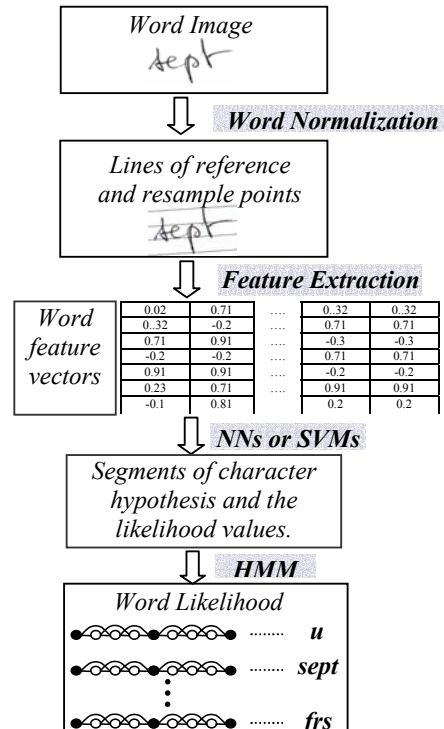


Figure 1: A hybrid OHR system.

Many current systems use Discrete Hidden Markov Model based recognizer or a hybrid of Neural Network (NN) and Hidden Markov Model (HMM) for the recognition. Figure 1 show a hybrid NN/HMM OHR system. Online information captured by the input device first needs to go through some filtration, preprocessing and normalization processes. After normalization, the writing is usually segmented into basic units (normally character or part of character) and each segment is classified and labeled.

Using HMM search algorithm in the context of a language model, the most likely word path is then returned to the user as the intended string [1]. Segmentation process can be performed in various ways. However, observation probability for each segment is normally obtained by using a neural network (NN) and the probabilities of transitions within a resulting word path are estimated by a Hidden Markov Model (HMM). This research aimed to investigate the usage of support vector machines (SVM) in place of NN in a hybrid SVM/HMM recognition system

The main objective is to further improve the recognition rate by using support vector machine (SVM) at the segment classification level. This is motivated by successful earlier work by Ganapathiraju [4] in a hybrid SVM/HMM speech recognition (SR) system and the work by Bahlmann [8] in OHR. Ganapathiraju obtained better recognition rate compared to hybrid NN/HMM SR system. In this work, SVM is first developed and used to train an OHR system using character databases.

SVM with probabilistic output are then developed for use in the hybrid system. Eventually, the SVM will be integrated with the HMM module for word recognition. Preliminary results of using SVM for character recognition are given and compared with results using NN reported by Poisson [9]. The following databases were used: IRONOFF, UNIPEN and the mixture IRONOFF-UNIPEN databases. This paper is organized as follows: Section 2 introduces SVM and the various implementations. The description of character recognition experiments with SVM, databases used in the experiments and the experiments results are discussed in section 3. Section 4 contains a summary that concludes this paper.

## 2. SUPPORT VECTOR MACHINE (SVM)

SVM in its basic form implement two class classifications. It has been used in recent years as an alternative to popular methods such as neural network. The advantage of SVM, is that it takes into account both experimental data and structural behavior for better generalization capability based on the principle of structural risk minimization (SRM). Its formulation approximates SRM principle by maximizing the margin of class separation, the reason for it to be known also as large margin classifier.

The basic SVM formulation is for linearly separable datasets. It can be used for non-linear datasets by indirectly mapping the nonlinear inputs into to linear feature space where the maximum margin decision function is approximated. The mapping is done by using a kernel function. Multiclass classification can be performed by modifying the 2 class scheme.

### 2.1. SVM formulation

For a set of  $l$  linearly separable data and its class  $\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$  where  $x_i \in R^d$  and  $y_i \in \{\pm 1\}$ , the maximum margin classifier is  $f(x) = \text{sgn}(w \cdot x + b)$  where  $w$  and  $b$  are parameters that maximize the margin with respect to the two classes. A new form of the classifier, expressed with input and output vectors information is as follows:

$$f(x) = \text{sgn}\left(\sum_{i=1}^l \alpha_i y_i (x_i \cdot x) + b\right)$$

Here parameter  $\alpha$  for each corresponding input vectors needs to be found in order to find the maximal margin classifier. The  $\alpha$  values are mostly zero and those inputs with non-zero  $\alpha$ 's are called the support vectors and they contribute strongly towards the decision function.

If the data set is not linear, the decision function used is :  $f(x) = \text{sgn}\left(\sum_{i=1}^N \alpha_i y_i K(x_i, x) + b\right)$  where a

kernel  $K$  is used in the mapping of the non-linear input space into a linear space. The maximal margin classifier is found in the linear space. There are a few possible kernels that can be chosen :

(a) Linear kernel:  $K(x, y) = x \cdot y$ ,

(b) Polynomial kernels:  $K(x, y) = (x \cdot y + 1)^d$ ,

(c) Radial basis function kernel:

$$K(x, y) = e^{-\frac{|x-y|^2}{2\sigma^2}}$$

(d) Hyperbolic tangent kernel:

$$K(x, y) = \tanh(ax \cdot y - b)$$

The value of  $\alpha$  obtained is constraint to be positive for perfectly separable case and between 0 and C in the case of non-linearly separable data. The value C is the penalty term and needs to be chosen prior to training the SVM.

### 2.2. SVM implementation

SVM training involves solving a convex quadratic programming (QP) problem with equality and inequality constraints. The solution solves for nonzero parameters  $\alpha$  and extracts the support vectors corresponding to it. A number of methods of SVM training have been developed over the years to improve on approximation accuracy, memory requirement and training time. Other issues include finding the best training model using appropriate kernel and the hyper parameters. In addition, basic SVM only handle two-class classification. Multiclass

classification requires training of many two class classifiers and in classification, voting schemes are used for selecting the correct class. Implementing SVM training involves (a) Selecting the parameter C, kernel function and any kernel parameters, (b) Solving the dual QP or alternative formulation using appropriate algorithm to obtain  $\alpha$ 's and the support vectors, (c) Calculating threshold b using the support vectors.

Selecting the parameter values C, the kernel and its parameters is called model selection. Choosing kernel parameters can be done by minimizing some generalization error or performance measures such as k-fold cross-validation or leave-one-out (LOO) estimates. SVM have been made popular by the availability of stable implementation packages. There are a few implementation packages available publicly and have been popularly used as reported by many researchers. Among them are LIBSVM, SVMTool and SVMLight.

### 3. CHARACTER RECOGNITION USING SVM

Preliminary experiments were conducted to investigate the usage of SVM in online character recognition. The databases used and the recognition results are described.

#### 3.1 Databases

In the experiments we have used the IRONOFF database, UNIPEN database and a combination of the two databases together. IRONOFF contains both online and offline handwriting information collected by IRCCyN in Nantes, France. It contains 4,086 isolated digits, 10,685 isolated lower case letters, 10,679 isolated upper case letters and 410 EURO signs. It also contains 31 346 isolated words from a 197 word lexicon (French: 28 657 and English: 2 689). In the experiments only the isolated digits and letters are being used.

UNIPEN online database are available from the International Unipen Foundation. It consists of various training and benchmark datasets for online handwriting which includes characters and words. In the experiments only benchmark set 1a (16,000 isolated digits), 1b (28000 isolated upper case characters) and 1c (61000 isolated lowercase characters) are used. A combination of IRONOFF and UNIPEN database called IRONOFF-UNIPEN is also used.

#### 3.2 Experiments on SVM for character recognition

For the experiments, a feature extractor module was developed to extract 7 local features for each point of the online signal in the example character. An example character is first normalized in size, slant and rotation angle and spatially resampled to obtain a

uniform sample of 50 points. For each point the following 7 features were extracted: (a) normalized x coordinate, (b) normalized y coordinate, (c) direction angle  $\theta(x)$  of the curve according to the x axis (cosine  $\theta(x)$ ), (d) direction angle  $\theta(x)$  of the curve according to the y axis direction (sine  $\theta(x)$ ), (e) curvature according to x axis, (f) curvature according to y axis and (g) the position of stylus (up or down).

These 7 features are chosen since they are simple to obtain and have been used by Poisson [9] in other similar experiments using TDNN and MLP NN. Therefore, for each example character there are 350 feature values which are the inputs of the SVM. For our character recognition, we use modified LIBSVM with RBF kernel, since RBF kernel have been shown to generally give better recognition result [7]. Grid search on each dataset of the databases were done in order to choose the best values for the C and gamma parameters (representing  $\frac{1}{2\sigma^2}$  in the original RBF

kernel formulation) for the final SVM models. It is observed that C values between 2 and 8 and gamma values between  $2^{-7}$  and  $2^{-5}$  yielded the best character recognition rate. We have chosen a single pair of C = 8 and gamma =  $2^{-5}$  value for our training on all databases since the results obtained shows that individual grid search on the datasets yields almost similar C and gamma values for majority of the datasets.

Tests of SVM on IRONOFF- UNIPEN datasets have shown that recognition rate for digits and upper case is higher than lower case since handwritten lower case characters differs significantly from person to person. Table 1 shows recognition results for IRONOFF-UNIPEN database. Training time and the number of support vectors varies according to the training size. Table 2 shows the result when comparing MLP NN, TDNN and SVM for IRONOFF-UNIPEN database.

*Table 1: Detail Recognition performance of SVM on IRONOFF- UNIPEN datasets.*

Data Set	Training Set	Test Set	Test set (%)	nSV	Training time (s)
Digit	13451	6270	98.68	3014	497
Lowercase	42778	20172	93.76	15696	5897
Uppercase	25662	11621	95.13	10035	2808

*Table 2: Recognition rates and parameters using MLP, TDNN and SVM on IRONOFF- UNIPEN datasets.*

Data Set	MLP		TDNN		SVM	
	Free par.	Rec Rate	Free par.	Rec Rate	nSV	Rec Rate
Digit	36110	<b>97.9</b>	3790	<b>98.4</b>	3014	<b>98.68</b>
Lowercase	37726	<b>91.3</b>	8926	<b>92.7</b>	15696	<b>93.76</b>
Uppercase	37726	<b>93.0</b>	8926	<b>94.5</b>	10035	<b>95.13</b>

For IRONOFF-UNIPEN database, the recognition rate using SVM is the highest, compared with TDNN and MLP NN. However, in term of parameter storage, TDNN is a clear winner due to the weight-sharing scheme. Without any compression, SVM model requires the most space since each support vector (SV) consist of many feature values (in our case 350). However, space saving can be achieved by storing only the original online signals and the pen-up/pen-down status corresponding to the SV in a compact manner. During recognition, the model will be expanded dynamically as required. Table 3 shows the comparison of recognition rates between all the three methods using all three databases in which SVM outperforms the other two methods in all three cases.

Experiments using SVMs with probabilistic output were also performed on the same datasets for comparison. Table 4 gives the result.

**Table 3: TDNN vs. SVM comparison for all datasets of IRONOFF and UNIPEN databases**

Data Set	IRONOFF database			UNIPEN database		
	MLP	TDNN	SVM	MLP	TDNN	SVM
Digit	98.2	98.4	<b>98.83</b>	97.5	97.9	<b>98.33</b>
Lowercase	90.2	90.7	<b>92.47</b>	92.0	92.8	<b>94.03</b>
Uppercase	93.6	94.2	<b>95.46</b>	92.8	93.5	<b>94.81</b>

**Table 4: SVM distance vs. probabilistic SVM based recognition for IRONOFF and UNIPEN databases**

Data Set	IRONOFF database		UNIPEN database	
	SVM	SVM prob.	SVM	SVM prob.
Digit	<b>98.83</b>	98.68	98.33	<b>98.35</b>
Lowercase	<b>92.47</b>	92.42	94.03	<b>94.14</b>
Uppercase	<b>95.46</b>	95.45	94.81	<b>94.85</b>

#### 4. CONCLUSION

In all the experiments, the results have shown that at character level, SVM recognition rates are significantly better due to structural risk minimization implemented by maximizing margin of separation in the decision function. However, the increase in recognition rate is not without some impact. SVM model size is characterized by the number of support vectors obtained in the training. Storing these support vectors for recognition requires larger memory as compared to NN weights since each support vector is a multidimensional feature vector.

The number of support vectors can be reduced by selecting better C and gamma parameter values through a finer grid search and by reduced set selection [5][6]. The comparison of recognition results of SVM with probabilistic output and SVM distance output shows that both are comparable. In some datasets, SVM distance gives slightly higher while in some others the probabilistic output gives higher recognition rates.

Work on integrating the SVM character recognition framework into the HMM based word recognition framework is on the way. In the hybrid system, word preprocessing and normalization needs to be done before SVM is then used for character hypothesis recognition and word likelihood computation using HMM. It is envisaged that, due to SVM's better discrimination capability, word recognition rate will be better than in a NN/HMM hybrid system.

#### 5. REFERENCES

- [1] Homayoon S.M. Beigi, "An Overview of Handwriting Recognition," Proceedings of the 1<sup>st</sup> Annual Conference on Technological Advancements in Developing Countries, Columbia University, New York, July 24-25, 1993, pp. 30-46.
- [2] Bengio, Y. and LeCun, Y., "Word Normalization for On-Line Handwritten Word Recognition," in Proc. 12 th Int'l. Conf. Pattern Recognition, vol. 2, pp.409-413, Jerusalem, Oct. 1994.
- [3] Bengio, Y., LeCun, Y., LeRec: "An ANN/HMM Hybrid for On-line Handwriting Recognition", Neural Computing, vol 7, no. 5, 1995.
- [4] A. Ganapathiraju., "Support Vector Machines for Speech Recognition," PhD Thesis, Mississippi State University, January 2002.
- [5] C.J. Burges. "Simplified support vector decision rules". In L. Saitta, editor, Proceedings, 13th Intl. Conf. on Machine Learning, pages 71-77, San Mateo, CA, 1996. Morgan Kaufmann.
- [6] T. Downs, K.E. Gates, A. Masters., "Exact Simplification of Support Vector Solution", In Journal of Machine Learning Research 2 (2001) 293-297.
- [7] S. S. Keerthi and C.-J. Lin, "Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel" , Neural Computation., July 1, 2003; 15(7): 1667 - 1689.
- [8] C. Bahlmann, B. Haasdonk, H. Burkhardt., "Online Handwriting Recognition with Support Vector Machine – A Kernel Approach", In proceeding of the 8th Int. Workshop in Handwriting Recognition (IWHFR), pp 49-54, 2002.
- [9] E. Poisson, C. Viard-Gaudin, P.M Lallican, "Multi-modular architecture based on convolutional neural networks for online handwritten character recognition", ICONIP2002, 2002.