Handwritten Symbol Recognition using an Ensemble of SVM Classifiers

Ronald Clark*, Quik Kung*, M.A. Van Wyk†
University of the Witwatersrand
* {ronald.clark, quik.kung}@students.wits.ac.za
† ma.vanwyk@wits.ac.za

***Abstract*—Support Vector Machines (SVM) have proven to be highly accurate in classifying handwritten mathematical symbols - especially when a diverse range of features is used. This paper investigates the classification of handwritten mathematical symbols using an SVM method and an ensemble of three different feature sets in order to minimise the number of training samples required and achieve accurate classification rates. The architecture of the system consists of pre-processing, symbol segmentation and classification. Three SVMs are used, each operating on different feature sets: sample point coordinates (SVM 1) turning angles and their derivatives (SVM 2) and global features (SVM 3). The symbol classifications are combined using various decision fusing techniques. The system was tested using a small set of 252 samples consisting of 41 classes or 6 samples per class. The results yielded a 97.20% correct classification rate using feature set 1 while a rate of only 90.91% was obtained using a single high-dimensional SVM combining the three feature sets. The ensemble configuration further improved the classification rate to 98.601% using a simple average-based decision fusion scheme. As such, the proposed SVM ensemble considerably increases the classification accuracy when only a few training samples are available.**

NOMENCLATURE

Ω	Set of all symbol classes.
h_i	Possible symbol classes (ie. the elements of Ω)
p_{ji}	Confidence that the input of the j'th classifier belongs to class i.
$m_k(h_i)$	Proposition that the sample belongs to class i. In this paper it is assumed to be identical to the confidence output p_{ji}.
$P(h_j)$	Probability that the j'th classifier labels a sample with the class proposition h_j.
$P(\mathbf{h}\|c_k)$	Likelihood of the propositions given input class c_k.
$P(c_k\|\mathbf{h})$	Posterior probability of the class label given the propositions $\mathbf{h}$.
$\mathbf{x_i}$	Feature vector belonging to the ith sample.
$\mathbf{p_j}$	Vector representing the soft-decision output of an SVM.
L	Total number of classifiers.
c	Total number of symbol classes.
$\mathbf{p_i}$	Coordinate vector of sample point i.
$\phi(\mathbf{x})$	Map to higher-dimensional feature space.
$K(\mathbf{x}, \mathbf{x_i})$	Kernel function defining the inner-product space.
$\mathbf{w}$	Normal vector defining the optimal hyperplane.
b	Offset defining the optimal hyperplane.
α_i	A Lagrange multiplier.
ϵ	Error term used in finding the SVM decision boundary.

I. INTRODUCTION

Handwriting recognition is the process of converting characters drawn as a series of graphical marks into their symbolic representation which can be further processed by a computer system. This process may be carried out in an *online* or *offline* manner. In the *online* case (which is considered in this paper) recognition is performed at the same time as the writing process which means that information related to the writing dynamics and stroke ordering are available. The complete symbol recognition process involves three steps:

1) Pre-pocessing (de-skewing, de-hooking, conversion to equidistant samples, smoothing etc.)
2) Segmentation to isolate symbols
3) Classification of symbols

The classification of symbols can be done using either a parametric or non-parametric classifier. Parametric classifiers operate on a number of specially-selected features that have been extracted from the symbol to perform classification while non-parametric classifiers simply operate on the entire input data set. Parametric classifiers often achieve better recognition rates and classification times compared to non-parametric classifiers and are therefore the preferred method in many systems [1].

As such, this paper describes a parametric character recognition process which involves two primary steps: feature extraction and classification [2]. Most modern systems make use of either Support Vector Machines (SVM's) or Artificial Neural Networks (ANNs) as these techniques have proven very effective for online symbol recognition [3]–[5]. SVMs have been successfully used in face detection, handwritten digit recognition and data mining [6].

The accuracy of the SVM can be improved in two ways: by increasing the number of features or increasing the number of samples used for training. The training process, however, is very computationally-expensive which leads to a tradeoff between the desired accuracy and acceptable training time. Furthermore, this problem is exacerbated by the fact that the number of samples required to achieved a certain classification accuracy as well as the training time per sample increases with the dimensionality of the feature vectors. This can render the use of even a moderately-sized training set unfeasible.

Numerous studies have investigated the idea of combining multiple SVMs to improve classification accuracy through techniques such as boosting and bagging. For example, Kim et al. obtained an improvement in performance of 1.81% using a boosted combination of 10 multi-class SVMs to achieve a classification accuracy of 97.83% [6]. Even so, a relatively large number of samples was still needed to achieve this accuracy with a training set of 3828 and a test set of 1797 symbols.

This paper proposes the use of an ensemble of three SVMs, each operating on a unique low-dimensional feature set with the goal of minimising the number of training samples required while maximising the classification accuracy. The recognition of handwritten mathematical symbols to illustrate the performance gains achieved by the proposed ensemble of classifiers.

The paper is structured as follows. In Section II and III, a brief background is given of the SVM classifier and techniques for combining the output of multiple classifiers. In Section IV the pre-processing and symbol segmentation techniques are explored and the feature extraction is described. Lastly, in Section VI and Section VIII, the classification rates for the individual SVMs, a single optimised SVM as well as the ensemble configuration are analysed and discussed and relevant conclusions are drawn.

II. Support Vector Machines

SVMs work by finding a boundary in the feature space which maximises the distance between feature vectors belonging to two distinct input classes [7]. The decision boundary usually takes the form of a linear function which separates the two classes. In linearly inseparable problems, a non-linear decision surface is created by lifting the feature space into a higher dimensional space which allows a linear separating hyperplane to be found [8]. This hyperplane corresponds to a non-linear decision surface in the original feature space. The mapping is denoted by $\phi(\mathbf{x})$ which represents the map to the higher dimensional space where the data are linearly separable.

By using the kernel function $\mathbf{K}(\mathbf{x}, \mathbf{x}_i) = \phi(\mathbf{x_i})^{\mathbf{T}}\phi(\mathbf{x})$, the decision function of the SVM can be represented by:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i \mathbf{K}(\mathbf{x}, \mathbf{x_i}) + b \tag{1}$$

where $f(x)$ is the decision output, y_i is the label of the training symbol $\mathbf{x_i}$ and $\mathbf{x}$ is the symbol to be classified.

The parameters α_i and b are found during training which is performed by solving the following optimisation problem:

$$\min_{w,b} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{\mathbf{i=1}}^{\mathbf{n}} \epsilon_{\mathbf{i}} \tag{2}$$
$$\text{subject to } y_i(\mathbf{w}^{\mathbf{T}}\phi(\mathbf{x}_i) + b) \geq 1 - \epsilon_i$$

Many kernel functions exist but a well-performing kernel, used in many optical character recognition (OCR) systems, is the radial basis function (RBF):

$$\mathbf{K}(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma||\mathbf{x} - \mathbf{x_i}||^2),\ \gamma > 0 \tag{3}$$

The constant C in Equation 2 is the penalty parameter of the error term and the constant γ in Equation 3 is a kernel parameter. Both of these parameters have a significant effect on the accuracy of the trained system and need to be carefully set prior to the training process. The method used for selecting these parameters is mentioned in Section VI.

Although SVMs are binary classifiers, multi-class classification is easily achieved by combining SVMs in a one-against-others or one-against-one scheme [9]. Although SVM training time is proportional to the square of the number of samples and thus relatively slow, actual classification is very fast and can be performed in real-time [9].

A. Parameter Selection

As described in Section II, there are two parameters that need to be set when using an RBF kernel: C and γ which need to be carefully selected prior to the training process.

The selection of these parameters can be automated by using the grid-search method described in [7]. This method selects the combination of parameters which give the best cross-validation accuracy during training by carrying out a brute force search of all the possible parameter combinations. This optimisation process only needs to be carried out once for a particular training set.

III. Combining Classifiers

Throughout this paper it is assumed that the output of the j'th SVM classifier is a vector of scores $\mathbf{p_j} = [p_{j0}p_{j1}...p_{jn}]$ which approximate the posterior probabilities of the input sample belonging to a certain symbol class c_i given the observed feature vector, ie. $p_{ji} = P(c_i|\mathbf{x})$. A decision rule is thus needed to determine the final symbol class based on these probability values.

A. Dempster's Rule of Combination

Dempster's rule considers a number of mutually exclusive and exhaustive propositions h_i, $i = \{1...n\}$ which form part of a universal set Ω. Each classifier indicates its opinion by producing a mass of belief function $m_k(h_i)$ over Ω which is an independent indication of the classifiers belief that the proposition is correct.

The combined masses of belief can then be found using Dempster's rule of combination:

$$\begin{aligned} m_{12}(A) =& \frac{1}{1-K} \sum_{B \cap C = H} m_1(B) m_2(C) \\ K =& \sum_{B \cap C = \phi} m_1(B) m_2(C) \end{aligned} \tag{4}$$

The output quantity $m_{12}(A)$ represents a third mass function which combines pieces of evidence from the individual classifiers to produce stronger support for the most likely propositions.

The quantity $1 - K$ in Equation 4 is a normalisation coefficient which gives a measure of the conflict between the sources. If this quantity is near zero, the classifiers are in total disagreement and the Dempster rule is no longer valid.

B. Naive Bayes

The Bayes scheme also assumes that the individual classifiers produce independent predictions for each class type. If $P(h_j)$ denotes the probability that the j'th classifier labels a sample $\mathbf{x}$ (of class c_k) with the class proposition h_j, the likelihoods of the proposed classes can be calculated as follows [10]:

$$P(\mathbf{h}|c_k) = P(h_1, h_2, ..., h_L|c_k) = \prod_{i=1}^{L} P(h_i|c_k) \tag{5}$$

The posterior probabilities are then obtained by calculating [10]:

$$P(c_k|\mathbf{h}) = \frac{P(c_k)P(\mathbf{h}|c_k)}{P(\mathbf{h})} = \frac{P(c_k)\prod_{i=1}^{L} P(s_i|c_k)}{P(\mathbf{h})} \tag{6}$$

which can be used to classify the input $\mathbf{x}$. As the quantity $P(\mathbf{s})$ does not depend on the class type, the decision can be made using the quantity [10]:

$$\mu_k(\mathbf{x}) \propto P(c_k) \prod_{i=1}^{L} P(s_i|c_k) \tag{7}$$

C. Majority Vote

The majority vote ensemble technique chooses as the final decision the class that appears most often in the selections made by the component classifiers. By denoting the decision of j'th classifier as $d_{j,k} \in \{0, 1\}$, $j = 1, ..., L$ and $m = 1, ..., c$ where L is the number of classifiers and c is the number of symbol classes, the decision output will be class k if [11]:

$$\sum_{j=1}^{L} d_{j,k} = \max_{m=1}^{c} \sum_{j=1}^{L} d_{j,m} \tag{8}$$

If any "ties" result, the output of the classifier with the highest measure is taken as the final decision.

D. Average

The average method simply finds the average of the confidence outputs of the individual classifiers and assigns the class label with the highest average confidence.

E. Product

In this scheme, the maximum of the product of the classifier outputs for each class is used as the class label.

IV. System Architecture

The system uses a modular architecture. The input module captures strokes as an ordered series of (x_i, y_i) data point coordinates. These data points are preprocessed to reduce noise and decrease the number of points per stroke. The symbol segmentor then groups strokes based on a simple distance threshold to form symbols.

The classifier module operates on these symbols by extracting chosen features and sending the resulting feature vectors to the appropriate SVM for classification as illustrated in Figure 1.

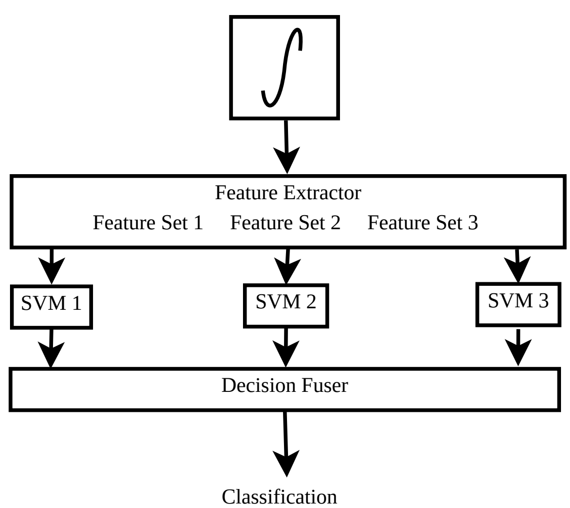


Figure 1. Ensemble SVM classifier

A. Pre-processing

The pre-processing stage involves filtering and re-sampling of the data points. Filtering is carried out by replacing the raw coordinates by a weighted sum of the neighbouring points. As in [12], three coefficient values are used:

$$\mathbf{p}_i^* = 0.25\mathbf{p}_{i-1} + 0.5\mathbf{p}_i + 0.25\mathbf{p}_{i+1} \tag{9}$$

This smoothing technique is computationally inexpensive and has proven to be very effective [12]–[14]. After the stroke has been smoothed, new samples are obtained by generating points that are equidistant with respect to arc length.

B. Symbol Segmentation

Individual strokes are grouped into symbols using a method similar to that of Ernesto [12]. In order to determine whether two strokes belong to the same symbol, the minimum distance between the sample points of the two strokes is compared to a threshold value d_{th}. The threshold value is dynamically adjusted according to the height of the second stroke to accomodate for symbols of various sizes (for example superscripts and subscripts). The threshold is determined as follows:

$$d_{th} = \frac{1}{10}\max(width,\ height) \tag{10}$$

If the distance between the end points of the strokes is lower than the threshold value, the strokes are concatenated to form a single continuous stroke.

C. Classification

Many methods exist for creating an ensemble of classifiers, however, the most important consideration is to ensure that the classification performance of the individual SVMs are independent and differ as much as possible from each other [15]. This is usually done by using different training sets for different SVMs which are obtained using techniques such as bagging, boosting or randomisation [6].

In this system, instead of varying the training sets, each SVM is trained using a unique set of features. That is, each vector contains different information about the symbol and not a different training set.

D. Feature Extraction

The features used for classification are similar to those proposed in [12]. The first two feature sets (FS) are derived from the local features of the stroke (s) of points $(p_1, \ldots, p_n)$ and the third FS is obtained by including the total number of points. The points are used in the order in which they are written, making them sensitive to writing direction. The feature sets have been chosen to capture the greatest variation and are as follows:

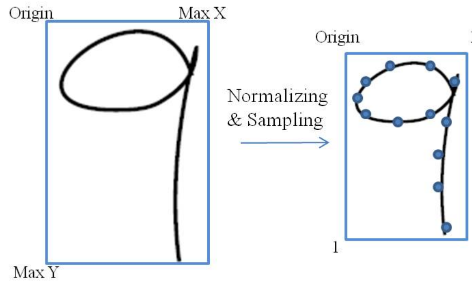


Figure 2. Normalizing and Sampling of a Symbol for FS 1.

- FS 1
  - 20 co-ordinate points: (x_i, y_i) of p_i
- FS 2
  - 19 turning angles: $\frac{\theta_i}{2\pi}$ where the turning angle is $\theta_i = \angle p_{i-1}\bar{p}_i p_{i+1}$
  - 18 derivatives of the turning angle: $\frac{(\theta_{i+1}-\theta_i)}{2\pi}$ and $\frac{(\theta_{i-1}-\theta_i)}{2\pi}$
- FS 3
  - Center of gravity: $x_g = \sum_{i=1}^{n} x_i/n$ and $y_g = \sum_{i=1}^{n} y_i/n$
  - Total length: l
  - Accumulated angle: $\theta_a = \sum_{i=1}^{n} \theta_i/2\pi$

FS 1 is chosen for its simplicity and direct approach of co-ordinate comparisons to be performed by the SVM. It takes the x and y co-ordinates of the strokes, with the origin at the top left corner, scales and normalizes the values, depending on the size of the written symbol, and stores these points as its feature vector or FS 1. The normalisation of the symbol correctly factors the symbol size in order to match to the training set provided. This feature can be seen in Figure 2 where a written symbol is sampled, preprocessed (smoothed) and then normalised before re-sampling.

FS 2 calculates the turning angles by comparing to a point to the one before and after it. The turning angle is obtained by the cosine rule and thus calculates the interior angle. The derivative of the angle is then calculated with respect to the neighbouring points. This FS is aimed at differentiating between symbols with similar structural characteristics in terms of point co-ordinates but which differ in angular integrity, such as sharp corners as opposed to gradual changes in direction. The process of determining the angle between three points in a stroke is shown in Figure 3.

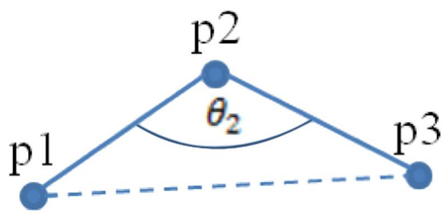


Figure 3. Calculation of Angle at a point for FS 2.

FS 3 consists of the center of gravity, the total length and the sum of all the angles of each stroke in a symbol. The centre of mass is found by summing all the x and y co-ordinates and dividing by the total number of points. The total length of the strokes is taken as the length from point to point after the symbol has been smoothed and the total angle is the summation of the constituent angles. FS 3 easily distinguishes between symbols with similar structure but with points clustered in a certain region. The total length and accumulated angles of the symbols are global properties to further strengthen this feature for SVM classification.

V. Testing Procedure

The system was trained on a small set of 252 samples consisting of the following 41 classes:

- Digits: 0-9
- Symbols: infinity (∞)
- Letters: a-d, i, m, k, s, x-z
- Greek letters: α, β, θ, ∂, ϵ, μ, σ, ω and π
- Operators: addition (+), subtraction (-), division (/), parenthesis, square root ($\sqrt{}$), summation ($\sum$), and integral ($\int$)
- Relations: equality ($=$), less than ($<$), greater than ($>$)

The samples for both the testing and training data sets were written by the same user on a Wacom Intuous3 6×8 tablet. The symbols were chosen randomly from the Aster database of mathematical expressions [16].

An SVM with an RBF kernel was used as it achieved better cross-validation rates during training compared to the linear, Gaussian and polynomial kernels. The classifiers were implemented and trained using the libSVM software library [17] which provides excellent tools for tuning the kernel and training parameters. The grid-search method was used to find the optimal SVM parameters. The output of the grid-search is shown in Figure 4 where it can be seen that the the optimal values of C and γ were 2^{10} and 2^{-5}, respectively.

VI. Results and Discussion

A complete set of correctly recognised symbols, extracted from the testing results, is shown in Figure 5.

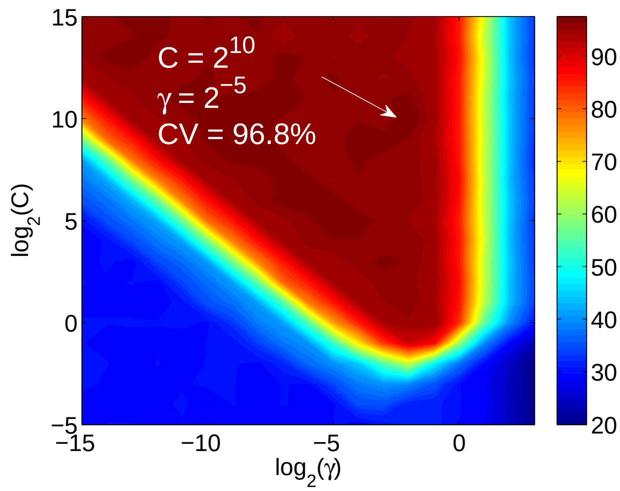


Figure 4. Output of the grid-search used for tuning the SVM parameters. A maximum cross validation accuracy of 96.8% is obtained when $C = 2^{10}$ and $\gamma = 2^{-5}$.

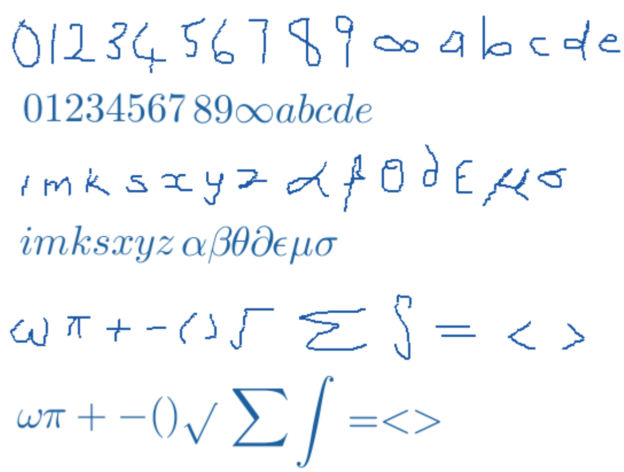
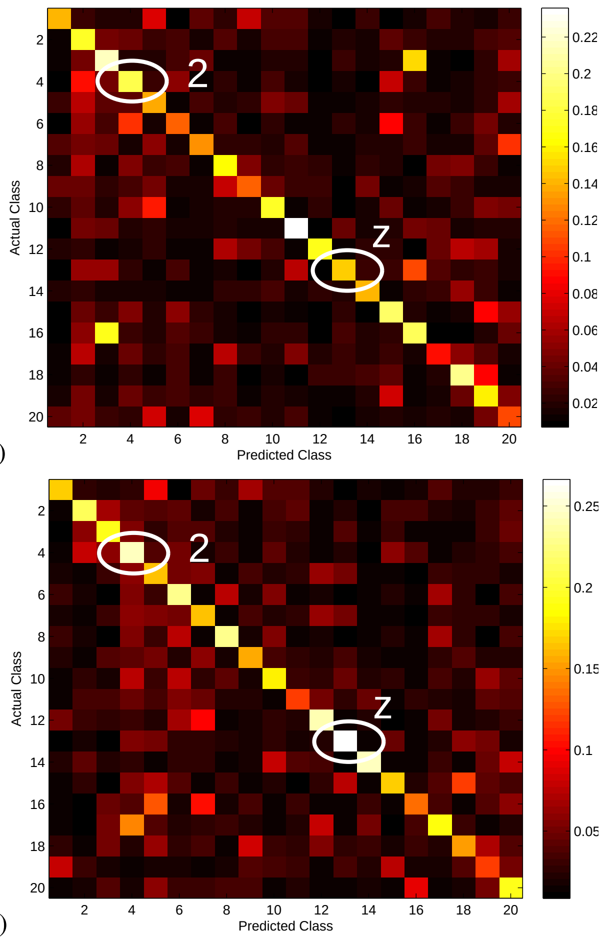


Figure 5. A set of correctly recognised symbols obtained during testing. The symbols were sourced from the Aster database of mathematical expressions. The symbols shown in this image are representative of the raw input to the system, ie. the results of the preprocessing stage are not shown. The typeset symbols have been rendered using LaTeX.

The aim of using an SVM ensemble is to extend the set of correctly recognised symbols beyond that of a single FS while still keeping the required number of training samples as low as possible. The different features enable the system to distinguish key characteristics in certain symbols that the other feature may not. A pertinent example is the digit '2' and the letter 'z'. Considering only the co-ordinate features, it can easily mistake the letter for the digit or vice verse. However, when considering the turning angle and the derivative, the top right corner of the letter 'z' will create a vast difference for the SVM classifier compared to the co-ordinate vector. This result is confirmed by the confusion matrices in Figure 6 which shows the probability outputs obtained from the SVM for each testing sample. The confidence level for the digit "2"



(a)

(b)

Figure 6. Confusion matrix of FS1 (a) and FS2 (b) for a subset of 20 testing symbols obtained from the probability output of the SVM.

is twice as high (≈ 0.22) for FS2 as it is for FS1 (≈ 0.11).

Table I
CLASSIFICATION RATES ACHIEVED BY COMBINING 3 SVMS WITH DIFFERENT FEATURE SETS USING A TRAINING SET AND TESTING SET EACH CONTAINING 252 SYMBOLS.

Decision Fuser	Correct Classification Rate (%)
FS1	97.203
FS2	70.63
FS3	88.11
Single SVM	90.91
Product	97.902
Average	98.601
Bayes	96.5
Dempster	97.202
Majority Vote	84.61

The recognition rates for the different feature sets are shown in Table I. The first four rows report the rates achieved by using the different feature sets independently and as a single classifier using all the features. The remaining rows show the result of using the three SVMs in an ensemble configuration with different decision-fusing schemes.

Table I shows that the classification rate achieved using only FS1 is higher than that of the single optimised SVM. This is most probably due to the severely limited size of the set of symbols used during training which leads to poor generalization performance for the high-dimensional classifier. As FS1 has significantly less features (40 compared to 79) it requires fewer training samples to achieve the same (and even better) classification accuracy which is a key advantage of using an ensemble of SVMs operating on small feature sets.

Out of the five decision fusion methods evaluated in this paper, the majority vote clearly performs the worst with a classification accuracy of only 84.61%. This is similar to the result achieved by Gorgevik et al., who attribute the poor performance of voting cooperation schemes to the limited information that is used about the member classifiers as no consideration is given to confidence outputs or second choices [6].

In this case, the "simple" average and product cooperation schemes outperform the more complicated naive Bayes method and Dempster rule. This is contrary to the result obtained in [6] where the simple cooperation schemes only achieve average recognition rates. Again, this is mainly due to the small training set size which limits the accuracy of the prior probabilities used in the Bayes scheme as well as the the output posterior probabilities generated by the SVM which are used by both the Dempster rule and the Bayes scheme. For example, Gorgevik et al. use a minimum training set size of 10000 samples to derive these values [2].

VII. FUTURE WORK

An important factor affecting the performance of the ensemble technique is the independence of the features and feature sets used by the classifiers. The independence of the features was, however, not verified in this paper as an adequate method for quantifying the independence could not be found. Three techniques that could possibly be used include principal component analysis, factor analysis and linear discriminant analysis which are commonly used for dimensionality reduction.

Furthermore, an intelligently-weighted sum ensemble could improve on the accuracy achieved by the average ensemble technique. In this case the output of the ensemble would be $w_1 \times$FS1 + $w_2 \times$FS2 + $w_3 \times$FS3 with $w_1 + w_2 + w_3 = 1$. Optimal values for w_1, w_2 and w_3 could then be found on the training set, possibly giving better results than the average.

VIII. CONCLUSION

Although SVMs have been shown to achieve high classification rates for handwritten symbols, they generally require a large number of training samples to achieve satisfactory performance. Because of the writer dependent nature of online handwritten symbol recognition, these samples need to be generated by the end user of the system which can be a time-consuming and tedious process.

In this paper, the use of an ensemble of SVM classifiers is investigated as the symbol recognition component of a handwritten mathematical expression recognition system. A number of decision fusion methods are considered to combine the outputs of the individual classifiers.

The results show that, for a small training set, a classifier using a feature vector with fewer components can increase the classification accuracy. This accuracy can be further enhanced by combining the output of multiple SVMs, each performing classification on a different set of features extracted from the same symbol. In this case, the optimal performance is achieved by a product-based combination of the individual SVM outputs.

IX. Acknowledgments

The authors would like to thank their colleagues Merelda Wu and Matthew Looi for their valuable suggestions and comments regarding this paper.

References

[1] D. Blostein. *Recognition of Mathematical Notation*, chapter 22. World Scientifc Publishing Company, 1996.

[2] Dejan Gorgevika, Dusan Cakmakov, and Vladimir Radevski. Handwritten digit recognition by combining support vector machines using rule-based reasoning. In *Proceedings of the 23rd International Conference on Information Technology Inferfaces*, pages 139–144, 2001.

[3] Y. LeCun. Energy-based models in document recognition and computer vision. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, 2007.

[4] L. Bottou. Comparison of classifer methods: a case study in handwritten digit recognition. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 1994.

[5] T. M. English. comparison of neural network and nearest-neighbor classifers of handwritten lower-case letters. In *Proceedings of the Fifth IEEE International Conference on Neural Networks*, 1993.

[6] Hyun-Chul Kim, Shaoning Pang, Hong-Mo Je, Daijin Kim, and Sung Yang Bang. Pattern classification using support vector machine ensemble. *Journal of Pattern Recognition*, pages 160–163, 2002.

[7] A. Ben-Hur and J. Weston. A users guide to support vector machines. Technical report, Department of Computer Science. Colorado State University, 2010.

[8] B. Keshari and S.M. Watt. Hybrid mathematical symbol recognition using support vector machines. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, 2007.

[9] I.-S. Oh and C. Suen. A class-modular feed-forward neural network for handwriting recognition. *Pattern Recognition*, 35:229–244, 2002.

[10] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons, 2004.

[11] Nicholas Stepenosky, Deborah Green, John Kounios, and Christopher M. Clarkand Robi Polikar. Majority vote and decision template based ensemble classifiers trained on event related potentials for early diagnosis of alzheimer's disease. In *ICASSP*, 2006.

[12] Ernesto Tapia. Understanding mathematics: A system for the recognition of on-line handwritten mathematical expressions. Master's thesis, Freie Universität Berlin, 2005.

[13] R. Zanibbi, D. Blostein, and J. R. Cordy. Baseline structure analysis of handwritten mathematics notation. In *Proceedings of the IEEE ICDAR conference*, 2001.

[14] X.-D. Tian, H.-Y. Li, X.-F. Li, and L.-P. Zhang. Research on symbol recognition for mathematical expressions. In *Proceedings of the First International Conference on Innovative Computing, Information and Control*, 2006.

[15] Hyun-Chul Kim, Shaoning Pang, Hong-Mo Je, Daijin Kim, and Sung Yang Bang. Constructing support vector machine ensemble. *Pattern Recognition*, 36:2757–2767, 2003.

[16] T.V. Raman. Audio system for technical readings. Technical report, Cornell University, 1994.

[17] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:2701–2707, 2011.