

© 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Ensemble Classifier Composition: Impact on Feature Based Offline Cursive Character Recognition

Ashfaque Rahman and Brijesh Verma

Abstract— In this paper we propose different ensemble classifier compositions and investigate their influence on offline cursive character recognition. Cursive characters are difficult to recognize due to different handwriting styles of different writers. The recognition accuracy can be improved by training an ensemble of classifiers on multiple feature sets focussing on different aspects of character images. Given the feature sets and base classifiers, we have developed multiple ensemble classifier compositions using three architectures. Type-1 architecture is based on homogeneous base classifiers and Type-2 architecture is composed of heterogeneous base classifiers. Type-3 architecture is based on hierarchical fusion of decisions. The experimental results demonstrate that the presented method with best composition of classifiers and feature sets performs better than existing methods for offline cursive character recognition.

I. INTRODUCTION

Off-line Cursive Character Recognition refers to the process of transforming the image of a handwritten character into a representation (e.g. character class, ASCII code etc.) that is useful for different computer applications. Character recognition is a part of handwriting recognition process with a number of real world applications including postal address recognition, document authentication, and interpreting historical documents. The applications are of significant interest in various financial institutions, postal services, and law enforcement agencies.

Despite its potential applications, the recognition of cursive handwritten characters has not been satisfactory in comparison [1]–[7]. Diversity of handwriting styles is identified to be the main reason for the unsatisfactory recognition rate. This brings in a lot of research in recent times [8]–[17] to investigate features for cursive character recognition invariant to handwriting styles. A survey of different feature extraction methods is available in [8]–[10]. We aim at identifying a suitable combination of features for character recognition. We thus investigated a set of complementary features that focus on different aspects of a character image. The details on the features are presented at Section II.

While the features focus on particular aspects of characters, the accuracy can be further improved by combining the decisions produced by different features [11]–[12][14]. From classification point of view this is called an

ensemble classifier [20]–[23]. An ensemble classifier generates the base classifiers in a way so that they complement each other and produce better accuracy than their base counterparts. The accuracy depends to a lot of extent on how the feature sets are combined in ensemble architecture. In this paper we investigate a number of ensemble architectures to combine the different feature sets to obtain the best recognition accuracy. We have confined our experiments to C-Cube data set [19] that is widely used in the literature [12][17]–[18].

The research presented in the paper aims to investigate the following: (i) explore different feature extraction methods for cursive character recognition, (ii) explore different ensemble architectures to combine the different feature sets in order to improve accuracy, (iii) identification of classifier and feature set combinations that perform best on C-Cube handwriting data set, (iv) provide a comparison of the presented work and other existing works on C-Cube data set.

The paper is organized as follows. Section II presents the cursive character recognition features investigated in the research. The ensemble architectures are presented in Section III. The experimental platform is presented in Section IV. The investigation results and comparisons appear in Section V. Finally Section VI concludes the paper.

II. HANDWRITING RECOGNITION FEATURES

The features used in the research are presented previously in the literature. We have used a total of nine features as used in [12]. The feature extraction algorithms are presented next.

A. Camastra Features

This feature set consists of local and global features that are extracted from the binary image of a character [17]. The character image is first divided into 4×4 sub-images/cells arranged in a grid. In order to achieve shift invariance some cells are allowed to overlap. The local features are computed from the sub-images. The first feature computes the percentage of foreground pixels in the cell with respect to the total number of foreground pixels in the character image. This is called *grey feature*. Let n_i be the number of foreground pixels in cell i and N be the total number of foreground pixels in the character. The grey feature for cell i is set to n_i/N .

The second feature is a directional feature. Two directions are considered for feature extraction: 0° and 90° i.e. along the horizontal and vertical axis. Let the width and height of

Ashfaque Rahman is with the CQUniversity, Australia (phone: +61749306508; e-mail: a.rahman@cqu.edu.au).

Brijesh Verma is with the CQUniversity, Australia (phone: +61749309058; e-mail: b.verma@cqu.edu.au).

the cell be w_i and h_i pixels respectively. The process identifies the number of black pixels along each row (0° direction) and each column (90° direction) in the cell. The directional feature in cell i is computed as:

$$d_i = \frac{1}{2} \left(1 + \frac{1}{h_i w_i^2} \sum_{j=1}^{h_i} m_j^2 - \frac{1}{h_i^2 w_i} \sum_{j=1}^{w_i} m_j^2 \right) \quad (1)$$

where m_j is the number of black pixels along row/column j .

The feature set also includes two global features that provide information about the shape of the character and the relative position of the character with respect to the baseline. The overall shape information is represented by the width/ratio of the character. The fraction of the character below the baseline acts as the second global feature. A total of $4 \times 4 \times 2 = 32$ local features and 2 global features are computed for each pattern providing a 34 dimensional feature set.

B. Concavities Measurement

Concavity features are computed from a character image in this case [15]. The character image is first normalized to a 18×15 matrix. The image is then partitioned into six zones. A search is conducted for each white pixel in a zone. The search looks for black pixels in four freeman directions. The outcome of the search for each white pixel is coded and incrementally updates a vector of dimension 13 that is initialized with zeros. The vector has two indices: upper and lower index. The upper index represents the number of directions where black pixels were reached and the lower index represents the direction where black pixel was not found. When the search finds black pixel in all four directions an additional search is conducted in four auxiliary directions to make sure that the pixel falls within a contour. When the search for all the white pixels in every zone is finished the vectors are concatenated to produce a feature vector of length $6 \times 13 = 78$.

C. Image Projections

Image projection features are obtained by projecting the image along a particular direction. In this case [14] two types of projections are obtained: diagonal and radial. Image is first normalized to a 32×32 matrix. Diagonal projections are simply the projections of the image onto the two diagonal lines: $+45^\circ$ and -45° . Radial projections are computed by grouping foreground pixels that are at the same radial distance from the centre of the image [14]. The radial distance is measured in units of pixels and thus pixels that are at the same radial distance from the centre lie on the edge of a square. In order to achieve rotational invariance the image is partitioned into four quadrants: top, bottom, left, and right. A total of $32 + 32 = 64$ diagonal projection features and $4 \times 16 = 64$ radial projection features are computed and concatenated to a 128 dimensional projection feature vector.

D. MAT based Gradient Directional Features

The gradient components of a grayscale character image are computed to build this feature set [16]. The binary character

image is converted to gray scale image using Medial Axial Transformation (MAT) algorithm. The grayscale image is then normalized within the range of $[0,1]$. Sobel operators along horizontal and vertical axis is then applied on the normalized image to obtain gradient character images along x and y axis respectively. Gradient magnitude and phase is then computed for each pixel in the gradient character image. Image directions for each pixel with nonzero magnitude are then quantized into one of eight directions at $\pi/4$ intervals. The image is divided into $4 \times 4 = 16$ equally spaced sub-images and the number pixels in each direction of each sub-image are considered as a feature. A total of $16 \times 8 = 128$ features are computed from the entire image.

E. Binary Gradient Directional Features

This feature extraction method for gradient directional features is the same as that of MAT-based directional features with the only exception that no MAT transform is conducted. The gradient features are extracted directly from the binary character image. A feature vector of length 128 is computed from each character image.

F. Median Gradient Features

First few steps of the feature extraction algorithm in this case are different from the MAT-based directional feature computation process. As 2D median filter is first applied on the normalized image. The Robert operators along horizontal and vertical axis is then applied to the filter image to compute gradients along x and y axis. The remaining steps are same as before and 128 dimensional feature vector is computed from a character image.

G. Structural Features

In this feature set, the structural properties of a handwritten character are represented by histograms and profiles [13]. The binary character image is scaled to a 32×32 matrix. The horizontal histogram, the vertical histogram, the radial histogram, the in-out profile and the out-in profile is then computed from the normalized matrix. Horizontal and vertical histograms are computed by the number of foreground pixels in each row (32 rows) and column (32 columns) of the normalized matrix respectively. The radial histogram is computed along 72 directions at five degree intervals by computing the number of foreground pixels. The in-out profile computes the location of the first foreground pixel from the centre along 72 directions at five degree intervals. The out-in profile on the other hand computes the location of the first foreground pixel from the edge along 72 directions at five degree intervals. A total of $32 + 32 + 72 + 72 + 72 = 280$ features are computed for each pattern providing a 280 dimensional feature set.

H. Multi-zone Features

This feature set is computed by dividing the character image into a set of non-overlapping rectangular segments and computing the ratio of foreground and background pixels in each segment [12]. In order to achieve robustness different

division arrangements are considered. In each arrangement the image is divided into $R \times C$ non-overlapping rectangular segments. Following division arrangements are used in the research: $3 \times 1, 1 \times 3, 2 \times 3, 3 \times 2, 3 \times 3, 1 \times 4, 4 \times 1, 4 \times 4, 6 \times 1, 1 \times 6, 6 \times 2, 2 \times 6$ and 6×6 . A total of $3 + 3 + 6 + 6 + 9 + 4 + 4 + 16 + 6 + 6 + 12 + 12 + 36 = 123$ features are computed from each pattern.

I. Edge Maps

Features are extracted in this case [12][14] by first obtaining the edge map in different directions, dividing the modified image into multiple zones and computing the proportion of foreground pixels in each zone. The image thinning operation is applied first and the resulting image is scaled to 25×25 matrix. The Sobel operator is then applied to extract edge maps in four directions: horizontal, vertical, $+45^\circ$ diagonal and -45° diagonal. The original image and the edge maps are then divided into 25 sub-images of 5×5 pixels. The percentage of foreground pixels in each sub-image represents a feature. A total of $5 \times 25 = 125$ features are computed from each pattern.

III. ENSEMBLE CLASSIFIER ARCHITECTURES

In order to investigate the influence of ensemble classifier compositions we have developed three architectures with the feature sets described above. Let μ_1, \dots, μ_M be the M base classifiers and f_1, \dots, f_N be the N feature sets considered in the ensemble classifier architecture. We assume that the M base classifiers are heterogeneous in nature. A base classifier μ_i trained on feature set f_j is indexed by its training parameters defined as ρ_{μ_i, f_j} where $1 \leq i \leq M$ and $1 \leq j \leq N$. The three ensemble architectures we considered in the research are presented in Figure 1, Figure 2, and Figure 3.

In the first architecture (Figure 1) all the feature sets f_1, \dots, f_N are extracted from a pattern and a base classifier μ_i then learns the feature sets separately. The training parameter ρ of the base classifier varies with the feature set. The decisions provided by the base classifiers are fused using majority voting to produce the final classification verdict. Same base classifier (with different training parameters) is used with all the feature sets. This ensemble architecture is suitable to identify the subset of feature sets that is suitable for a particular type of base classifier.

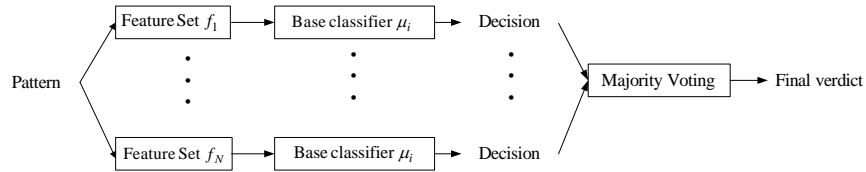


Figure 1: Type-1 ensemble classifier where decisions from a base classifier, trained on the feature sets, are fused using majority voting to the final classification verdict.

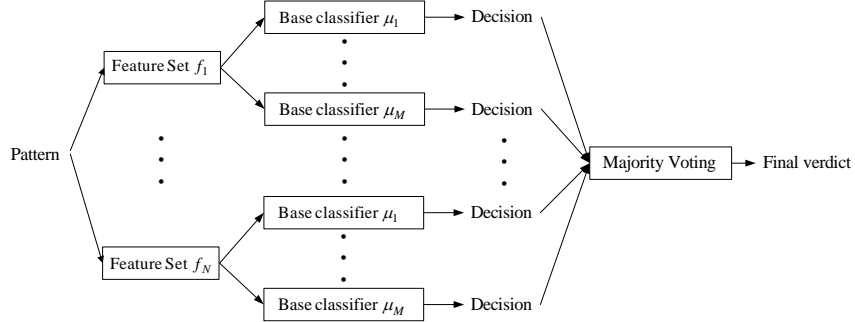


Figure 2: Type-2 ensemble classifier where the decisions from multiple heterogeneous base classifiers trained on the feature sets are fused using majority voting to the final classification verdict.

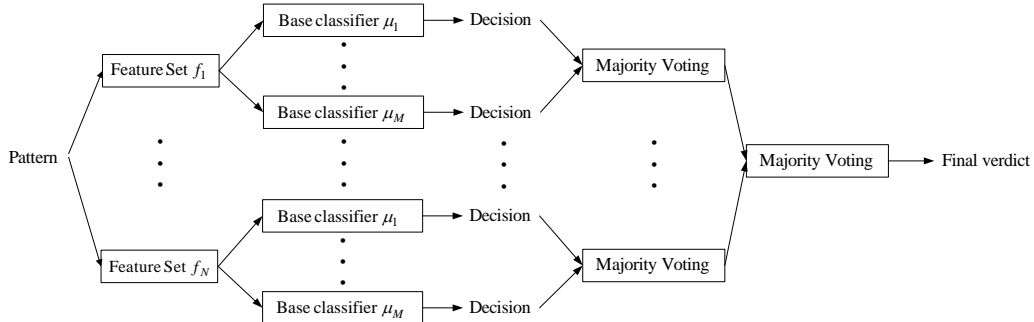


Figure 3: Type-3 ensemble classifier where the decisions from multiple heterogeneous base classifiers trained on the feature sets are fused in two levels using majority voting to the final classification verdict.

In the second architecture (Figure 2) a set of heterogeneous base classifiers are trained on the feature sets extracted from a pattern. The decisions produced by the different base classifiers on the feature sets are fused using majority voting to produce the final classification verdict. This architecture is suitable to identify the subset of feature sets that is suitable for a particular combination of base classifiers of different types.

The third ensemble architecture (Figure 3) is based on hierarchical fusion where decisions are fused at two levels. A set of heterogeneous base classifiers are trained on each feature set. A base classifier provides a decision on a pattern. These decisions produced on a feature set are fused using majority voting in the first level. The decisions produced on all the feature sets in the first level are combined using majority voting at the second level. This architecture is helpful to identify the influence of hierarchical fusion over flat fusion as used in Type-2 architecture.

While evaluating these architectures we used different base classifiers and their combinations. We used different subsets of classifier set $\{\mu_1, \dots, \mu_M\}$ and the feature sets $\{f_1, \dots, f_N\}$ to find the combination that performs best given a data set.

IV. EXPERIMENTAL SETUP

We used C-Cube data set [19] to evaluate the presented ensemble methods. The data set contains 57,293 character images that include both upper case and lower case letters. The division of the data set into training and test set is also provided in [19]. The training set contains 38,160 character images and the test set contains 19,133 character images. There are a number of advantages of using this data set. As the data set has a defined partitioning it becomes easier to compare against other existing works conducted on identical partitioning. The data set also provides some pre-computed features obtained using the algorithm in [17]. There are some global features readily available in the data set that can be used along with other features and as reported in [12] the addition of these global features contribute to 1%–2% improvement in classification accuracy. We thus evaluated the presented methods on the training and test sets of the C-Cube data set.

We used three different base classifiers in the ensemble architecture namely k -Nearest Neighbour (k -NN), Multi Layer Perceptron (i.e. neural network NN) and Support Vector Machine (SVM). We used k within the range of 1 to 9 with the k -NN classifier to obtain the best performing k . The NN is trained in all cases with the following configuration: (i) training algorithm: resilient backpropagation, (ii) activation functions: log sigmoid, (iii) learning rate: 0.001, (iv) momentum: 0.15, (v) epochs: 10000, (vi) goal: 0.0. The number of hidden units was varied within the range of 50 to 500 to obtain the best performances. The SVM was trained in all cases with radial basis kernel. The parameter g was varied within the range of 0 to 10 for obtaining the best classification performance. All

the experiments were conducted on MATLAB.

As mentioned in Section II we used nine feature sets. We used the following feature sets as provided by the authors in [12] in the experiments: *Camastra*, *Binary Gradient*, *Concavities*, *Projection*, *Mat Gradient*, *Median Gradient*, and *Structural*. We developed codes to generate the *Multi zone* and *Edge map* feature sets to use in the experiments.

V. RESULTS AND DISCUSSION

We have conducted a number of experiments to evaluate the presented methods and also to compare the performance against the existing works. We have evaluated all possible combinations of the classifiers and feature sets on the C-Cube data set and in the section we present the best performing combinations. The test set is evaluated under two scenarios: (i) Upper and lower case letters considered as separate classes i.e. 52 classes, and (ii) Upper and lower case letters considered as a unique class i.e. 26 classes. We have presented the classification accuracies for both cases in the following sections. The diversity among the base classifiers were obtained using KW-variance [24][24].

A. Single classifier Single Feature Evaluation

In this experiment we have trained each classifier separately on each feature set. Table I represents the test accuracies obtained for different feature sets with different classifiers. In general SVM performs better than the other two classifiers. Some of the feature sets namely *Camastra*, *Binary gradient*, and *Projection* performs better than other feature sets for all classifiers. The training parameters namely k , HU (Hidden Units), and g corresponding to best performing base classifiers is presented in Table II. In general higher values of k reveal better performance for the k -NN classifier. The hidden units were small for some feature set (e.g. 50 for *Concavities*) and large for others (e.g. 500 for *Camastra*). The value of g also varied a lot with 0.1 for feature set like *Structural* and 6.2 for feature set like *MAT gradient*.

B. Type-1 Ensemble Classifier with Multiple Feature Sets

We have evaluated Type-1 ensemble classifier by using the k -NN, NN, and SVM classifiers separately on different feature sets. All possible subsets of nine feature sets were used with each classifier separately under ensemble architecture Type 1. The cardinality of the sub sets are varied from 3 to 9 and the best performing combinations for each base classifier are presented in Table III. Ensemble classifiers with k -NN, NN and SVM classifier performs best with 8, 8 and 7 feature sets achieving 83.59%, 88.14% and 88.89% accuracies respectively. SVM ensemble performs better than the other two ensembles. Note that diversity always increases with increasing number of feature sets.

C. Type-2 Ensemble Classifier with Multiple Feature Sets

In this experiment we have evaluated Type-2 ensemble classifier by using the k -NN & NN, k -NN & SVM, NN & SVM, and k -NN & NN & SVM classifier combinations on

different feature sets. All possible subsets of the nine feature sets were used with each of these three classifier combinations separately under ensemble architecture Type 2. The feature sub set length varies from 2 to 9 for dual classifier combinations and varies within the range of 1 to 9 for the three classifier combinations. The best performing combinations are presented in Table IV. Ensemble classifiers with k -NN & NN, k -NN & SVM, NN & SVM, and k -NN & NN & SVM classifier combinations perform best with 6, 5, 6 and 6 feature sets achieving 87.90%, 88.03%, 89.93% and 89.17% accuracies respectively. NN & SVM ensemble performs better than the other three ensembles. The diversity increases with increasing number

of feature sets as well.

D. Type-3 Ensemble Classifier with Multiple Feature Sets

In this experiment we have evaluated Type-3 ensemble classifier. At least three classifiers are required for the purpose of majority voting and we thus used k -NN, NN and SVM. All possible subsets of nine feature sets were used with these classifier combinations. The decisions produced by these three classifiers for each feature set were fused first using majority voting. These first level decisions were fused in a second level using majority voting. The best performing combination is presented in Table V where a total of 8 feature sets achieve 88.04% classification accuracy.

Table I: Classification accuracy of individual classifiers

Table I: Classification accuracy of individual classifiers								
Feature Set	k -NN		NN		SVM			
	Up	Low Separate	Up	Low Joined	Up	Low Separate	Up	Low Joined
<i>Binary Gradient</i>	70.52		77.89		73.21		79.91	
<i>Camastra</i>	72.44		79.68		75.00		81.95	
<i>Concavities</i>	67.60		74.44		73.38		80.24	
<i>Projection</i>	56.75		63.58		67.06		73.26	
<i>Mat gradient</i>	64.15		71.26		73.03		79.89	
<i>Median Gradient</i>	68.66		75.68		71.94		78.66	
<i>Structural</i>	62.89		70.22		69.31		76.04	
<i>Multi zone</i>	67.06		74.11		72.57		79.25	
<i>Edge map</i>	61.82		68.82		65.90		71.98	

Table II: Best performing parameters of individual classifiers

Feature Set	k -NN	NN	SVM
	k	HU	g
<i>Binary Gradient</i>	8	250	1.3
<i>Camastra</i>	8	500	2.1
<i>Concavities</i>	7	50	0.9
<i>Projection</i>	9	250	0.1
<i>Mat gradient</i>	8	300	6.2
<i>Median Gradient</i>	9	200	4.0
<i>Structural</i>	9	250	0.1
<i>Multi zone</i>	7	200	0.3
<i>Edge map</i>	9	250	0.3

Table III: Best performing Type-1 ensemble classifiers

Classifier	No. of features	Best performing feature set	Accuracy		Diversity		
			Up	Low Separate		Up	Low Joined
k -NN	8	<i>Camastra, Binary Gradient, Concavities, Mat Gradient, Median Gradient, Structural, Multi zone, Edge map</i>	76.16		83.59		0.1044
NN	8	<i>Camastra, Binary Gradient, Concavities, Mat Gradient, Median Gradient, Structural, Multi zone, Edge map</i>	81.09		88.14		0.0962
SVM	7	<i>Camastra, Binary Gradient, Concavities, Mat Gradient, Median Gradient, Multi zone, Edge map</i>	82.27		88.89		0.0783

Table IV: Best performing Type-2 ensemble classifiers

Classifier	No. of features	Best performing feature set	Accuracy		Diversity
			Up Low Separate	Up Low Joined	
k -NN & NN combo	6	<i>Camastra, Binary Gradient, Concavities, Mat Gradient, Median Gradient, Multi zone</i>	80.67	87.90	0.1034
k -NN & SVM combo	5	<i>Camastra, Binary Gradient, Concavities, Median Gradient, Multi zone</i>	81.21	88.03	0.0892
NN & SVM combo	6	<i>Camastra, Binary Gradient, Concavities, Mat Gradient, Median Gradient, Multi zone</i>	83.03	89.93	0.0884
k -NN & NN & SVM combo	6	<i>Camastra, Binary Gradient, Concavities, Mat Gradient, Median Gradient, Multi zone</i>	82.19	89.17	0.0991

Table V: Best performing Type-3 ensemble classifiers

Classifier	No. of features	Best performing feature set	Accuracy		Diversity	
			Up	Low Separate		Up
k -NN & NN & SVM combo	8	<i>Camastra, Binary Gradient, Concavities, Projection, Mat Gradient, Median Gradient, Structural, Multi zone</i>	81.05		88.04	0.0788

Table VI: Base classifier ranking in terms of average classification accuracy

	k -NN		NN		SVM	
	Up Low Separate	Up Low Joined	Up Low Separate	Up Low Joined	Up Low Separate	Up Low Joined
<i>Average</i>	65.77	72.85	71.27	77.91	74.49	81.10
<i>Rank</i>	3		2		1	

Table VII: Base classifier ranking in terms of feature set

Feature Set	Average		Rank
	Up Low Separate	Up Low Joined	
<i>Binary Gradient</i>	74.12	81.07	2
<i>Camastra</i>	75.60	82.51	1
<i>Concavities</i>	72.73	79.52	3
<i>Projection</i>	64.61	71.12	9
<i>Mat gradient</i>	70.16	77.02	6
<i>Median Gradient</i>	71.76	78.51	4
<i>Structural</i>	68.18	75.18	7
<i>Multi zone</i>	71.21	77.99	5
<i>Edge map</i>	66.19	72.67	8

E. Performance Comparison

The best performing ensemble classifiers of Type-1, Type-2, and Type-3 under different base classifier combinations are presented in Table III, Table IV, and Table V. Type-3 architecture performs worse than the two other architectures. This is due to the fact that the first level fusion is conducted only on three decisions and there were little diversity among the base classifiers.

NN & SVM combination under Type-2 ensemble architecture performs better than any other ensemble classifier. It can be observed that the best accuracy do not correspond to best diversity. The best performance of the NN & SVM combination over the others can be explained from the rankings (1 being the best) of base classifiers and feature sets in Table VI and Table VII. The NN & SVM combination is composed of best six feature sets. Out of the three base classifiers k -NN performs the 6.94% and 11.32% worse than NN and SVM respectively. Inclusion of k -NN thus influences the reduction of accuracy. This is why the ensemble classifiers including k -NN & NN combination, k -NN & SVM combination, and k -NN & NN & SVM combination performs inferior to NN & SVM combination. Edge map is the second worst performing feature set as evidenced from Table VII. Edge map feature set is involved in best performing k -NN combo, NN combo and SVM combo. This leads to inferior performance of these ensemble classifiers against NN & SVM combo.

The best performance of the presented ensemble classifier is compared against that of some published works on C-Cube data set and presented in Table VIII. Note that the presented method performs better than others. The performance of the presented ensemble classifier is better than that in [12] because of a number of reasons. Only NN classifiers are used in [12] and as per the performance analysis in, NN combo performs inferior to NN & SVM combo. All the nine feature sets are used in [12]. Inclusion of *Edge map*, *Projection* and *Structural* feature sets reduces classification accuracy of the ensemble. The presented

ensemble outperforms all the other pervious works. This is because these previous works are based on either NN or SVM only. The presented ensemble uses combination of NN and SVM and as evidenced empirically they are better than other combinations.

VI. CONCLUSION

In this paper we have investigated different ensemble classifier architectures with a combination of a number of feature sets for cursive character recognition. We can draw a number of conclusions from the discussions presented above about the C-Cube data set: (i) Based on the average classification performance *Camastra* feature set performs best (82.51%) and *Projection* feature set performs worst (71.12%); (ii) The NN & SVM classifier combination with *Camastra*, *Binary Gradient*, *Concavities*, *Mat Gradient*, *Median Gradient*, *Multi zone* achieves 89.93% classification accuracy that is better than any other base classifier and feature set combination. The inferior performance of other combinations is due to the inclusion of either k -NN classifier (Rank: 3, lowest) or *Edge map* (Rank: 8, second lowest) feature set in them. (iii) The presented method performs better than currently available methods in the literature. Use of only one type (NN/SVM) of base classifier and use of all the features independent of their rank can be attributed to the inferior performance of the existing methods. In future we would like to incorporate the presented method with cursive word recognition to improve accuracy.

Table VIII: Performance comparison of presented and existing methods

Method	Accuracy
34D-SVM + Neural GAS [17]	86.20
34D-MLP [17]	71.42
MLP Ensemble [12]	89.34
SVM-26 class [18]	89.61
This paper	89.93

REFERENCES

- [1] S. N. Srihari, "Automatic handwriting recognition", Encyclopedia of Language & Linguistics, Elsevier, 2006.
- [2] B. Gatos, I. Pratikakis, A.L. Kesidis, S.J. Perantonis, "Efficient off-line cursive handwriting word recognition", Tenth International Workshop on Frontiers in Handwriting Recognition, 2006.
- [3] A. L. Koerich, A. S. Britto, L. E. S. de Oliveira and R. Sabourin, "Fusing high- and low-level features for handwritten word recognition", Tenth International Workshop on Frontiers in Handwriting Recognition, 2006.
- [4] N. R. Howe, T. M. Rath and R. Manmatha, "Boosted decision trees for word recognition in handwritten document retrieval", The 28th SIGIR Conference on Research & Development in Information Retrieval, pp. 377-383, 2005.
- [5] J. Sadri, C. Suen, and T. D. Bui, "A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings", Pattern Recognition, vol. 40, pp. 898 – 919, 2007.
- [6] H. Lee and B. Verma, "Binary Segmentation with Neural Validation for Cursive Handwriting Recognition," IEEE International Joint Conference on Neural Networks, pp. 1730–1735, 2009.
- [7] H. Lee and B. Verma, "Over-segmentation and Neural Binary Validation for cursive handwriting recognition," IEEE International Joint Conference on Neural Networks, pp. 1–5, 2010.
- [8] N. Arica and F.T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 31, May. 2001, pp. 216–233.
- [9] H. Fujisawa, "Forty years of research in character and document recognition--an industrial perspective," Pattern Recognition, vol. 41, Aug. 2008, pp. 2435–2446.
- [10] N. Arica and F.T. Yarman-Vural, "Optical Character Recognition for Cursive Handwriting," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, Jun. 2002, pp. 801–813.
- [11] B. Verma, P. Gader, and W. Chen, "Fusion of multiple handwritten word recognition techniques," Pattern Recognition Letters, vol. 22, Jul. 2001, pp. 991-998.
- [12] R. M. O. Cruz, G. D. C. Cavalcanti, and T. I. Ren, "An ensemble classifier for offline cursive character recognition using multiple feature extraction techniques," International Joint Conference on Neural Networks (IJCNN), pp. 744–751, Barcelona, Spain, 2010.
- [13] E. Kavallieratou, K. Sgarbas, N. Fakotakis, and G. Kokkinakis, "Handwritten word recognition based on structural characteristics and lexical support," International Conference on Document Analysis and Recognition, pp. 562–567, 2003.
- [14] Y. –C. Chim, A. A. Kassim, and Y. Ibrahim, "Dual classifier system for handprinted alphanumeric character recognition," Pattern Analysis and Applications, vol. 4, no. 1, pp. 155–162, 1998.
- [15] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Automatic recognition of handwritten numerical strings: a recognition and verification strategy," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 24, no. 11, pp. 1438–1454, 2002.
- [16] P. Zhang, "Reliable recognition of handwritten digits using a cascade ensemble classifier system and hybrid features," Ph. D. Thesis, Concordia University, Montreal, P. Q., Canada, 2006.
- [17] F. Camastra, "A svm-based cursive character recognizer," Pattern Recognition, vol. 40, no. 12, pp. 3721–3727, 2007.
- [18] F. Camastra, M. Spinetti, and A. Vinciarelli, "Offline Cursive Character Challenge: a New Benchmark for Machine Learning and Pattern Recognition Algorithms," International Conference on Pattern Recognition (ICPR), pp. 913–916, 2005.
- [19] C-Cube, <http://www.idiap.ch/project/ccc/>, Last accessed January, 2011.
- [20] A. Rahman and B. Verma, "A Novel Ensemble Classifier Approach using Weak Classifier Learning on Overlapping Clusters," IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1–5, 2010.
- [21] R. Polikar, "Ensemble based systems in decision making," IEEE Circuits and Systems Magazine, vol. 6, no. 3, pp. 21–45, 2006.
- [22] L. Chen and M. S. Kamel, "A generalized adaptive ensemble generation and aggregation approach for multiple classifiers systems," Pattern Recognition, vol. 42, pp. 629–644, 2009.
- [23] A. Rahman and B. Verma, "A novel ensemble classifier approach using weak classifier learning on overlapping clusters," The International Joint Conference on Neural Networks (IJCNN), pp. 1–7, 2010.
- [24] L. I. Kuncheva and C. J. Whitaker, "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy," Machine Learning, vol. 51, no.2, pp. 181–207, 2003.
- [25] R. Kohavi and D. H. Wolpert, "Bias plus variance decomposition for zero-one loss functions," Proc. Int. Conf. on Machine Learning, pp. 275–283, Bari, Italy.