A KNN-SVM Hybrid Model for Cursive Handwriting Recognition

Cleber Zanchettin, Byron Leite Dantas Bezerra and Washington W. Azevedo

Abstract—This paper presents a hybrid KNN-SVM method for cursive character recognition. Specialized Support Vector Machines (SVMs) are introduced to significantly improve the performance of KNN in handwritten recognition. This hybrid approach is based on the observation that using KNN in the task of handwritten characters recognition, the correct class is always one of the two nearest neighbors of the KNN. Specialized local SVMs are introduced to detect the correct class among these two different classification hypotheses. The hybrid KNN-SVM recognizer showed significant improvement in terms of recognition rate compared with MLP, KNN and a hybrid MLP-SVM approach for a task of character recognition.

I. INTRODUCTION

Handwriting is one of the most natural communication method among people. The handwritten character recognition has been a concern of the scientific community [29] [12]. This is a complex task due to variations of existing styles in handwriting, the personal style of each writer, since for each writer the form of a character can be done in different ways and even the same writer can represent the character in different ways [12]. Another problem in recognizing handwritten characters is the similarities between different characters, for example, U and V, Q and O, among others.

The handwriting recognition task is traditionally divided into online and offline recognition. In online recognition a time series of coordinates, representing the movement of the pen-tip, is captured, while in the offline case only an image of the text is available. Because of the greater ease of extracting relevant features, online recognition generally yields better results [36]. Another important distinction is between recognizing isolated characters or words, and recognizing whole lines of text. Lastly, handwriting recognition can be split into cases where the writing style is constrained in some way, for more challenging scenario where it is unconstrained. Despite more than 30 years of handwriting recognition research [36], [29], [44], [6], developing a reliable, general-purpose system for unconstrained text line recognition remains an open problem.

In this article, working with the problem of handwriting character recognition, we present the advantage of using Support Vector Machines (SVMs) [42] to improve the performance of an ICR system (Intelligent Character Recognition) based on KNN. This simple combination seems to be promising when compared to more complex classifiers and is investigated as an alternative to build system for unconstrained text line recognition.

In Section II is presented a description of character recognition and we point out the main problems of handwriting recognition. In Section III are presented the most common types of classifiers used for character recognition found in the literature. Section IV describes the database used in experiments. In Section V is presented the proposed technique. In Section VI experiments and results are presented. The conclusion is given in the final section.

II. HANDWRITTEN CHARACTER RECOGNITION

Character recognition is a classification task that has been studied extensively by the scientific community since the invention of the computer [6] and consists in recognizing a set of characters from an image, separating them into 10 classes, in the case of digits, or 26 classes, in the case of letters of the Western alphabet.

There are some problems that hinder the implementation of character recognition. In some cases the scanned image is of low quality, thus is necessary to perform a preprocessing to eliminate image noises. Another problem that occurs is the existence of distorted characters, especially when dealing with handwritten documents, due to the characteristics of the writer's handwriting.

Moreover, another difficulty to consider is the similarity between some characters, such as "I" and "J", "Q" and "O", "U" and "V", among others, that may hinder the classification of recognized characters. In Figure 1 is presented a sample of characters with confusion.

Fig. 1. Characters with structural similarity.

To build an efficient handwritten recognition system is important to choose correctly the extracted features to be used. In [33] is performed a study to evaluate a set of feature extraction techniques proposed in [34] for handwritten letters recognition. Additionally, the authors proposed a technique based on the projection of the image outline on the sides of a regular polygon built around each character. The
feature vector is formed by the perpendicular distances taken from each side of the polygon to the contour of the image. In the evaluation process the proposed approach is compared using two polygons (square and hexagon) and two different amounts of projection lines taken from each side of the polygon, with two versions of coding bit maps (standard and tuned). The discriminatory power of each case is examined through the use of a MLP neural network (Multi-Layer-Perceptron).

In [41] is proposed a methodology based on a new feature extraction technique based on recursive subdivision of the character image so that the result of sub-images in each iteration has a balanced number (approximately equal) of pixels in the foreground, as far as it is possible. In the experiments two databases of handwritten characters (CEDAR and LIC) and two databases of handwritten digits were used (MNIST and CEDAR). The classification step was performed using Support Vector Machine (SVM) with Radial Basis Function (RBF).

In [10] the authors present a new approach for recognizing cursive characters using multiple feature extraction algorithms and an ensemble classifier. Several feature extraction techniques, using different approaches, were evaluated. Two techniques, Modified Edge and Multi Zoning Maps, were proposed. Based on the results, a combination of feature sets is proposed in order to achieve high recognition performance. This combination was motivated by the observation that the sets of characteristics are independent and complementary. The ensemble was conducted by combining the outputs generated by the classifier in each set of features separately. The database used for the experiments was the C-Cube, a three-layer MLP network, trained with the Resilient Backpropagation.

In [1] an hybrid MLP-SVM method for unconstrained handwritten digits recognition is proposed. This hybrid architecture is based on the idea that the correct digit class almost systematically belongs to the two maximum MLP outputs and that some pairs of digit classes constitute the majority of MLP substitutions (errors). Specialized local SVMs are introduced to detect the correct class among these two classification hypotheses. The hybrid MLP-SVM recognizer achieves a recognition rate of 98.01% in real mail zipcode digits recognition task, a performance better than several classifiers reported in recent researches.

In [7] the authors present a cursive character recognizer that performs the character classification using SVM and neural gas. The neural gas is used to verify whether lower and upper case version of a certain letter can be joined in a single class or not. Once this is done for every letter, the character recognition is performed by SVMs. SVMs compare notably better, in terms of recognition rates, with popular neural classifiers, such as learning vector quantization and MLP. SVM recognition rate is among the highest presented in the literature for cursive character recognition.

In [34] and [41] was observed similarities between some letters (e.g. ‘B and D’, ‘H and N’, ‘O and Q’). In [10] was detected a high error rate in characters that have two completely different ways of writing (e.g., ‘a and A’, ‘f and F’). In [7] the author uses SVM classifiers to handwrite recognition.

### III. Handwritten Classifiers

Statistical classifiers and artificial neural networks have been the most widely used classification methods [24], [25]. In statistical classifiers, the features are of the form n-tuples or vectors. The purpose of these classifiers is to estimate the probability of belonging to a character analysis of the possible classes. The techniques used for classification can be divided into parametric and nonparametric. Parametric classifiers include Linear Discriminant Function (LDF) [11] and Quadratic Discriminant Function (QDF) [11], for example. Among the nonparametric classifiers the K-Nearest Neighbor [42][40] and decision trees are largely used.

The kNN is a statistical approach without training phase. The main idea is to compare a training database with the pattern X to be classified. The k patterns in the database nearest to the pattern X are analyzed. The most frequent class in these k patterns will be the class of X pattern.

To compare the patterns, it is common to use the Euclidian distance (1), Manhattan distance (2) or Minkwoski distance (3). The k constant is a small number. The distances equations are:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$  \hspace{1cm} (1)

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$  \hspace{1cm} (2)

$$d(x, y) = \left( \sum_{i=1}^{n} (x_i - y_i)^p \right)^{\frac{1}{p}}$$  \hspace{1cm} (3)

Where $x$ is the feature vector of the pattern that we want to recognize, $y$ a pattern of the training database, $n$ the feature vector size and $p$ is a constant defined empirically.

The hidden Markov chains HMM (Hidden Markov Models) are statistical classifiers that perform a double stochastic process, being a fundamental stochastic process is not observable (hidden), but it can be observed through another set of stochastic processes that yields the subsequent observations [31]. Applications of HMM initially appeared in speech recognition [21] and were later extended to model characters [5] and words [49] [13].

Artificial neural networks are another widely used for handwritten recognition. The used models include the multilayer perceptron, networks of radial basis function, probabilistic neural networks, support vectors machines and polynomial classifiers [25]. Besides the classifier, neural networks can also be used to extract features and to do so the images of its patterns are directly inserted in its entry.

The main difference between statistical classifiers and artificial neural networks is that the network parameters are optimized in a supervised discriminative learning process that seeks to separate the patterns into different classes [16]. When the network structure is properly assigned and the training set has many elements, the artificial neural networks are able to provide a high accuracy in pattern classification of unknown test set, however, training is slow.
and runs the risk of network loses its ability to spread and become over-specialized [24], [25].

The classifier LVQ (Learning Vector Quantization), used in some studies [24], [25] is considered hybrid since it uses the 1-NN rule for classification and adjusts its parameters in the same way that neural networks. A new type of classifier, SVM (Support Vector Machine) has emerged in this area and therefore appears in some recent works [37], [28], [22], [46]. SVM is based on statistical learning theory and optimization of quadratic programming. Because it is a binary classifier, multiple SVMs may be combined to form a classification system [24], [22], [46]. Systems that employ SVMs have outperformed the results obtained using traditional techniques such as artificial neural networks, however, they require more memory and have higher computational cost. Therefore, hybrid approaches have been employed, using a combination of classifiers with simpler classifiers at the beginning and more complex in later levels. Thus, SVM would be responsible for resolving the most difficult cases that are rejected by the simpler classifiers [45].

In recent years much research has resulted in the design of classification systems using different approaches for combining multiple classifiers [26], [23] [32]. The combination method can use local performance estimates [47], Local Learning Algorithms [4], Adaptive Mixtures of Local Experts [20] or to aggregate the decisions obtained from individual classifiers to obtain better final decisions from a statistics standpoint statistics [18]. Another feature used to increase the effectiveness of the systems is the use of a modular classifier, which consists of multiple classifiers, each one specialized in a class of problem [27].

IV. C-CUBE DATABASE

The C-Cube is a public database available for download on the Cursive Character Challenge website (http://ccc.idiap.ch). The database consists of 57293 files, including uppercase and lowercase letters, manually extracted from the CEDAR and United States Post Service (USPS) databases. As reported by Camastra [8], this database has three advantages:

1) The database is already divided into training and test sets, so the results of different researchers can be compared rigorously;
2) The database contains not only images but also their feature vectors extracted using the algorithm proposed by Camastra [7];
3) The results obtained using the methods of the state of the art still leave significant space for significant improvement.

The dataset is divided into 38160 (22274 lower case and 15886 upper case) images for training and 19133 (11161 lower case and 7972 upper case) images for test. All images are binary and with variable size. For each image, additional information are provided such as distance between baseline and upper line, distance of the upper extreme from the baseline and distance of the lower extreme from the baseline. The number of samples for each class is variable and was selected according to its frequency in documents extracted from the CEDAR and USPS datasets. Figures 2 and 3 show the distribution of the letters in the lower and upper case versions, respectively. It can be seen that there is a big difference in the number of pattern among the letters.

Fig. 2. Lower case letter distribution in the C-Cube Database [33].

Fig. 3. Upper case letter distribution in the C-Cube Database [33].

To perform the experiments each character image passed a feature extraction process that can be defined by the extraction of the most important information to perform the classification. There are several feature extraction techniques proposed in literature and its choice can be considered the most important factor to achieve high accuracy rates [40].

In [10] two different experiments were made (All the experiments were conducted using three layers MLP trained using the Resilient Backpropagation algorithm): First splitting upper and lower cases and other experiment with both. For the latter, as some letters present the same shape in both upper and lower cases versions, they were joined into a single class. Camastra [7] used a clustering analysis to verify whether the upper and lower case versions of the same
letters are similar in shape. The letters (c, x, o, w, y, z, m, k, j, u, n, f, v) presented the highest similarity between the two versions and were joined into a single class.

The classification results for the split and joined cases are shown in Tables I and II, respectively. The results are ordered by the recognition rates. The proposed modified Edge Maps algorithm presented the overall best result. Most feature sets presented better accuracy for the upper case letters with the exception of the method proposed by Camastra that performed better for lower letter case. This feature set also presented the best accuracy (84.37%) for the lower letter case. It can be seen that the methods based on gradients and the modified edge maps presented the best results. These methods have in common the use of directional information. The Camastra 34D feature set also uses directional features.

<table>
<thead>
<tr>
<th>Method</th>
<th>No. Nodes</th>
<th>Upper Case (%)</th>
<th>Lower Case (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge</td>
<td>490</td>
<td>86.52</td>
<td>81.13</td>
</tr>
<tr>
<td>Binary Grad.</td>
<td>490</td>
<td>86.35</td>
<td>79.89</td>
</tr>
<tr>
<td>MAT Grad.</td>
<td>300</td>
<td>85.77</td>
<td>79.22</td>
</tr>
<tr>
<td>Median Grad.</td>
<td>360</td>
<td>85.10</td>
<td>79.48</td>
</tr>
<tr>
<td>Camastra 34D</td>
<td>400</td>
<td>79.63</td>
<td>84.37</td>
</tr>
<tr>
<td>Zoning</td>
<td>450</td>
<td>84.46</td>
<td>78.07</td>
</tr>
<tr>
<td>Structural</td>
<td>320</td>
<td>81.94</td>
<td>77.70</td>
</tr>
<tr>
<td>Concavities</td>
<td>530</td>
<td>73.35</td>
<td>81.89</td>
</tr>
<tr>
<td>Projections</td>
<td>500</td>
<td>71.73</td>
<td>79.90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>No. Nodes</th>
<th>Recognition Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge</td>
<td>490</td>
<td>82.49</td>
</tr>
<tr>
<td>Binary Grad.</td>
<td>490</td>
<td>81.46</td>
</tr>
<tr>
<td>MAT Grad.</td>
<td>300</td>
<td>80.83</td>
</tr>
<tr>
<td>Median Grad.</td>
<td>360</td>
<td>79.96</td>
</tr>
<tr>
<td>Camastra 34D</td>
<td>400</td>
<td>79.97</td>
</tr>
<tr>
<td>Zoning</td>
<td>450</td>
<td>78.60</td>
</tr>
<tr>
<td>Structural</td>
<td>320</td>
<td>77.07</td>
</tr>
<tr>
<td>Concavities</td>
<td>530</td>
<td>74.90</td>
</tr>
<tr>
<td>Projections</td>
<td>500</td>
<td>73.85</td>
</tr>
</tbody>
</table>

V. AN HYBRID ARCHITECTURE COMBINING KNN-SVM

One of the problems hindering the task of handwritten character recognition, is the similarity between different characters (for example, O and Q, U and V, among others) causing confusion at moment of classifying such patterns.

The main idea is to increase the KNN recognition rate, sensible to different classes with similar attributes, using the SVM as a decision classifier, similar to Bellili et al. proposal [1]. The adaptation here is to get the two most frequently classes in the k nearest neighbor and use the SVM to decide between these two classes.

It is a good technique to be used where one wrong classification brings high costs because it uses the two best classical techniques for it. As it depends of kNN, the main disadvantage is the processing time. The KNN-SVM hybrid model is shown in Figure 4.

This combination of methods results in the specialization of SVMs in the local areas around the surface of separation between each pair of characters that constitute the majority of the errors of the kNN (confusion). Thus, SVMs are introduced only for pairs of classes that constitute the greatest confusion of kNN. This process is detailed in the next section.

VI. EXPERIMENTS AND RESULTS

In this section are published the results of several experiments performed in the C-Cube test dataset. All of them follows the same experimental methodology of previous proposed work whose results were reported over this same dataset. Therefore, it is possible to compare the proposed method against other ones. Additionally, the preliminary and final experiments follow a cross validation methodology in order to grant a high confidence for the reported rates.

A. KNN adjustment process

The k parameter was adjusted empirically based on the error of the expected result of the classifier for each k tested. Normally, the k parameter is varied from 1 to the square root of cardinality of the dataset. After several simulations, it was chosen the value 13 for k, because this value yields the best classification rates, independently of the letter case (upper, lower or mixed), and the usage or not of some feature extraction technique.

Concerning the distance adopted in the kNN method, the Euclidean distance was chosen in the experiments where the character feature extraction technique proposed by Camastra [7] were used. Otherwise, the following procedure was adopted:

- the input character image is scaled to 30 pixels of height and 35 pixels of width, given a bi-
dimensional vector of 1050 pixels;
- each element of the vector is set to 0 if the respective pixel in the image is white, and 1 for a black pixel;
- so, two binary vectors of 1050 elements each, are the input to a function which measures the Jaccard distance [40] between those images.

Two different experiments were performed: at first dividing upper and lower case, and another experiment with both cases. The average results for the spitted and combined cases are presented in Tables III and IV.

### TABLE III
**RECOGNITION RATE FOR THE UPPER AND LOWER CASE SEPARATED**

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Distance function</th>
<th>Case</th>
<th>Recognition rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>kNN</td>
<td>Euclidean</td>
<td>Uppercase</td>
<td>88.65</td>
</tr>
<tr>
<td>kNN</td>
<td>Euclidean</td>
<td>Lowercase</td>
<td>81.07</td>
</tr>
<tr>
<td>kNN</td>
<td>Jaccard</td>
<td>Uppercase</td>
<td>75.25</td>
</tr>
<tr>
<td>kNN</td>
<td>Jaccard</td>
<td>Lowercase</td>
<td>72.24</td>
</tr>
</tbody>
</table>

### TABLE IV
**RECOGNITION RATE FOR THE JOINT CASE**

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Distance function</th>
<th>Case</th>
<th>Recognition rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>kNN</td>
<td>Euclidean</td>
<td>Uppercase</td>
<td>82.26</td>
</tr>
<tr>
<td>kNN</td>
<td>Jaccard</td>
<td>Uppercase</td>
<td>70.55</td>
</tr>
</tbody>
</table>

**B. Analysis of confusion**

With the kNN model trained for each case (uppercase, lowercase and uppercase + lowercase) their confusion matrices were generated. Figure 5 illustrates the part referring to major confusion occurred in experiments with uppercase letters. Due to the large amount of classes, 26 classes for the case uppercase/lowercase and 52 classes for the uppercase + lowercase, it is impractical to display the complete confusion matrices for all cases. Therefore, we can analyze pairs of characters from which the classifier achieved the highest number of confusion.

**C. SVM training process**

Different SVMs were derived from pairs of the classes (e.g. (U, V), (m, n), (N, n) ... etc.) constituting the majority of the confusions observed with the kNN classifier, as shown in figure 6. Different kernel functions (linear, polynomial and RBF) were tested and the best performances were obtained by trained SVMs with the RBF kernel function.

**D. Hybrid architecture kNN-SVM**

Based on the confusion matrices analyzed, the hybrid architecture, taking into account the kNN classifier (with k=13) based on the Euclidean distance and Camasta [7] feature extraction technique, was constructed as follows:

- Uppercase: kNN + 23 SVMs;
- Lowercase: kNN + 32 SVMs;
- Uppercase+ Lowercase: kNN + 33 SVMs.

The choice of pairs of classes was based on the amount of errors taking as minimum 10% the size of the test set.

**TABLE V**

<table>
<thead>
<tr>
<th>Case</th>
<th>kNN (%)</th>
<th>kNN+SVM (%)</th>
<th>Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uppercase</td>
<td>88.65</td>
<td>89.65</td>
<td>1.00</td>
</tr>
<tr>
<td>Lowercase</td>
<td>81.07</td>
<td>84.68</td>
<td>3.61</td>
</tr>
<tr>
<td>Uppercase+</td>
<td>82.26</td>
<td>83.76</td>
<td>1.50</td>
</tr>
<tr>
<td>Lowercase</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These results show that our hybrid kNN-SVM recognizer improves significantly the performance in terms of recognition and error rate compared with a single kNN model for characters classification task. One can also observe that the results obtained by our method were better than the recognition rates of all the technique of extracting features presented in Tables I and II. It is worth emphasizing the simplicity and the training speed, compared with techniques of extracting features more complex [49] requiring, in most cases, high computational cost and longer training.

The best results obtained in recent years, for C-Cube the database, are displayed in Table VI.

In [38], the HVQ with temporal pooling algorithm is a partial implementation of the work of [15] on hierarchical temporal memory (HTM). This biologically-inspired model
places emphasis on the temporal aspect of pattern recognition, and consequently parses all images as ‘movies’. The hierarchy itself is a full 4 level tree of degree 4 that processes a $32 \times 32$ pixel input character image. During training, each node receives input from the layer below, with leaf nodes receiving a $4 \times 4$ raw pixel image that is moved one pixel at a time across the node's receptive field, in a process known as sweeping. As the sweep progresses, we count how frequently one pattern follows another. This information is then used to create temporal groups that collect together patterns that have most frequently succeeded another during training. The same process of temporal pooling is repeated at each level up to the root node, where images are classified according to their character values. During recognition, an image is again swept across the leaf node sensors, as each non-root node estimates the membership probability of its input for each of its temporal groups. This information is propagated up to the root, which then outputs the most probable character classification.

A method for increasing the recognition rates of handwritten characters by combining MLP and SVM was presented in [48]. The experiments demonstrated that the combination of MLPs networks with SVMs experts pairs of classes that constitute the greatest confusion of MLP, had improved performance in terms of recognition rate. Besides the MLP classifier is generally more robust than kNN, in some cases this behavior may not be verified. In our experiments it was observed the kNN classifier achieving better results than MLP classifier in the character recognition task. The first hypothesis this happen is the C-Cube dataset has a heterogeneous and unbalanced distribution concerning the number of samples of each class, which is a known problem for MLP classifiers. Another problem for MLP is that the accuracy decreases when the number of classes the model have to distinguish enlarges.

Observing the results obtained by the models presented in Table VI, we observe that despite the simplicity of the proposed hybrid classifier, we achieved better results than MDF-RBF [39], MDF-SVM [39], 34D-MLP [7] and MLP+SVM [48] in addition to results that can be considered satisfactory in comparison with other models HVQ-32 [38], HVQ-16 [38], 34D-RBF [39] and 34D-SVM + Neural GAS [7] that are more complex and take longer to be trained. Thus, we proved that by using a technique of extracting simple features as input to a kNN+SVM classifier, we have as main advantages:

1. Fast training process: in the kNN we just need to find the best value for $k$ and the best function to
the target sample;

2. Similar or better results to models of high complexity.

One weakness of the proposed model, using the kNN is the classification computational cost since, for every image included in the training set of some class letter, we need to
run the feature extraction algorithm and compute the distance from the feature vector against the feature vector extracted for the unknown image. In order to minimize this overhead, we recommend to prune the number of samples of each class.

As a future work, several hypothesis may be explored to chose when some new image should be included or not in the training set of some class. Independently of the hypothesis, the idea is always keep the training set of each class with the maximum entropy but with limited likelihood between elements of the same class (in order to prevent elements so much similar) and, in addition, minimizes the similarity between classes.

VII. CONCLUSIONS

In this paper, a method for increasing the recognition rates of handwritten characters by combining kNN and SVM is proposed. The experiments demonstrated that the combination of kNN with SVMs experts pairs of classes that constitute the greatest confusion of kNN, have improved performance in terms of recognition rate. The results showed a significant improvement from 1.00% to 3.61% in recognition rate for all cases tested (uppercase, lowercase and uppercase + lowercase).

As a proposal for future research, we intend to use different feature extraction techniques combined with classifier ensemble in our hybrid kNN-SVM. This action might improve the character recognition rates for our recognizer and, moreover, allow the proposed model to be used in the cursive word recognition task. Due to C-Cube properties, all analysis in this work were focused in the character recognition task.

Another direction we would like to pursue is the use of clustering analysis [7] to verify whether the upper and lower case versions of the same letters are similar in shape. With this, we could reduce the number of confusions in the kNN and hence achieving better recognition rates. Finally, some strategies to minimize the recognition response time of the kNN may be investigated, such as filtering out redundant (very similar) images of the same class.

ACKNOWLEDGMENT

This work was supported by FACEPE (Brazilian Research Agency).

REFERENCES


