Character Recognition Experiments using Unipen Data

Marc Parizeau, *Alexandre Lemieux*, and *Christian Gagné*

Laboratoire de vision et systèmes numériques(LVSN),
Département de génie électrique et de génie informatique,
Université Laval, Ste-Foy (Qc), Canada, G1K 7P4.

Abstract

This paper presents experiments that compare the performances of several versions of a Regional-Fuzzy Representation (RFR) developed for Cursive Handwriting Recognition (CHR). These experiments are conducted using a common Neural Network (NN) classifier, namely a Multi-Layer Perceptron (MLP) trained with backpropagation. Results are given for Sections 1a (isolated digits), 1c (isolated lower-case), and part of Section 3 (lower-case extracted from phrases) of the Unipen database. Data set Train-R01/V07 is used for training while DevTest-R01/V02 is used for testing. The best overall representation yields recognition rates of respectively 97.0% *and* 85.6% *for isolated digits and lower case, and* 84.4% *for lower-case extracted from phrases.*

1 Introduction

Feature extraction is assuredly one of the important cornerstone of any pattern classification system. Indeed, no matter how sophisticated are the classifier and learning algorithm, poor feature extraction and selection will always lead to poor system performance. Of course, one could envision a classifier that inputs raw data and automatically learns to extract discriminant features from this data, but such an approach underlies colossal difficulties that are usually overwhelming in practice. Moreover, most classification methods require that patterns be represented in a fixed dimensional feature space that is often incompatible with raw data.

In this paper, we focus on the problem of on-line handwriting recognition [1, 2, 3, 4] where we wish to classify on-line handwritten characters that somehow have already been segmented. This problem is one example where raw data cannot be directly interpreted as a fixed dimensional feature space. Our specific objective is twofold: first to improve upon a previously published handwriting representation (feature space) [5], and second to report new results on several sections of the Unipen handwriting database [6].

To estimate the relative performance of our different handwritten character representations, we work on the same Unipen data sets using a single neural network classifier, namely a multilayer perceptron with backpropagation [7], trained with a fixed set of parameters. In this way, even if backpropagation may not be the best nor the fastest learning algorithm for all situations and problems, we assume in this study that its limitations will not change the relative ordering of the different representations, nor that it will affect greatly their performance gains on a given data set. Besides, it is our experience that, when used correctly, it can in fact perform very well on large noisy data sets like Unipen that contains broad within class deviations.

The paper is organized as follows. Section 2 first presents an overview of the original handwriting representation, and then proceeds with a detailed description of the proposed enhancements. Then, Section 3 describes both the experimental protocol and the results of the recognition experiments conducted on the different representations constructed from these enhancements. Finally, Section 4 concludes this paper with a discussion of possible ways to further improve upon our “fuzzy-regional representation”.

2 Fuzzy-regional representation

Our representation for handwriting recognition is based on a window of attention [5], as illustrated in Figure 1. In the context of this paper, it is assumed that characters are already segmented and that this window corresponds to the character’s bounding box. In a previous work [8], the same representation was used in a different context, where window size and window position (attention focus) were changed dynamically in order to learn to segment cursive words. The main idea behind this former work was to be able to initialize a window of attention somewhere near a character and have a segmentation process fine tune its position and size in order to correctly locate the character. For

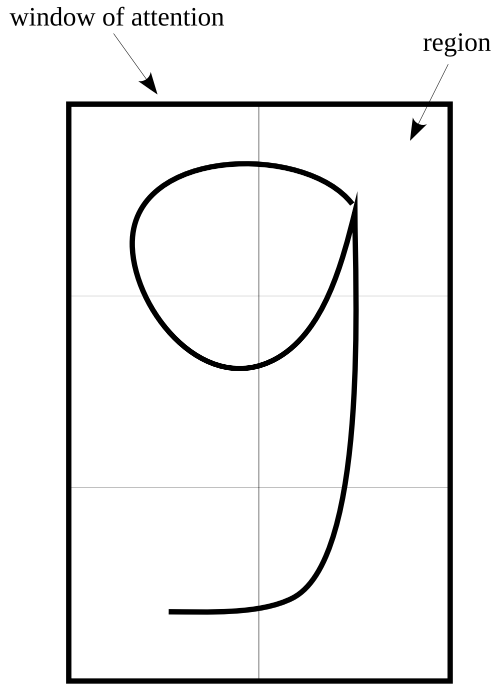

Figure 1. Window based regional representation.

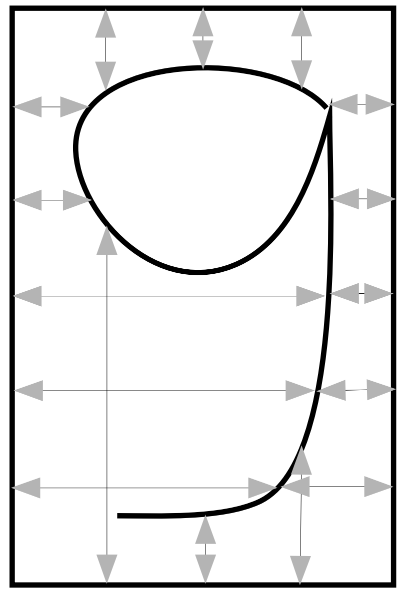

Figure 2. Horizontal and vertical distances between window content and window edges.

this process to converge, the system needs a high performance representation/classifier to detect when this window of attention actually matches the character's bounding box. Thus the object of the present work.

The representation is said to be "regional", because the window of attention is subdivided into a grid of distinct regions (in the example of Figure 1, there are 3×2 regions). Within each of these regions, 7 fuzzy variables are extracted [5]: the first three measure the degree to which region content is *rectilinear*, *curved clockwise*, and *curved counter-clockwise*, while the other four measure the degree to which region content is *horizontal*, *vertical*, and *oblique* with positive or negative slopes. These 7 variables are then assembled to form a fuzzy vector, and the basic handwriting representation is simply the concatenation of all regional fuzzy vectors plus a set of relative horizontal and vertical densities associated with rows and columns of regions, and two global variables that measures the aspect ratio of the window (character bounding box) [5].

The following paragraphs present several enhancements to this basic representation. Their relative performances will be compared in Section 3.

Curvature variables The first experiment conducted with the above basic representation was to test whether the added complexity of the regional curvature variables really improved the recognition rate of the classifier. Indeed, although this information may contain useful information, it also increases the size of the input space by $3 \times n$ dimensions, n being the number of regions, which can make classifier training and generalization harder.

Experimental results have shown that the contribution of the curvature variables is very significant both for isolated digits ($+1.5\%$) and isolated lower-case ($+4\%$).

Region overlapping The process of overlaying a grid of non overlapping regions over the window of attention in effect creates spatial discontinuities that may induce noise in the representation. The second experiment was thus to build different grids of overlapping regions, using overlapping factors of 2.5%, 5%, 10%, and 15%, and to test their effect on recognition rates. Surprisingly, results have shown that the basic fuzzy regional representation is mostly unaffected by region overlapping (within $\pm 0.05\%$ of error).

Distance measurements Another possible improvement for the basic representation was to add information about horizontal and vertical distances between the content of the window of attention and its edges, as illustrated by Figure 2, where these distances are sampled at regular intervals and normalized with respect to window size. For the experiments described in Section 3, we have chosen to sample each edge 5 times which adds to the feature space $4 \times 5 = 20$ new dimensions.

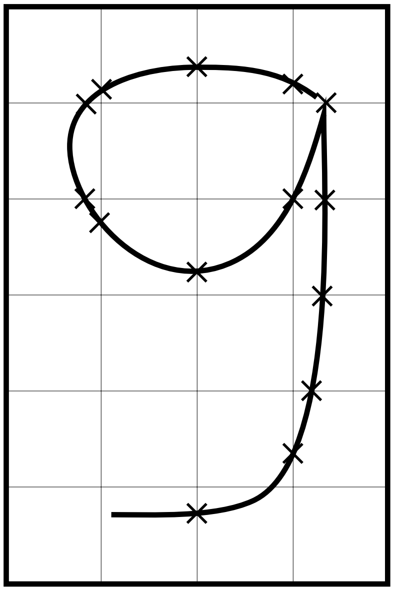

Figure 3. Number of horizontal and vertical intersections.

Intersections A final improvement consists in adding new variables for counting the number of times that imaginary horizontal and vertical lines intersect the script. These imaginary line are sampled at fixed intervals, as illustrated by Figure 3 (intersections are marked with an $\times$). Just like for the distance measurements, 5 horizontal and 5 vertical lines were used for the experiments described in Section 3, which adds to the feature space another $2 \times 5 = 10$ new dimensions.

3 Experimental results

In order to compare relative performances of the proposed improvements to the basic regional fuzzy representation, the same classifier was trained using the same procedure and from the same raw data. Each specific handwriting representation was extracted from Section 1a, Section 1c, and part of Section 3 of the Unipen data sets [6]. Training data is from Unipen Train-R01/V07 and testing data is from DevTest-R02/V02. Table 1 summarizes the sizes of these data sets. The "Total char" column represents the total number of characters found in each section. The "Bad char" column represents the total number of samples that were removed from these data sets. These bad samples are segments that cannot be possibly recognized for one of the following reasons: 1) they have 0 width and are not instances of digit 1 or characters i or l, 2) they have 0 height, and 3) they have 0 width and 0 height. Hence, they are not counted in the recognition rates reported below. Certainly, there are many other unrecognizable (badly written or mislabeled) characters in the Unipen data sets but none were removed except for those flat obviously erroneous samples. It should be noted that section 3 of Unipen not only contains lower-case characters but also upper-case, digits, and punctuation marks taken in the context of phrases. The "Total char" column in Table 1 for Section 3 only counts the lower-case characters (the other character types were not used for our experiments).

Table 1. Unipen data set statistics: train **is Train-R01/V07 and** test **is DevTest-R02/V02.**

Data Set		Total char	Bad char
Section 1a (digits)	train	$15\,953$	4
	test	$8\,598$	34
Section 1c (lower-case)	train	$61\,359$	44
	test	$37\,470$	23
Section 3 (lower-case)	train	$40\,092$	0
	test	$26\,560$	0

A multilayer perceptron classifier was used with standard on-line backpropagation training [7] and a single hidden layer of between 50 and 90 neurons. The learning rate and momentum were fixed at respectively 0.1 and 0.25. Training was controlled using a cross-validation procedure where 75% of the training set was used for training and 25% for validation. The minimum number of training epochs was fixed at 35 for digits, and 40 for lower-case characters. The total number of training epochs varied from 65 to 125.

Our experimental recognition results are summarized in Tables 2, 3, and 4 for Sections 1a, 1c, and 3 of the Unipen data sets, respectively. Representation type "RFR" is the plain *Regional Fuzzy-Representation*. Type "RFR-base" is the reference representation that combines RFR with horizontal and vertical region densities, and aspect ratio variables [5]. Two grid sizes are compared: 3×2 (three lines and 2 columns) and 3×3. Column "Feature Size" gives the total dimension of the feature space, and column "Hidden Neurons" specifies the number of neurons on the hidden layer of the classifier. For each representation type and grid size, several classifiers were trained on the training sets. Columns "Mean Rec. Rate" and "Max Rec. Rate" give respectively the average and maximum recognition rates obtained on the testing set. Finally, column "Max Shift RFR-base" presents the increase in recognition rate relative to the reference representation (RFR-base) with the 3×2 grid.

Results for digits show that the proposed enhancements to the RFR-base representation are indeed significant at $+1.5\%$ for both the 3×2 and 3×3 grid sizes, and that this increase in performance is mainly due to the distance variables, especially in the case of the higher resolution grid

Table 2. Experimental results for Unipen DevTest-R02/V02, Section 1a (isolated digits).

Representation Type	Grid Size	Feature Size	Hidden Neurons	Mean Rec. Rate (%)	Max Rec. Rate (%)	Max Shift RFR-base
RFR	3×2	42	50	94.5	94.8	-0.7%
RFR-base[a]	3×2	49	60	95.4	95.5	–
	3×3	71	60	96.1	96.2	+0.7%
RFR-base + distances (5)	3×2	69	60-70	96.5	96.6	+1.1%
	3×3	91	60-70	96.7	96.8	+1.3%
RFR-base + intersect (5)	3×2	59	60	96.0	96.2	+0.7%
	3×3	81	60-70	96.1	96.3	+0.8%
RFR-base + dist(5) + inter(5)	3×2	79	60-70	96.9	97.0	+1.5%
	3×3	101	70-80	96.6	97.0	+1.5%

[a] The results for RFR-base are very similar to those reported in [5] where an another classifier was used: 95.3% for a 3×2 grid and 96.1% for a 4×3 grid (no results were given for the 3×3 grid in that paper).

Table 3. Experimental results for Unipen DevTest-R02/V02, Section 1c (isolated lower-case).

Representation Type	Grid Size	Feature Size	Hidden Neurons	Mean Rec. Rate (%)	Max Rec. Rate (%)	Max Shift RFR-base
RFR	3×2	42	80	77.7	78.0	-2.5%
RFR-base	3×2	49	80	80.3	80.5	–
	3×3	71	80	83.0	83.1	+2.6%
RFR-base + distances (5)	3×2	69	80	83.4	83.5	+3.0%
	3×3	91	80	84.4	84.8	+4.3%
RFR-base + intersect (5)	3×2	59	80	82.8	82.9	+2.4%
	3×3	81	80	84.3	84.4	+4.0%
RFR-base + dist(5) + inter(5)	3×2	79	80	84.9	85.3	+4.8%
	3×3	101	90	85.6	85.6	+5.1%

(3×3). Also, to the authors knowledge, the obtained recognition rates of up to 97.0% are the highest reported performance on Section 1a of Unipen DevTest-R02/V02.

As for isolated lower-case, performance gains are even higher at 5.1% for Section 1c and 5.9% for Section 3. Of course, the baseline performance was also much lower, but the performance increase is nevertheless very much significant. Moreover, results show that for lower-case letters, the gains from both the distance and intersection variables is around three times those of the densities variables. Also, it is interesting to note that grid size 3×2 is almost as good as grid 3×3 when using both the distance and intersection variables, which is not the case for the reference representation, nor for the plain RFR representation. Thus, contrary to our past experience where we used to work with two different grid size for digits and lower-case, it seems that we can now use a common feature space of similar dimension (79 vs 71) to what we were using before for lower-case letters, and achieve considerably higher performance.

4 Conclusion

This paper has presented some significant improvements to the RFR-base handwriting representation. We call this new representation RFR-DI. Recognition rates for digits using RFR-DI have reach 97.0% which is probably not very far from what a typical human could achieve on Section 1a of Unipen DevTest-R02/V02. Indeed, a low scale reading experiment [5, 9] with humans has resulted with the conclusion that at least 1% of that data set is completely unrecognizable. This leaves a small although very difficult 2% to reach for!

Recognition rates achieved on isolated lower-case characters are less impressive, being at the 85% level. On the other hand, we have not seen any published results higher then these on the same complete DevTest-R02/V02 data sets. Furthermore, results as high as for digits are most probably impossible to achieve. The problem with recognition of isolated lower-case script (apart from some very badly written samples in Unipen) is that without context, many letter pairs can easily be confused. For example, dotless i and l, f and t, n and m, n and r, v and u, y and g,

Table 4. Experimental results for Unipen DevTest-R02/V02, Section 3 (isolated lower-case in the context of phrases).

Representation Type	Grid Size	Feature Size	Hidden Neurons	Mean Rec. Rate (%)	Max Rec. Rate (%)	Max Shift RFR-base
RFR	3×2	42	55-70	76.8	77.3	-1.2%
	3×3	63	55-75	79.7	79.8	+1.3%
RFR-base	3×2	49	55-75	78.0	78.5	–
	3×3	71	60-80	81.2	81.3	+2.8%
RFR-base + distances (5)	3×2	69	55-75	82.1	82.5	+4.0%
	3×3	91	65-85	82.8	83.3	+4.8%
RFR-base + intersect (5)	3×2	59	55-70	81.6	81.9	+3.4%
	3×3	81	60-80	82.9	83.4	+4.9
RFR-base + dist(5) + inter(5)	3×2	79	70-85	84.0	84.1	+5.6%
	3×3	101	75-95	84.3	84.4	+5.9%

etc. If we accept the best two hypotheses produced by the classifier, then the best result on Section 3 leaps by almost 6% to 90.1%, which shows that RFR-DI, although better than RFR-base, is not quite good enough to disambiguate the somewhat similar shapes of lower-case scripts.

To surpass the current level of performance, we are currently investigating ways to achieve what we qualified as "usually overwhelming" in the introduction: to automatically discover an optimal representation from raw data using genetic programming techniques [10], given only some very basic assumptions like curvature and orientation variables, and regional decomposition. So far, however, we have not been able to "discover" a better representation than RFR, although we are very close. For example, our genetically engineered representation has reached the 93% mark on isolated digits, before leveling off.

5 Acknowledgements

This research was supported by NSERC-Canada and FCAR-Québec grants to M. Parizeau, and by an FCAR-Québec scholarship to C. Gagné.

References

[1] R. Plamondon, D.P. Lopresti, L.R.B. Schomaker, R. Shrihari, "Online Handwriting recognition", *Wiley Encyclopedia of Electrical and Electronics Engineering*, John G. Webster (editor), vol. 15, pp. 123-146, John Wiley & Sons, 1999.

[2] I. Guyon, M. Schenkel, J. Denker, W. Hubbard, "Overview and Synthesis of On-Line Cursive Handwriting Recognition Techniques", *Handbook on Optical Character Recognition and Document Image Analysis*, World Scientific Publishing, 1996.

[3] T. Wakahara, H. Murase, K. Odaka, "On-Line Handwriting Recognition", *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1181-1194, 1992.

[4] C.C. Tappert, C.Y. Suen, T. Wakahara, "The State of the Art in On-Line Handwriting Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-12, no. 8, pp. 787-808, 1990.

[5] J.F. Hébert, M. Parizeau, N. Ghazzali, "A new fuzzy geometric representation for on-line isolated character recognition", *Proc. of the 14th International Conference on Pattern Recognition*, Brisbane (Australie), pp. 1121-1123, 1998.

[6] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, S. Janet, "UNIPEN Project of On-Line Data Exchange and Recognizer Benchmarks", *Proc. of the 14th International Conference on Pattern Recognition*, Jerusalem (Israël), pp. 29-33, 1994.

[7] S. Haykin, *Neural Networks: A Comprehensive Foundation* (chap. 6), IEEE Press, 1994.

[8] J.F. Hébert, M. Parizeau, N. Ghazzali, "Learning to Segment Cursive Words using Isolated Characters", *Proc. of the Vision Interface Conference*, Trois-Rivières (Canada), pp. 33-40, 1999.

[9] J.F. Hébert, "Architecture neuronale hybride pour l'apprentissage incrémental des connaissances: application à la reconnaissance d'écriture cursive", Ph.D. thesis, Université Laval, 2000.

[10] W. Banzhaf & al., "Genetic Programming : An Introduction : On the Automatic Evolution of Computer Programs and Its Applications", Morgan Kaufmann Publishers, 1998.