

Document downloaded from:

<http://hdl.handle.net/10251/68388>

This paper must be cited as:

Álvaro Muñoz, F.; Sánchez Peiró, JA.; Benedí Ruiz, JM. (2014). Offline Features for Classifying Handwritten Math Symbols with Recurrent Neural Networks. 22nd International Conference on Pattern Recognition (ICPR 2014). IEEE. doi:10.1109/ICPR.2014.507.



The final publication is available at

<http://dx.doi.org/10.1109/ICPR.2014.507>

Copyright IEEE

Additional Information

Offline Features for Classifying Handwritten Math Symbols with Recurrent Neural Networks

Francisco Álvaro, Joan-Andreu Sánchez, José-Miguel Benedí
Pattern Recognition and Human Language Technologies
Universitat Politècnica de València, Spain
Email: {falvaro, jandreu, jbenedi}@dsic.upv.es

Abstract—In mathematical expression recognition, symbol classification is a crucial step. Numerous approaches for recognizing handwritten math symbols have been published, but most of them are either an online approach or a hybrid approach. There is an absence of a study focused on offline features for handwritten math symbol recognition. Furthermore, many papers provide results difficult to compare. In this paper we assess the performance of several well-known offline features for this task. We also test a novel set of features based on polar histograms and the vertical repositioning method for feature extraction. Finally, we report and analyze the results of several experiments using recurrent neural networks on a large public database of online handwritten math expressions. The combination of online and offline features significantly improved the recognition rate.

I. INTRODUCTION

Mathematical notation represents a very valuable source of information in many fields. Automatic recognition of mathematical expressions started several decades ago [1]. However, handwritten math recognition has lately become a very active research field [2] due to the development of tactile-devices and human-based interfaces. Furthermore, in the last years several competitions on math recognition have been organized with increasing participation of research groups from all over the world [3].

Mathematical expression recognition is commonly divided into three primary tasks [2]: symbol segmentation, symbol recognition and structural analysis. These subproblems can be tackled sequentially or jointly, but in any case a good math symbol classifier is crucial for the performance of the model. In this paper we focus on classifying handwritten math symbols.

Several features and classifiers have been proposed for handwritten math symbol recognition [2]. Generally, the proposals are focused on online recognition [4], [5], [6] but it is difficult to find a paper focused on offline recognition. Some authors have combined online and offline information and it has proven to significantly improve the recognition results [7], [8], [9]. However, we are not aware of a public database of offline handwritten math expressions or any paper focused on offline handwritten math symbol recognition that provides comparable results [10]. For these reasons, in this paper we want to study the performance of several offline features for this problem using a large publicly available dataset.

Handwriting recognition has been studied for years and math expressions are a particular case in that field. For this reason, the features developed for text recognition have been successfully applied to math symbol classification [9].

Thereby, in this work we want to explore several well-known text offline features for math symbol recognition [11], [12], [13]. Furthermore, we also want to test a novel set of features based on polar shape descriptors [14], [15].

There was a lack of large public dataset of handwritten formulae and this caused that some of those approaches were difficult to compare. In this work we report an extensive experimentation using the dataset of the last CROHME competition [3] in order to provide comparable results. We used a Recurrent Neural Network (RNN) classifier because it has proven to outperform Hidden Markov Models (HMM) in this task [6]. We report experiments with online features, offline features and their combination. Results show that offline features outperform online features and that their combination significantly improves the recognition rate.

The remainder of the paper is organized as follows. First, a review of related work is presented in Section II. Then, a set of online features is defined in Section III. The different offline features are described in Section IV. Section V presents the RNN symbol classifier and the experimentation is reported and analyzed in Section VI. Finally, conclusions and future work are presented in Section VII.

II. RELATED WORK

In the literature, most of the proposals regarding handwritten math symbols classification are focused on online recognition. Several classifiers have been used like elastic matching [4], HMM [5], [9] or RNN [6]. The advanced architecture Bidirectional Long Short-Term Memory (BLSTM) for RNN has outperformed other classifiers in several tasks like handwritten text recognition [16] or handwritten math symbol recognition [6]. For this reason we used this classifier to carry out the experiments.

Regarding offline features, many authors have combined online and offline feature extraction to obtain a hybrid system in order to improve the recognition results [7], [8], [9]. In addition to the combination of two independent sets of features, a hybrid feature set that directly merges online and offline information has been studied in [6]. This set improved the classification accuracy of HMM but results with RNN remained the same.

In this work we study three sets of well-known features that have been extensively used for handwriting text recognition. We refer to these features by the name of the group or university that has developed them: PRHLT, FKI and RWTH. They are detailed in Section IV. Furthermore, we also wanted

to assess the performance of a set of features based on polar descriptors because, in the last years, several approaches based on polar shape descriptors [14] have been published in the mathematical expression recognition field. For example, polar features have been used to detect layout classes for symbols [17], symbol segmentation [18] or spatial relationships classification [19]. There are also proposal for handwritten text recognition using polar histograms [15].

Since the MathBrush database [20] was released more papers are reporting comparable results [4], [18], [6]. These papers presented online math symbol recognition experiments. Recently, the CROHME 2013 competition [3] has merged several datasets (including MathBrush) such that it represents the largest handwritten mathematical expression recognition corpus currently available. We used the training and test sets defined in the competition in order to provide comparable results.

III. ONLINE FEATURES

The CROHME 2013 database is online and there is no publicly available such a large database of offline handwritten math expressions. We use the online features to obtain a comparable experiment with previous published results and, hence, have a better understanding of the offline features performance.

A mathematical symbol is represented as a sequence of points in the space. We compute the 7 time-based features defined in [21] for each point $p = (x, y)$:

- Normalized coordinates: (x, y) normalized values such that $y \in [0, 100]$ and the aspect-ratio of the sample is preserved.
- Normalized first derivatives: (x', y') .
- Normalized second derivatives: (x'', y'') .
- Curvature: k , the inverse of the radius of the curve in each point.

Finally, for each point p we obtain a feature vector $f_{\text{on}}(p)$ as:

$$f_{\text{on}}(p) = [x, y, x', y', x'', y'', k]$$

It should be noted that no resampling is required prior to the feature extraction process because first derivatives implicitly perform writing speed normalization [21].

IV. OFFLINE FEATURES

Although the CROHME 2013 database is online, we can easily produce the images representing the mathematical symbols. Then, a wide set of offline features can be extracted from the images. In the following sections we describe the image generation process and the different offline sets of features used for classification.

A. Image Generation

Given an online sample encoding a math symbol as a sequence of points in the space, we generated the image representation as follows. We set the image height to H pixels and we kept the aspect ratio (up to $5H$ to prevent symbols

like a fraction bar from creating too wide images). Then we rendered the image representation of each symbol by using linear interpolation between each two consecutive points in every stroke. The final image is produced after smoothing it using a mean filter with a window sized 3×3 pixels.

B. PRHLT Features

In this section we describe the offline features that the Pattern Recognition and Human Language Technologies (PRHLT) group has used for many handwritten text recognition tasks [13]. These features have also been successfully applied to math symbol recognition [9].

Given the image of a math symbol (Fig. 1.a) it is transformed into a sequence of feature vectors as follows. First, the image is divided into a grid of small square cells sized a small fraction of the image height ($H/20$). Then for each cell we compute three different values (Fig. 1): normalized gray level (b), horizontal gray-level derivative (c) and vertical gray-level derivative (d). The feature extraction was extended to a 5×5 window centered at the current cell to obtain smoothed values. The values are weighted by a two-dimensional Gaussian function in b) and a unidimensional Gaussian function in c) and d). The derivatives are computed by least squares fitting a linear function. Finally, we stack the three different images that result after computing these values for each cell (see Fig. 1.e). Thus, each column from left to right represents a feature vector f_{prhlt} of 60 dimensions.

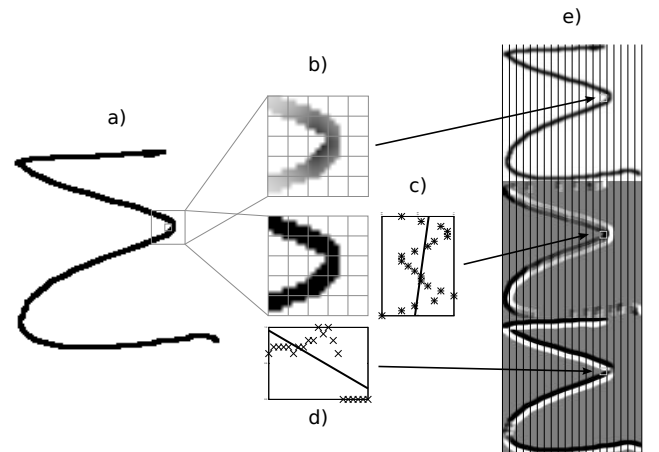


Fig. 1. Example of PRHLT offline features computation. Given an image of a math symbol (a) we compute the normalized gray level (b), horizontal gray-level derivative (c) and vertical gray-level derivative (d). Each column of the stacked images (e) represents a feature vector.

C. FKI Features

The FKI features [11] are a well-known set of geometric features that has been used for years in handwriting recognition. Given a binary image of height H , we compute 9 geometrical values c_i for each column x as:

- Number of black pixels in the column:

$$c_1(x) = \sum_{y=1}^H I(x, y)$$

- Center of gravity of the column:

$$c_2(x) = \frac{1}{H} \sum_{y=1}^H y \cdot I(x, y)$$

- Second order moment of the column:

$$c_3(x) = \frac{1}{H^2} \sum_{y=1}^H y^2 \cdot I(x, y)$$

- Position of the upper contour in the column:

$$c_4(x) = \min\{y \mid I(x, y) = \text{black}\}$$

- Position of the lower contour in the column:

$$c_5(x) = \max\{y \mid I(x, y) = \text{black}\}$$

- Orientation of the upper contour in the column:

$$c_6(x) = \frac{c_4(x+1) - c_4(x-1)}{2}$$

- Orientation of the lower contour in the column:

$$c_7(x) = \frac{c_5(x+1) - c_5(x-1)}{2}$$

- Number of black-white transitions in the column:

$$c_8(x) = \text{NT}_{\text{black} \rightarrow \text{white}} I(x, y); 1 \leq y \leq H$$

- Number of black pixels between the upper and lower contours:

$$c_9(x) = \sum_{c_4(x) < y < c_5(x)}^H I(x, y)$$

Finally, given a binary image of size $W \times H$, for each column x such that $1 \leq x \leq W$ this definition will generate a 9-dimensional feature vector $f_{\text{fki}}(x)$ as:

$$f_{\text{fki}}(x) = [c_1, c_2, \dots, c_9]$$

D. RWTH Features

The RWTH features have been extensively used in handwriting recognition [12]. Given an image of height H , this set of features is simply computed using a sliding window of width w from left to right (see Fig. 2a). Then, the $w \times H$ values of each extracted window are projected to D dimensions by using Principal Component Analysis (PCA). Finally each projected window represents a feature vector f_{rwth} .

Lately, the vertical repositioning method [22] has provided very good results in handwriting recognition [23], hence, we have also tried this method. We first compute the vertical center of gravity of each extracted window. Then, we reposition the window such that its vertical center is aligned with the computed center of gravity. Fig. 2 shows an example of the RWTH features computed with and without vertical repositioning.

E. Polar Features

Shape descriptors are a well-known representation that have been used for multiple applications [24]. Bing *et al.* [15] proposed a feature extraction based on polar histograms for handwritten text recognition. We defined similar features but we use circles instead of ellipses and equally spaced distances instead of log-distances.

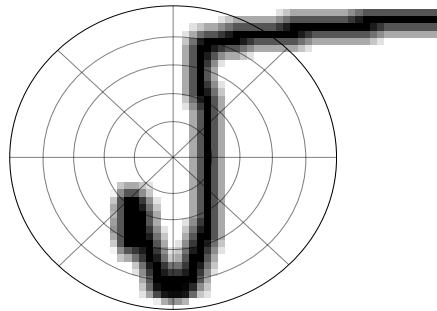


Fig. 3. Example of polar descriptor with $n = 5$ circles and $m = 8$ arcs in a particular column of a square root symbol.

We define a polar descriptor centered in a point p as follows. We draw n circles with radii equally spaced up to the maximum radius r . Moving counterclockwise, draw radii dividing each circle into m equal arcs. This descriptor is encoded as a matrix such that each row represents a circle and each column represent the angle starting from 0 degrees. Figure 3 shows an example of polar descriptor.

The idea is to use a sliding polar descriptor instead of the sliding window used in the previously described RWTH features. Given a descriptor, each of the $n \times m$ bins contains the number of foreground pixels that fall into that bin normalized by the total number of pixels in the descriptor. Thus, each feature vector f_{polar} is computed as the projection of that polar histogram to D dimensions using PCA. The feature vectors are computed setting the center of the polar descriptor on each column of the input image at row $H/2$ (centered).

We also wanted to apply the vertical repositioning method to these features. Given a column x and radius r , the standard polar features are centered in $p = (x, H/2)$. In this case, following the methodology described in RWTH features, we compute the vertical centroid y' of the window from column $(x - r)$ to $(x + r)$ and then the descriptor center is set to $p = (x, y')$. Fig. 4 shows an example of the polar histograms computed with and without vertical repositioning.

V. RECURRENT NEURAL NETWORKS

Sequential classification has traditionally been tackled using HMMs. However, in the last years RNNs have brought more attention in the research community. RNNs are a connectionist model containing a self-connected hidden layer. The recurrent connection provides information of previous inputs such that the network can benefit from past contextual information [25]. The LSTM advanced RNN architecture allows that cells can access to context information over long periods of time. This is achieved by using a hidden layer composed of recurrently connected subnets, called memory blocks [16].

Bidirectional RNNs [26] have two separate hidden layers that allow the network to access context information in both time directions: one hidden layer processes the input sequence forward and the other processes it backward. The combination of bidirectional RNNs and the LSTM architecture results in BLSTM-RNNs. These networks have outperformed standard RNNs and HMMs in handwriting text recognition [16] and math symbol recognition [6] and also they are faster than HMMs in terms of classification speed.

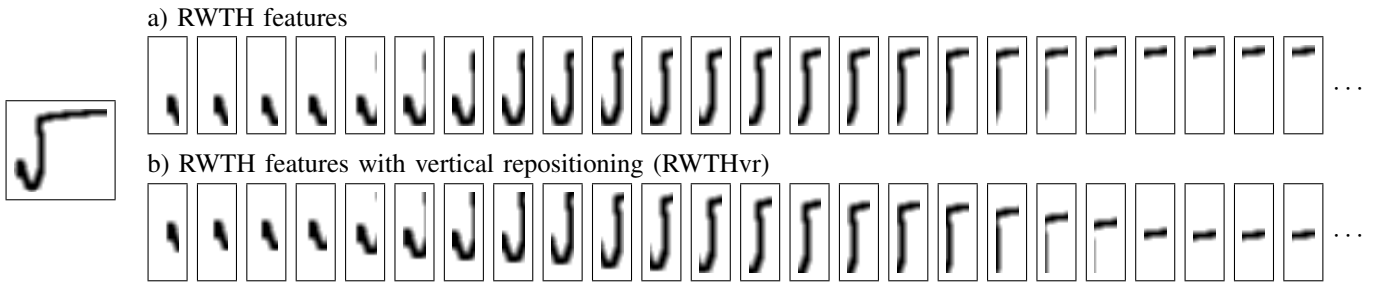


Fig. 2. Example of the RWTH offline features for a square root symbol using a sliding window of $w = 11$ pixels. Each window represents a feature vector after projecting it to D dimensions with PCA.

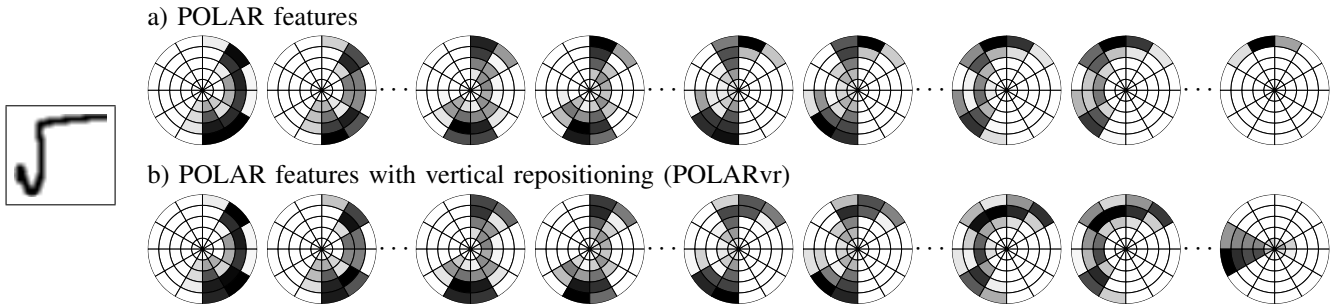


Fig. 4. Example of polar features for a square root symbol using $n = 5$ circles and $m = 12$ arcs. The gray-scale colors of the bins in the descriptors represent the values of the polar histogram, from zero (white) to the maximum value in that descriptor (black).

In this work we used BLSTM-RNN for math symbol classification. The RNN was trained in a frame-based approach, i.e., given an input sequence s composed of n vectors such that $s = \{f_1, f_2, \dots, f_n\}$, the network computes for each class c the posterior probability $P(c | f_i)$, where f_i represents a feature vector. Then, we obtained the posterior probability per symbol after the following normalization:

$$P(c | s) = \frac{1}{n} \sum_{i=1}^n P(c | f_i) \quad (1)$$

where the probability per symbol for class c is computed as its average probability per frame.

VI. EXPERIMENTS

In this section we detail the experimentation carried out to assess the performance of the different features proposed for handwritten mathematical symbol classification. We used the RNNLIB library [27] for training the RNN-BLSTM and the tools for computing the different features are also available¹.

A. CROHME 2013 Database

The recent CROHME 2013 competition [3] released a large resource for mathematical expression recognition as a result of combining and normalizing several datasets. This database contains 8,835 online handwritten math expressions for training and 671 math expressions for testing. There are about 86K math symbols for training and 6K symbols for testing distributed in 101 classes.

We used the train and test sets of the competition in order to provide comparable results. Moreover, we needed a validation

set for tuning parameters of some features and also for training the RNN classifier. Hence, we first extracted a validation set by taking 10% of the samples of each class in the original train set, and the remaining 90% were used for training. This way we also kept proportionally the distribution of the classes.

B. RNN-BLSTM Configuration

We used the same configuration of the RNN-BLSTM classifier for every experiment. The size of the input layer was determined by the feature set and the output layer size was 101, i.e., the number of symbol classes. The forward and backward hidden layers each contained 100 LSTM memory blocks. The network weights were initialised with a Gaussian distribution of mean 0 and standard deviation 0.1. The network was trained using online gradient descent with a learning rate of 0.0001 and a momentum of 0.9. This configuration has obtained good results in both handwritten text recognition [16] and handwritten math symbol classification [6].

The experimentation was carried out in two steps. First, we trained the network until the error had ceased to improve on the validation set for 50 epochs. As the RNN results depend on the random initialization, we performed four experiments with the validation set in order to compute the average number of epochs \bar{e} required to obtain the best network. Then we trained four RNNs using the full training set during \bar{e} epochs. Finally, we classified the test set with the RNNs trained with the full training set and the average recognition rate is reported.

C. Results

In this section we report the classification performance of every set of features while recognizing the CROHME 2013 test set using a RNN-BLSTM classifier. The online features

¹Source code available in the software section of www.prhlt.upv.es

do not need any parameter to be tuned, so we carried out the experiment as defined in the previous section. Using the online samples, we rendered the images of every symbol as described in Section IV-A where the image height was set to 40 pixels.

Regarding offline features, the PRHLT features computed 60-dimensional vectors for each column of the rendered images and the FKI features extracted 9 features per column. On the other hand, RWTH and POLAR features had some parameters to be tuned. We tried several sizes w of the sliding window in RWTH features. The polar features radius was set to half the height of the image, in this case $r = H/2 = 20$, and we tested different values of circles (n) and arcs (m). We also tried different number of dimensions for the PCA projection.

The parameters of the features in the final experiment were obtained according to the error obtained with the validation set. The RWTH features were extracted using a sliding window of width $w = 5$ pixels and projected to 30 dimensions, whereas the POLAR features were computed with $n = 5$ circles, $m = 12$ arcs and also projected to 30 dimensions. This configuration was also used with the vertical repositioning method of both sets of features.

Finally, given that we have computed both online and offline features, the next natural step was to combine both classification results in order to obtain better results. We applied a naive Bayes classifier (assuming uniform class priors), which aimed at balancing the relative reliability of the online (a) and offline (b) features by using a weight parameter (α):

$$\Pr(c | a, b) = \alpha \cdot \Pr(c | a) + (1 - \alpha) \cdot \Pr(c | b) \quad (2)$$

where the probability for one model is computed as defined in Eq. (1). We set $\alpha = 0.5$ for every combination.

The results are shown in Table I such that there are three sets of results: using online features, using offline features and their combination. We report top-1, top-2 and top-5 recognition rates. Furthermore, there are several classes that produce many classification errors because they have very similar shape but different semantic [18], [6]. Hence, we also computed top-1 recognition rate where those similar classes were merged (TOP'_1). The symbols merged in this error were: $\{1, |, \iota, \text{comma}\}$, $\{P, p\}$, $\{S, s\}$, $\{C, c\}$, $\{X, x, \times\}$, $\{V, v\}$ and $\{o, 0\}$. It should be noted that we are dealing with the handwritten version of these symbols.

TABLE I. CLASSIFICATION RESULTS WITH RNN AND SEVERAL SETS OF FEATURES: ONLINE, OFFLINE AND THEIR COMBINATION.

Features	TOP ₁	TOP ₂	TOP ₅	TOP' ₁
Online	82.5 ± 0.3	92.3 ± 0.1	96.8 ± 0.1	88.3 ± 0.2
PRHLT	83.9 ± 0.4	93.4 ± 0.2	97.6 ± 0.1	89.9 ± 0.4
FKI	84.1 ± 0.2	93.6 ± 0.1	97.8 ± 0.1	90.3 ± 0.2
RWTH	83.4 ± 0.3	93.1 ± 0.2	97.5 ± 0.0	89.6 ± 0.3
RWTH _{vr}	82.9 ± 0.2	92.2 ± 0.3	97.4 ± 0.1	88.8 ± 0.2
POLAR	81.2 ± 0.4	91.2 ± 0.3	96.7 ± 0.1	87.2 ± 0.3
POLAR _{vr}	80.8 ± 0.3	90.4 ± 0.3	96.4 ± 0.1	86.6 ± 0.4
On. + PRHLT	87.1 ± 0.2	95.0 ± 0.1	98.3 ± 0.1	92.7 ± 0.2
On. + FKI	86.8 ± 0.4	95.3 ± 0.2	98.4 ± 0.1	92.6 ± 0.2
On. + RWTH	86.7 ± 0.3	94.9 ± 0.2	98.4 ± 0.1	92.6 ± 0.1
On. + RWTH _{vr}	86.5 ± 0.2	94.7 ± 0.2	98.4 ± 0.1	92.3 ± 0.0
On. + POLAR	86.1 ± 0.4	94.5 ± 0.2	98.2 ± 0.1	91.9 ± 0.1
On. + POLAR _{vr}	86.0 ± 0.2	94.2 ± 0.2	98.1 ± 0.1	91.7 ± 0.1

D. Discussion

The experimentation summarized in Table I produced several interesting results. First, although online data normally yields better results than offline data [28], we obtained higher recognition rate values with offline features than using online features. Results show that online features obtained on test set a 82.5 top-1 recognition rate whereas PRHLT, FKI and RWTH offline features improved up to 83.9, 84.1 and 83.4 respectively. The difference in performance is significative. In our opinion, an important factor for these results could be that the CROHME database comprises samples from many different writers and datasets. This can cause that the online samples were more sensitive to the writing style, whereas the image representation of the symbols could present less variance.

Regarding the performance of the offline features, there are several results to take into account. First, we can see that PRHLT, FKI and RWTH features obtained the best results outperforming online features. On the other hand, the proposed POLAR features provided a good performance although not as good as the rest of offline features, even slightly worse than the online results.

We would like to remark the results obtained by the FKI features because they obtained the best results with only 9 features, while the rest of offline features required 30 or 60 dimensions and more complex calculations. We think that FKI features obtained such a good performance, because the BLSTM-RNNs can take advantage of context. Hence, it does not require features to include context in their representation, like RWTH or POLAR, because the model itself can manage it. Another result that may suggest this conclusion is that the recognition rate of RWTH features in validation set worsened with wider sliding windows.

With respect to the vertical repositioning method, this technique has really improved the results in handwritten text recognition [23]. Nevertheless, in this task results did not present any improvement or they were even slightly worse. Our intuition is that this could be caused due to we are tackling the classification of isolated handwritten math symbols. This means that each sample is already segmented, whereas in handwriting text recognition the segmentation is a complex problem solved implicitly by the model, like HMMs or BLSTM-RNNs [16].

The combination of the classification results of online features and offline features led to significant improvements in the achieved recognition rate. We combined the posterior probabilities of both results according to Eq. (2) with $\alpha = 0.5$. Every combination between online and offline systems obtained significantly better error rates, with relative improvements up to 20%.

Finally, we could see that there is an important gap between top-1 and top-2 results, and top-5 recognition rates are really high. This behaviour shows that even though there is still room for improvement, the classifier provides very good results. We expect that with the integration of this classifier in a mathematical expression recognition system, the contextual information of the expression could help to solve many misclassifications. This can also be seen according to the reported TOP'₁ recognition rate, because the hybrid classification achieved almost

93% recognition rate if those similar shaped classes are not considered as errors. In addition to the classes detailed in TOP₁ computation, many errors were caused by very similar classes like: $\{5, s\}$, $\{t, +\}$, $\{comma, \}$, $\{q, 9\}$ or $\{z, 2\}$. There are not published results yet using this database, but an experiment using the described online features and BLSTM-RNNs is reported in [6] on a smaller dataset such that it outperformed previous publications [4], [18].

VII. CONCLUSION

In this paper we have studied the performance of several sets of offline features for handwritten math symbol classification. We tested different well-known features, a novel set of features based on polar histograms and the vertical repositioning method. We reported experiments on a large public database of a recent competition [3]. This database provides online data, hence, we also obtained results with online features in order to provide a better analysis.

Results show that offline features provided better results than online features, and the their combination produced up to 20% relative improvement. FKI offline features presented the best performance due to they obtained the best recognition rates using only 9 features. Finally, vertical repositioning method did not improve results in this task.

Future work will be focused on improving the discrimination of symbols with very similar shape. Furthermore, it would be interesting to continue developing the polar histogram features because they seem promising. There are still several configurations that should be tested: log-distances vs equally spaced distances and circles vs ellipses. Also it could be convenient to use more than one polar descriptor on each column because this type of descriptor is more sensitive to angular variations of points near to its center than to variations of points farther away.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for helpful discussion about the polar features. This work was partially supported by the Spanish MEC under the STraDA research project (TIN2012-37475-C02-01) and the FPU grant (AP2009-4363), by the Generalitat Valenciana under the grant Prometeo/2009/014, and through the EU 7th Framework Programme grant tranScriptorium (Ref: 600707).

REFERENCES

- [1] R. H. Anderson, "Syntax-directed recognition of hand-printed two-dimensional mathematics," in *Symposium on Interactive Systems for Experimental Applied Mathematics: Proceedings of the ACM Symposium*. New York, NY, USA: ACM, 1967, pp. 436–459.
- [2] R. Zanibbi and D. Blostein, "Recognition and retrieval of mathematical expressions," *International Journal on Document Analysis and Recognition*, vol. 15, no. 4, pp. 331–357, 2012.
- [3] H. Mouchère, C. Viard-Gaudin, R. Zanibbi, U. Garain, and D. H. Kim, "ICDAR 2013 CROHME: Third International Competition on Recognition of Online Handwritten Mathematical Expressions," in *Document Analysis and Recognition, Int. Conference on*, 2013, pp. 1428–1432.
- [4] S. MacLean and G. Labahn, "Elastic matching in linear time and constant space," *Document Analysis Systems, IAPR Workshop on*, 2010.
- [5] L. Hu and R. Zanibbi, "HMM-Based Recognition of Online Handwritten Mathematical Symbols Using Segmental K-Means Initialization and a Modified Pen-Up/Down Feature," *International Conference on Document Analysis and Recognition*, vol. 0, pp. 457–462, 2011.
- [6] F. Álvaro, J.-A. Sánchez, and J.-M. Benedí, "Classification of On-line Mathematical Symbols with Hybrid Features and Recurrent Neural Networks," in *International Conference on Document Analysis and Recognition*, 2013, pp. 1012–1016.
- [7] H.-J. Winkler, "HMM-based handwritten symbol recognition using on-line and off-line features," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, 1996, pp. 3438–3441.
- [8] B. Keshari and S. Watt, "Hybrid mathematical symbol recognition using support vector machines," in *International Conference on Document Analysis and Recognition*, vol. 2, 2007, pp. 859–863.
- [9] F. Álvaro, J.-A. Sánchez, and J.-M. Benedí, "Recognition of on-line handwritten mathematical expressions using 2D stochastic context-free grammars and hidden Markov models," *Pattern Recognition Letters*, vol. 35, no. 0, pp. 58–67, 2014.
- [10] A.-M. Awal, H. Mouchère, and C. Viard-Gaudin, "The problem of handwritten mathematical expression recognition evaluation," *Frontiers in Handwriting Recognition, Int. Conf. on*, vol. 0, pp. 646–651, 2010.
- [11] U.-V. Marti and H. Bunke, "Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 01, pp. 65–90, 2001.
- [12] P. Dreuw, D. Rybach, C. Gollan, and H. Ney, "Writer adaptive training and writing variant model refinement for offline arabic handwriting recognition," in *Document Analysis and Recognition. International Conference on*, 2009, pp. 21–25.
- [13] A. Toselli, A. Juan, and E. Vidal, "Spontaneous Handwriting Recognition and Classification," in *Proc. of the 17th International Conference on Pattern Recognition*, Cambridge, UK, August 2004, pp. 433–436.
- [14] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 4, pp. 509–522, 2002.
- [15] B. Su, X. Ding, L. Peng, and C. Liu, "A novel baseline-independent feature set for arabic handwriting recognition," in *Document Analysis and Recognition, International Conference on*, 2013, pp. 1250–1254.
- [16] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 5, pp. 855–868, 2009.
- [17] L. Ouyang and R. Zanibbi, "Identifying layout classes for mathematical symbols using layout context," in *IEEE Western New York Image Processing Workshop*, 2009.
- [18] L. Hu and R. Zanibbi, "Segmenting Handwritten Math Symbols Using AdaBoost and Multi-scale Shape Context Features," *Int. Conference on Document Analysis and Recognition*, vol. 0, pp. 1180–1184, 2013.
- [19] F. Álvaro and R. Zanibbi, "A shape-based layout descriptor for classifying spatial relationships in handwritten math," in *ACM Symposium on Document Engineering (DocEng)*, 2013, pp. 123–126.
- [20] S. MacLean, G. Labahn, E. Lank, M. Marzouk, and D. Tausky, "Grammar-based techniques for creating ground-truthed sketch corpora," *International Journal on Document Analysis and Recognition*, vol. 14, pp. 65–74, 2011.
- [21] A. Toselli, M. Pastor, and E. Vidal, "On-Line Handwriting Recognition System for Tamil Handwritten Characters," in *Pattern Recognition and Image Analysis*, 2007, vol. 4477, pp. 370–377.
- [22] A. Giménez, I. Khoury, and A. Juan, "Windowed Bernoulli Mixture HMMs for Arabic Handwritten Word Recognition," in *Frontiers in Handwriting Recognition, Int. Conference on*, 2010, pp. 533–538.
- [23] P. Doetsch, M. Hamdani, H. Ney, A. Giménez, J. Andrés-Ferrer, and A. Juan, "Comparison of Bernoulli and Gaussian HMMs Using a Vertical Repositioning Technique for Off-Line Handwriting Recognition," in *Frontiers in Handwriting Recognition (ICFHR), International Conference on*, 2012, pp. 3–7.
- [24] M. Yang, K. Kpalma, and J. Ronsin, "A Survey of Shape Feature Extraction Techniques," in *Pattern Recognition*, P.-Y. Yin, Ed., Nov. 2008, pp. 43–90.
- [25] B. Pearlmutter, "Learning state space trajectories in recurrent neural networks," in *International Joint Conference on Neural Networks*, vol. 2, 1989, pp. 365–372.
- [26] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *Signal Processing, IEEE Transactions on*, pp. 2673–2681, 1997.
- [27] A. Graves, "RNNLIB: A recurrent neural network library for sequence learning problems," <http://sourceforge.net/projects/rnnl/>.
- [28] R. Plamondon and S. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, pp. 63–84, 2000.