ICFHR 2014 Competition on Recognition of On-line Handwritten Mathematical Expressions (CROHME 2014)

Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, Utpal Garain

ICFHR 2014 Competition on Recognition of On-line Handwritten Mathematical Expressions (CROHME 2014)

H. Mouchère and C. Viard-Gaudin
LUNAM Université de Nantes
IRCCyN/IVC – UMR CNRS 6597
{harold.mouchere, christian.viard-gaudin}
@univ-nantes.fr

R. Zanibbi
Department of Computer Science
Rochester Institute of Technology
Rochester, NY, USA
rlaz@cs.rit.edu

U. Garain
Computer Vision and Pattern Recognition (CVPR)
Indian Statistical Institute
utpal@isical.ac.in

***Abstract*— We present the outcome of the latest edition of the CROHME competition, dedicated to on-line handwritten mathematical expression recognition. In addition to the standard full expression recognition task from previous competitions, CROHME 2014 features two new tasks. The first is dedicated to isolated symbol recognition including a reject option for invalid symbol hypotheses, and the second concerns recognizing expressions that contain matrices. System performance is improving relative to previous competitions. Data and evaluation tools used for the competition are publicly available.**

Keywords-component; mathematical expression recognition, online handwriting, spatial relations, performance evaluation.

I. INTRODUCTION

CROHME, which stands for Competition on Recognition of On-line Handwritten Mathematical Expressions, is now a well-established annual competition. This edition is the fourth in a series started in 2011 at ICDAR in Beijing [1], and then continued with ICFHR in Bari [2], and recently ICDAR in Washington [3]. There are a number of explanations for the sustained interest in these competitions. First, handwritten math recognition is a very challenging problem, which significantly increases difficulties associated with handwriting recognition. Not only is the number of symbol classes large (it has been fixed to 101 for CROHME 2014), but more importantly, the two dimensional nature of the notation, as compared to the left-right layout of a standard text, greatly complicates the segmentation and interpretation stages. Second, the availability of an efficient mathematical recognition tool would be of a tremendous interest for everyone who is struggling to input expressions in a document or math-aware search engine [4]. And third, nowadays, many touching devices, from small digital pads to large white boards, are available and allow the user to write and store the corresponding ink traces.

With this 2014 edition, we have introduced two major innovations. One of them is the introduction of a task related to isolated symbol classification (*Task 1*). It can be considered as a sub-task of the main problem of recognizing mathematical expressions. This task should be more accessible for newcomers to CROHME, since it requires only the design and training of a classifier. However, we have required that this classifier has the capability to reject invalid symbol segmentation hypotheses, where the input does not correspond to an actual symbol. The second change is a new task dealing with the recognition of matrices (*Task 3*), which opens the competition to new mathematical domains. On the other hand, we have kept the main task dedicated to general mathematical expressions largely unchanged, by keeping the same symbol set, expression grammar, and training set (*Task 2*). Of course the test sets for all three tasks are new, with new writers and expressions.

Table I summarizes the evolution of the CROHME competition in terms of tasks, expression grammars and symbol class sets, data sets for the traditional expression recognition task (*Task 2*) and participants. Roughly the same number of organizations participated in CROHME 2014 as CROHME 2013, but with the addition of two new tasks.

TABLE I. CROHME EVOLUTION

	2011	**2012**	**2013**	**2014**
Expression Grammars	2	3	2	1
# Symbol Classes	60	75	101	101
Task 2 Training expressions	921	1 341	8 836	8 836
Task 2 Test expressions	348	486	671	986
Participations for *Task 1*	-	-	-	8
Participations for *Task 2*	4	6	8	7
Participations for *Task 3*	-	-	-	2

II. COMPETITION PROTOCOL AND DATA

New test data for all three tasks, along with training data for the new matrix recognition task (*Task* 3) were collected by the three organizing labs: IRCCyN/IVC (France), RIT/DPRL (USA), and ISI/CVPR (India), ensuring a large variability in writing styles. Test expressions were selected from Wikipedia articles.

Different capture devices were used to collect expressions by the three labs. Large white boards and tablet PC were used at IVC, expressions were drawn by finger on iPads at the DPRL, and a Wacom tablet with stylus was used at ISI. The training data are already publicly available on the TC11 website[1].

A. Isolated Symbol Recognition (Task 1)

Given the importance of accurate symbol classification for handwritten math recognition, a new task to evaluate symbol recognizers has been proposed. In this new task, symbols are extracted from complete handwritten expressions, and have not been written as single symbols.

[1] http://www.iapr-tc11.org/mediawiki/index.php/CROHME:_Competition_on_Recognition_of_Online_Handwritten_Mathematical_Expressions

Since symbols need to be segmented before being classified, we also consider mis-segmented (i.e. invalid) symbols. To produce samples of mis-segmented symbols, we have randomly selected sequences of one to four strokes written in time-order that do not correspond to valid symbols. This produces both under- and over-segmented symbols. Mis-segmented symbols should be assigned to a reject class (junk). As seen in Table II, the proportion of valid symbols to junk in the test set is around 1:1 (i.e. roughly as invalid as valid symbols). We compare symbol classifiers two ways: 1) over valid symbols only, and 2) over both valid symbols and mis-segmented symbols. Table II provides additional information about the data sets used for Task 1. A script was provided that allowed participants to generate junk samples automatically from the training data provided for Task 2.

As suggested by Tables II and IV, symbols used for Task 1 were generated from test expressions in Task 2.

TABLE II. ISOLATED SYMBOL DATASETS (TASK 1)

Training Set	Valid Symbols	85 781
	Invalid Symb. (Junk)	User-specified (script)
Test Set	Valid Symbols	10 061
	Invalid Symb. (Junk)	9 161

B. Matrix Recognition (Task 3)

This task has been introduced to foster new research projects, since to date, few systems have been proposed to handle matrix notations. An expression with matrices can be considered as an expression with nested sub-expressions arranged in a tabular layout (i.e. in rows and columns). Thus to recognize expressions with matrices, systems must first detect the presence of one or more matrices, then their cells, the arrangement in rows and columns, and their contents, which can be any mathematical expression consistent with the grammar in task 2. Hence, a new representation has been proposed allowing multi-level evaluation that considers matrices, rows, columns, cells, and of course symbols.

TABLE III. MATRIX DATASETS (TASK 3)

Training Set	Matrices	362
	Number of cells	2 332
	Number of symbols	4 281
Test Set	Matrices	175
	Number of cells	1 075
	Number of symbols	2 101

As can be seen from Table III the average number of cells per matrix is around 6, corresponding to a simple 2×3 matrix, and there are about two symbols per cell on average. Examples of matrix expressions used in the competition are shown in Figure 1.

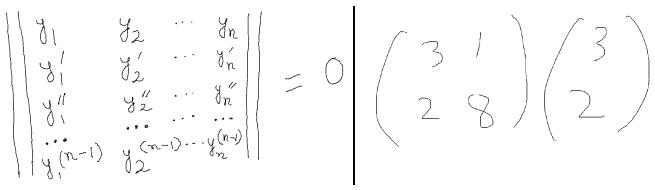

Figure 1. Two examples of expressions with matrices.

C. Mathematical Expression Recognition (Task 2)

Task 2 considers recognition of complete handwritten expressions, requiring systems to identify symbol locations and identities, along with the spatial layout of symbols. This is the core task of the competition. We have kept the same constraints as in 2013 the competition [3] focusing on ***Part 4*** from CROHME 2013, to allow for tracking progress.

TABLE IV. MATHEMATICAL EXPRESSIONS DATASETS (TASK 2)

Training Set	Expressions	8 836
	Number of symbols	85 781
Test Set	Expressions	986
	Number of symbols	10 061

III. SYSTEM DESCRIPTION

This section describes the eight participating teams that submitted a total of 16 systems to participate for the different tasks. Table V presents the affiliation and the tasks participation of each one. A brief description of each system is provided in the remainder of this section.[2]

TABLE V. CROHME PARTICIPANTS

System	Affiliation	Tasks
I	Universitat Politècnica de València	1 – 2 – 3
II	University of São Paulo	1 – 2
III	MyScript	1 – 2 – 3
IV	Rochester Institute of Technology, DRPL	1 – 2
V	Rochester Institute of Technology, CIS	1 – 2
VI	Tokyo University of Agriculture and Technology	1 – 2
VII	University of Nantes, IRCCyN	2
VIII	ILSP/Athena Research and Innovation Center	1

A. Universitat Politècnica de València (System I)

Two systems are proposed by F. Alvaro from Pattern Recognition and Human Language Technology (PRHLT) research center, Universitat Politècnica de València.

Task 1. Given an isolated math symbol represented by a set of strokes, seven online features were computed for each point. Then, an image of the symbol was generated using linear interpolation, setting the image height to 40 pixels and keeping the aspect ratio. Nine offline features were extracted from each column of the rendered image. Finally, a BLSTM-RNN classifier was trained for each set of features (online and offline) and the outputs of both classifiers were combined. A detailed description of this classifier and feature extraction (PRHLT + FKI) can be found in [5]. The rnnlib library[3] was used and the feature extraction tools are also publicly available as open-source from www.prhlt.upv.es.

Tasks 2 and 3. The seshat[4] system parses expressions using two-dimensional stochastic context-free grammars. It is an advanced model evolution of the system presented in [6]. The parser first computes several hypotheses for

[2] Note that systems IV and VII are from organizing labs.

[3] A. Graves, RNNLIB: A recurrent neural network library for sequence learning problems, http://sourceforge.net/projects/rnnl/

[4] https://github.com/falvaro/seshat

symbol segmentation and recognition at stroke level using several stochastic sources (symbol classifier, segmentation model and duration model). Symbol classification is performed by a combination of online and offline features with a BLSTM-RNN classifier [5] as in task 1. Then, the parser builds new hypotheses by combining smaller sub problems using a spatial relationship classifier (a Gaussian Mixture Model with geometric features). Finally, after parsing the most likely derivation tree represents the recognized math expression. Hence, symbol recognition, segmentation and the structural analysis are globally determined using the grammar and contextual information.

For the matrix recognition task, we used the best system trained for task 2 with a grammar that accounts for matrices. There are mainly two differences in this task. First, the number of rows/columns must match when combining hypothesis in order to create a matrix. Second, the spatial relationships between rows or columns take into account the differences between the centers of each row/column.

B. University of São Paulo (System II)

This system is proposed by Frank Aguilar.

Tasks 1 and 2. The recognition process is divided into three steps: (1) symbol hypotheses generation, (2) baseline tree extraction and (3) syntactical parsing. In the first step, each stroke is combined with its three nearest neighbor strokes. Each combination is considered as a symbol hypothesis. A symbol classifier with a reject option [7] is used to label each hypothesis with its corresponding symbol label or as a junk class (not a symbol). In the second step, several partitions, using the symbol hypotheses with the lowest cost, are generated and a baseline structure tree [8] is built for each partition. A cost that combines the symbol hypothesis costs and relation costs is assigned to each baseline tree. In the final step, the latex code of each baseline tree is extracted and a parser evaluates if it belongs to a predefined mathematical language grammar. The legal expression with lowest cost is selected for output.

C. Myscript, formerly Vision Objects (System III)

Task 1. The *MyScript* Isolated Symbol recognizer uses a two stage process: a feature extraction process followed by Neural Network (Multi-layer perceptron) classifiers that output a list of symbol candidates with recognition scores. The chosen feature set uses a combination of dynamic and static information. Dynamic information is extracted from the trajectory of the pen and is based on information such as the position, direction and curvature of the ink signal. Static features are computed from a bitmap representation of the ink and are typically based on projections and histograms.

Tasks 2 and 3. The *MyScript* Math recognizer is an on-line recognizer that processes digital ink. The overall recognition system is built on the principle that segmentation, recognition and interpretation have to be handled concurrently and at the same level in order to result in the best candidate.

The Math recognition engine analyzes the spatial relationships between all the parts of the equation, in conformity with the rules laid down in its grammar, to determine the segmentation of all its parts. The grammar is defined by a set of rules describing how to parse an equation, each rule being associated with a specific spatial relationship. For instance a fraction rule defines a vertical relationship between a numerator, a fraction bar and a denominator. The matrix recognition uses a specific post-processing to retrieve the rows and columns.

The Math recognizer has also a symbol expert that estimates the probabilities for all the parts in the suggested segmentation. This expert is based on feature extraction stages, where different sets of features are computed. These feature sets use a combination of on-line and off-line information. The feature sets are processed by a set of character classifiers, which use Neural Networks and other pattern recognition paradigms.

The Math recognition engine includes a statistical language model that uses context information between the different symbols depending on their spatial relationships in the equation. Statistics have been estimated on hundreds of thousands of equations. A global discriminant training scheme on the equation level with automatic learning of all classifier parameters and meta-parameters of the recognizer is employed for the overall training of the recognizer. The recognizer has been trained on writing samples that have been collected from writers in several countries.

D. Rochester Institute of Technology, DRPL (System IV)

This system was developed by K. Davila Castellanos and L. Hu from the Document and Pattern Recognition Lab (RIT) with some contributions by F. Alvaro (UPV). The systems are available as open source online.[5]

Task 1. The symbol classifier by K. Davila is an SVM with a Gaussian Kernel trained for probabilistic classification [10] using a combination of on-line and off-line features. On-line features include normalized line length, number of strokes, covariance of point coordinates, and the number of points with high variation in curvature and total angular variation. Off-line features include: normalized aspect ratio; the count, position of the first and position of the last times that traces intersect a set of lines at fixed horizontal and vertical positions (crossings); 2D fuzzy histograms of points and fuzzy histograms of orientations of the lines.

Task 2. The segmenter considers strokes in time series, using AdaBoost with confidence-weighted predictions to determine whether to merge each pair of strokes [9]. The same classifier used in Task 1 is used for Task 2. The parser is a modified version of Eto and Suzuki's MST-based parsing algorithm [11]. The parser recursively groups vertical structures (e.g. fractions, summations and square roots), extracts the dominant operator (e.g. fraction line) in each vertical group, and then detects symbols on the main baseline and on baselines in superscripted and subscripted regions. For each baseline, an MST is built over symbol candidates and the spatial relations between symbols. Spatial relationships are classified using bounding box geometry and a shape context feature for regions around symbol pairs [12].

[5] https://github.com/DPRL

E. Rochester Institute of Technology, CIS (System V)

This system is proposed by Wei Yao and Fan Wang from the Chester F. Carlson Center for Imaging Science (CIS).

Tasks 1 and 2. The segmentation method is based on Hu's work [9]. Given an expression with n strokes, the segmentation method considers merging or splitting the n-1 stroke pairs in time order. For each stroke pair 405 features are measured: geometric features (23), multi-scale shape context features (180), and classification probabilities (202). Features are scaled and then reduced to 100 dimensions using PCA. Using LIBSVM[6], an SVM with a radial basis function kernel is trained using cross validation and grid-search to obtain the best observed parameter pair (C, γ).

For classification, five features are used: pen-up/down, the density in the center of a 3×3 matrix containing symbol strokes, normalized y-coordinate, sine of curvature and cosine of vicinity from slope [13]. Features were then scaled.

Segmented and classified symbols are then used as input for parsing. The first parsing stage detects "strong" geometric relationships (above, below and inside). To solve the above/below spatial relationships, we first search for symbols that horizontally overlap horizontal lines. Then we search for root symbols, to identify contained expressions. The expression is then split into several sub-expressions (numerator and denominator of a fraction, base of a square root, and remaining symbols) that do not contain the previous relationships. Then, shape feature Polar Histograms [12] are used to detect "weak" geometric relationships (Right, superscript and subscript). A polar histogram layout descriptor in 15 (distance) × 20 (angle) resolution is used to generate 300 features for each two adjacent symbols. After PCA reduction, SVM classifier is employed to classify relationships.

F. Tokyo Univ. of Agriculture and Technology (System VI)

This system was developed at Nakagawa Laboratory by Anh Duc Le, Truyen Van Phan, and Masaki Nakagawa.

Task 1. The system normalizes each input symbol pattern with the Line Density Projection Interpolation (LDPI) normalization method. Then a 512D feature vector is extracted based on Normalization-Cooperated Gradient Feature (NCGF). The feature vector is reduced to 300D by FLDA. The reduced vector is classified by a Generalized Learning Vector Quantization (GLVQ) classifier. For classifying invalid symbols more accurately, patterns are clustered into 64 clusters by using the LBG algorithm for training.

Task 2. The system is an upgrade of the system used in CROHME 2013. It contains 3 processes: symbol segmentation, symbol recognition, and structure analysis. We propose body box and a method to learn structural relations from training patterns without any heuristic decisions by using two SVM models. In structure analysis, we employ stroke order to reduce the complexity of the parsing algorithm to $O(n^3|P|)$ where n is the number of symbols, and P the number of production rules. The detail of our system is presented in [14]. We did not use more data than the CROHME 2013 training part to train the system and the CROHME 2013 testing part for validation.

G. University of Nantes, IRCCyN (System VII)

Tasks 2. This system simultaneously optimizes expression segmentation, symbol recognition, and 2-D structure recognition under the restriction of an expression grammar [15]. This strategy is adopted to prevent errors from propagating from one step to another. The approach transforms the recognition problem into a search for the best possible interpretation of a sequence of input strokes.

The symbol classifier is a classical neural network, a Multilayer Perceptron, which has the capability to reject invalid segmentation hypotheses, unlike most existing systems. The originality of this system stems from the global learning schema, which allows training a symbol classifier directly from mathematical expressions. The advantage of this global learning is to consider the 'junk' examples (i.e. invalid symbol segments) and include them in the training set for the symbol classifier's reject class. Furthermore, contextual modeling based on structural analysis of the expression is employed, where the models are learnt directly from expressions using a global learning scheme. This global approach is applicable for other 2-D languages, for example, flowchart recognition [16].

H. Institute for Language and Speech Processing/Athena Research and Innovation Center (System VIII)

Task 1. This system is based on a template elastic matching distance [17]. In this method, the pen-direction features are quantized using the 8-level Freeman chain coding scheme and the dominant points of the stroke are identified using the first difference of the chain code. The distance between two symbols results from the difference of the respective chain codes of the variable speed normalization of dominant points weighted by the local length proportions of the strokes.

IV. RESULTS

The results which are displayed in this section give only a partial view of the behavior of the different systems, a much more detailed analysis can be achieved by exploiting the information provided by the label graph evaluation tools [18]. Additional tables will be prepared and displayed on the CROHME website[7].

A. Isolated Symbol Recognition (Task 1)

Two important behaviors of the proposed classifiers are evaluated: the accuracy of the classifier with respect to the true symbols, i.e. ignoring the junk class, and the ability of the classifier to accept true symbols while rejecting junk samples. The first issue is evaluated considering only the true symbols with the classical *Top N* recognition rate, the mean position of the true label is also provided. To evaluate the reject option of the classifier, in addition to the previous values, we compute the True Acceptance Rate (*TAR* is also

[6] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[7] http://www.isical.ac.in/~crohme/

the recall rate for symbols) and the False Acceptance Rate (*FAR*). These rates do not depend of the ratio between the number of symbol and junk samples. Table VI presents these rates for all submitted systems. When a system does not include the reject option, n.r.o. will be displayed.

TABLE VI. ISOLATED SYMBOL RECOGNITION (TASK 1)

System	Without Junk (101 classes)		With Junk (102 classes)		With Junk (2 classes)	
	Top1	True Mean Position	Top1	True Mean Position	TAR	FAR
I	91.24	1.20	n.r.o.			
	89.79	1.25	84.14	1.28	80.29	**6.44**
II	82.72	1.60	79.11	1.50	82.49	14.63
III	91.04	1.19	**85.54**	1.23	**87.12**	10.39
IV	88.66	1.27	83.61	1.29	83.52	9.03
V	85.00	1.37	71.19	1.48	86.84	36.85
VI	84.31	1.53	n.r.o			
	82.08	1.63	76.24	1.57	77.98	16.47
VIII	77.25	1.68	n.r.o			

As expected, performances are decreasing when the junk class, which covers many different configurations, has to be managed. The decrease is most of the time around 5% but reaches 14% for one system. Interestingly, the top 3 systems (III, I and IV) are based on different classifiers, respectively MLP, RNN, and SVM. The operating points that define the trade-off between TAR and FAR are a bit different for the different systems and it turns out that there is a larger difference in the FAR rates which are approximately ranging from 6% to 37% than in the TAR rates which are comprised between 78% and 87%.

B. Matrix Recognition (Task 3)

It has been possible to extend the approach based on the label graph metric used in the last CROHME competition. To this end, new object labels have been added to represent the specificities of matrices: cell, row, column, and matrix. In addition to the classical full expression recognition rate, and symbol and layout recall rates (as with task 2), it is also possible to extract recall rates for this four specific objects of interest.

TABLE VII. MATRIX RECOGNITION (TASK 3)

	System I	System III
Expression Rate	31.15	53.28
Symbol Reco Rate	87.43	89.81
Matrix Recall Rate	73.14	92.57
Row Recall Rate	70.59	92.00
Column Recall Rate	50.84	69.16
Cell Recall Rate	55.35	71.07

Only two systems took part in this task. It can be noticed that this task is harder than regular mathematical expression recognition. However, for the first edition of this task, as presented in section II-B, the complexity of the expressions was moderated. As a matter of fact, at the expression level the best rate is 53.28%, somehow below the best rate displayed in Table IX. For the matrix case, the spatial relationships are more elaborated since every piece of ink has to be assigned the right symbol, cell, row and column for a perfect recognition. Looking at the other rows of Table VII, it is worth to note that rows are more correctly extracted than columns for both systems.

C. Mathematical Expression Recognition (Task 2)

A first experiment, presented in Table VIII compares on the same test set (Test set 2013), the performances of the previous systems [3] with the current ones (2014). Most of the systems succeed to improve their performances on the last year test set. This is an encouraging sign of progress, tempered to some degree by the possibility of using, intentionally or not, this test set as a validation set for tuning the meta-parameters of the 2014 systems.

TABLE VIII. MATHEMATICAL EXPRESSION RECOGNITION (TASK 2, ON TEST SET 2013, 671 EXPRESSIONS)

System	2013 Syst. [3]	2014 Systems			
	Correct (%)	Correct (%)	Segments (recall%)	Seg+Class (recall%)	Tree Relations (recall%)
I	23.40	30.70	93.01	83.33	83.93
II	9.39	15.05	74.84	63.58	57.73
III	**60.36**	**62.15**	**98.27**	**93.47**	**94.82**
IV	14.31	19.52	86.44	74.98	71.57
V	-	17.44	88.52	74.88	65.34
VI	19.97	25.04	90.96	76.13	75.42
VII	18.33	17.59	89.64	71.47	70.50

The results on the new test set (2014) are provided in Tables IX and X. The rates at the expression level with no error, and with at most one to three errors are given in Table IX. In addition to the correct rate at the expression level, table X also provides recall and precision metrics for symbol segmentation ('Segments'), detection of symbols with their correct classification ('Seg+Class'), and spatial relationship detection. A correct spatial relationship between two symbols requires that both symbols are correctly segmented and in the right relationship. For spatial relationships, we use metrics based on edges in the symbol layout trees ('Tree Rels.').

TABLE IX. CORRECT EXPRESSION RECOGNITION RATE (SYSTEMS 2014, TEST SET 2014, 10061 EXPRESSIONS). EXPRESSION LEVEL EVALUATION

System	Correc (%)t	<= 1 error	<= 2 errors	<= 3 errors
I	37.22	44.22	47.26	50.20
II	15.01	22.31	26.57	27.69
III	**62.68**	**72.31**	**75.15**	**76.88**
IV	18.97	28.19	32.35	33.37
V	18.97	26.37	30.83	32.96
VI	25.66	33.16	35.90	37.32
VII	26.06	33.87	38.54	39.96

To evaluate the complexity of 2014 versus 2013 test sets, we can compare each of the systems on these two datasets in Tables VIII and IX. We can observe a large stability, except for systems I and VII, which show an improvement around 8 % on the new test set. One hypothesis is that the new test is a little bit simpler than the former although the mean number

of symbols per expression has slightly increased from 9.06 to 10.20.

TABLE X. MATHEMATICAL EXPRESSION RECOGNITION (SYSTEMS 2014, TEST SET 2014, 10061 EXPRESSIONS). OBJECT LEVEL EVALUATION

System	Segments (%)		Seg+Class (%)		Tree Rels. (%)	
	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.
I	93.31	90.72	86.59	84.18	84.23	81.96
II	76.63	80.28	66.97	70.16	60.31	63.74
III	**98.42**	**98.13**	**93.91**	**93.63**	**94.26**	**94.01**
IV	85.52	86.09	76.64	77.15	70.78	71.51
V	88.23	84.20	78.45	74.87	61.38	72.70
VI	83.05	85.36	69.72	71.66	66.83	74.81
VII	89.43	86.13	76.53	73.71	71.77	71.65

It is also interesting to note that the large differences at expression level between the systems (cf. Table IX, from 15.01% to 62.68%) are attenuated at the object level (cf Table X) in term of the segmentation (from 76.63% to 98.42%), recognition (from 66.97% to 93.91%) or layout (from 60.31% to 94.26%).

Table XI shows the raw Hamming distances between labeled graphs and the average percentage of mislabeled node and edge labels per expression (ΔBn) along with a variation designed to weight segmentation, classification and parsing decisions more equally (ΔE). It allows a more detailed analysis, e.g. we can see how are recognized the miss-segmented symbols. For example the system IV has less stroke label errors than system VII even if the seg+class rates are comparable.

TABLE XI. MATHEMATICAL EXPRESSION RECOGNITION (TASK 2, ON TEST SET 2014). STROKE AND STROKE TO STROKE LEVEL EVALUATION. (13 796 STROKES FROM 10 061 SYMBOLS, 288 660 STROKE PAIRS)

System	Label Hamming Distances				μ error (%)	
	Strokes	Pairs	Seg	Rel	ΔB_n	ΔE
I	2026	7174	2540	4634	6.23	11.98
II	4523	17449	7772	9677	14.42	25.77
III	845	2489	836	1653	2.21	4.56
IV	3406	12768	5328	5328	9.98	19.17
V	2742	13477	4386	9091	10.26	18.24
VI	4039	13325	5670	7655	10.05	19.43
VII	3558	11795	4388	7407	9.38	17.66

V. CONCLUSION

Eight groups from five different countries took part in this fourth edition of the CROHME competition. Concerning the results, for task 1, related to math symbol classification with a reject class, we can mention the close outcome since three systems are separated by less than 2%. The new task 3 about matrix detection and recognition attracts only 2 participants but shows the feasibility and the difficulty of this task. Seven systems participated in the main task 2 related to full expression recognition. The best system (III, *MyScript*) reaches 62.68% which is comparable to the previous competition in the same condition (using external private training data). The best system using only the CROHME datasets (I, *UPV*) improves significantly its results up to 37.22%, maybe because of a stronger symbol classifier.

The CROHME committee wants to highlight that some of the proposed systems are open sources (UPV, RIT), we hope that this will help the progress in this research domain.

REFERENCES

[1] H. Mouchère, C. Viard-Gaudin, D. H. Kim, J. H. Kim and U. Garain, "CROHME 2011: Competition on Recognition of Online Handwritten Mathematical Expressions", in Proc. ICDAR:1497-1500, 2011.

[2] H. Mouchère, C. Viard-Gaudin, D. H. Kim, J. H. Kim and U. Garain, "ICFHR 2012 Competition on Recognition of On-Line Mathematical Expressions (CROHME 2012)", in Proc. ICFHR: 811-816, 2012.

[3] H. Mouchère, C. Viard-Gaudin, R. Zanibi, D. H. Kim, J. H. Kim and U. Garain, "ICDAR 2013 CROHME: Third International Competition on Recognition of Online Handwritten Mathematical Expressions", in Proc. ICDAR, 2013.

[4] R. Zanibbi and D. Blostein, "Recognition and Retrieval of Mathematical Expressions," Int'l. J. Document Analysis and Recognition (IJDAR), 15(4):331-357, 2012.

[5] F. Alvaro and J.A. Sanchez and J.M. Benedi, "Offline Features for Classifying Handwritten Math Symbols with Recurrent Neural Networks," International Conference on Pattern Recognition, 2014.

[6] F. Alvaro and J.A. Sanchez and J.M. Benedi, "Recognition of On-line Handwritten Mathematical Expressions Using 2D Stochastic Context-Free Grammars and Hidden Markov Models," Pattern Recognition Letters, vol. 35, pp. 58-67, 2014.

[7] F. Julca–Aguilar, N. S. T. Hirata, C. Viard–Gaudin, H. Mouchère, S. Medjkoune. "Mathematical symbol hypothesis recognition with rejection option". in Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR 2014, in press.

[8] R. Zanibbi, D. Blostein, and J. Cordy. "Baseline structure analysis of handwritten mathematics notation". In Proc.. Sixth Intern. Conference on Document Analysis and Recognition, pages 768–773, 2001.

[9] L. Hu and R. Zanibbi, "Segmenting Handwritten Math Symbols Using AdaBoost and Multi-Scale Shape Context Features", International Conference on Document Analysis and Recognition, pp. 1212-1216, Washington DC, USA, Aug. 2013.

[10] K. Davila, S. Ludi and R. Zanibbi. "Using Off-line Features and Synthetic Data for On-line Handwritten Math Symbol Recognition". Proc. ICFHR 2014, in press.

[11] Y. Eto and M. Suzuki, "Mathematical formula recognition using virtual link network". in Proc. ICDAR:762-767, 2001.

[12] F. Alvaro, R. Zanibbi, "A shape-based layout descriptor for classifying spatial relationships in handwritten math". ACM Symposium on Document Engineering, pp. 123-126, 2013.

[13] M. Eichenberger-Liwicki and H. Bunke, "Feature selection for HMM and BLSTM based handwriting recognition of whiteboard notes," International Journal on Pattern Recognition and Artificial Intelligence 23(5), pp. 907– 923, 2009.

[14] D. Le, T. Van Phan, and M. Nakagawa, "A System for Recognizing Online Handwritten Mathematical Expressions and Improvement of Structure Analysis", DAS, 11th IAPR Workshop on Document Analysis Systems, April 2014, France.

[15] A.-M. Awal, H. Mouchère and C. Viard-Gaudin, "A global learning approach for an online handwritten mathematical expression recognition system", in Pat. Recogn. Letters, vol 35, pp. 68-77, 2014.

[16] A.-M. Awal, G. Feng, H. Mouchère and C. Viard-Gaudin, "First Experiments on a new Online Handwritten Flowchart Database", in Proc. Document Recognition and Retrieval XVIII, 2011.

[17] F. Simistira, V. Katsouros, and G. Carayannis, "A Template Matching Distance for Recognition of On-Line Mathematical Symbols", in Proc. ICFHR, pp. 415-420, 2008.

[18] R. Zanibbi., H. Mouchère, C. Viard-Gaudin. "Evaluating structural pattern recognition for handwritten math via primitive label graphs", Proc Document Recognition and Retrieval XX,. USA, 2013.