ICFHR 2016 CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions

H. Mouchère, C. Viard-Gaudin
UBL 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
CVPRU, Indian Statistical Institute
Kolkata, 700108, India
utpal@isical.ac.in

***Abstract*—This paper presents an overview of the 5th Competition on Recognition of Online Handwritten Mathematical Expressions (CROHME). As in previous years, the main task is formula recognition from handwritten strokes (Task 1). Additional tasks include classification of isolated symbols (Task 2a), classification of isolated valid and invalid symbols (Task 2b), a new task on parsing formula structure from valid handwritten symbols (Task 3), and parsing expressions with matrices (Task 4, experimental). In total, eleven (11) research labs registered for the competition, with six (6) teams submitting results. Innovations for this CROHME included providing a corpus of formulae from Wikipedia to train language models, and an online system for result submission. The highest recognition rates were obtained by MyScript corporation (Task 1. 67.65%, 2a. 92.81%, 2b. 86.77%, 3. 84.38%, and 4. 68.40%). Using only provided training data, the highest recognition rates were obtained by WIRIS corporation (Task 1. 49.61%, Task 3. 78.80%, Task 4. 56.40%), the Tokyo University of Agriculture and Technology (Task 2a. 92.28%), and RIT (Task 2b. 83.34%). The competition results suggest that recognition of handwritten formulae remains a difficult structural pattern recognition task.**

***Keywords*-mathematical expression recognition, handwriting recognition, spatial relations, performance evaluation.**

I. Introduction

Recognition of mathematical expressions has been an active area of research for over two decades [1]. While the earliest work on online handwritten math recognition was completed in the 1960's (e.g., by Anderson at MIT), by the 1980's research on math notation recognition concentrated on images of printed expressions. Since that time, both offline and online handwritten expression recognition have gradually attracted more attention. Although several results were reported in the literature, progress was unclear due to the absence of common datasets and evaluation metrics. To overcome this bottleneck, the Competition on Recognition of Handwritten Mathematical Expressions (CROHME) was introduced as a shared task at the ICDAR 2011 conference held in Beijing.[1]

After receiving a very encouraging response from the research community, CROHME has been held regularly, with four editions of this event already having been successfully completed. A recent paper provides an overview of previous CROHMEs, and highlights the contributions of this event to research in math recognition and structural pattern recognition [2].

This fifth edition of CROHME was organized to continue encouraging activity in handwritten math recognition research, and to improve available data, tools and benchmarks for research in the area. For CROHME 2016, four tasks are defined:

- **Task-1** Formula Recognition: Isolated formula recognition from handwritten strokes (main task)
- **Task-2** Symbol Recognition: Classification of isolated math symbols
- **Task-3** Structure Recognition: Parsing formula structure from given (valid) handwritten symbols
- **Task-4** Matrix Recognition: Recognition of expressions containing matrices from handwritten strokes

Task-1 is the main task. Task-2 is split into two sub-tasks; In Task-2a only valid symbols are provided as input, while in Task-2b incorrect symbol segmentations (*junk*) are mixed with valid symbols. Task-3 has been introduced to evaluate spatial relationship recognition (i.e., structure recognition) independently of symbol recognition. Task-4 remains an experimental task for recognizing expressions with matrices.

Eleven groups registered for the competition, and six groups submitted results. For the main task, the highest recognition rate is 5% higher than in CROHME 2014 (67.65% vs. 62.68%). This is a positive trend, but these rates suggest that recognizing handwritten math remains a difficult problem, likely due to the complex interactions involved in segmenting, classifying, and relating symbols, and the number of decisions that need to be made correctly in order to recognize a non-trivial formula [2].

The rest of the paper is organized as follows. Section-II presents the competition protocol and data. Participant-provided system descriptions are given in Section-III. The evaluation results and related discussion are presented in Section-IV, and Section-V concludes the paper.

[1] http://www.icdar2011.org

II. Competition Protocol and Data

A. Data

Thanks to the release of data from previous CROHME competitions, more than 10,000 labeled handwritten formulae are publicly available.[2] We created training, validation and some test sets from this data. The validation sets allow benchmarking against test sets from the last two CROHME competitions, and/or for use in preventing over-fitting by machine learning algorithms (e.g., neural nets). This use of existing data allowed our data collection efforts to focus on providing new test sets for the main task, Task 1 (Formula Recognition) and for the experimental Task 4 (Matrix Recognition). Table I shows which data have been used for each task.

Table I
Tasks in the Competition and Data Sets used in each Task. * indicates new data sets created for the competition.

	Training Set	Validation Set	Test Set
Task 1 - Formulae	Train 2014 8 836 expr.	Test 2014 986 expr.	**2016 Test*** 1 147 expr.
Task 2 - Symbols	Train 2014	Test 2013	Test 2014
2a. Valid	85 802 symb.	10 061 symb.	10 019 symb.
2b. Valid+Junk	74 284 junk	9161 junk	**8416 junk***
Task 3 - Structure (new for 2016)	Train 2014 8 836 expr.	Test 2013 671 expr.	Test 2014 986 expr.
Task 4 - Matrices (experimental)	M. Train 2014 362 expr.	M. Test 2014 175 expr.	**M. Test 2016*** 250 expr.

For Tasks 2a and Task 3, previous CROHME data sets from 2014 were used to limit the data preparation effort of the organizers. For Task 2b, the 'valid' symbols were the same as in the test set for CROHME 2014, but the set of 'junk' symbols was recomputed using a different random seed for the 'junk' generation script.

Data Collection. The new data sets for Tasks 1 and 4 were collected and labeled by the organizers at the University of Nantes in France. As the Wikipedia corpus used to define test expressions in the previous competition was made available to participants (see below), we used a new source of expressions: arXiv paper sources from 2000 and 2001 processed and made available for the KDD 2003 Cup [3]. Following the same selection process as in previous competitions [2], a corpus of 1147 expressions was selected, according to the same expression grammar and symbol frequency constraints used to create the CROHME 2014 Test set. Expressions for Task 4 with matrices were randomly selected from the same corpus as in 2014, so some expressions are the same as in the 2014 Matrix Test set. Some modifications were made manually to increase the complexity of some very simple matrices, and to reduce the frequency of symbols '0' and '1'.

[2]http://tc11.cvc.uab.es/datasets/CROHME-2014_2

Both individual formulae and expressions with matrices were collected from 50 writers using three types of devices with different resolutions: 1) pen-based 12" tabletPC, 2) 27" touch-screen using a finger to draw, and 3) a pen-based interactive whiteboard using a video projector. During acquisition, a GUI shows an expression to copy (rendered by LaTeX) and this LaTeX string is stored with the handwritten strokes in a raw InkML file. To generate ground truth for each expression, the LaTeX string is parsed to count how many symbols have to be found, and from which classes. This symbol information is then used to guide automatic symbol recognition and parsing of the ink, searching for the known symbols and relationships. This strategy is simple but effective for most symbols, but requires manual checking followed by editing to correct mistakes. On average, the full ground-truthing process takes one minute per expression.

Wikipedia Formulae. State-of-the-art handwritten math recognizers use language models such as Stochastic Context-Free Grammars [2]. To support parameter fitting for language models, we made available over 592,000 formulae from English Wikipedia in LaTeX and Presentation MathML formats,[3] taken from the recent NTCIR-12 Math IR competition [4].

Data Usage by Participants. All participants used the provided training data sets. WIRIS was the only system to make use of the provided Wikipedia formula corpus to train their language models, while MyScript was the only system that made use of an additional private training set (approx. 30,000 formulae). RIT used a synthetic data generation technique to produce a training set five times larger than the original for Task 2 [5]. All participants used the validation data sets to prevent over-fitting by machine learning algorithms and/or model selection.

B. Evaluation Metrics and Tools

As in the last two CROHME competitions, updated and improved versions of the CROHMELib and LgEval libraries[4] were made available to participants, and used to compute results for the competition [2]. Included are tools that provide performance metrics at the stroke, symbol, and expression level, along with automated error analyses such as confusion matrices and *confusion histograms* that tabulate errors in symbol and/or relationship recognition for target subgraphs [2].

Formula Representation: Symbol Layout Trees (SLTs). Formulae are represented by labeled adjacency graphs over strokes (*label graphs*). This allows errors to be unambiguously identified, even for conflicting segmentations. For CROHME, label graphs represent formulae as *Symbol Layout Trees* (SLTs) [1], with symbols defined by labeled stroke groups, and spatial relationships by labeled directed edges

[3]http://ntcir-math.nii.ac.jp/data/

[4]https://www.cs.rit.edu/~dprl/Software.html

between symbols. SLTs provide a simple representation of formula appearance.

In a label graph adjacency matrix, labels on the diagonal define the symbol class associated with each stroke, while off-diagonal elements represent stroke groupings and spatial relationships. Symbols are represented by bidirectional edges between all stroke pairs in the symbol, with each edge labeled by the symbol class. For spatial relationships, a labeled edge is defined from every stroke of the parent symbol to every stroke of the child symbol. For matrices, we generalize label graphs to include sets of labels on nodes and edges [2]. This allows a stroke to belong to more than one type of object and/or level of structure. For example, a single stroke '2' may be a symbol, in a cell, belonging to a row, a column, and a matrix. We simply assign all of these types to the stroke, and identify the segmentation of objects using cliques sharing a label (e.g., all strokes labeled as a 'Row' sharing bi-directional 'Row' edges are an object).

Formula Recognition Metrics. For Tasks 1, 3 and 4, metrics provided by LgEval include conventional formula and symbol recognition rates, along with finer-grained *recall* and *precision* metrics for the segmentation and classification of symbols *and* relationships. Recall is the same as the recognition rate, while precision measures the accuracy of returned results, for example the percentage of returned symbol segments that are correct. A relationship is correctly detected (segmented) when two symbols with a parent-child relationship in the Symbol Layout Tree are found, ignoring the symbol and relationship labels. Hamming distances over stroke labels in label graphs are also provided for fine-grained analysis [2].

Isolated Symbol Results and Metrics. For Tasks 2a and 2b, each inkML test file contains a single symbol with an associated identifier ('UI' tag). Participants submitted a CSV file containing one line for each test file. Each line provides the symbol identifier followed by a ranked list of the Top-10 classification candidates. A CROHMELib tool (`evalSymbolIsole.py`) was used to compute symbol recognition rates along with the average rank of the correct symbol class (*TMP:* True Mean Position), where any target class not appearing in the Top-10 is treated as rank 11. For Task 2b where some inputs were 'junk' symbols (i.e. incorrectly segmented symbols), the tool also provides the symbol true positive rate (*TAR:* True Acceptance Rate) and false positive rate (*FAR:* False Acceptance Rate).

Ranking. The expression recognition rate was used for Tasks 1, 3 and 4, while the symbol recognition rate was used for Tasks 2a and 2b.

C. Submission

Previously CROHME systems were submitted directly to organizers, requiring significant efforts to configure and run. For CROHME 2016, recognition results were provided by participants through a web-based evaluation system[5] implemented in Django.[6] Multiple submissions were permitted, with the highest obtained recognition rates used to rank systems. After a participant made a submission, the recognition rate obtained was visible to the participant. A publicly visible rankings page was automatically updated with the names of participants hidden during the competition. After logging in, a participant could see their place in the rankings, but not the names of other participants.

Unfortunately a bug required organizers to upload results for participants in many cases. Despite its shortcomings, the submission interface greatly reduced the effort to compile results, and motivated participants to correct and improve their results. We plan to repair the submission system and make it available after the competition, as done for other document analysis competitions (e.g., ICDAR 2015 Robust Reading[7] [6]).

III. Participating System Descriptions

The system descriptions in this Section were provided by the participants of CROHME 2016. We list system descriptions alphabetically by group name. Eleven participants registered for CROHME 2016, but several withdrew, taking Task-1 from 8 to 5 participants; Task-2 from 11 to 3; Task-3 from 6 to 4; and Task-4 from 3 to 2. We are uncertain what the cause for this was, whether it was technical difficulties, concerns about performance, or other reasons.

MyScript. (Tasks 1-4) The MyScript Math recognizer system is built on the principle that segmentation, recognition and interpretation have to be handled concurrently and at the same level in order to produce the best candidates. The recognition engine analyzes the spatial relationships between all parts of the equation, in conformity with the rules laid down in its grammar, to determine the segmentation of all of 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 symbol recognizer extracts both dynamic and static features from the online signal. Dynamic information is extracted from the trajectory of the pen, including direction and curvature. Static features are computed from a bitmap representation of the ink and are typically based on projections and histograms. The features are then processed by a combination of Deep MLP and Recurrent Neural Networks.

The recognition engine includes a statistical language model that evaluates the contextual probabilities between symbols in the equation. The recognizer is trained on about

[5] http://crohme2016eval.cs.rit.edu
[6] https://www.djangoproject.com/
[7] http://rrc.cvc.uab.es

30,000 additional handwritten samples collected from writers in different countries, using discriminative training and automatic learning of all meta-parameters.

Rochester Institute of Technology (RIT). (Task 2) Our approach for on-line recognition of handwritten math symbols uses adaptations of off-line features [5]. Our feature set is based on shape descriptors, in order to obtain greater tolerance to variations in writing direction and stroke order. Currently these features are: Crossings, 2D histograms of stroke segment lengths, 2D histograms of stroke segment orientations, and 2D histograms of visual words. Currently, we use 2D histograms of stroke segment lengths computed locally and k-means clustering to create our visual word dictionary.

Tokyo University of Agriculture and Technology. (Tasks 1, 2 & 3) The systems are developed by Nakagawa Laboratory. For classifying isolated math symbols, an offline classifier (CNN) and several online classifiers (LSTMs) are combined [7]. For parsing formula structure, the CYK algorithm is employed. Stroke order is employed to reduce the complexity of the parsing algorithm and additional grammar rules are employed to deal with the multiple stroke order variations. Details of the system are presented in Le et al. [8]. For recognition of mathematical expressions, our previous system from CROHME 2014 was improved by integrating the new symbol recognizer and retraining weight parameters.

University of Nantes. (Task 1) Our system recognizes handwritten mathematical expressions by merging multiple 1D sequences of labels produced by a sequence labeler. The proposed solution aims at rebuilding a 2D expression from several 1D-labeled paths. An online math expression is a sequence of strokes which is later used to build a graph considering both temporal and spatial information for strokes. In this graph, nodes correspond to strokes and edges denote the relationship between a pair of strokes. Next, we select 1D paths from the graph with the expectation that these paths can catch all the strokes and the relationships between pairs of strokes. As an advanced and strong sequence classifier, BLSTM networks with a local CTC output layer are adopted to label the selected 1D paths. We assign different weights to these 1D-labeled paths, and then merge them to rebuild a label graph. After that, an additional post-process is performed to complete edges automatically. No language models have been used.

University of São Paulo. (Tasks 1 & 3) In this approach, the recognition problem is modeled as a graph parsing problem. The method is divided into two stages: (1) hypotheses graph generation and (2) graph parsing. In the first stage, multiple symbols (labeled stroke groups) and relations between symbols are identified and stored in a structure called a *hypotheses graph*. To this end, symbol and relation classifiers are trained using both valid and invalid symbols and relations extracted from complete mathematical expressions, as described in [9]. A graph grammar is used to define a language of graphs representing mathematical expressions. In the second stage, a top-down parsing algorithm uses a graph grammar to parse the hypotheses graph and identify the subgraph containing the best interpretation, taking symbol and spatial relation scores into account [10].

WIRIS math. (Tasks 1-4) The handwritten math expression recognition engine is part of WIRIS Editor,[8] a solution for editing math formulas. It is based on an integrated approach such that symbol segmentation, symbol classification and the structure of the math expression are globally determined. The recognition process is guided by a probabilistic grammar that accounts for the structural probability between symbols and expressions. The statistical language model has been estimated using both the expressions of the training dataset and formulae from English Wikipedia provided by the organizers of the competition.

The system is completed with specialized modules for specific tasks: spatial relations classification, symbol segmentation and symbol classification. Mathematical symbol classification is performed using neural networks and a combination of several sets of online and offline features, like the input sequence of points, histograms of oriented gradients or the offline representation of the symbol hypothesis. The remaining models are also statistical classifiers such that all probabilistic sources of information are estimated from training data.

Finally, the global recognition engine parameters have been tuned for each task of the competition (expressions, matrices and structure). Matrix recognition requires additional considerations, like matching dimensions of rows and columns and handling spatial information for proper segmentation of cells.

IV. Results

Symbol Recognition (Tasks 2a & 2b). Table II shows that recognizing isolated handwritten mathematical symbols is still non-trivial. Indeed, some classes are difficult or impossible to discriminate. For example, without context it is difficult to separate 'x', 'X', and '$\times$' (times); 'o', 'O', and '0'; and 'p' and 'P.' Commas and dots require relative size information to avoid ambiguity - otherwise commas may be frequently confused with '1' or '/' for example. These ambiguities place an upper bound on accuracy [5], and recognition rates may are higher if highly ambiguous classes are merged. The tools provided for the competition allow these rates to be computed (omitted for space).

The average rank for the correct symbol class (TMP) is high for all three systems, with the location of the correct class slightly below the 1st (top-1) position. When an isolated symbol classifier is used in a system recognizing full expressions, it will need to discriminate between correctly and incorrectly segmented symbols. As presented in Table II

[8]http://www.wiris.com/editor/demo

Task 2b, a substantial drop-off in classification accuracy occurs when invalid symbols are added along with a reject (*junk*) class. RIT obtains a better true acceptance rate (TAR) than MyScript, but with a higher false acceptance rate (FAR), indicating that the RIT system is biased toward accepting more symbols as valid.

Table II
RESULTS FOR SYMBOL RECOGNITION TASKS (TASKS 2A & 2B). *Top-1:* RECOGNITION RATE, *TMP:* TRUE MEAN POSITION (AVERAGE RANK OF THE CORRECT CLASS), *TAR:* TRUE ACCEPTANCE RATE FOR VALID SYMBOLS, *FAR:* FALSE ACCEPTANCE RATE FOR JUNK SYMBOLS.

	TASK 2A (VALID) (101 classes)		TASK 2B (VALID + JUNK) (102 classes)		(Symb. vs. Junk)	
	Top-1	TMP	Top-1	TMP	TAR	FAR
MyScript	**92.81**	**1.13**	**86.77**	**1.19**	89.82	**11.16**
Tokyo	92.27	1.15	-	-	-	-
RIT	88.85	1.25	83.34	1.31	**95.86**	19.71

Isolated Expression Recognition (Tasks 1 & 3). Table III presents expression level results for Task 1 and Task 3, where formulae are parsed from handwritten strokes (Task 1) or provided symbols (Task 3). We provide structure recognition rates along with fully correct recognition rates. Correct structure recognition requires only symbol segmentation and the detection of relationships between symbols to be valid; to be fully correct an expression must have correct structure and correct labels for symbols and relationships. Also shown in Table III are the expression recognition rates obtained if one or two label errors are permitted in the label graph adjacency matrix for an expression.

Fully correct recognition rates are much higher in Task 3, as the symbols are provided, and spatial relationships are constrained by symbol types (e.g., '+' cannot have a superscript). We see that MyScript finds the correct unlabeled graph for 88% of the test expressions in Task 1, and 90.7% of test expressions in Task 3. It is interesting that the difference between these structure recognition rates is small.

Symbol level metrics for Task 1 are shown in Table IV, including recall and precision for segmentation, and for both correct segmentation and classification. The MyScript and WIRIS systems perform best, but even the weakest-performing system at the expression level have a competitive symbol recall rate. Univ. Nantes low expression rates may be due to not using an expression grammar for parsing. Most systems have higher symbol precision than recall, suggesting that they often under-segment symbols when errors are made.

Recall and precision metrics are also provided for detecting (segmenting) and classifying spatial relationships in Table IV. For Tasks 1 and 3, there are six spatial relationships: *Above, Below, Subsc, Super, Right,* and *Inside*. The spatial relationship performance metrics for Task 1 are lower and vary more than those for symbols. This is because segmenting and classifying relationships is strictly harder,

Table III
RESULTS FOR ISOLATED FORMULA PARSING. IN ADDITION TO RECOGNITION RATES FOR SYMBOL LAYOUT TREES (SLTS) WITH CORRECT STRUCTURE, AND CORRECT STRUCTURE AND LABELING, RATES FOR FORMULAE WITH $n \leq 2$ INCORRECT LABELS FOR STROKES AND DIRECTED STROKE EDGES ARE SHOWN.

TASK 1: ISOLATED FORMULAE (TEST 2016)

	Structure	Structure + Labels		
	Rec.Rate	Rec.Rate	$\leq$ 1 err.	$\leq$ 2 err.
MyScript	**88.14**	**67.65**	**75.59**	**79.86**
Wiris	74.28	49.61	60.42	64.69
Tokyo	61.55	43.94	50.91	53.70
São Paolo	57.02	33.39	43.50	49.17
Nantes	21.45	13.34	21.02	28.33

TASK 3: ISOLATED W. PROVIDED SYMBOLS (TEST 2014)

	Structure	Structure + Labels		
	Rec.Rate	Rec.Rate	$\leq$ 1 err.	$\leq$ 2 err.
MyScript	**90.67**	**84.38**	**85.90**	**87.62**
Wiris	86.61	78.80	80.42	82.75
São Paulo	69.27	64.81	67.34	70.69
Tokyo	70.99	61.46	63.89	66.84

as both parent and child symbols need to be segmented correctly for a relationship to be correctly detected. As one would expect, for the Top-3 systems the most common spatial relationship confusions are between right-adjacency and subscript, followed by right-adjacency and superscript.

The most frequently mis-recognized symbols for each system are presented in Table VI. Table VII provides the most frequent symbol pair relationships (bigrams) that are confused by each system. In both Tables, we have provided the frequencies for the Top-5 most frequent test set patterns. For all participants, the most common symbol recognition errors correspond to the most frequent symbols ('2' and'1') and ambiguous symbols (x, $\times$). As in the past [2], recognizing a '1' above a fraction line as a single '1' is a common error (see Table VII).

Matrix Recognition (Task 4). The results for the second experimental matrices recognition task are presented in Table V. Systems perform better than in the last competition, but it is not clear whether this is due to algorithms or the new data set. As in the last CROHME, the alignment of elements in columns is harder than for rows. Symbol recall rates for both systems are lower than their results for Task 1 (see Table IV), because matrix layout makes symbol detection more difficult.

Table V
TASK 4: MATRIX RECOGNITION RESULTS.

	MyScript	Wiris
Expression Rate	**68.40**	56.40
Symbol Recall	**94.86**	87.03
Matrix Recall	**97.52**	85.67
Row Recall	**95.61**	87.16
Column Recall	**90.71**	82.22
Cell Recall	**87.49**	84.68

Table IV
TASK 1: SYMBOL-LEVEL RESULTS. THERE ARE 101 SYMBOL CLASSES, AND SIX SPATIAL RELATIONSHIPS. *Seg:* SEGMENTATION, *Seg+Class:* SEGMENTATION & CLASSIFICATION, *Rec:* RECALL, *Prec:* PRECISION.

	SYMBOLS				SPATIAL RELATIONSHIPS			
	Seg (%)		Seg+Class (%)		Seg (%)		Seg+Class (%)	
	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.
MyScript	**98.89**	**98.95**	**95.47**	**95.53**	**96.81**	**96.82**	**95.11**	**95.11**
Wiris	96.49	97.09	90.75	91.31	91.98	92.62	90.17	90.79
Tokyo	91.62	93.25	86.05	87.58	84.41	85.98	82.11	83.64
São Paulo	92.91	95.01	86.31	88.26	83.51	86.26	81.48	84.16
Nantes	94.45	89.29	87.19	82.42	76.34	71.67	73.20	68.72

Table VI
TASK 1: FREQUENT SYMBOL RECOGNITION ERRORS AND TOP-5 TEST SET SYMBOL FREQUENCIES.

	1st	#	2nd	#	3rd	#
MyScript	c	(23)	1	(22)	q	(18)
Wiris	1	(65)	z	(55)	2	(44)
Tokyo	x	(99)	1	(86)	2	(77)
São Paulo	1	(70)	$\times$	(77)	2	(72)
Nantes	1	(115)	2	(42)	$\times$	(39)
Freq. in	$-$	(1120)	2	(898)	1	(891)
Test set	$+$	(763)	x	(695)		

Table VII
TASK 1: MOST FREQUENT SYMBOL BIGRAM ERRORS AND TOP-5 MOST FREQUENT SYMBOL BIGRAMS IN THE TEST SET.

	1st	#	2nd	#	3rd	#
MyScript	$\sum_a$	(11)	$\lim_n$	(8)	$\sum_i$	(8)
Wiris	$x)$	(16)	$z)$	(14)	$(z$	(14)
Tokyo	dx	(29)	$x)$	(22)	$\frac{1}{}$	(22)
São Paulo	$--$	(36)	$\sqrt{+}$	(26)	$\frac{1}{}$	(24)
Nantes	$\frac{1}{}$	(88)	$x+$	(46)	$--$	(44)
Freq. in	$\frac{1}{}$	(198)	$\frac{}{2}$	(156)	$+1$	(126)
Test set	-1	(124)	$--$	(124)		

V. CONCLUSION

The recognition of handwritten math expressions is still a challenge after six years of competitions. High recognition rates for individual tasks were obtained, but correctly parsing a handwritten expression requires many individual segmentation and classification decisions to be made correctly in tandem. The system with the best performance for the main task is again MyScript corporation, which made use of a large private dataset to train their handwritten symbol parser. The best results obtained using only provided data were from WIRIS corporation (Tasks 1, 3 and 4), the Tokyo University of Agriculture and Technology (Task 2a), and RIT (Task 2b).

For online handwritten math recognition, a key direction for future research is improving structure recognition: even from provided symbols in Task 3, for the best submission almost 10% of recognized formulas have incorrect structure. Improving detection of invalid (junk) symbols is an important related direction. Improvements in symbol and spatial relationship classification can also be made, including better discrimination between right-adjacent and sub/super-scripted relationships.

REFERENCES

[1] R. Zanibbi and D. Blostein, "Recognition and retrieval of mathematical expressions," *Int'l. J. Document Analysis and Recognition*, vol. 15, no. 4, pp. 331–357, 2012.

[2] H. Mouchère, R. Zanibbi, U. Garain, and C. Viard-Gaudin, "Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011–2014," *Int'l. J. Document Analysis and Recognition*, vol. 19, no. 2, pp. 173–189, 2016.

[3] J. Gehrke, P. Ginsparg, and J. Kleinberg, "Overview of the 2003 KDD cup," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 2, pp. 149–151, 2003.

[4] R. Zanibbi, A. Aizawa, M. Kohlhase, I. Ounis, G. Topić, and K. Davila, "NTCIR-12 MathIR task overview," in *Proc. NTCIR-12: Evaluation of Information Access Technologies*, Tokyo, 2016, 10 pp.

[5] K. Davila, S. Ludi, and R. Zanibbi, "Using off-line features and synthetic data for on-line handwritten math symbol recognition," in *Proc. ICFHR*, Crete, Greece, 2014, pp. 323–328.

[6] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, and S. Lu, "ICDAR 2015 competition on robust reading," in *Int'l. Conf. Document Analysis and Recognition*, Nancy, France, 2015, pp. 1156–1160.

[7] H. D. Nguyen, A. D. Le, and M. Nakagawa, "Deep Neural Network for recognizing online handwritten mathematical symbols," in *Proc. 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Malaysia, 2015.

[8] A. D. Le, T. V. Phan, , and M. Nakagawa, "A system for recognizing online handwritten mathematical expressions and improvement of structure analysis," in *Proc. IAPR Workshop on Document Analysis Systems*, France, 2014.

[9] F. Julca-Aguilar, C. Viard-Gaudin, H. Mouchère, S. Medjkoune, and N. Hirata, "Mathematical symbol hypothesis recognition with rejection option," in *Proc. ICFHR*, Crete, Greece, 2014, pp. 500–505.

[10] F. Julca-Aguilar, H. Mouchère, C. Viard-Gaudin, and N. S. T. Hirata, "Top-down online handwritten mathematical expression parsing with graph grammar," in *Proc. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP)*, 2015, pp. 444–451.