ICFHR 2020 Competition on Offline Recognition and Spotting of Handwritten Mathematical Expressions - OffRaSHME

Da-Han Wang
Fujian Key Laboratory of Pattern Recognition and Image Understanding
School of Computer and Information Engineering, Xiamen University of Technology, Xiamen, China
wangdh@xmut.edu.cn

Fei Yin, Jin-Wen Wu
National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences, Beijing, China
{fyin, jinwen.wu}@nlpr.ia.ac.cn

Yu-Pei Yan, Zhi-Cai Huang, Gui-Yun Chen, Yao Wang
Fujian Key Laboratory of Pattern Recognition and Image Understanding
School of Computer and Information Engineering, Xiamen University of Technology, Xiamen, China

Cheng-Lin Liu
National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences, Beijing, China
liucl@nlpr.ia.ac.cn

Abstract—This paper presents the competition on Offline Recognition and Spotting of Handwritten Mathematical Expressions (OffRaSHME) held at the 17th International Conference on Frontiers in Handwriting Recognition (ICFHR 2020). Handwritten Mathematical Expression Recognition (HMER) has wide potential applications and is capturing increasing attention in recent years. Previous HMER competitions mainly focused on online datasets or offline datasets that are converted from online data. In this competition, we have collected a dataset of offline handwritten mathematical expressions by scanning papers that contain expressions. Moreover, we labeled the offline dataset at symbol level, i.e., the bounding boxes of each symbol are also provided, to facilitate the research of HMER. At last, 19,749 offline handwritten mathematical expressions are collected for training, and 2,000 ones are provided for evaluating the participating systems. In the competition, 7 teams submitted 8 systems for the task of offline HMER, among which 5 systems only use the provided datasets without any extra data while 3 systems use extra data. The winner team achieved a recognition accuracy of 79.85% (without extra data) and 81.85% (with extra data) on the offline formula recognition task.

Keywords—offline handwritten mathematical expression, offline formula recognition, symbol level annotation, competition

I. Introduction

Handwritten Mathematical Expression Recognition (HMER) has wide potential applications in many areas such as education, office automation and conference systems, and hence is steadily capturing increasing attentions from the community in recent years [1-3]. HMER is also an important sub-problem in the research of document analysis and recognition. The challenges of HMER mainly lies in the complicated two-dimensional (2D) structures and spatial relations contained in Handwritten Mathematical Expressions (HMEs). There are still a lot of technical problems in HMER that requires further research.

The six previous Competitions on Recognition of HMEs (CROHME) had been organized at ICDAR 2011, 2013 & 2019, and ICFHR 2012, 2014 & 2016 to push forward the research of HMER, and achieved significant success [4-9]. However, it is noticed that, previous CROHME competitions except the one held at ICDAR 2019 only released online data of HMEs (i.e., stroke data with traces) and focus on the online HMER problem. In ICDAR 2019 CROHME, the organizers expanded the set of inputs to include both online and offline handwritten formulas. However, the offline HMEs are generated using online traces and may be different from real HMEs written on papers/documents. In the real scenarios, offline HMER may face the problems of blurring, noises, lacking of strokes, complex background, etc., and hence is much more challenging and requires further investigation.

On the other hand, in the application of computer-assisted scoring that has potential significance in the education area, the technique of handwritten expression spotting can be used to perform automated scoring by retrieving the input formula (standard answer) from the test examination papers. The base problem of this task is the handwritten formula matching problem, which essentially is a structural pattern matching problem. In addition to computer-assisted scoring, the retrieval of input formulas on the web or database is also a problem worth studying. These kinds of tasks have not been conducted in previous CROHME competitions.

In our original plan, we planned to organize two tasks. Task 1 is the offline handwritten formula recognition, which is the base task in the OffRaSHME, while Task 2 is handwritten formula spotting, which aims to explore the prospects of formula recognition in computer-assisted scoring and formula retrieval. However, due to the effects of COVID-19, we are unable to collect the datasets for Task 2, and hence had to cancel Task 2 finally. In the following, we mainly introduce the details of Task 1.

For the competition, we have collected a dataset of offline HMEs by scanning papers that contain expressions. This kind of offline HMEs is expected to be much more challenging and open a large room for research and improvement. We also asked people to write expressions that have the same classes

978-1-7281-9966-5/20/$31.00 ©2020 IEEE
DOI 10.1109/ICFHR2020.2020.00047

Authorized licensed use limited to: UNIVERSITY OF BATH. Downloaded on May 14,2021 at 08:16:37 UTC from IEEE Xplore. Restrictions apply.

with the HMEs used in ICDAR 2019 CROHME competition [9] on papers, which can support better comparison research.

The newly collected dataset is the first dataset of offline HMEs collected by writing on and scanning from papers, and can be an extension to previous datasets. Moreover, we labeled the offline dataset at symbol level. Unlike the offline HMEs used in ICDAR 2019 CROHME competition, which are generated from online ones and can be easily annotated at symbol level, directly annotating offline HMEs without stroke information will be non-trivial.

Finally, 7 teams submitted 8 systems for the task, among which 5 systems only use the provided datasets without any extra data while 3 systems use extra data. The recognition performance was evaluated on closed datasets using the evaluation protocol that is consistent with previous CROHME competitions. The winner team achieved an recognition accuracy of 79.85% (without extra data) and 81.85% (with extra data) on the offline formula recognition task, demonstrating the challenge of the task.

In the following, we first introduce the datasets and the evaluation protocal in Section II, and then presents the participating systems in Section III. The recognition results are described in Section IV, and concluding remarks are addressed in Section V.

II. DATASETS AND EVALUATION PROTOCAL

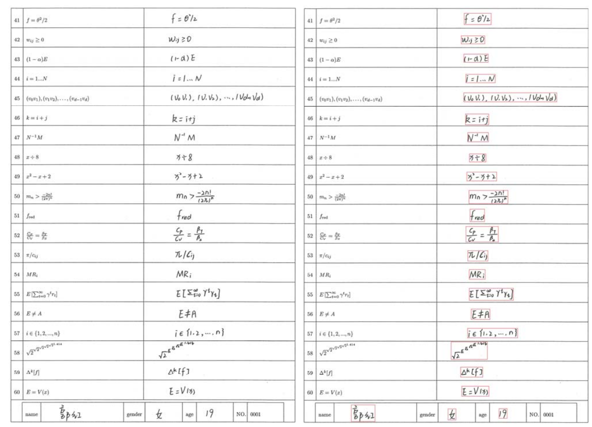
We have collected two subsets for training. The first subset contains the HMEs that have the same set of expressions with the HMEs used in ICDAR 2019 CROHME competition. For this subset, we try our best to reserve as much formulas as possible. Formulas with conflicting nodes in LG format label in the original dataset, wrongly written formulas and formulas that exceed the prescribed writing area are not used. Finally, we obtained 9,749 available offline handwritten expressions in this subset. The offline HMEs will be annotated at symbol level: the latex transcripts, the corresponding label graph and bounding boxes of symbols will be provided. In collection of this dataset, we randomly sorted the mathematical expressions in the ICDAR 2019 CROHME competition, and asked about 70 people to write these formulas, each people writing 150 formulas.

The second subset were collected using the Wikipedia formula corpus provided by CROHME. We have asked 400 people to write mathematical expressions on papers, with each people writing 150 formulas. We then sorted the 60,000 expressions randomly and selected 10,000 as the training subset. For this subset, the latex transcript and the label graph are provided but the bounding boxes of symbols are not provided (the bounding boxes of symbols will be provided after the competition is finished).

For evaluating the participating systems, another randomly selected 2000 expressions are used as the test set. In the following, the details of data acquisition and data annotation will be described.

A. Data Acquisition

We first generated 60,000 mathematical expressions in the latex format, and then transformed them into PDF documents as in Figure 1. Then the writers were asked to write the corresponding expressions on the papers, which were scanned into colored offline images using a commonly used scanner (Ricoh MP 9003SP) with 600 dpi resolution, as shown in Figure 1(a). To extract the offline expressions, we need to get the bounding boxes of the written expressions. Toward this purpose, we adopt the connected components analysis (CCA) method followed by a component grouping step to capture the bounding boxes of the handwritten expressions. In some cases, there exists errors such as missing strokes, or larger bounding boxes caused by small noisy dots, etc. Hence, we develop a tool to edit the bounding boxes. At last, according to the bounding boxes of the handwritten expressions, we can extract the expressions directly, as shown in Figure 2.



(a) (b)

Figure 1. Examples of handwritten expression collection and extraction.

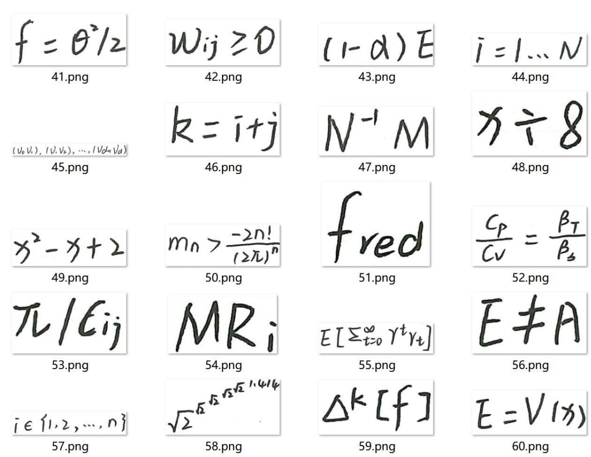


Figure 2. The collected offline handwritten mathematical expressions.

B. Data Annotation

To annotate the offline handwritten formulas in symbol level, we modify the open-sourced LabelMe [10] to finish the work. Figure 3 shows the main steps. The formula image and the symbol Label Graph (LG) file are loaded simultaneously (Figure 3 (a)). Then the bounding boxes of each symbol can be annotated by the users (Figure 3(b)), while the corresponding symbol class (provided by the LG file) is

Authorized licensed use limited to: UNIVERSITY OF BATH. Downloaded on May 14,2021 at 08:16:37 UTC from IEEE Xplore. Restrictions apply.

automatically related to the annotated bounding box (Figure 3(c)). The correspondence between the bounding box and the symbol class can also be edited conveniently. When all the bounding boxes and the corresponding symbol classes are labeled, which are saved in one TXT file, we obtain the symbol level annotation of the sample.

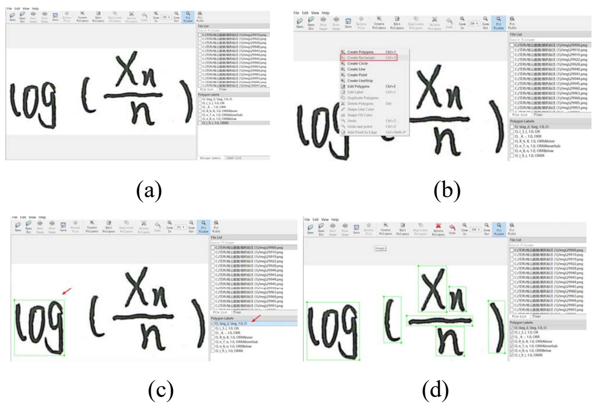


Figure 3. The procedure of symbol level annotation.

C. Evaluation Protocol

Consistent with previous CROHME competitions, formula recognition results are evaluated in the format of symbol label graph. Symbol label graph metrics include both the math symbols and the structural relationships. The evaluation on validation dataset are available to participants offline with the CROHMELib and LgEval libraries [3]. Participating systems that recognize formula images into LaTeX codes or other format representations are required to convert the results into symbol label graphs. LaTeX format results can be converted into symbol label graphs with the LgEval libraries.

III. PARTICAIPATING SYSTEMS

In the following, we describe the submitted methods provided by the participating teams. For this competition, each team has 3 opportunities to submit their recognition results, and only the results with the highest performance are used for final ranking. Hence, the team names are used as the system names.

*A. **USTC-iFLYTEK***

Authors: Changjie Wu, Chen Yang, Qing Wang, Jianshu Zhang, Jun Du from National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP Lab), University of Science and Technology of China, and Mingjun Chen, Huihui Zhu, Jiajia Wu, Jinshui Hu from iFLYTEK Research

The system is based on encoder-decoder models with attention mechanism. An enhanced densely connected convolutional network is used as the encoder. As for the decoder, a string decoder [11] and a tree decoder [12] are combined. The string decoder generates the math LaTeX string and the tree decoder generates the sub-tree structures of math formula. The bounding box provided in the official dataset are utilized to decompose and reconstruct math formulas as data augmentation. During the first submission, all the models are trained using only official training dataset. During the second and third submissions, 200,000 private math formula datasets are used to pre-train the encoder-decoder models. A LaTeX text dataset provided in the NTCIR-12 MathIR dataset is also used to train a RNN-based language model.

*B. **HCMUS-I86***

Authors: Vinh-Loi Ly, from University of Science, Ho Chi Minh City, Vietnam, and The-Anh Vu-Le, from Vietnam National University, Ho Chi Minh City, Vietna.

In the preprocessing stage, the image is inverted by changing from the original light background, dark foreground to dark background, light foreground; Then the image is resized to a fixed height H (for the experiments, H = 64), keeping the aspect ratio; The image is then padded to a fix width W (for the experiments W = 14 * 64 = 896, which fits all training samples. Finally, the image is normalized according to ImageNet statistics.

In the recognition stage, both the encoder and the decoder of the Transformer architecture are adopted. More precisely, visual feature maps are extracted from images using a backbone network while one-hot vectors are obtained from one-hot encodings of the latex representation of the text. Then, the feature maps are passed to the Transformer Encoder and combined with one-hot vectors in the Transformer Decoder to output a sequence with the same length.

Several CNN architectures are varied for the backbone and it is found that using DenseNet-161 with its last block dropped would reduce overfitting and give feature maps with higher resolution.The Cross Entropy loss function and the Adam optimizer are used to train the model. Also, starting from 0.001, the learning rate is reduced by tenth every 5 epochs that the Expression Recognition Rate does not increase.

In the post-processing stage, since the network is not enforced with any grammar, there are errors in the closing of opened brackets. A simple heuristic is implemented to fix these errors. More specifically, for every opening bracket that does not have the corresponding closing brackets, the respective closing brackets are appended to the end of the sequence chronologically. The merged characters are also split in the preprocessing stage to the original format.

*C. **SCUT-DLVCLab***

Authors: Zhe Li, Tianwei Wang, Songxuan Lai, Lianwen Jin(instructor), from School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China

In this system, offline handwritten mathematical expression recognition is regarded as an image-to-sequence problem and an encoder-decoder framework is used to recognize expressions, including symbol prediction and structure phasing. ResNet is used as the encoder to extract the features. The attention-based decoder relied on RNN and iteratively generated the target LaTeX sequences from features. Finally, the LaTeX sequences are converted into the symbol label graphs. Two important techniques are applied in the training stage, including scale augmentation and drop attention, which are in detail described in the paper [13]. This methods followed the rules of the competition and didn't use any extra datasets or corpus.

Authorized licensed use limited to: UNIVERSITY OF BATH. Downloaded on May 14,2021 at 08:16:37 UTC from IEEE Xplore. Restrictions apply.

D. ***TUAT***

Authors: Thanh-Nghia Truong, Cuong Tuan Nguyen, and Masaki Nakagawa, from the Nakagawa laboratory, Tokyo University of Agriculture and Technology.

This system uses an end-to-end deep neural network that is trained using weakly supervised learning. The network has three parts: an encoder using a Convolutional Neural Network to encode CNN features, a decoder using gated recurrent units with attention, and a symbol classifier to improve the localization and classification of the CNN features. Besides, the model ensemble method is used to increase the results. More details of this method can be found in the paper [14]. Moreover, a math language model is used to improve the performance of our models.

For the data, the end-to-end network and the math language model are trained using the given dataset. Since the organizing committee did not explicitly divide the dataset into training and test subsets, the TUAT team divided the dataset into three parts to train, validate, and test the models according to their own needs.

E. ***SYSU***

Authors： Chungkwong Chan， from School of Mathematics, Sun Yat-Sen University

The system reduces offline recognition to online recognition. At first, the stroke extraction algorithm described in [15] is applied to recover strokes from an input image. It works by manipulating a graph based representation of skeleton with heuristic rules. After that, the end-to-end trainable pure online recognizer introduced in [1] is used to recognize the strokes. It is a recurrent neural network which adopts the standard attention based encoder-decoder framework. Pairs of extracted trajectory and normalized LaTeX code were used to train the network, positions of symbols were not used. Finally, recognition results are converted from LaTeX to symLG. Only the official dataset was used, and positions of symbols were not used.

F. ***MLE***

Authors: the MLE team, from Multi-country team of ML enthusiasts

This system is based on the multi-scale-attention model (image to latex task) described in [17]. For model training, the official dataset and CROHME 2011-2019 datasets are used. For image pre-processing, all images are converted to binary images by pixel threshold. After that, the images are filtered with kernel size (5, 5) for removing small linked-components, and expanded to the same size of 512×1536 by adding zero-padding. For the training neural network model, Adam and SGD optimizers are used with learning rate 0.0001, batch size 4 for each iteration, and l2 regularization with the weighting parameter 0.0001. We performed data augmentation on-the-fly, which includes random rotation [-5 degree, +5 degree], shear [-5 degree, +5 degree] and zoom [0.8, 1.2]. Training took approximately 14 hours on a NVidia GeForce RTX 2080.

G. ***IVTOV***:

Authors: the IVTOV team, from SRK (Ukraine)

The recognition engine is built on the principle of image vectorization and further online recognition. In the first step, the image is binarized by thresholding and the binarized image is filtered by removing small-area linked-components from both the foreground and the background. After that, Lee thinning is used to skeletonize the foreground and extract segments from the skeleton. At the final stage of vectorization, graph theory and heuristics is used for constructing traces that made up an online document for the second stage. As a recognition engine, the online recognition system that participated in CROHME 2019 is used. Online recognition is based on BLSTM-CTC architecture in combination with PCFG and statistical language model, which is described in [16]. This recognizer was trained specifically on the official competition train set, in-house online dataset, and dataset obtained by double converting the samples from online to offline and vice versa using the vectorization algorithm mentioned above.

IV. RECOGNITION RESULTS

In this section, we briefly analyze the recognition results of different systems. As mentioned above, each team has 3 opportunities to submit their recognition results, and only the highest performance will be used for ranking. In this competition, participants using additional training data or formula corpus are not ranked. We encourage contestants to innovate some new algorithms. Submitted systems trained with extra datasets are listed in another table only for reference, and will not take part in the final ranking.

Table 1. Recognition results with standard datasets

Rank	**Team**	**Correct**	**≤1 s.err**	**≤2 s.err**	**Structure**
1	USTC-iFLYTEK	79.85	89.85	92.00	92.55
2	SCUT-DLVCLab	72.90	86.05	88.80	89.45
3	TUAT	71.75	82.70	85.80	86.60
4	HCMUS-I86	66.95	79.65	83.60	84.00
5	SYSU	61.35	77.30	81.55	82.90

Table 2. Recognition results with extra datasets

Team	**Correct**	**≤1 s.err**	**≤2 s.err**	**Structure**
USTC-iFLYTEK	81.85	90.50	92.30	93.05
IVTOV	61.90	73.00	75.95	77.10
MLE	46.85	61.15	65.80	67.45

Table 1 shows the final recognition results of the submitted systems trained with standard datasets, while Table 2 shows the results of the systems trained with extra datasets. The recognition rates when one to two errors in the symbol LG are permitted given in the table shows the room for improvement of each system. In addition, the table also lists the structural recognition rate of different systems when ignoring the symbol category.

From Table 1, we can see that the USTC-iFLYTEK achieved an accuracy of 79.85% with standard datasets. In the ICDAR 2019 CROHME competition, the USTC-iFLYTEK achieved the highest performance with 80.73% on online dataset and 77.15% on converted offline dataset. The highest performance of this competition and the ICDAR 2019 CROHME competition does not vary much (comparable). It should be noticed that, in this competition, the symbol level annotations, which are expected to improve the performance, are provided and the USTC-iFLYTEK utilized the annotations for training. Besides, it is interesting to observe a part of teams

Authorized licensed use limited to: UNIVERSITY OF BATH. Downloaded on May 14,2021 at 08:16:37 UTC from IEEE Xplore. Restrictions apply.

that only use official data outperforms some teams that use extra data. The highest accuracy using official data is only 2.00% lower than the highest accuracy using extra data. Data augmentation of formula corpus and images is a very effective way to improve system performance. Nonetheless, system design could cause a large difference in accuracy (from 61.35% to 79.85% in Table 1 and 46.85% to 81.85% in Table 2).

Table 3. Recognition errors with official dataset

Teams	1st		2nd		3rd		1st		2nd		3rd	
	S	**#E**	**S**	**#E**	**S**	**#E**	**R**	**#E**	**R**	**#E**	**R**	**#E**
USTC-iFLYTEK	'2'	53	')'	49	'p'	44	'Right'	211	'Sub'	73	'Inside'	37
SCUT-DLVCLab	'1'	94	')'	93	'x'	73	'Right'	345	'Sub'	147	'Sup'	48
TUAT	')'	103	'1'	93	'2'	84	'Right'	523	'Sub'	108	'Sup'	52
HCMUS-I86	'1'	136	')'	126	'2'	118	'Right'	612	'Sub'	186	'Sup'	95
SYSU	')'	148	'2'	119	'1'	115	'Right'	494	'Sub'	161	'Sup'	137

S:symbol **R**:relation **#E**: the number of the recognition error

Table 4. Recognition errors with extra dataset

Teams	1st		2nd		3rd		1st		2nd		3rd	
	S	**#E**	**S**	**#E**	**S**	**#E**	**R**	**#E**	**R**	**#E**	**R**	**#E**
USTC-iFLYTEK	'2'	46	')'	45	'X'	43	'Right'	212	'Sub'	70	'Inside'	35
IVTOV	')'	203	'-'	170	'2'	163	'Right'	736	'Sub'	365	'Sup'	194
MLE	'1'	301	'-'	286	')'	271	'Right'	1103	'Sub'	558	'Sup'	316

In Table 3 and Table 4, we further list the number of the top three incorrectly recognized symbols and structures. For different recognition systems, the main error is that a certain structural relationship in the symbol relationship tree is incorrectly recognized. Moreover, it can be seen that, the offline handwritten formulas in this competition are collected from clean paper. Even though, the performance in this competition is still far from satisfactory. Hence, offline HMER is a challenging task, and there is still a large room for improvement.

V. CONCLUSIONS

This paper presents the ICFHR 2020 OffRaSHME competition that focuses on the offline recognition of handwritten mathematical expressions (HMEs). A large dataset of offline HMEs were collected by asking people to write on papers and scanning papers that contain expressions. Moreover, the offline dataset were annotated at symbol level, to facilitate research of fine-grained HME recognition. Seven teams have submitted their systems in the competition. The winner team achieved an recognition accuracy of 79.85% using the provided datasets, which demonstrates the challenge of the task. In the future, we will collect offline HMEs from real scenarios with more complex conditions, where offline HMER may face the problems of blurring, noises, missing strokes, complex background, etc. The competition for Task 2, handwritten formula spotting, will also be organized in the future once the datasets are prepared, to push forward the research of offline HMER.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China under Grants 61773325, and by Natural Science Foundation of Fujian Province under Grants 2018J01574.

REFERENCES

[1] Jianshu Zhang, Jun Du, Lirong Dai: Track, Attend, and Parse (TAP): An End-to-End Framework for Online Handwritten Mathematical Expression Recognition. IEEE Trans. Multimedia 21(1): 221-233 (2019)

[2] Jin-Wen Wu, Fei Yin, Yan-Ming Zhang, Xu-Yao Zhang, Cheng-Lin Liu: Image-to-Markup Generation via Paired Adversarial Learning. ECML/PKDD (1) 2018: 18-34

[3] Harold Mouchère, Richard Zanibbi, Utpal Garain, Christian Viard-Gaudin: Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011-2014. IJDAR 19(2): 173-189 (2016)

[4] Harold Mouchère, Christian Viard-Gaudin, Dae Hwan Kim, Jin Hyung Kim, Utpal Garain: CROHME2011: Competition on Recognition of Online Handwritten Mathematical Expressions. ICDAR 2011: 1497-1500

[5] Harold Mouchère, Christian Viard-Gaudin, Dae Hwan Kim, Jin Hyung Kim, Utpal Garain: ICFHR 2012 Competition on Recognition of On-Line Mathematical Expressions (CROHME 2012). ICFHR 2012: 811-816

[6] Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, Utpal Garain, Dae Hwan Kim: ICDAR 2013 CROHME: Third International Competition on Recognition of Online Handwritten Mathematical Expressions. ICDAR 2013: 1428-1432

[7] Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, Utpal Garain: ICFHR 2014 Competition on Recognition of On-Line Handwritten Mathematical Expressions (CROHME 2014). ICFHR 2014: 791-796

[8] Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, Utpal Garain: ICFHR2016 CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions. ICFHR 2016: 607-612

[9] Mahshad Mahdavi, Richard Zanibbi, Harold Mouchère, Christian Viard-Gaudin, Utpal Garain: ICDAR 2019 CROHME + TFD: Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection. ICDAR 2019: 1533-1538

[10] Jianshu Zhang, Jun Du, Shiliang Zhang, Dan Liu, Yulong Hu, Jin-Shui Hu, Si Wei, Li-Rong Dai: Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition. Pattern Recognition 71: 196-206 (2017)

[11] Jianshu Zhang, Jun Du, Yongxin Yang, Yi-Zhe Song, Si Wei and Li-Rong Dai: A Tree-Structured Decoder for Image-to-Markup Generation. accepted, ICML 2020

[12] Zhe Li, Lianwen Jin, Songxuan Lai, Yuanzhi Zhu: Improving Attention-Based Handwritten Mathematical Expression Recognition with Scale Augmentation and Drop Attention. accepted, ICFHR 2020

[13] Thanh-Nghia Truong, Cuong Tuan Nguyen, and Masaki Nakagawa, Improvement of End-to-End Offline Handwritten Mathematical Expression Recognition by Weakly Supervised Learning, accepted, ICFHR 2020

[14] Chungkwong Chan: Stroke Extraction for Offline Handwritten Mathematical Expression Recognition. IEEE Access 8: 61565-61575 (2020)

[15] Dmytro Zhelezniakov, Viktor Zaytsev, Olga Radyvonenko: Acceleration of Online Recognition of 2D Sequences Using Deep Bidirectional LSTM and Dynamic Programming. IWANN (2) 2019: 438-449

[16] Jianshu Zhang, Jun Du, Lirong Dai: Multi-Scale Attention with Dense Encoder for Handwritten Mathematical Expression Recognition. ICPR 2018: 2245-2250

Authorized licensed use limited to: UNIVERSITY OF BATH. Downloaded on May 14,2021 at 08:16:37 UTC from IEEE Xplore. Restrictions apply.