Training an End-to-End System for Handwritten Mathematical Expression Recognition by Generated Patterns

Anh Duc Le and Masaki Nakagawa
Department of Computer Information
Tokyo University of Agriculture and Technology
Email: leducanh841988@gmail.com, nakagawa@cc.tuat.ac.jp

***Abstract*—Motivated by recent successes in neural machine translation and image caption generation, we present an end-to-end system to recognize Online Handwritten Mathematical Expressions (OHMEs). Our system has three parts: a convolution neural network for feature extraction, a bidirectional LSTM for encoding extracted features, and an LSTM and an attention model for generating target LaTex. For recognizing complex structures, our system needs large data for training. We propose local and global distortion models for generating OHMEs from the CROHME database. We evaluate the end-to-end system on the CROHME database and the generated databases. The experiential results show that the end-to-end system achieves 28.09% and 35.19% recognition rates on CROHME without and with the generated data, respectively.**

Keywords*—Online Handwritten Mathematical Expression Recognition, End-to-End Model, Encoder-Decoder Model, Patterns Generation

I. Introduction

Recognition of online handwritten mathematical expression (OHME) is one of the current challenges concerning handwriting recognition. It can be divided into three main processes. First, a sequence of input strokes is segmented into hypothetical symbols (symbol segmentation). Then hypothetical symbols are recognized by a symbol classifier (symbol recognition). Finally, structural relations among the recognized symbols are determined and the structure of the expression is analyzed by a parsing algorithm in order to provide the most likely interpretation of an input OHME (structural analysis). The recognition problem requires not only segmentation and recognition of symbols but also analysis of two-dimensional (2D) structures and interpretation of the structural relations. Ambiguities arise in all stages of the process.

Many approaches have been proposed for recognizing OHMEs especially during last two decades. They are summarized in the survey papers [1, 2] and the recent competition papers [3]. Most of them follow three interdependent processes as mentioned above. These processes can be handled independently [2] or jointly [4, 5, 6, 7]. In the following, we will review a few recent approaches participated in the recent Competition on Recognition of Online Handwritten Mathematical Expressions (CROHME).

A system for recognizing OHMEs by using a top-down parsing algorithm was proposed by MacLean et al. [4]. The incremental parsing process constructs a shared parse forest that presents all recognizable parses of the input. Then, the extraction process finds the top to n^{th}-most highly-ranked trees from the forest. By using horizontal and vertical order, this method reduces infeasible partitions and makes the method independent from stroke order. However, the worst-case number of sub-partitions that must be considered during parsing and the complexity of the parsing algorithm are still quite large as $O(n^4)$ and $O(n^4|P|)$, respectively. This system incorporates a correction mechanism to help users to edit recognition errors.

A global approach allowing mathematical symbols and structural relations to be learned directly from expressions was proposed by Awal et al. [5]. During the training phase, symbol hypotheses are generated without using a language model. The dynamic programming algorithm finds the best segmentation and recognition of the input. The classifier learns both the correct and incorrect segmentations. The training process is repeated to update the classifier until the classifier recognizes the training set of OHMEs correctly. Furthermore, contextual modeling based on structural analysis of the expression is employed, where the models are learnt directly from expressions using the global learning scheme.

A formal model for OHME recognition based on 2D Stochastic Context Free Grammar (SCFG) and Hidden Markov Model (HMM) was proposed by Alvaro et al. [6]. HMM uses both online and offline features to recognize mathematical symbols. The Cocke-Younger-Kasami (CYK) algorithm is modified to parse an input OHME in two dimensions (2D). They use the range search to improve time complexity from $O(n^4|P|)$ to $O(n^3 log n|P|)$. To determine structural relations among symbols and sub-expressions, a Support Vector Machine (SVM) learns geometric features between bounding boxes.

Le et al. presented a recognition method based on SCFG [7]. Stroke order is employed to reduce the search space and the CYK algorithm is employed to parse a sequence of input strokes. Therefore, the complexity of the parsing algorithm is still $O(n^3|P|)$, like that of the original CYK algorithm. They extended the grammar rules to cope with multiple symbol variations and proposed a concept of body box with two SVM models for classifying structural relations. The experiments showed the good recognition rate and practical processing time.

A modified version of the Minimum Span Tree (MST) based parsing algorithm was presented by Hu et al. [8]. The

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.151

parser extracts MST from a directed Line-of-Sight graph. The time complexity of this parsing method is lower than the time complexity of the CYK parsing method. This parser achieved good result of structure analysis on OHME patterns assuming correct segmentation and symbol recognition.

Mouchere et al. have been organizing CROHME for fair evaluation based on common databases. The above systems have shown good performance for the recent CROHME databases. However, they require ground-truth of OHMEs in different levels such as stroke, symbol, and structure. The collection and preparation of ground-truth for OHMEs are time-consuming tasks. The CROHME training set currently contains 8835 OHMEs from five different databases. It is hard to increase the number of OHMEs because it takes time and effort to collect and make ground-truth. Recently, Zhang et al. proposed a method using BLSTM for interpreting 2D languages such as OHMEs [9]. The method is an end-to-end model which requires only input data and their corresponding Latex or MathML. It is able to produce the result from input handwriting by using a BLSTM model. However, the performance is still lower than the above systems.

Recently, an attention-based encoder-decoder model has been successful in machine translation [10] and image caption generation [11]. It outperforms traditional methods in many tasks of sequence to sequence problems. Y. Deng et al. extended this model to recognize images of printed MEs to LaTex [12]. This model shows an encouraging result on printed MEs patterns. For OHMEs, however, the problem is hard since symbols and structures of OHMEs have more variations and distortions than those of printed MEs.

In this paper, we present an end-to-end system employing convolution neural network based on the attention-based encoder-decoder model. In our knowledge, this is the first work that employs the attention-based encoder-decoder for handwriting recognition. This system requires a large data for training, so we propose local and global distortion models to generate OHMEs from the CROHME database.

The rest of this paper is organized as follows. The end-to-end system for recognition of OHMEs is presented in Section 2. The local and global distortion models for data generation are described in Section 3. The experimental results are presented and discussed in Section 4. Conclusions are drawn in Section 5.

II. Overview of the End-to-End Recognition System

The structure of the end-to-end recognition system is shown in Figure 1. It has three parts: a convolution neural network for feature extraction from the image of an OHME, a bidirectional LSTM for encoding extracted features, and a LSTM and attention model for generating the target LaTex. They are described in the following sections.

A. Feature Extraction by CNN

Features are extracted from the image of an OHME by a convolution neural network which contains multiple layers of convolution and max-pooling layers. This is a standard CNN without recent techniques such as dropout, maxout, etc. An input image (H x W) is divided into (K x L) equal squares. In this paper, the size of square is (8 x 8). CNN takes a square and produces a feature vector with D elements. As a result, we obtain a sequence of feature vectors ($F_1, F_2, ..., F_{KxL}$) from an input image (H, W), where (H, W) and (K, L) are image sizes and reduced sizes and D is the depth of features. The order of the feature extraction is from left to right and from top to bottom.

B. Encoder

An encoder encodes the sequence of feature vectors into a sequence of outputs ($E_1, E_2, ..., E_{KxL}$). We employ a bidirectional LSTM which contains a forward LSTM and a backward LSTM.

C. Decoder

A decoder generates one symbol at a time. At each time step t, the decoder predict symbol $\underline{y_t}$ based on the current output O_t and the context vector C_t. O_t is calculated from the previous hidden state of the decoder h_{t-1}, the previous decoded vector O_{t-1}, and the previous symbol y_{t-1}. C_t is computed by weighted sum of the sequence of outputs and their weights produced by an attention model.

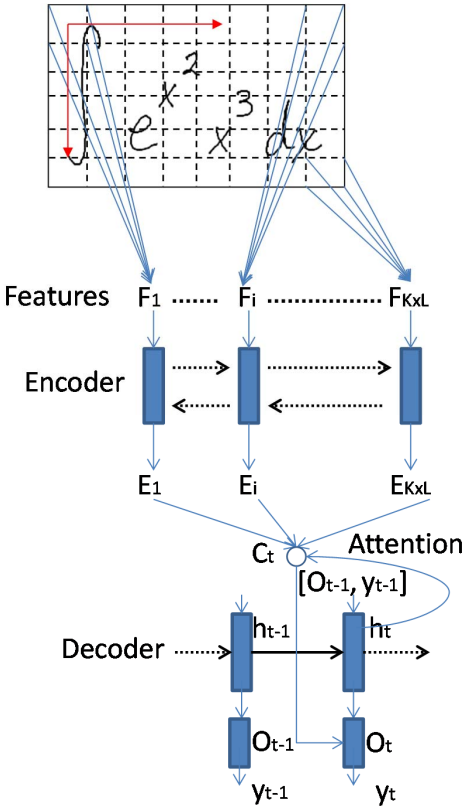


Fig. 1. Structure of the end-to-end model

III. Patterns Generation

Patterns generation was successfully applied for Japanese handwritten text recognition by Chen et al. [13]. In this work, we extend their model into local and global distortions. OHMEs are distorted by combination of local and global

distortions. The local distortion is applied for symbols in an OHME, while the global distortion is applied for the whole OHME. The local distortion includes shear, shrink, perspective, shrink plus rotation, and perspective plus rotation. The global distortion includes scaling and rotation. The process of distortion is shown in Figure 2. First, all symbols in an OHME are distorted by the same distortion models. Then, the OHME is distorted by scaling and rotation models sequentially. The distortion models are described in the following sections.

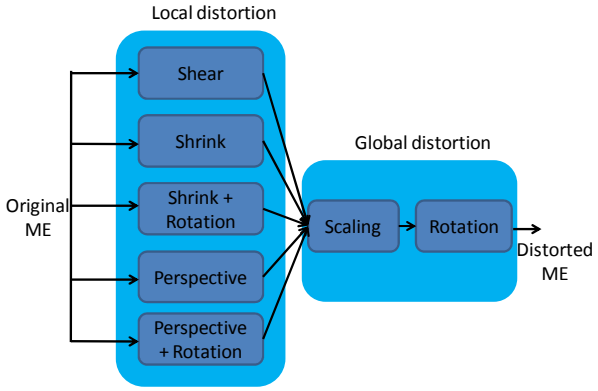


Fig. 2. Process of distortion model for patterns generation.

A. Local Distortion

Shear is a transformation that moves points in an axis by a distance increasing linearly with the other axis. The shear includes vertical and horizontal shear models. They are calculated by Eqs. (1) and (2).

The shrink and perspective are both similar to the shear with different transformation equations. The vertical and horizontal shrink models are described in Eqs. (3) and (4) respectively. The vertical and horizontal perspective models are shown in Eqs. (5) and (6), respectively.

The shrink plus rotation model applies shrink and rotation models sequentially. It is similar to the perspective plus rotation models. The rotation model is shown in Eq. (7).

$$\begin{cases} x' = x + y\tan\alpha \\ y' = y \end{cases} \quad (1) \qquad \begin{cases} x' = x \\ y' = y + x\tan\alpha \end{cases} \quad (2)$$

$$\begin{cases} x' = y\left(\sin\left(\frac{\pi}{2} - \alpha\right) - \left(\frac{x\sin(\alpha)}{100}\right)\right) \\ y' = y \end{cases} \quad (3)$$

$$\begin{cases} x' = x \\ y' = x\left(\sin\left(\frac{\pi}{2} - \alpha\right) - \left(\frac{y\sin(\alpha)}{100}\right)\right) \end{cases} \quad (4)$$

$$\begin{cases} x' = \frac{2}{3}\left(x + 50\cos\left(4\alpha\frac{x-50}{100}\right)\right) \\ y' = \frac{2}{3}y\left(\sin\left(\frac{\pi}{2} - \alpha\right) - \left(\frac{y\sin(\alpha)}{100}\right)\right) \end{cases} \quad (5)$$

$$\begin{cases} x' = \frac{2}{3}x\left(\sin\left(\frac{\pi}{2} - \alpha\right) - \left(\frac{x\sin(\alpha)}{100}\right)\right) \\ y' = \frac{2}{3}\left(y + 50\cos\left(4\alpha\frac{y-50}{100}\right)\right) \end{cases} \quad (6)$$

$$\begin{cases} x' = x\cos\beta + y\sin\beta \\ y' = x\sin\beta + y\cos\beta \end{cases} \quad (7)$$

where (x', y') is the new coordinate transformed by any of local distortion models, α is the angle of shear, shrink, and perspective distortion models and β is the angle of rotation distortion model. The local distortion model and its parameters are presented by (id, α, β) where id is the identifier of the distortion model from 1 to 5, α and β are from -10° to 10°.

Figure 3 show examples of local distortion models with α = 10° and β = 10°.

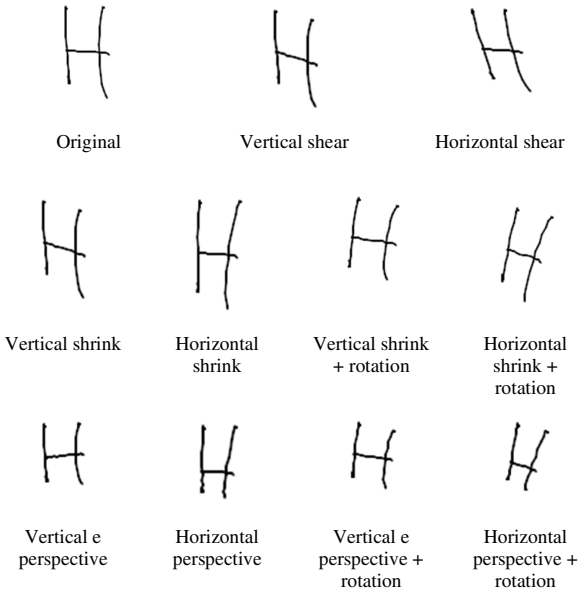


Fig. 3. Examples of local deformation by shear, shrink and perspective transformations.

B. Global Distortion

Global distortion distorts an OHME in baseline and size. We employ rotation and scaling models. The rotation model is similar to the local distortion. Scaling model is shown in Eq. (8). An example of the global distortion is shown in Figure 4.

$$\begin{cases} x' = kx \\ y' = ky \end{cases} \quad (8)$$

where k is the scaling factor. The parameters of the global distortion model are presented by (k, γ) where γ is the angle of the global rotation distortion model, k is from 0.7 to 1.3, and γ is from -10° to 10°.

Fig. 4. Examples of global distortion by scaling and rotation models.

C. Patterns generation

To generate an OHME, we first randomize five variables (*id, α, β, k, γ*). Then, all symbols in an OHME are distorted by local distortion models with (*id, α, β*). Then, the OHME is distorted by the global distortion model with (*k, γ*). Figure 5 shows some generated OHMEs from the original OHME that appeared in Figure 3.

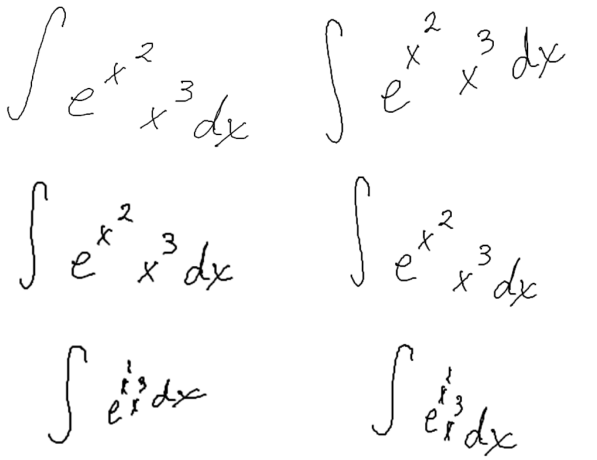
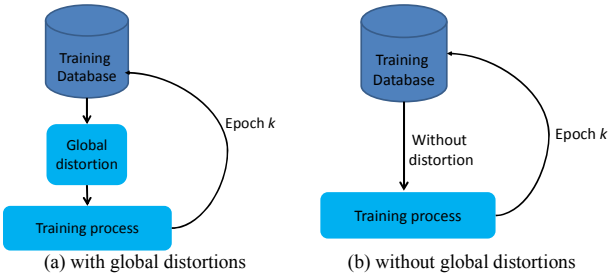


Fig. 5. Sample OHMEs generated by combination of local and global distortion models.

IV. Evaluation

First, we trained the end-to-end system by the CROHME training set. We repeatedly employ the training set to train the system. The training terminates when no increase of recognition rate is observed after 10 epochs. The resultant system is referred as the baseline system. Then, we created the two new training datasets G_CROHME1 and G_CROHME2 by patterns generation which is detailed in the next section. For each generated dataset, we trained the end-to-end system with applying global distortions of different values for parameters at every epoch as shown in Figure 6(a). We also trained the system without global distortion as shown Figure 6(b). Namely, training with global distortions uses images from training set with global distortions applied at the beginning of every epoch while training without distortion employ the same images from the training set for every epoch. Then, we evaluated all the systems on the CROHME 2014 test set.



(a) with global distortions (b) without global distortions

Fig. 6. The training process of the end-to-end model with and without distortions.

Next, we compared the performance of the best end-to-end system in the above with the other systems which participated CROHME 2014.

A. Databases

We use the CROHME 2014 database [11]. Organized at ICHFR 2014, CROHME 2014 was a contest in which OHME recognition algorithms competed. It allows the performance of the proposed system to be compared with others under the same conditions. There were seven participants. The CROHME 2014 database contains 8,835 OHMEs for training and 986 OHMEs for testing. The number of symbol classes is 101.

We generated more patterns by using the above-mentioned distortion models. We prepared two new training sets named as G_CROHME1 and G_CROHME2. G_CROHME1 and G_CROHME2 were created by generating 3 and 5 new OHMEs from every OHME in the CROHME training set, respectively. They also include original OHMEs from the CROHME training set. The number of OHMEs and generated OHMEs for each training set are shown in table I.

TABLE I. Description of Training Sets

	CROHME training set	**G_CROHME 1**	**G_CROHME 2**
# of OHMEs	8,835	35,340	53010
# of generated OHMEs	0	26,508	44,180

We employ the CROHME 2013 test set for validation and the CROHME 2014 test set for evaluation.

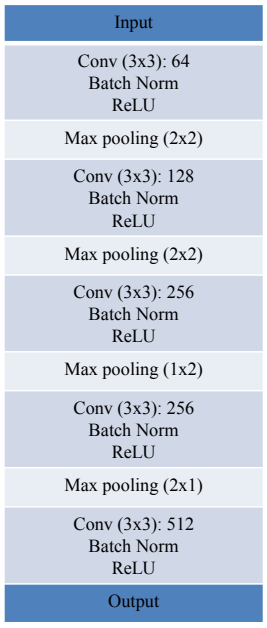


Fig. 7. Structure of CNN feature extraction. The parameters of the convolution and max pooling layers are denoted as "Conv (filter size): number of filters" and "Max pooling (filter size)", respectively.

B. End-to-end system configuration

A CNN with convolution, batch norm, ReLU, and max-pooling layers was employed for feature extraction as shown in Figure 7. A single layer bidirectional LSTM and a single layer LSTM are used for the encoder and decoder, respectively. The size of hidden states of the encoder and decoder is 256 and 512, respectively. We used mini-batch stochastic gradient descent to learn the parameters. The initial learning rate was set to 0.1. The training process was stopped when the recognition rate on validation set stopped improving after 10 epochs. The system was implemented by using Torch and the Seq2seq-attn NMT system [14]. All the experiments were performed on a 4GB Nvidia Tesla K20.

C. Results

The first experiment evaluated the performance of the end-to-end systems trained on the CROHME training set, G_CROHME1, and G_CROHME2. An OHME is recognized correctly in terms of expression level if all of its symbols, relations and its structure are recognized correctly. For measurement, we use expression recognition rate which counts OHMEs recognized at the expression level over all the testing OHMEs. The training process is shown in Figure 6(b). Table II shows the recognition rate on validation and testing sets by using different training sets. The recognition rates on both validation and testing set increase when the number of training patterns increases.

TABLE II. PERFORMANCE OF END TO END SYSTEM ON DIFFERENT TRAINING SETS.

Rec. rate	CROHME training set	G_CROHME 1	G_CROHME 2
Validation(%)	17.16	19.55	21.64
Testing(%)	18.97	21.10	26.27

One of techniques to prevent over-fitting and improve generalization of neural models is to use distortions at the beginning of every epoch. In this experiment, we employed the global distortion model described in Section III.B and the same data in Table II. The training process is shown in Figure 6(a). The results are shown in Table III. Similarly, the recognition rates increase when the number of training patterns increases.

TABLE III. PERFORMANCE OF END TO END SYSTEM WITH DISTORTION ON TRAINING.

Rec. rate	CROHME training set	G_CROHME 1	G_CROHME 2
Validation(%)	23.25	30.04	30.10
Testing(%)	28.09	34.99	35.19

Table IV shows our best recognition result and the results of the systems which participated in the CROHME 2014 competition. The four factors are measured in the evaluation, namely, *Sym Seg* as symbol segmentation rate, *Sym Seg + Rec* as symbol segmentation and recognition rate, *Rel Tree* as rate of structural analysis (termed "relation tree"), and *Exp Rec* as expression recognition rate. The end-to-end system produces latex format, so that we obtain only expression recognition rate. The best end-to-end system is ranked third after systems I and III.

TABLE IV. COMPARISION OF END TO END MODEL AND THE RECOGNITION SYSTEMS ON CROHME 2014 (%)

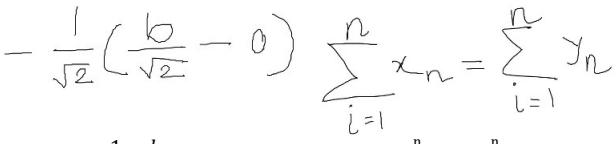
Method \ Measure	Sym Seg	Sym Seg + Rec	Rel Tree	Exp Rec
I	93.31	86.59	84.23	37.22
II	76.63	66.97	60.31	15.01
III	98.42	93.91	94.26	62.68
IV	85.52	76.64	70.78	18.97
V	88.23	78.45	61.38	18.97
VI	83.05	69.72	66.83	25.66
VII	89.43	76.53	71.77	26.06
End-to-end	N/A	N/A	N/A	35.19

Finally, we evaluate the end-to-end system by structure recognition rate. Structure recognition rate is calculated by the percent of OHMEs whose structure is recognized correctly irrespective of symbol labels. For example, the two OHMEs ($x^2 + 1$ and $x^3 + 7$) share the same structure. Table V shows the structure recognition rates of the end-to-end systems trained on the CROHME training set, G_CROHME1, and G_CROHME2. It shows that the end-to-end systems can learn well the structures of OHMEs. If we want to improve the expression recognition rates of the end-to-end systems, the remaining problem is how to improve the symbol recognition inside the end-to-end systems.

TABLE V. STRUCTURE RECOGNITION RATE OF END-TO-END SYSTEMS WITH DISTORTION ON TRAINING SET.

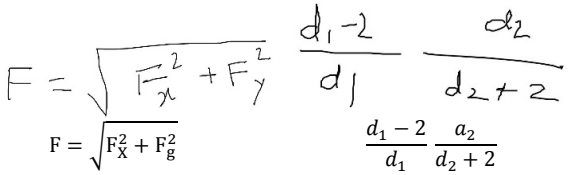
Rec. rate	CROHME training set	G_CROHME 1	G_CROHME 2
Testing(%)	51.52	58.22	56.69

Figure 8 shows examples recognized correctly and incorrectly by the end-to-end system trained on G_CROHME2.



$$-\frac{1}{\sqrt{2}}\left(\frac{b}{\sqrt{2}}-0\right) \qquad \sum_{i=1}^{n} x_n = \sum_{i=1}^{n} y_n$$

a). Correctly recognition



$$F = \sqrt{F_x^2 + F_y^2} \qquad \frac{d_1 - 2}{d_1} \quad \frac{a_2}{d_2 + 2}$$

b) Incorrectly recognition

Fig. 8. Examples recognized correctly and incorrectly by our system.

V. CONCLUSION

In this paper, we have presented the end-to-end system for recognizing OHMEs. We proposed a combination of the local and global distortion models for patterns generation. The efficiencies of the proposed local and global distortion models are demonstrated through the experiments. The recognition rate is improved when we increase the number of training patterns. It achieves 28.09%, 34.99% and 35.19% by using distortion on the CROHME training set, G_CROHME1, and G_CROHME2, respectively. It shows that the end-to-end system is a potential system to compare with existing systems of OHME recognition.

There still remain problems to improve the expression recognition rate of the end-to-end system as follows. First, we should generate more OHMEs whose structures are more varied and employ a larger memory GPU for training. Then, we should improve the symbol recognition inside the end-to-end system by employing tree-structured LSTM [15] or decomposable attention model [16].

ACKNOWLEDGMENT

This research has been supported by JSPS fellowship under the number 15J08654.

REFERENCES

[1] K. Chan and D. Yeung, Mathematical Expression Recognition: A Survey, International Journal of Document Analysis and Recognition, pp. 3-15, 2000

[2] R. Zanibbi and D. Blostein, Recognition and Retrieval of Mathematical Expressions, International Journal of Document Analysis and Recognition , pp.331-357, 2012.

[3] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, and U. Garain, ICFHR 2014 competition on recognition of on-line handwritten mathematical expressions (CROHME 2014). Proc. Int'l Conf. Frontiers in Handwriting Recognition, pp. 791-796, 2014.

[4] S. MacLean and G. Labahn, A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets. International Journal of Document Analysis and Recognition, vol. 16, pp. 139-163, 2013.

[5] A. M. Awal, H. Mouchère, and C. Viard-Gaudin, A global learning approach for an online handwritten mathematical expression recognition system, Pattern Recognition Letters, 2014, pp. 68–77.

[6] F. Alvaro, J. Sánchez, and J. Benedí, Recognition of On-line Handwritten Mathematical Expressions Using 2D Stochastic Context-Free Grammars and Hidden Markov Models, Pattern Recognition Letters, pp. 58-67, 2014.

[7] A.D. Le and M. Nakagawa, A system for recognizing online handwritten mathematical expressions by using improved structural analysis, International Journal of Document Analysis and Recognition, , vol. 19, pp 305–319, 2016.

[8] L Hu, R Zanibbi, MST-based visual parsing of online handwritten mathematical expressions, 15th International Conference on Frontiers in Handwriting Recognition, 2016, pp. 337-342.

[9] T. Zhang, H. Mouchere, C. Viard-Gaudin: Using BLSTM for interpretation of 2-D languages. Case of handwritten mathematical expressions. Document Numérique 19(2-3): 135-157 (2016).

[10] Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[11] Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of The 32nd International Conference on Machine Learning, 2048‑2057.

[12] Y. Deng, A. Kanervisto, and A. M. Rush, What You Get Is What You See: A Visual Markup Decompiler, arXiv preprint http://arxiv.org/pdf/1609.04938v1.pdf

[13] B. Chen, B. Zhu and M. Nakagawa: Training of an On-line Handwritten Japanese Character Recognizer by Artificial Patterns, Pattern Recognition Letters, Vol. 35, No. 1, pp.178-185.

[14] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M.Rush, OpenNMT: Open-Source Toolkit for Neural Machine Translation, ArXiv e-prints, eprint = {1701.02810}.K. S. Tai, R. Socher, and C. Manning, Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075, 2015.

[15] K. S. Tai, R. Socher, and C. Manning, Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075, 2015.

[16] A. P. Parikh, O. Tackstrom, D. Das, J. Uszkoreit, A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933 , 2016.