2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Context-Aware Mathematical Expression Recognition: An End-to-End Framework and A Benchmark

Wenhao He[1,4], Yuxuan Luo[2], Fei Yin[1], Han Hu[2], Junyu Han[2], Errui Ding[2], Cheng-Lin Liu[1,3,4]
[1] National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences,
95 Zhongguan East Road, Beijing 100190, P.R. China
[2] Institute of Deep Learning, Baidu Research, Beijing, P.R. China
[3] CAS Center for Excellence in Brain Science and Intelligence Technology,
95 Zhongguan East Road, Beijing 100190, P.R. China
[4] University of Chinese Academy of Sciences, Beijing, P.R. China
Email: {wenhao.he, fyin, liucl}@nlpr.ia.ac.cn, {luoyuxuan, huhan02, hanjunyu, dingerrui}@baidu.com

Abstract—**In this paper we propose a novel end-to-end framework for mathematical expression (ME) recognition. The method uses a convolutional neural network (CNN) to perform mathematical symbol detection and recognition simultaneously incorporating spatial context, and can handle multi-part and touching symbols effectively. To evaluate the performance, we provide a benchmark that contains MEs both from real-life and synthetic data. Images in our dataset undergo multiple variations such as viewpoint, illumination and background. For training, we use pure synthetic data for saving human labeling effort. The proposed method achieved 87% accuracy of total correct for clear images and 45% for cluttered ones.**

I. Introduction

As an essential module in an Optical Character Recognition (OCR) system for technical papers, mathematical expression (ME) recognition has long been studied, traced back to forty years ago [1]. Recent research [2] [3] [4] mostly lays emphasis on online ME recognition, where stroke sequences are available, while offline recognition is receiving increasing attentions for application needs in digitization and retrieval of paper documents. Offline ME recognition also faces increasing challenge because more and more documents are captured by hand-held cameras (including mobile phones) instead of flat scanners.

A typical mathematical expression recognition system contains two stages: symbol recognition and structure analysis [1] [5]. The former stage tells where and what kinds of mathematical symbols are and the latter one parses structural relations among symbols based on the given locations and category information. Some previous works [6] [7] [8] treated the two stages separately. Without using structure information in the symbol segmentation and recognition stage, such methods are prone to errors.

There have been two main approaches for integrating symbol recognition and structure analysis in ME recognition. The Hidden Markov Model (HMM) has been used for performing symbol recognition and structural analysis simultaneously in online handwritten ME recognition [1], where the input is a sequence of strokes. In offline ME recognition, graph grammars and Stochastic Context-Free Grammars (SCFG) [9] [10] [11] have been proven effective in modeling the structure information of MEs. These methods, however, still cannot overcome the symbol segmentation problem when there are broken and touched symbols.

In recent years, deep neural networks, especially the Convolutional Neural Network (CNN) [12], have been widely used with great success in computer vision problems including OCR. The CNN is potentially applicable to ME recognition but still needs careful consideration in integrating symbol segmentation, recognition and structure analysis.

In this paper, we propose a novel context-aware framework for offline ME recognition based on CNN. The proposed method can employ context information embedded in convolutional features to locate and recognize symbols, and can segment touching and multi-part symbols. To reduce the human labeling effort in data collection, we train the CNN model using synthetic data, in which the ground truths are easily aligned with images. To test the proposed method, we provide a database containing 416 ME images taken by mobile phones in various conditions (See Fig 1). We evaluate the proposed method from two aspects: symbol-level detection performance (precision, recall and F_1 score) and expression-level rate of at most N symbol errors. On real test images, the proposed method achieved 87% expression-level accuracy of total correct on clear images and 45% on cluttered ones.

The reminder of this paper is organized as follows: Section 2 reviews related works. Section 3 describes the proposed end-to-end method for ME recognition. Section 4 introduces the training and test datasets. Section 5 presents our experimental results, and Section 6 offers concluding remarks.

II. Related Work

In the context of offline ME recognition, many works have contributed to the related issues of symbol segmentation, recognition, structural analysis and benchmark. For symbol

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

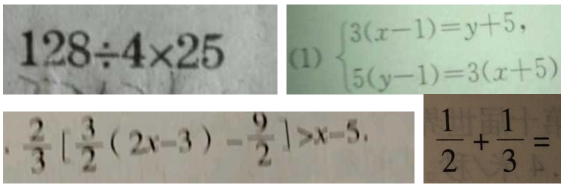


Fig. 1. ME images taken by mobile devices

segmentation, connected components and projection projection cutting methods are widely adopted [1] [7] [8] [10] [13], but these segmentation methods usually face two problems: (1) Symbols composed of multiple components need a merging strategy [1]; (2) Touching symbols need splitting [14]. For symbol recognition, manually designed features [15] extracted from segmented symbols are utilized for classification. However this entails a post-processing for disambiguating symbols like '—' and fixed groups like 'log' and 'sin' because the features contain no context information. For structural analysis, works on ME grammars like [9] [10] [11] are widely used to tackle both symbol recognition and structural analysis. For the benchmarks in offline ME recognition, previous work can be found in [16] [17] [18]. Images from these databases are mostly selected from scanned documents, PDF files or synthetic data which have clean background, uniform illumination and standard viewpoint.

CNN based object detection methods fall into two groups: proposal based [19] [20] [21] and sliding window based [22] [23]. The proposal based method is much like traditional segmentation-recognition pipeline in ME recognition which cannot avoid problems in pre-segmenting symbols. We there adopt a sliding window based method as the base to design the context-aware framework.

III. Proposed Method

A. Overview

The block diagram of our system for ME recognition is illustrated in Fig 2. In the test phase, image pyramid is fed into the network and after several layers of convolution, down-sampling and up-sampling, the output feature maps are sent to three tasks. Each task outputs a map with the same height and width as the input feature maps. Given a point i located at (w_i, h_i) in feature maps, the detection task gives a confidence score s that a symbol locates at i, the regression task gives a 4-dimensional vector $\{x_1, y_1, x_2, y_2\}$ representing the bounding box of the symbol at i and the recognition task gives the symbol category with maximum probability at i. The detection and regression tasks are designed to locate symbols, so we combine the output of these two tasks into bounding boxes each of which is represented by a 5-dimensional vector $p = \{s, x_1, y_1, x_2, y_2\}$ and apply non-maximum suppression (NMS) to all the bounding boxes with s above a threshold. After getting each symbol's bounding box, the final step is to decide their categories by combining recognition results with suppressed bounding boxes. The multi-task structure aims to split the complicate ME recognition into simpler sub-tasks and the shared convolutional features greatly reduces computation.

B. Region of Interest (RoI) Sampling

Unlike test stage which takes images of arbitrary resolution, to generate training samples, we first perform RoI sampling. First, we crop large patches containing mathematical symbols and sufficient context information. And then each patch is cropped and resized to 240×240 with a mathematical symbol in the center that roughly has diagonal line of 48 pixels. The output ground truth in training is a 6-channel map sized of 60×60. The first channel is a 0-1 matrix in which the positive labeled region is a circle with radius r_d, located in the center of a mathematical symbol bounding box. The radius r_d is proportional to the bounding box diagonal line length with the ratio of 0.3. The following 4 channels are filled with distances from top-left and bottom-right corners to the center of a bounding box. The last channel is modified from the first one by replacing the positive value 1 to K which represents the category of the symbol in the corresponding region. In addition, if multiple symbols fall into a patch, we treated symbols that have similar scales to the centered symbol as positive.

C. Network Architecture

The network architecture is shown in Fig 3 and each convolution layer is connected with a ReLU neuron. The Inception1, Inception2-1 to Inception2-5 are modified from the Inception layer introduced in [24]. The up-sampling in $Concat\&Resize$ block is realized by bilinear interpolation. Since receptive fields vary between layers, the output of Inception1 is concatenated with those of Inception2-1, Inception2-3 and Inception2-5 to assist this model to process symbols of different scales. Each $Output$ block is composed of three tasks introduced above.

The detection task intends to give a confidence map of symbols. Given K samples with the ground truth 0-1 matrix y^* and predicted confidence map $\hat{y}$, the detection cost function is displayed in Eq(1). We adopt quadratically smoothed Hinge Loss here instead of Euclidean Loss used in [22] for better binary classification performance.

$$L_{det}\left(\hat{y}, y^*\right) = \frac{1}{2K} \sum_{i=1}^{K} \max\left(0, \operatorname{sign}\left(0.5 - y_i^*\right) \cdot \left(\hat{y}_i - y_i^*\right)\right)^2 \tag{1}$$

For the cost function L_{reg} of regression task, given the ground truth $\hat{d}$ and predicted d^*, we have:

$$L_{reg} = \frac{1}{2K} \sum_{i=1}^{K} \left\| \hat{d}_i - d_i^* \right\|^2 \tag{2}$$

For the last recognition task, we adopt softmax loss denoted as L_{cls}. Given C kinds of categories, the ground truth g^* and predicted distribution $\hat{g}$, we have Eq(3).

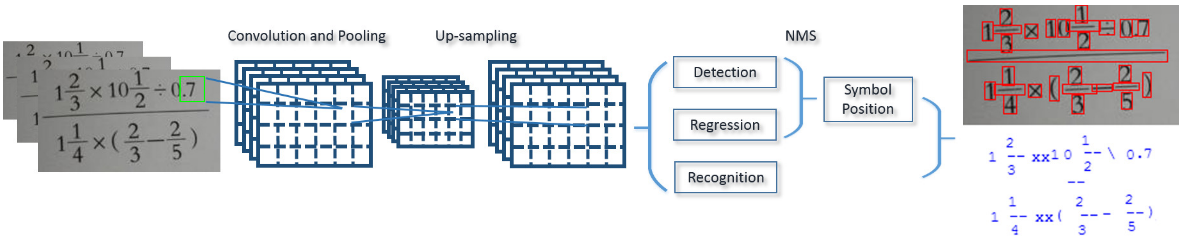
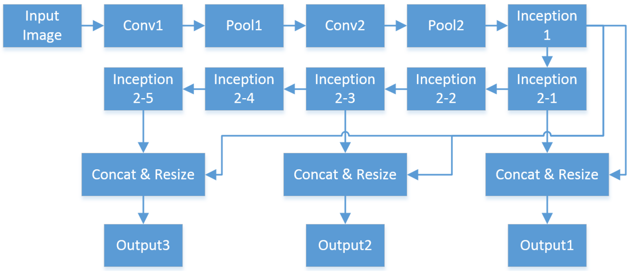
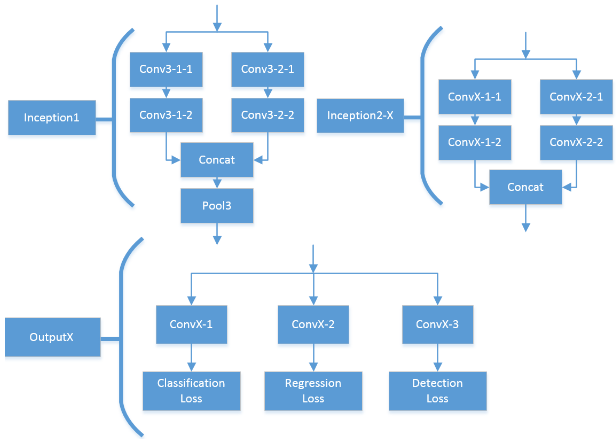


Fig. 2. The context-aware model pipeline



(a)



(b)

Fig. 3. Network structure of the proposed model. (a) Overall structure. (b) Detailed structures of sub network.

$$L_{cls} = \frac{1}{K} \sum_{i=1}^{K} \sum_{j=1}^{C} 1\left\{g_i^* = j \text{ and } g_i^* > 0\right\} \cdot \log \hat{g}_i \qquad (3)$$

It should be noticed that, we ignore background region for it has already been classified in detection task.

To train this network, first we randomly choose one of the three output pipelines and resize the training samples and ground truths provided by RoI sampling to fit the receptive field. Parameters owned exclusively by the other two pipelines will not be trained. In the test phase, three output pipelines will give results simultaneously. Benefiting from the different receptive field of each output pipeline, we will consume much less time for we need smaller image pyramid.

D. Multi-task Fusion

In test phase, the network takes an image pyramid as input and produces a 7-dimensional vector for each pixel. Denote the 7-dimensional vector p as $p = \{s, x_1, y_1, x_2, y_2, c, c_s\}$. The first element s is the confidence score of being a symbol, and x_1, y_1, x_2, y_2 denote the left, top, right, bottom of the bounding box centered by pixel p, respectively. The elements c and c_s represent predicted symbol category and its probability, respectively. Candidate outputs are filtered out by selecting vectors p with s higher than a fixed threshold. Then non-maximum suppression (NMS) using the first 5 elements in p is performed for the candidate bounding boxes to get bounding boxes for each symbol. Since we get bounding boxes of symbols, next we will predict the category for each symbol.

Given a bounding box B_0 after NMS, reconsider candidate vectors p whose bounding boxes have more than 0.5 Intersection-over-Union (IoU) ratio with B_0. And the category C for this box is given by a simple voting strategy below.

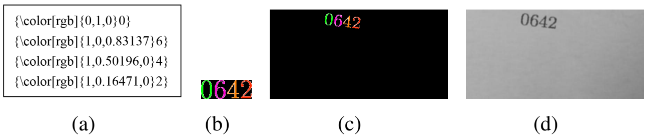
$$C = \arg\max_i \left(\sum_i c_s | c = i \right) \qquad (4)$$

The reason why we employ a voting strategy is that the bounding box B_0 is selected according to detection score without enough recognition evidence, so directly giving the recognition result of the highest detection score pixel may not be accurate.

IV. Data Preparation

We consider 97 classes of symbols which cover all the digits and letters, as well as widely used mathematical symbols. All the 97 symbols are listed in Table 1. To generate synthetic data for training, firstly we generate ME LaTeX strings and assign different colors to symbols according to categories. Secondly we convert the LaTeX strings into images. Thirdly, get the bounding boxes of symbols by the mapping between symbols and colors. Fourthly, distort images by translation, rotation, degradation and adding textured background from real pages. Fig 4 displays the proceeder of generating a ME image.

The test set is divided into three subsets S_1, S_2 and S_3. S_1 contains images captured under controlled conditions, S_2 contains images selected without constrain and S_3 contains all synthetic MEs. The statistics of symbols for the training and test sets are shown in Table 2.



```
{\color[rgb]{0,1,0}0}
{\color[rgb]{1,0,0.83137}6}
{\color[rgb]{1,0.50196,0}4}
{\color[rgb]{1,0.16471,0}2}
```

Fig. 4. Proceeder of generating a ME image. (a) ME LaTeX string. (b) Origin color ME image. (c) Distorted ME image. (d) ME image blended with real page texture.

TABLE I
THE 97 SYMBOLS CLUSTERED INTO 6 GROUPS

Symbol Type	Digit and Letter	Operator	Pair Symbol	Other
Content	0-9, a-z, A-Z	$+, -, \times, \div, =, >, <, \geq, \geqslant, \leq, \leqslant, \neq, \ngtr, \nless, \ngeq, \ngeqslant, \nleq, \nleqslant, \approx, \%$	$(,), [,]$	$\pi, :, , .., \cdots ,$ □, ■, △, ▲

V. EXPERIMENTS

A. Implementation Details

We train our model with no pre-trained model. We set the loss weight for recognition 10 times higher than the other two tasks for the first 2 epochs and 10 times lower afterwards. We also set the hard negative ratio introduced in [22] to be 0.6 during training stage. In the test stage, we resize an input image into pyramid of multi-scale which deals with symbol size varying from 6 pixels to 100 pixels. The whole network is trained on a single NVIDIA Tesla K40 GPU with 11GB graphic memory and the platform we adopted is Caffe [25].

Note that we perform dense cropping for bounding boxes of fraction lines considering it requires deeper model to detect the whole fraction line, so we try to detect part of them instead and combine them with simple rules. As for small scale symbols, like dot and minus sign, we pad their bounding boxes satisfying that they have similar height to symbols nearby. Fig 5 shows the special pre-precessed bounding boxes.

TABLE II
STATISTICS OF THE DATA SET

Data Type	Synthetic	Real-life
Train Set	378000	/
Test Set S_1	/	214
Test Set S_2	/	202
Test Set S_3	2500	/

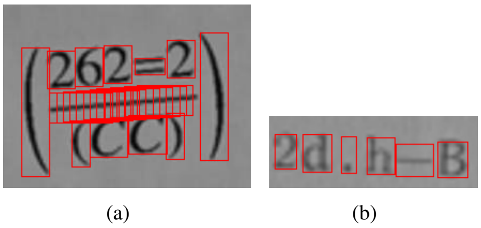


Fig. 5. (a) Fraction line is cut into dense overlapped pieces. (b) Bounding boxes of dot and minus are padded horizontally.

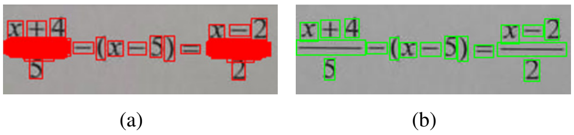


Fig. 6. (a) The fraction line pieces before grouping. (b) The grouped fraction line.

TABLE III
COMPARISON OF CONTEXT-AWARE NETWORK AND RCNN. (A) THE RECALL, PRECISION AND F1 SCORE (RPF) OF THE WHOLE TEST SET. (B) THE RPF OF THE WHOLE TEST SET WITH LESS SYMBOLS EVALUATED

(a)

RPF	S_1	S_2	S_3
OUR	95.49/96.29/95.88	88.25/90.35/89.28	98.00/97.20/97.59
RCNN	74.69/84.79/79.42	46.52/79.91/58.80	85.60/83.87/84.72

(b)

RPF	S_1	S_2	S_3
OUR	95.62/96.22/95.91	88.35/91.21/89.75	98.11/97.25/97.67
RCNN	83.62/87.79/85.65	46.33/83.01/59.46	89.03/85.38/87.16

To combine fraction line pieces, we first aggregate them into several groups G, such that no box in a group overlaps with boxes in another, and then we can assume that each group G_i in G contains only one fraction line. After that, denote boxes in G_i as h, and the boundary of the fraction in G_i is given by following expressions. The subscript numbers 1 to 4 refer to the left, right, top and bottom of the bounding box, respectively. The Fig 6 shows the result of fraction line grouping.

$$\begin{cases} G_1 = \min(h_1) \\ G_2 = \max(h_2) \\ G_3 = \operatorname{mean}(h_3) \\ G_4 = \operatorname{mean}(h_4) \end{cases} \tag{5}$$

B. Comparison with RCNN

We compare the performance of our context-aware network with RCNN [20], a proposal based detection architecture, for which we employ Maximally-Stable-Extremal-Regions (MSER) method to extract proposals. To get a fixed size of training and testing samples, we pad origin proposals into squared ones and resize them into 32 × 32. The padded proposals that overlap with ground truth higher than a symbol-related threshold are regarded to be positive, otherwise are treated as negative samples.

It may require more careful pre-processing for proposals of certain symbols like '$= \div \geq ()$', so we also give another result with less symbols evaluated. The recall, precision and F_1 scores are displayed in Table 3. In comparison, the proposed method outperforms RCNN on all three datasets. The expression level evaluation results are shown in Fig 7. In expression level evaluation, given $N = 0$ we will get the ratio of all correct. From Fig 7 we can see the all correct ratio is 87% for S_1 and 45% for S_2.

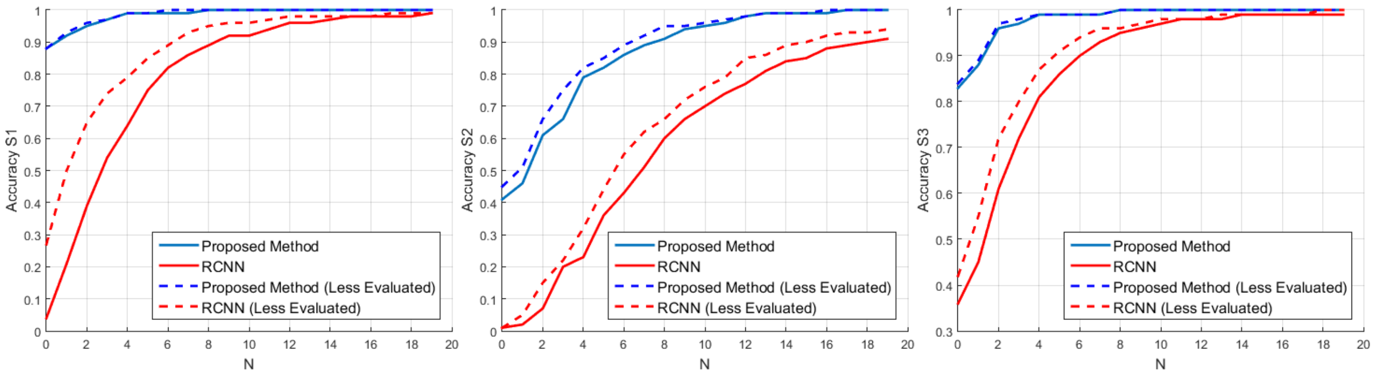


Fig. 7. Expression level evaluation results of S_1 (top left), S_2 (top right) and S_3 (bottom left). The solid lines represent origin results and the dashed ones represent results with less symbol evaluated.

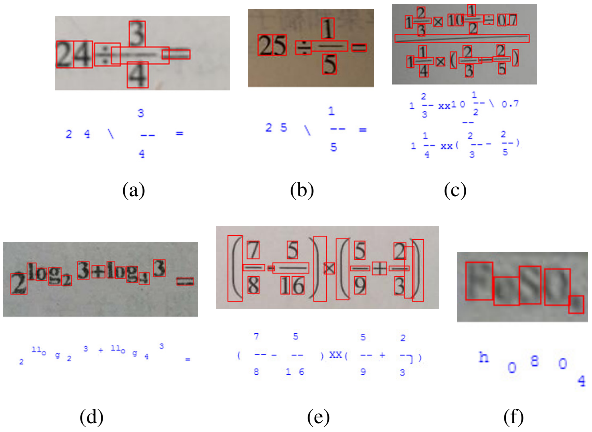


Fig. 8. Results of context-aware framework for hard tackling cases. (a) Auto-cut touching symbols. We use ‘- -’ to represent fraction line and \ to represent ‘÷’; (b) (c) Auto-combine multi-parts symbols like ‘÷’ ‘=’. ‘xx’ that represents multiplication sign is used to distinguish from letter ‘x’; (d) Ambiguous symbol ‘1’, ‘o’ in ‘log’ is correctly recognized. We use ‘ll’ here to distinguish from digit ‘1’; (e) (f) Failed cases for long symbols like brackets and blurred ones

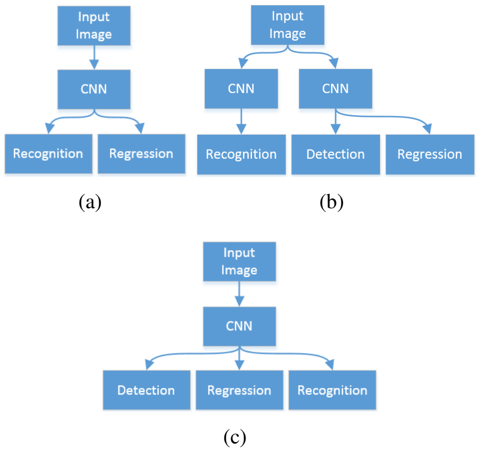


Fig. 9. (a) The structure without shared convolutional features. (b) The structure with no detection. (c) The proposed method

C. Case Analysis

The purpose of context-aware framework is to solve the existing problems in ME recognition. The Fig 8 displays some hard-tackling cases for previous methods and results of our model. The results in Fig 8 display the effectiveness of proposed method to handle multi-parts symbols like ‘÷’, ‘:’, touching case in Fig 8.a and ambiguous symbols like fraction line, minus sign, letter ‘l’ and digit ‘1’. We also display some failed cases like long and blurred symbols. For long symbols like bracket, regression task tends to output poor results and for blurred images like Fig 8.f, we human could recognize the symbols ‘$FeSO_4$’ depending on our much more abundant prior knowledge.

D. Effectiveness of Multi-task Learning

To test the effectiveness of triple tasks, we have designed two extra networks. The first one removes detection task and takes the background category into recognition. This no-detection network is designed to demonstrate the importance of detection task though the recognition task could also handle detection. The second network trains recognition task separately with the same convolution structure as detection and regression tasks. The latter network intends to prove the benefit of shared feature design. Simplified structures of three networks for comparison are shown in Fig 9.

The results are shown in Table 4. By comparison we can come to the conclusion that triple-task structure indeed benefits the performance. The no-detection network gives a much lower precision which indicates the idea that splitting a complicated task into two indeed helps improve the overall performance. For the unshared feature structure, precision and recall are both slightly lower than that of the proposed method. It is obvious that the shared feature structure could provide enough information and save much more computation than the other.

VI. Conclusion

In this paper, we present a context-aware end-to-end system for ME recognition, by introducing a CNN based structure with multi-task learning to perform mathematical symbols detection and recognition simultaneously. Experiments verify

TABLE IV
RECALL, PRECISION AND F1 SCORE OF DIFFERENT OUTPUT STRUCTURES (PERCENTAGE) ON THE WHOLE TEST SET. M_1 TO M_3 REPRESENT THE PROPOSED NETWORK, NO-DETECTION AND UNSHARED FEATURE STRUCTURE RESPECTIVELY.

RPF	S_1	S_2	S_3
M_1	95.49/96.29/95.88	88.25/90.35/89.28	98.00/97.20/97.59
M_2	94.10/65.18/77.01	86.22/61.63/71.88	94.52/65.72/77.53
M_3	93.45/94.73/94.08	86.65/90.25/88.41	97.68/96.96/97.31

that the context information contained by convolutional features helps improve the performance. We also build a database supplying ME images taken under various circumstances. Compared with existing ME benchmarks, ours focuses on images taken by hand-held cameras and we believe that the end-to-end model and dataset can assist to stimulate further research in this area.

Though being able to locate and recognize the symbols in MEs, the proposed method is not able to interpret the ME structure and we will consider structural analysis in our future work.

ACKNOWLEDGMENTS

This work has been supported in part by the National Basic Research Program of China (973 Program) Grant 2012CB316302,the National Natural Science Foundation of China (NSFC) Grant 61411136002, the Strategic Priority Research Program of the Chinese Academy of Sciences (Grants XDA06040102 and XDB02060009).

REFERENCES

[1] K.-F. Chan and D.-Y. Yeung, “Mathematical expression recognition: a survey,” *International Journal on Document Analysis and Recognition*, vol. 3, no. 1, pp. 3–15, 2000.

[2] H. Mouchere, C. Viard-Gaudin, D. H. Kim, J. H. Kim, and U. Garain, “Crohme2011: Competition on recognition of online handwritten mathematical expressions,” in *Proceedings of the 11th International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 1497–1500.

[3] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, and U. Garain, “Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014),” in *Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 791–796.

[4] F. Simistira, V. Katsouros, and G. Carayannis, “Recognition of online handwritten mathematical formulas using probabilistic svms and stochastic context free grammars,” *Pattern Recognition Letters*, vol. 53, pp. 85–92, 2015.

[5] H. M. Twaakyondo and M. Okamoto, “Structure analysis and recognition of mathematical expressions,” in *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 1. IEEE, 1995, pp. 430–437.

[6] R. Zanibbi, D. Blostein, and J. R. Cordy, “Recognizing mathematical expressions using tree transformation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1455–1467, 2002.

[7] H.-J. Lee and J.-S. Wang, “Design of a mathematical expression recognition system,” in *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 2. IEEE, 1995, pp. 1084–1087.

[8] J. Ha, R. M. Haralick, and I. T. Phillips, “Understanding mathematical expressions from document images,” in *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 2. IEEE, 1995, pp. 956–959.

[9] A. Grbavec and D. Blostein, “Mathematics recognition using graph rewriting,” in *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 1. IEEE, 1995, pp. 417–421.

[10] F. Alvaro, J.-A. Sánchez, and J.-M. Benedí, “Recognition of printed mathematical expressions using two-dimensional stochastic context-free grammars,” in *Proceedings of the 11th International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 1225–1229.

[11] D. Prusa and V. Hlavác, “Mathematical formulae recognition using 2d grammars,” in *Proceedings of the 9th International Conference on Document Analysis and Recognition*, vol. 2. IEEE, 2007, pp. 849–853.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[13] C. Faure and Z. X. Wang, “Automatic perception of the structure of handwritten mathematical expressions,” *Computer Processing of Handwriting*, pp. 337–361, 1990.

[14] M. Okamoto, S. Sakaguchi, and T. Suzuki, “Segmentation of touching characters in formulas,” in *Document Analysis Systems: Theory and Practice*. Springer, 1998, pp. 151–156.

[15] F. Álvaro and J. A. Sánchez, “Comparing several techniques for offline recognition of printed mathematical symbols,” in *Proceedings of the 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 1953–1956.

[16] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori, “Infty: an integrated ocr system for mathematical documents,” in *Proceedings of the 2003 ACM Symposium on Document Engineering*. ACM, 2003, pp. 95–104.

[17] I. T. Phillips, “Methodologies for using uw databases for ocr and image-understanding systems,” in *Proceedings of the 1998 Photonics West Electronic Imaging*. International Society for Optics and Photonics, 1998, pp. 112–127.

[18] X. Lin, L. Gao, Z. Tang, J. Baker, and V. Sorge, “Mathematical formula identification and performance evaluation in pdf documents,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 3, pp. 239–255, 2014.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[21] R. Girshick, “Fast r-cnn,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[22] L. Huang, Y. Yang, Y. Deng, and Y. Yu, “Densebox: Unifying landmark localization with end to end object detection,” *arXiv preprint arXiv:1509.04874*, 2015.

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *arXiv preprint arXiv:1312.6229*, 2013.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.