# A Deep Learning Technology based OCR Framework for Recognition Handwritten Expression and Text

**Tuanji Gong, Xuanxia Yao**
*School of Computer & Communication Engineering,University of Science and Technology Beijing, Beijing, China*

***Abstract***

*Recently Optical character recognition (OCR) based on deep learning technology has achieved great advance and broadly applied in various industries. However it still faces many challenging problems in handwritten text recognition and mathematical expression recognition, such as handwritten Chinese recognition, mixture of printed and handwritten Chinese characters, mathematical expression (ME), chemical equations. In traditional OCR, features selection played a vital role for recognition accuracy, while hand-crafted features are costly and time-consuming. In this paper, we introduce a deep learning based framework to detect and recognize handwritten and printed text or math expression. The framework consists of three components. The first component is DCN (Detection & Classification Network), which based on SSD model to detects and classify mathematical expression and text. The second component consists of text recognition and ME recognition models. The final component merges multiple outputs of the second stage into a whole text. Experiment results show that our framework achieves a relative 10% improvement in mixture of texts and MEs which are printed or handwritten in images. The framework has been deployed for recognition paper or homework at one online education platform.*

***Keywords:*** *deep learning,handwritten text, mathematical expression, OCR framework, online education*

## Ⅰ. Introduction

Optical character recognition (OCR) system allows us to convert content in image into electronic text, which we can edit and search. OCR has been broadly applied in many fields, such as identification card, check, license plate, and street number. In online education scenario, various applications employ OCR technique such as taking photograph to search questions and automatic scoring system. Taking photograph to search questions is a widespread used application which user can get similar questions and answers by taking photograph for a question on paper and search similar exercises. Automatic scoring system employs OCR technique to recognize offline handwritten answers. OCR in education scenario involves text, mathematical expression, chemical equation,as well as printed character and handwritten character.

In terms of input image, OCR is mainly divided into two categories: recognition of scanned document image and natural image. In term of data collection, handwriting character recognition is divided into two classifications: on-line handwriting character recognition and off-line recognition. In the offline character recognition case, the input is an image, while in the online character recognition domain it need recognize a sequence of strokes along the pen trajectory. In this work we focus on offline domain.

In online education scenario most images are scanned document images or camera images, so we focus on recognition of scanned document images or camera images. Since the first generation OCR system invented in 1960s, it has been studied more than 50 years. Traditional OCR system has two phases: detection and recognition. Typical detection technology used image filter and transform methods to process image. Recognition method usually includes hand-craft feature selection and a classification model to output text.

ME recognition is a branch of OCR and has significant difference with normal text recognition. Unlike normal text where only horizontal relationship is involved, ME has a left-right and up-down structure, including subscript, superscript, square root symbol, integral symbol etc. Early mathematical expression recognition used manual selection feature to feed a classification model, such as SVM, to recognize expression. The main shortcoming is that hand-crafted features are difficult and time-consuming and suffers from poor performance.

Recently deep learning technology has been achieved great success in many domains, such as  computer vision, speech recognition, and machine translation [1], and many researcher adopted deep learning technique into OCR research. Motivated by prominent performance of SSD model in image classification [2], Liao et al. proposed a fast text detect network based on SSD model, called Textbox [3], for detecting scene text in natural image. In mathematical expression recognition domain, attention-based encoder-decoder models were  employed to recognize mathematical expression in image  [4,5].

Although OCR has achieved high accuracy in scanned document images, it still suffers from low accuracy in some scanned or photo images such as homework assignment or test exam which contain printed, handwritten text and mathematical expressions. As ME recognition has distinct difference with normal text recognition, a single model hardly achieves good performance for text recognition and ME recognition simultaneously. For example, CRNN [6] model achieved high accuracy on text recognition, but it obtained poor accuracy on ME recognition. The model of image to latex [4] is good at ME recognition, not  text recognition as well.

To mitigate the issue, we propose a unified framework which consists of three components. The first component is a detection and classification network based on SSD model, which can not only detect text and mathematical expression but also classify text or ME. The second component is recognition stage which contains two recognition models, one is for text recognition, the other for ME recognition. Patch images with labels outputted by the first component are dispatched into corresponding recognition models according to label value, for example, patch images with text label are recognized by CRNN model [6] and patch images with ME label by Im2Latex [4]. The third component is the mergence stage which merges output texts or latex codes of second component into a whole text.

The rest is organized as follow. Section 2 introduces related work. We present our model and experiments in Section 3 and 4 respectively. In Section 5 we draw a conclusion. Our contributions are as follows: (1) We introduce a classification model to detect mathematical expression or text and handwritten or printed characters; (2) We design a unified framework to detect and recognize text/mathematical expression.

## Ⅱ. Related Work

OCR system usually consists of two stages: detection stage and recognition stage. From the view of input image, it is divided into two domains: scanned document recognition and scene text recognition. In former recognition, text detection is not too difficult to deal with and some traditional pre-processing methodsachieved high performance, which includes noise removal, binarization, skew detection, text line segmentation, character segmentation, and word segmentation [7]. Text-line detection methods consist of connected component labeling, X-Y cut algorithm [8], Run-length smearing, and Hough transform. Other text detection methods have used texture segmentation [9] and statistical properties of local image neighborhoods for the location of text in image [10].  In mathematical expression detection, expression extraction method based Parzen window and 2-D structures [11] was used widely. Chu et al. employed sign-detection  method to detect non-homogeneous regions and used additional features with centroid fluctuation information to detect text line [12]. Although above methods obtained high accuracy, but they need take a number of hand-crafted features and suffered from poor robustness.

OCR systems employing deep learning technology have achieved the state of the art in detection and recognition in natural image. These methods normally used convolution neural network (CNN) to extract feature [13-16] and employed RNN with CTC or attention based model to recognize text. A number of text detection models are based on object detection model. In [17], the authors proposed a text detection model based on the YOLO network [18] to identify text regions. TextBox [3], TextBox++ [19], and SegLink [20] methods are based on SSD-framework.

Detection and recognition of ME in image is an active research branch and attracts a lot of researcher [4, 5, 11, 12, 21-23]. It consists of mathematical expression detection and extraction, symbol recognition, analysis of page layout, and mathematical content explanation etc. Early ME recognition approaches based on trees [24] analyzed data structures, and approaches based on grammar[25] employed formal grammars and corresponding parsing algorithms to recognize math expression. Alavro et al. proposed a statistical framework employing two-dimensional stochastic context-free grammars [29]. Recently many deep learning based methods were proposed, which employed CNN to extract features and used attention-based model [4, 5, 26] or LSTM [27] model to recognize ME.

## Ⅲ. Framework

### 3.1 Pipeline framework

The framework consists of three-stage pipelines illustrated in Fig.1. The first stage is detection and classification network(DCN), which takes the image as input and detects the location of text box and classifies types of each text box. Each text box is one of three types: normal text, mathematical expression, and background. The outputs of DCN are fed into corresponding recognition models, for example, patches with text flag are dispatched to text recognition model based on CRNN, and patches with ME flag are sent to ME recognition model based on WYGIWYS [5]. After recognition, texts or LateX codes of patch are merged into a whole paragraph ordered by coordinate of patches from top to bottom and left to right.

### 3.2 Detection and classification network

#### 3.2.1 Architecture

The idea of DCN is the divided and conquer strategy and the task of DCN takes the image as input, and detects and classifies ME or text sub-regions, with labels of input image. The architecture of model is depicted in Fig.1.

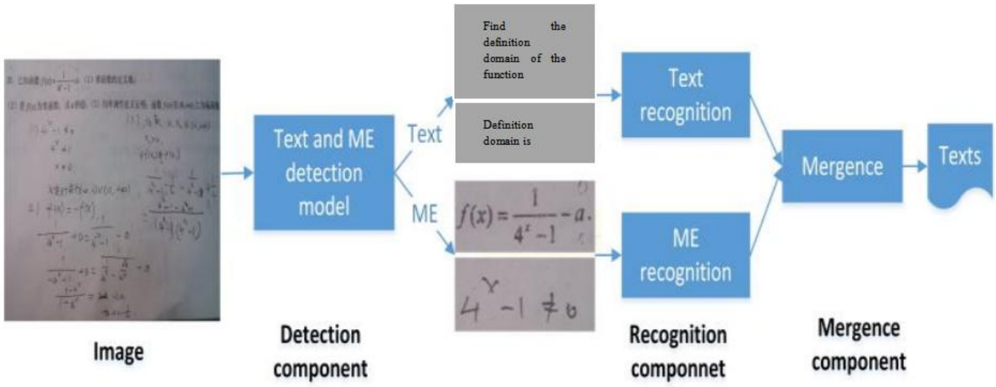

***Fig1: Architecture of framework. The architecture consists of three components: detection and classification component, recognition component and mergence component***

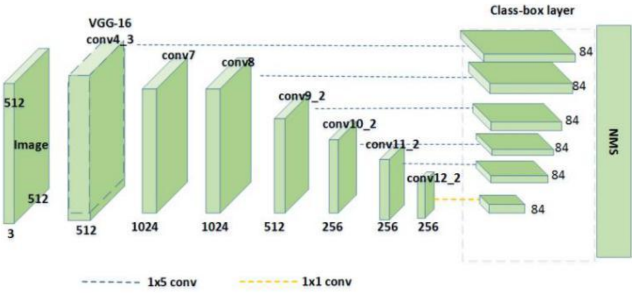

*Fig 2: The architecture of  Detection and Classification Network(DCN)*

DCN is a 28-layer fully convotional network. The kernel with blue line is 1x5 and one with yellow line is 1x1. On every feature map, a class-box layer predicts 84-d vector, which are the text/math express scores (3-d) and offsets(4-d) for 12 default boxes.

The proposed model inherits the popular VGG-16 architecture [28].The last two fully-connected layers  in VGG-16 are changed to convolution layers, followed by several convolutionand pooling layers. Because words and mathematical expressions in OCR images are usually horizontal and long objects. Following [3], we experiment several size convolution kernel, such as 1x5, 3x5,  standard 3x3 kernel, to find out the best kernel for the special scenarios.

Following the last and intermediate convolution layers, there are 6 class-box layers, which are used to detect various size feature map. These outputs are concatenated and passed through non-maximum suppression process.

3.2.2 Class-box layer

A class-box layer simultaneously predicts text or math expression presence, printed or handwritten character, and bounding boxes, conditioned on its input feature map, as shown in Fig.3. At every map location, it outputs two scores of text or math express and offsets to its associated default boxes in a convolution manner.  The structure of class-box layer is illustrated in Fig.3. In class-box layer we add an extra component to predict handwritten or printed character. The class-box layer consists of three components: detection of text or mathematical expression (TME), and detection of text box. Every component has two convolution layers and the convolutionkernel is 3x3. The output channel of last convolution layer for TME and HWP is $n_{anch} \times 3$ and is $n_{anch} \times 4$ for text box, where $n_{anch}$ is the number of anchors.

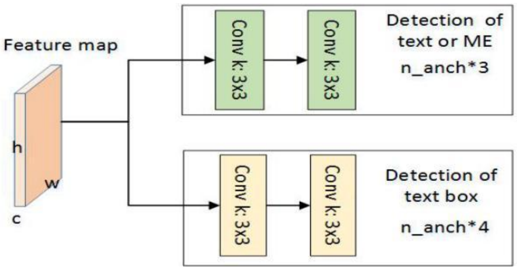

*Fig 3: The network structure of class-box layer*

Feature map is fed into three convolution neural networks to detect text/ME, and coordinate of text or ME box. n_anch is the number of anchors.

3.2.3 Default boxes and aspect ratios

In OCR, words or mathematical expressions tend to have large, long aspect ratios. Similar to [19], we set 10 aspect ratios for default boxes, ranging from 1 to 10. As some images of mathematical expressions, such as fraction expression, have small aspect ratios with less than 1, we add two extra 1/2 and 1/3 default box.

### 3.2.4 Non-maximum suppression

NMS is applied to the aggregate outputs of all class-box layers to locate text or math expression.

### 3.2.5 Optimization

We extend the loss function in [3] with an extra loss item. Let $x$ be match indication matrix, and $c$ be the confidence. The objective loss function is made up of localization loss ( $L_{loc}$ ) and confidence loss ( $L_{conf}$ ), as follow:

$$L(x,c,l,g)=\frac{1}{N}(L_{conf}(x,c)+\alpha L_{loc}(x,l,g)) \tag{1}$$

Where $N$ stands for the amount of default boxes that match ground-truth boxes. $L_{loc}$ is the L1 regular loss [29] between recognized box( $l$ ) and truth box( $g$ ), $L_{conf}$ is the soft-max losses with 3 classes(text, math expression, and background class). In this work, smoothing control factor $\alpha$ is empirically set to 1.

### 3.3 Recognition model

In our framework, we adopt CRNN as text recognition and WYGIWYS [5] as ME recognition model. CRNN model is end-to-end trainable model combining CNN and RNN and can recognize arbitrary length sequences. We train two CRNN models: one is trained using printed text dataset, and the other using dataset with handwritten text.

WYGIWYS model integrates a convolution network and LSTM network with an attention mechanism for text and layout recognition. It uses a multi-layer CNN and a bidirectional LSTM to encode ME images and employs a LSTM with recurrent attention as the decoder to output the markup with LaTeX format.

### 3.4 Mergence stage

After recognition by various recognition models, outputs of various patches are merge into a whole text. The merge rule is as follows: detected patches are sorted by square summation of coordinate (SSC) of top-left point of the patch, e.g., $x^2+y^2$. If SSC of two patches are equal, sort again by x coordinate firstly and then by y coordinate.

## Ⅳ. Experiments

To evaluate effectiveness of our proposed method, we conduct experiments on a synthetic dataset and compare experiments with open resource OCR system and commercial OCR product in test dataset.

### 4.1 Synthetic Dataset

As far as we know, there is no existed dataset which annotates text bound box and text labels such as normal text or ME, handwritten or printed format, so we synthesize a dataset corresponding to the training condition. Synthetic dataset consists of two types of dataset, one is printed texts or math expressions, and the other is handwritten texts or expressions. An image is usually made up of multi-line texts and several mathematical expressions, and a line has one of three layouts:

(1) a line only includes normal texts
(2) a line only includes math expressions
(3) a line is made up of texts and math expressions.

Every image sample has a corresponding annotation file, and one line in it annotates a text block or math expression with 6 items segmented by white-space. The first two items are the left-top (x,y) coordinate of ground region, and the next two items are height and width of the block or expression, the fifth item is the class indicating texts with 1 or  expression with 2. The last item is the label with indicating printed format or handwritten format.

In synthetic dataset, the size of image is 512x512 and every image has 5-10 lines. Texts are sampled from files downloaded from Internet including novel, essay, book and every text's length ranges from 5 to 25. Printed text are rendered with one in 6 fonts, size of font are randomly selected from 20 to 30. The Image of Handwritten Chinese characters are sampled from CASIA offline HWDB1.1 dataset. Printed mathematical expressions are sampled from IM2LATEX-100K dataset and handwritten mathematical expressions are picked from CROHME dataset.

The synthetic dataset contains 1 million images with 50% handwritten format vs 50% printed format and is split into training set with 70%, validation set with 10% and test set with 20%.

### 4.1.1 Exam-paper dataset

The dataset contains 600 scanned images of handwritten exams of algebra subject in high school. Every image contains printed and handwritten texts and MEs and the size is 512x512. The content of every image has been transcript into digital text or Latex code. The dataset is divided into training set with 70%, validation set with 10%, and test dataset 20%.

## 4.2 Implementation details

### 4.2.1 DCN training

For training DCN model, we adopt Adam[30] method to optimize the model. The momentum value is 0.9. In pre-training stage, we trained our model with 200k epochs. The learning rate is set to 0.005, and it is decreased to 1/10 at every 20k iterations. In fine-tuning stage, the initial learning rate is set to 0.001, and then be reduced to 0.0001 at the 50kth iterations and terminated at 100k iterations. We use Dropout method to avoid over-fitting.

The experiment is conducted in Tensorflow framework on Geforce RTX 2080 GPU. The whole training time takes about 50 hours.

### 4.2.2 CRNN

The implementation of CRNN is based on Tensorflow implementation. The size of input is 200x32 and sequence length is 50. For printed dataset, after 200K epochs, the precision of word is 99.8% and sentence's precision is 87.2% , and for handwritten character, the precision of word is 90.3% and  sentence's precision is 68.4%.

### 4.2.3 WYGIWYS

We adopt the Tensorflow implementation of WYGIWYSand use the same training strategy. For IM2LATEX-100K dataset, the precision is 73.8%, and for CROHME dataset the precision is 57.24%.

## 4.3 Evaluation metric

Precision is the primary metric for OCR and F-measure is a complex metric that balances precision and recall rate and is widely applied measurement of text detection, so the evaluation metric use f-measure and precision. F-measure is defined as $F_1 = 2\frac{P \cdot R}{P+R}$ , where $P, R$ stands for precision and recall respectively.  Levenshtein distance is a metric for measuring the similarity between two sequences, as Eq. (3).

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & if\ \min(i,j)=0, \\ \min\begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise. \end{cases} \tag{3}$$

We adopt Levenshtein distance as measure metric to evaluate performance of recognition of mathematical expression.

## 4.4 Experiment and results

In the following we conduct three experiments to evaluate DCN performance and our framework performance.

### 4.4.1 Experiment I

In OCR system, the precision is the important evaluation metric. However, if patch is fed into wrong recognition model, it impairs the whole precision. In this experiment, we evaluate the performance of DCN component.

**Table 1 DCN performance.**

| **Component** | **Text or ME** | **Precision** | **Recall** | **F1** |
|---|---|---|---|---|
| Classification of Text or ME | Printed character | 0.91 | 0.84 | 0.83 |
| | HW character | 0.81 | 0.72 | 0.76 |

We train the model on training dataset and evaluate on test dataset. After training DCN model with synthetic dataset, we transfer Exam-paper dataset to retrain and evaluate it on Exam-paper test set. The precision for printed Text or ME is 0.91, recall is 0.84 and F1 is 0.87. For handwritten text, precision, recall, and f-measure are 0.81, 0.72, 0.76 respectively. The detail is shown in Table 1.

In order to evaluate the impact of kernel size for detecting text and math expression, we conduct three different convolution kernel with 1x5, 3x3, and 3x5 (height x width). We pick up printed text or math expression as test dataset. The printed text test dataset splits two sub-dataset: text dataset and math expression dataset. For text patch, the 1x5 kernel obtains the most performance, and for math expression, the kernel with 3x5 has the most performance in accuracy. The details are shown in Table 2. The possible reason is that text images in OCR training and test datasets largely have large aspect ratio, the 1x5 kernel bears widely receptive field and captures long text patches. As some math express patches, e.g. fractional expression, have small aspect ratio, 3x5 kernel balance between math expressions with big aspect ratios and those with small ratios.

**Table 2 Performance of different kernel size.**

| **Kernel Size(hxw)** | **Math Expression** | **Normal Text** |
|---|---|---|
| 1x5 | 0.88 | 0.97 |
| 3x3 | 0.85 | 0.92 |
| 3x5 | 0.90 | 0.95 |

### 4.4.2 Experiment II

In the experiment, we compare the performance of our pipeline framework with Tesseract and InftyReader. The test dataset use printed text or math expression as Tesseract (4.0) and InftyReader do not have model trained with handwritten characters and expressions.

Test dataset. We use Exam-paper test set to compare three models.

In the mixture input, we compare recognition performance with Tesseract and InftyReader. The results are shown in Table 3. In printed text images with multiple line Chinese texts, comparing to Tesseract, the precision of our framework is 0.95 with relatively 7% increase. For ME recognition, the test dataset contains 50% printed MEs and 50% handwritten MEs. The precision of our framework achieves 0.75 while InftyReader precision is 0.67.

**Table 3 Comparison proposed framework with Tesseract and InftyReader**

| Model | Text/ME | Precision |
|---|---|---|
| Tesseract | Printed character | 0.89 |

| | | |
|---|---|---|
| Our framework | Printed character | 0.95 |
| InftyReader | Math expression | 0.67 |
| Proposed framework | Math expression | 0.75 |

4.4.3 Experiment III

In this experiment, we compare the performance of framework with DCN and without DCN component. In our experiment, we evaluate and compare text recognition and ME recognition. For text recognition, the two models are DCN+2CRNN model with DCN and CRNN-M without DCN. The two CRNN models in DCN+2CRNN model are trained with 20K epoch by handwritten dataset and printed dataset respectively. The CRNN-M model is trained with 40K epoch by combination of handwritten and printed dataset. For ME recognition, there are two models as well as text recognition: DCN+2WYGIWYS model and WYGIWYS-M. The two WYGIWYS models in DCN+2 WYGIWYS model are trained with 20K epoch by CROHME dataset and IM2LATEX-100K respectively. The WYGIWYS-M model is trained with 40K epoch by combination of CROHME and IM2LATEX-100K. The results are shown in Table 4.

**Table 4 Analysis of performance with DCN component and without DCN.**

| Model | Description | Precision |
|---|---|---|
| DCN+2CRNN | Two models trained by handwritten and printed dataset respectively | 0.88 |
| CRNN-M | One model trained by mixture dataset | 0.79 |
| DCN+2WYGIWYS | Two split trained model | 0.77(Exact Match) |
| WYGIWYS-M | Mixture trained model | 0.71(Exact Match) |

For text recognition the precision of DCN+2CRNN model achieves 0.88 with an 11.3% relative increase than CRNN-M model which precision is 0.79. For mathematical expression recognition, comparison with WYGIWYS-M model, the precision of DCN+2WYGIWYS model is 0.77 and has a relative 8.4% increase.

## Ⅴ. Conclusion

In this work, we propose a multiple-stage deep learning based framework to detect and recognize printed and handwritten text or math expression. The framework consists of three components: DCN component, recognition component, and mergence component. The detection and classification stage employs a DCN component to detect text box and classify text or expression format in scanned documented image. The detected and classified patches are fed into corresponding recognition model and output strings of text or Latex code. After recognition, multiple strings are merged into a whole text.

In the future we will explore the end-to-end trainable network to detect and recognize text and mathematical expression.

## Acknowledgment

The work was done in Beijing sanhao company. We would like to thank Weize Zhang for helpful discussions. We also thank Yan Yan and Wei Ma for helpful comments.

## References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436-444, 2015.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, et al., "SSD: Single Shot MultiBox Detector," in European Conference on Computer Vision, 2016.

[3] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A Fast Text Detector with a Single Deep Neural Network," 2016.

[4] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, "Image-to-Markup Generation with Coarse-to-Fine Attention," 2017.

[5] Y. Deng, A. Kanervisto, and A. M. Rush, "What You Get Is What You See: A Visual Markup

Decompiler," 2016.

[6] B. Shi, X. Bai, and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, pp. 2298-2304, 2017.

[7] E. Borovikov, "A survey of modern optical character recognition techniques," arXiv: Computer Vision and Pattern Recognition, 2014.

[8] G. Nagy and S. C. Seth, "HIERARCHICAL REPRESENTATION OF OPTICALLY SCANNED DOCUMENTS," in international conference on pattern recognition, 1984.

[9] V. Wu, R. Manmatha, and E. M. Riseman, "Finding Text in Images," 1997, pp. 3-12.

[10] P. Clark and M. Mirmehdi, "Finding Text Regions Using Localised Measures," in british machine vision conference, 2000, pp. 675-684.

[11] J. Jin, X. Han, and Q. Wang, "Mathematical Formulas Extraction," in International Conference on Document Analysis & Recognition, 2003.

[12] W. T. Chu and F. Liu, "Mathematical Formula Detection in Heterogeneous Document Images," in Conference on Technologies and Applications of Artificial Intelligence, 2013, pp. 140-145.

[13] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep Features for Text Spotting," in european conference on computer vision, 2014, pp. 512-528.

[14] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading Text in the Wild with Convolutional Neural Networks," International Journal of Computer Vision, vol. 116, pp. 1-20, 2016.

[15] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting Text in Natural Image with Connectionist Text Proposal Network," european conference on computer vision, pp. 56-72, 2016.

[16] C. Bartz, H. Yang, and C. Meinel, "STN-OCR: A single Neural Network for Text Detection and Text Recognition," 2017.

[17] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic Data for Text Localisation in Natural Images," computer vision and pattern recognition, pp. 2315-2324, 2016.

[18] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," computer vision and pattern recognition, pp. 779-788, 2016.

[19] M. Liao, B. Shi, and B. Xiang, "TextBoxes++: A Single-Shot Oriented Scene Text Detector," IEEE Transactions on Image Processing, vol. 27, pp. 3676-3690, 2018.

[20] B. Shi, X. Bai, and S. J. Belongie, "Detecting Oriented Text in Natural Images by Linking Segments," computer vision and pattern recognition, pp. 3482-3490, 2017.

[21] W. He, Y. Luo, Y. Fei, H. Han, J. Han, E. Ding, et al., "Context-aware mathematical expression recognition: An end-to-end framework and a benchmark," in International Conference on Pattern Recognition, 2017.

[22] C. Liu, L. Zuo, X. Li, and X. Tian, "An improved algorithm for identifying mathematical formulas in the images of PDF documents," in IEEE International Conference on Progress in Informatics and Computing, 2016, pp. 252-256.

[23] J. Zhang, J. Du, and L. Dai, "Multi-Scale Attention with Dense Encoder for Handwritten Mathematical Expression Recognition," 2018.

[24] R. Zanibbi, D. Blostein, and J. R. Cordy, "Recognizing Mathematical Expressions Using Tree Transformation," IEEE Trans.pattern Anal.machine Intell, vol. 24, pp. 1455-1467, 2002.

[25] S. Lavirotte, "Mathematical Formula Recognition Using Graph Grammar," Spie Proceedings, vol. 3305, pp. 44-52, 1998.

[26] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, et al., "Watch, Attend and Parse: An End-to-end Neural Network Based Approach to Handwritten Mathematical Expression Recognition," Pattern Recognition, vol. 71, pp. 196-206, 2017.

[27] A. D. Le and M. Nakagawa, "Training an End-to-End System for Handwritten Mathematical Expression Recognition by Generated Patterns," in international conference on document analysis and recognition, 2017, pp. 1056-1061.

[28] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image

Recognition," international conference on learning representations, 2015.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 39, pp. 1137-1149, 2015.

[30] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," international conference on learning representations, 2015.