ISBN: 978-981-18-3959-7
2022 the 12th International Workshop on Computer Science and Engineering (WCSE 2022)
doi: 10.18178/wcse.2022.06.030

Mathematical Expression Character Recognition Based on Object Detection

Chunmei Xing [1], Jun Yue [1+] and Chunjie Zhou [1]

[1] School of Information and Electrical Engineering, Ludong University, Yantai, 264025, China

Abstract. The recognition accuracy of mathematical characters is one of the key factors affecting mathematical expression recognition. The Traditional mathematical expression recognition consist of three stages which are character segmentation, character recognition and structural analysis. Such methods are cumbersome and with the error accumulation. To solve this problem, a method to use object detection technology combining the two steps of character segmentation and character recognition in expression recognition is proposed in this paper, by which the expression is simplified and the accuracy of character recognition is improved. Firstly, a dataset (PME) containing 130 character categories and a total of 2000 formula pictures is established; and then, the SE (squeeze and excitation) block is integrated into the backbone network (CSPDarkNet53) to perform feature recalibration; finally, three sizes feature maps are obtained to predict different size characters respectively. The experiments results show that mAP (mean average precision) is 98.62% on PME, which is 1.52%, 8.95% and 4.5% higher than that of benchmark model YOLOv4, contrast model UP-DETR and BiDet, which proves the effectiveness of this method.

Keywords: mathematical character recognition, CNN, attention, object detection.

1. Introduction

The development of online education is particularly rapid as the impact of COVID-19(Coronavirus disease 2019). Online education contains a large number of mathematical expressions, most expressions are saved in fixed formats that cannot be edited or reuse such as pictures. Therefore, the research of mathematical expression recognition is of great practical significance to achieve resource sharing and online learning.

Early mathematical expression recognition was not mature enough and largely relied on predefined grammar. In 1968, Anderson [1] first proposed the recognition of mathematical expression, and he proposed the concept of expression recognition based on syntax orientation. In the 1980s, researchers began to decompose the mathematical expression recognition into small problems and solve them one by one. Okamoto [2] divided expression recognition into image scanning, symbol segmentation, symbol recognition, and structural analysis, etc. Lin [3] used the global threshold method to segment the characters, and then used the template matching method to identify the split characters.

With the development of neural networks, researchers began to apply neural networks [4] [5] to the recognition of mathematical expression. Fang [6] improved the distinguishability of features in a supervised learning manner by constructing a joint loss, expanded inter-class differences and reduced intra-class similarity. Kang [7] proposed a human-in-the-loop solution for the uncertainty of expression characters and structural complexity of handwritten mathematical expressions. Fu [8] proposed a single-point sticking symbol segmentation method based on the feature of character's outline for the character sticking problem, and the final sticking character segmentation accuracy can reach 87.25%.

At present, expression recognition has made great progress, but there are still problems such as cumbersome steps and poor recognition, so the recognition method needs to be further improved.

+ Corresponding author. Tel.: +86-13562559603; fax: +0535-6681245.
E-mail address: yjzcqxcm@163.com.

2. Related Works

Attention Mechanism. In general terms, attention models [9] [10] are roughly divided into spatial attention models and channel attention models. Each region in the image isn't with equal importance in the spatial attention model, so the key of the model is to find the most important parts in the image and process them [11]. While the channel attention model requires that the neural network can intelligently judge the importance of different channels. Based on SENet [12], the backbone network CSPDarkNet53 integrates the SE block deeply to locate and recognize characters in the mathematical expression in this paper.

Object Detection. Object detection is an important branch of computer vision, which focuses on finding important content in the picture and determine their locations and categories. Object detection has been widely used in image classification, semantic segmentation, etc. The object detection technology is adopted in expression recognition to achieve character segmentation and character recognition, simultaneously, the recognition of expressions is simplified.

3. SE-YOLOv4 Method

In this paper, YOLOv4 with faster processing speed is selected as the benchmark network. On this basis, SE-YOLOv4 is proposed in this paper to improve the accuracy of the network to meet the requirements of expression character recognition. The technical route of this paper is shown in the upper of Fig. 1 and the lower is an explanation of the modules used in the upper.

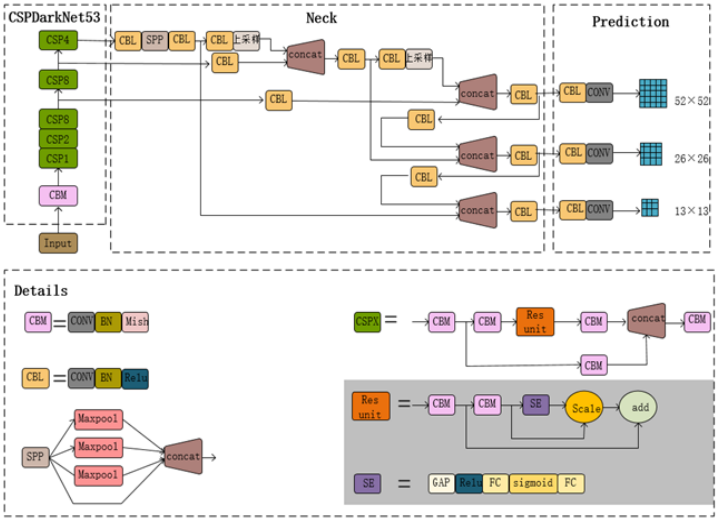

Fig. 1: Technology roadmap.

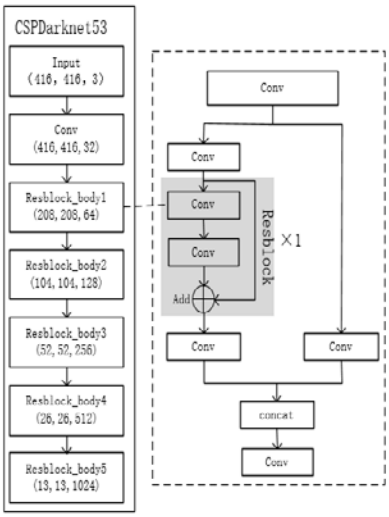

Fig. 2: CSPDarkNet53 structure.

As shown in the dotted box on the left in Fig. 1, CSPDarkNet53 extracts features from the input. The *X* in green *CSPX* can be replaced by 1, 2, 8, 8, 4. In order to recalibrate features, SE block is integrated into the CSPDarkNet53, which is shown in the shaded part in the lower right, and the integrated details of SE block are shown in Fig. 3. Then the CSPDarkNet53 outputs three different sizes feature maps which are send to *Neck*. The *Neck* handles the feature information with upsampling, etc., as shown in the dotted box in the middle. At last, the information generated by the *Neck* is sending to *Prediction* for predict, as shown in the dotted box on the right.

The structure of CSPDarkNet53 is shown in Fig. 2. CSPDarknet53 is mainly composed of five Resblock_bodies, and each Resblock_body contains several Resblocks. The internal structure of the first Resblock_body is shown in the dashed box on the right. The overall structure of the five Resblock_bodies are same, only the number of Resblocks is different. From top to bottom, the number of resblocks contained in resblock_body is 1, 2, 8, 8, 4 respectively.

3.1. SE Block

SE block consists of squeeze operation and excitation operation. It is a computing unit, and it can obtain the weight of each channel automatically by learning, which shows the importance of each channel. Then

valuable features are emphasised and ones with less value are suppressed according to the obtained weight. SE block can be built upon any given transformation:

$$F_{tr}:X\rightarrow U,X\in R^{H'\times W'\times C'},U\in R^{H\times W\times C} \tag{1}$$

Where the F_{tr} is convolution operator, X represents the input, U represents the output, H',W',C' represents the input's height, width and number of channels respectively. H, W, C represents the output's height, width and number of channels respectively.

Suppose $M=[m_1,m_2,\cdots,m_c]$ is the learned set of filter kernels, where m_c denotes the parameters of the c-th filter. Next the output after convolution transformation can be written as $U=[u_1,u_2,\cdots,u_c]$, where

$$u_c=m_c*X=\sum_{t=1}^{c'} m_c^t*x^t \tag{2}$$

Where * represents convolution, c' represents the number of channels of input, $X=[x^1,x^2,\cdots,x^{c'}]$, $m_c=[m_c^1,m_c^2,\cdots,m_c^{c'}]$, m_c^t is a two-dimensional convolution kernel which means that a single channel of m_c acts on the channel of the corresponding input X. For the convenience of calculation, the bias is omitted here, so the output is generated by adding all channels, and the correlation between channels is implicitly embedded in m_c.

- Squeeze: D is obtained by shrinking U according to its spatial dimensions, the c-th element d_c of D is calculated by :

$$d_c=\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}u_c(i,j) \tag{3}$$

  Where $d_c\in R^c$.

- Excitation: In order to utilize the information obtained in the above squeeze operation, an excitation operation is performed next which aims to capture the dependencies between channels. The specific formula is shown in formula (4).

$$T=\sigma(W_2\delta(W_1,D) \tag{4}$$

  Where δ represents the Relu function, σ represents the Sigmoid function, $W_1\in R^{\frac{C}{r}\times C}$, $W_2\in R^{C\times\frac{C}{r}}$, C represents the number of channels, r is the reduction rate , taking the value 1 /16.

  The final output $\tilde{X}=[\tilde{x_1},\tilde{x_2},\cdots,\tilde{x_c}]$ is obtained by rescaling U by the function T：

$$\tilde{x_c}=F_{scale}(u_c,t_c)=t_cu_c \tag{5}$$

  Where $F_{scale}(u_c,t_c)$ represents the channel-wise multiplication between the scalar t_c and the feature map $u_c\in R^{H\times W}$.

3.2. SE-YOLOv4 Module

SE block is integrated into CSPDarknet53, we take the first Resblock_body as an example to illustrate the improved network structure, which is shown in Fig. 3.

The input will be calculated along the left and right branches respectively. The right branch performs a convolution operation with a convolution kernel size of 1 × 1, the number of convolution kernels is 64. Then it will be concatenated with the results on the left branch, that is, *Concat* operation in Fig. 3. The left branch performs a convolution firstly, and then the *Resblock* module is entered. In this module, two convolution operations are performed in sequence, and then the SE block is entered to calculate the channel-wise weight. Formula 5 will be adopted to process the obtained weights and the convolutional results, that is, the *Scale* operation in Fig. 3. The result of the *scale* is added with the result of the first convolutional on the left to form a skip connect, that is, *Add* operation in Fig. 3. Then a fourth convolution operation is performed, and the result will be concatenated with the convolution result on the right. Finally the last convolutional

operation will be carried out. The purpose of operations above is to modify the number of channels. The overall structure and various parameters are shown in Fig. 4.

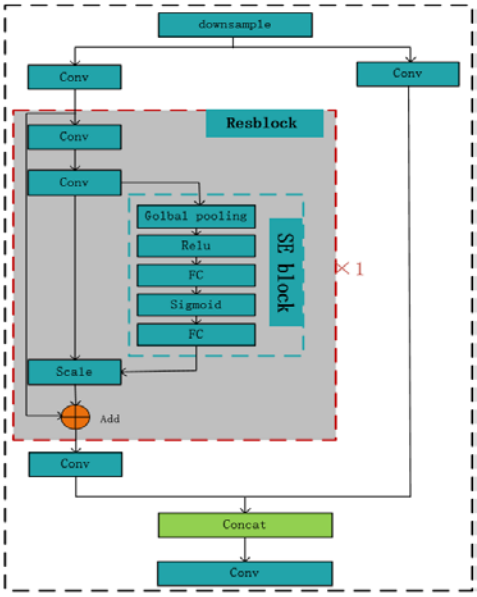

Fig. 3: Improved CSPDarknet53 structure.

Type	Filters	Size	Output	
Conv	32	3×3/1	416×416×32	
Conv	64	3×3/2	208×208×64	
CrossStagePartial				
Conv	32	1×1/1	208×208×32	
Conv	64	3×3/1	208×208×64	
Global Pooling	–	–	1×1×64	
FC	–	–	1×1×4	
FC	–	–	1×1×64	
Resdual	–	–	208×208×64	×1
Conv	128	3×3/2	104×104×128	
CrossStagePartial				
Conv	64	1×1/1	104×104×64	
Conv	64	3×3/1	104×104×64	
Global Pooling	–	–	1×1×64	
FC	–	–	1×1×4	
FC	–	–	1×1×64	
Resdual	–	–	104×104×128	×2
Conv	256	3×3/2	52×52×256	
CrossStagePartial				
Conv	128	1×1/1	52×52×128	
Conv	128	3×3/1	52×52×128	
Global Pooling	–	–	1×1×128	
FC	–	–	1×1×8	
FC	–	–	1×1×128	
Resdual	–	–	52×52×128	×8
Conv	512	3×3/2	26×26×512	
CrossStagePartial				
Conv	256	1×1/1	26×26×256	
Conv	256	3×3/1	26×26×256	
Global Pooling	–	–	1×1×256	
FC	–	–	1×1×16	
FC	–	–	1×1×256	
Resdual	–	–	26×26×256	×8
Conv	1024	3×3/2	13×13×1024	
CrossStagePartial				
Conv	512	1×1/1	13×13×512	
Conv	512	3×3/1	13×13×512	
Global Pooling	–	–	1×1×512	
FC	–	–	1×1×32	
FC	–	–	1×1×512	
Resdual	–	–	13×13×1024	×4

Fig. 4: Network parameter information.

4. Experiment and Result Analysis

4.1. Dataset Creation and Processing

In order to better identify mathematical characters, a print mathematical expression dataset PME is created in this paper, it contains a total of 130 categories which are Arabic numerals 0-9, small letters a-z, caption letters A-Z, etc., as shown in Table 1. PME dataset has a total of 2000 images, images are manually annotated with *LabelImg*. The split ratio of training set, validation set and test set is 7:1:2. All expressions are generated by manually writing screenshots. Some dataset pictures are shown in Fig. 5.

Table 1: Character category summary table

Categories	Characters
Small letter	a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
Capital letter	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
Arabic numerals	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Greek alphabet	α, β, π, ω, γ, δ, ρ, τ, ε, θ, μ, ∑, Δ, φ, σ
Calculating symbol	+, -, =, ÷, ×, ∫, ∩, ∩, /, *, ±
Relational symbol	≤, ≥, >, <, ≠, ≈,≫,≪ , ≅
Brackets	(,), [,], {, }
Others	——, !, ', ., ,, ∃, ⊂, ∇, ℃,℉, ∋,∈,∝, ϑ, √, ∂, ∞, ∀, ∏, |, ~, ∅, %, ⋯, ϵ, ≡, :

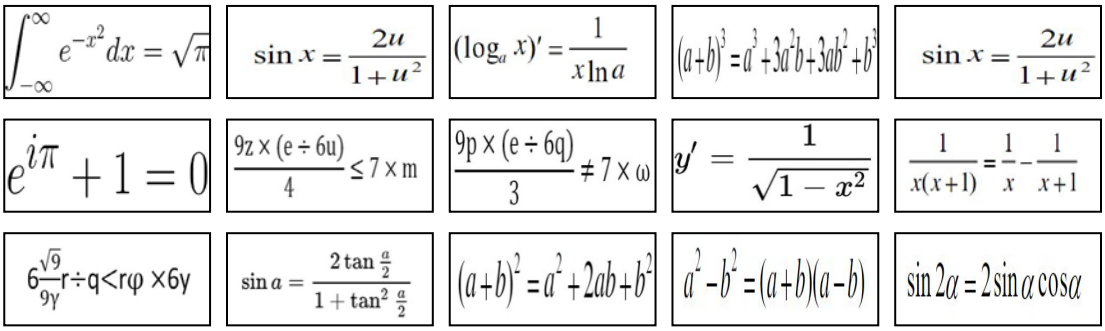

Fig. 5: Some dataset image examples.

4.2. Experiments Results and Analysis

During training, the input images are scaled to 416×416 uniformly. In order to reduce the over-fitting phenomenon and help the model to converge better, methods of mosaic, etc., are used to augment data in real time. The model will make predictions on three sizes feature maps: 52×52, 26×26, and 13×13, aiming to better detect characters of different sizes in the image. At last, the model performs non-maximum suppression operations to retain the most accurate prediction boxes. The network was trained by Adam optimizer, the model is optimized with an initial learning rate of 0.001 and 8 mini-batches.

In order to verify the effectiveness of SE_YOLOv4, the experiment result of SE_YOLOv4 is compared with the two models of BiDet[13] and UP-DETR[14], which is shown in Table 2.

Table 2: mAP of different systems on PME

Models	mAP (%)
UP-DETR	89.67
BiDet	94.12
YOLOv4	97.10
SE-YOLOv4 (ours)	98.62

As shown in Table 2, SE-YOLOv4 achieves mAP of 98.62% on the PME dataset. Compared with YOLOv4[15], the mAP is improved by 1.52%. Compared with UP-DETR and BiDet, the mAP is increased by 8.95% and 4.5% respectively. The experiments results show that the effect of the SE-YOLOv4 is better than the other two models, which verifies the effectiveness of the method in this paper.

Partial character recognition results are shown in Table 3.

Table 3: Partial mathematical character recognition accuracy

Category	AP(%)
!	93.27
%	100.00
*	100.00
+	99.36
,	100.00
-	96.67
/	100.00
1	100.00
3	99.67
<	97.43
A	100.00
J	98.14
[	95.91
e	100.00

Part of the visualization recognition results are shown in Fig. 6~Fig. 11. As we all know, the complex two-dimensional structure of the expression brings a lot of difficulties to the expression recognition. But in this paper, the expression's fractional structure(Fig. 9, Fig. 10, Fig. 11), square root structure(Fig. 8), subscript structure(Fig. 6, Fig. 7) and equal sign structure (Fig. 6, Fig. 7) have been recognized better by SE-YOLOv4 than other models . The effectiveness of this method is proved.

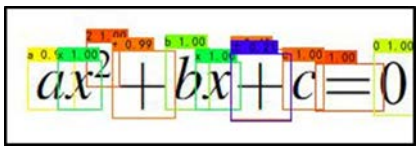

Fig. 6: Visualization results 1

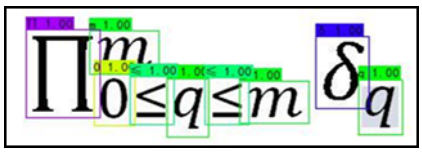

Fig. 7: Visualization results 2

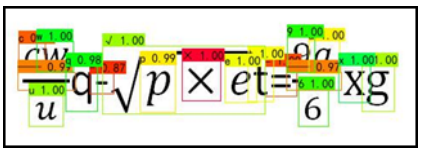

Fig. 8: Visualization results 3

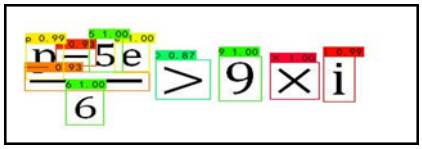

Fig. 9: Visualization results 4

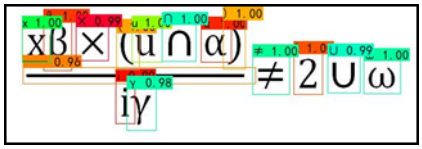

Fig. 10: Visualization results 5

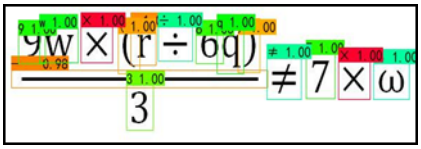

Fig. 11: Visualization results 6

5. Conclusions

This paper proposes the SE-YOLOv4 method, which simplifies the steps of mathematical expression recognition and improves the accuracy of character recognition. Firstly Dataset PME which contains 130 character categories and a total of 2000 pictures is established. Then, the object detection is adopted to mathematical expression recognition, omit character splitting, and simplify expression recognition. Next the SE block is integrated into CSPDarkNet53 to complete the extraction of complex features. Finally, the proposed model achieves a recognition precision of 98.62% on the PME test set, which proves the invalid of our method. In future work, we will supplement the dataset to contain more characters and research the overall structure of the expression, strive to recognize the expression which saved in image into editable expression.

6. Acknowledgements

The work was supported by the Yantai Key R&D Project (2019XDHZ2084,2018ZDCX003).

7. References

[1] R.H. Anderson. Syntex-directed recognition of hand-printed two-dimensional mathematics[A]. *Interactive Systems for Experimental Applied Mathematics[C]. New York: Academic Press*, 1968: 436-459.

[2] M. Okamoto, H. Imai, K. Takagi. Performance Evaluation of a Robust Method for Mathematical Expression Recognition[C]. *International Conference on Document Analysis and Recognition. IEEE Computer Society*, 2001: 438-446.

[3] Yanran Lin, and Lihong Yang. Research on Recognition of Printed Mathematical Formulas[J].*Data mining*,2020, 10(2):14.

[4] Li Y , Lin S , Liu J , et al. Towards Compact CNNs via Collaborative Compression[J]. 2021.

[5] Yin M , Sui Y , Liao S , et al. Towards Efficient Tensor Decomposition-Based DNN Model Compression with Optimization Framework[J]. 2021.

[6] D. Fang , and C. Zhang. Multi-Feature Learning by Joint Training for Handwritten Formula Symbol Recognition[J]. *IEEE Access*,2020,PP(99):1-1.Stroke order normalization for improving recognition of online handwritten mathematical expressions, 2019.

[7] Wenhui Kang, Jin Huang, Feng Tian , Xiangmin Fan, Jie Liu, Guozhong Dai. Human-in-the-Loop Based Online Handwriting Mathematical Expressions Recognition[J/OL].*Journal of Computer-Aided Design & Computer Graphics*: 1-14[2021-11-28].http://kns.cnki.net/kcms/ detail/11.2925.tp.20211007.1934.006.html.

[8] Pengbin Fu , Jianjun Li, Huirong Yang. Handwritten Formula Recognition Based on Segmentation of Adhesive Symbols and Multi-feature Fusion[J]. *Journal of Beijing University of Technology*,2021,47(08):842-853.

[9] Vaswani A , Ramachandran P , Srinivas A , et al. Scaling Local Self-Attention For Parameter Efficient Visual Backbones[J]. 2021.

[10] Hou Q , Zhou D , Feng J . Coordinate Attention for Efficient Mobile Network Design[J]. 2021.

[11] M. Jaderberg , K. Simonyan, A. Zisserman. Spatial transformer networks[C]//*Advances in neural information processing systems*, 2015:2017-2025.

[12] H. Jie, S. Li, S. Gang. Squeeze-and-Excitation Networks[C]// *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, 2018.

[13] Z. Wang, Z. Wu, J. Lu, et al. BiDet: An Efficient Binarized Object Detector[J]. *IEEE*, 2020.

[14] Z. Dai , B. Cai, Y. Lin, et al.UP-DETR: Unsupervised Pre-training for Object Detection with Transformers[J]. 2020.

[15] A. Bochkovskiy , C.Y. Wang , H. Liao.YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. 2020.