CCLSL: Combination of Contrastive Learning and Supervised Learning for Handwritten Mathematical Expression Recognition

Qiqiang Lin[1][0000−0002−8587−1651], Xiaonan Huang[1][0000−0001−8292−8862], Ning Bi[1,2][0000−0002−6296−9947], Ching Y Suen[3][0000−0003−1209−7631], and Jun Tan[1,2][0000−0002−2281−5599]

[1] School of Mathematics Computational Science, Sun Yat-sen University, Guangzhou, 510275, P.R.China
[2] Guangdong Province Key Laboratory of Computational Science Sun Yat-Sen University, Guangzhou, 510275, P. R. China
[3] Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, QC, H3G 1M8, Canada
{linqq9,wangchy53}@mail2.sysu.edu.cn
{mcsbn,mcstj}@mail2.sysu.edu.cn
suen@encs.concordia.ca

Abstract. Handwritten Mathematical Expressions differ considerably from ordinary linear handwritten texts, due to their two-dimentional structures plus many special symbols and characters. Hence, HMER(Handwritten Mathematical Expression Recognition) is a lot more challenging compared with normal handwriting recognition. At present, the mainstream offline recognition systems are generally built on deep learning methods, but these methods can hardly cope with HEMR due to the lack of training data. In this paper, we propose an encoder-decoder method combining contrastive learning and supervised learning(CCLSL), whose encoder is trained to learn semantic-invariant features between printed and handwritten characters effectively. CCLSL improves the robustness of the model in handwritten styles. Extensive experiments on CROHME benchmark show that without data enhancement, our model achieves an expression accuracy of 58.07% on CROHME2014, 55.88% on CROHME2016 and 59.63% on CROHME2019, which is much better than all previous state-of-the-art methods. Furthermore, our ensemble model added a boost of 2.5% to 3.4% to the accuracy, achieving the state-of-the-art performance on public CROHME datasets for the first time.

Keywords: Handwritten · Semantic invariant · Contrastive learning.

1 Introduction

Handwritten mathematical expression recognition (HMER) is more challenging than other handwritten forms such as handwritten digits and words [1–4] because handwritten mathematical expressions (HMEs) do not only have different writing styles but also include a large number of mathematical symbols, complex two-dimensional structures and the limitation of small trainable datasets.

The traditional grammar-based HMER model is divided into three steps [5–7] : symbol segmentation, symbol recognition, and structural analysis. However, it does not bring satisfication in the recognition of handwritten mathematical formulas with complex two-dimensional structures.

With continuous improvement of computing capability, deep learning has attracted more and more attention and reaped fruitful results in search technology, machine learning, machine translation, natural language processing, multimedia learning, recommendation and personalization technology, and other related fields. Ha et al. [8] firstly applied neural networks to recognize individual characters and symbols, and then Ramadhan et al. [9] established the convolutional neural network model to recognize mathematical formula symbols. Hai et al. [10] proposed a combination of convolutional neural networks and Long Short-Term Memory (LSTM) to effectively identify online and offline handwritten characters. However, these methods can only recognize single characters. Bahdanau et al. [11] proposed an Encoder-Decoder architecture framework based on attention mechanism, which made a significant breakthrough in machine translation. And then the Encoder-Decoder framework has gradually been applied to the field of mathematical expression recognition(MER).

Zhang et al. [4] firstly applied the Encoder-Decoder framework in the field of mathematical expression recognition with a proposal of an end-to-end offline recognition model, referred to as ”watch, attend and parse (WAP)”. Different from the previous models, WAP uses the attention mechanism to automatically segment symbols, so that the input HMEs images are modeled with the output of one-dimensional character sequences in LaTeX format. The original WAP model employs a fully convolutional networks(FCN) encoder and a recurrent neural network decoder using gated recurrent units(GRU) equipped with an attention mechanism as the parser to generate LaTeX sequences. Subsequently, Zhang et al. [12] further improved the WAP model, using DenseNet [14] network as the encoder, and proposed multi-scale attention model which solve the problem of mathematical symbol recognition well.

Truong et al. [15] proposed a weakly supervised learning method based on WAP model, which assisted the encoder to extract more useful high-level features by adding a symbol classifier near the encoder. In terms of the improvement of decoder, ”Bidirectionally Trained TRansformer (BTTR)” [16] replaces the GRU network decoder by a bi-directional Transformer decoder, and alleviates lack of coverage by employing positional encodings. An Attention aggregation based bi-directional Mutual Learning Network (ABM) is proposed by Biao et al. [17] to better learn complementary context information. Truong et al. [18] put forward a relation-based sequence representation, which reduced the ambiguity caused by the use of ”_” and ”{ }” and enhanced the recognition of offline handwritten mathematical expressions (HMEs) by reconstructing structural relations. Zhang et al. [19] proposed a tree-structured decoder to improve the decoding ability of dealing with complicated mathematical expressions.

Moreover, to improve the robustness of the recognizer with respect to writing styles, Wu et al. [20] proposed a novel paired adversarial learning method to extract semantic-invariant features. Le et al. [21] proposed the model of dual loss attention, which has two losses including decoder loss and context matching loss, in order to detect the semantic invariant features of the encoder from handwritten and printed mathematical expressions and improve the performance of LaTeX grammar for the encoder. However, there is a big difference in distribution between printed and handwritten MEs, so these methods cannot learn semantic-invariant features effectively. Therefore, we propose to explore how to effectively make full use of easily generated printed mathematical expressions (PMEs) to improve the recognition accuracy of HEMR model. Inspired by contrastive learning [32, 22–25], this paper proposes a new method based on BTTR model: a combination of self-supervised contrastive learning and supervised learning to enable the encoder to learn the semantic-invariant features between printed and handwritten. The contributions of this paper are listed below:

- With reference to self-supervised contrastive learning, we apply contrastive learning to feature extraction in printed and handwritten, so that the encoder can learn semantic-invariant features between the two different forms, and the extracted features are learned as similar as possible.
- Considering the case of shared encoder, large batchsize and large-scale datasets are required for contrastive learning, otherwise it is difficult to gain ideal results. In this paper, hybrid contrastive learning and supervised learning methods are employed to ensure the correct updating of parameters in the training process, and also guarantee the encoder learns semantic-invariant relationship between PMEs and HMEs.
- Extensive experiments on various CROHME benchmarks show that our method on both a single model and an ensemble model outperform state-of-the-art results.

2 Related Work

Contrastive learning. In recent years, contrastive learning has set off a wave of interest in the field of computer vision (CV). Models based on the idea of contrastive learning, such as MoCo [23], SimCLR [22], MoCov2 [24], SimSiam [25], emerge one after another. As a self-supervised representation learning method, contrastive learning has outperformed supervised learning in some tasks of CV. The main idea of contrastive learning is to narrow the distance between positive samples and expand the distance between negative samples. A pair of positive samples is usually obtained by two different random transformations of the same image. A typical model among them is SimCLR as shown in Fig. 1. SimCLR employs ResNet as the base encoder f(.) and add nonlinear projection head g(.), mapping representation to the space where the contrastive loss is applied. Inspired by these papers, we apply a contrastive learning architecture to handwritten mathematical expression recognition(HMER).

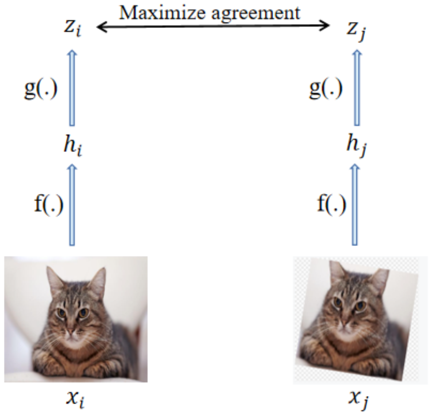

Fig. 1. The structure of SimCLR

BTTR. BTTR [16] uses DenseNet [12] as the encoder and transformer decoder as the decoder, which can perform both left-to-right (L2R) and right-to-left (R2L) decoding. The training phase is achieved by generating two target sequences (L2R and R2L) from the target LaTeX sequence, and computing the training loss for the same batch. Approximate joint search [27] is used during inference to improve recognition performance. This article improves on BTTR and uses it as the baseline.

3 Method

In this paper, we propose a method combining self-supervised contrast learning and supervised learning(CCLSL). HMEs come from CROHME training set, and we use Python and LaTeX provided by CROHME training set to generate images of printed mathematical expressions (PMEs). On a batchsize, we assume that there are N pairs of paired samples, each of which contains HMEs image and PMEs image of the same size and the same label, denoted as $x^{pair} = (x^h, x^p)$,and the corresponding label is marked Y^{pair}.

Our handwritten expression recognition system is shown in Fig. 3. CCLSL contains two parts, one is the encoder-decoder model based on supervised learning; the other part is the self-supervised contrastive learning model, which adds a projection head g(.) to the encoder for maximizing the similarity between corresponding printed and handwritten pixel features in the space where contrastive loss is applied. We define parameters of encoder block and decoder block as θ_e and θ_d respectively. Encoder-decoder parameters and projection head parameters are defined as θ and θ_g. In the given training set $D = (x^{pair}, Y^{pair})$, θ is updated by maximizing probability of prediction while (θ_e, θ_g) is updated by

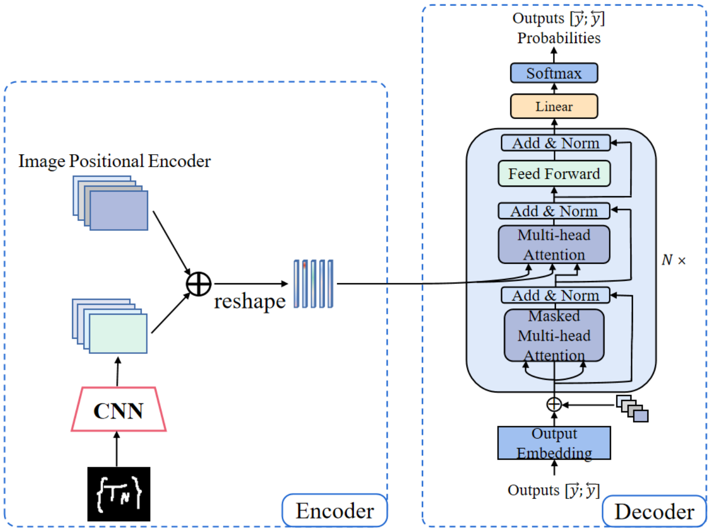

Fig. 2. The architecture of BTTR model. L2R and R2L sequences $[\overrightarrow{y};\overleftarrow{y}]$ are concatenated through the batch dimension as the input to the decoder.

maximizing the similarity of the corresponding pixel features of each printed and handwritten MEs in the contrastive space.

3.1 Encoder

In the encoder block, CNN network is used as the feature extractor of HMEs image, which is composed of DenseNet[14] and a 1×1 convolutional layer. The role of the last convolution layer is to adjust the size of image features to the embedded dimensions d model for subsequent processing. Image pair $x^{pair} = [x^h; x^p]$ is processed through CNN networks to obtain feature map $h^{pair} = [h^h; h^p]$, which is added to 2-D image positional encodings $E^{pair} = [E^P; E^p]$ to obtain feature map with positional information, and then is flattened to 1-D feature map $f^{pair} = [f^h; f^p]$, where $x^{pair} \in R^{2\times H\times W\times 1}$, $h^{pair} \in R^{2\times H'\times W'\times d}$, $E^{pair} \in R^{2\times H'\times W'\times d}$, $f^{pair} \in R^{2\times H'W'\times d}$.

3.2 Transformer Decoder

In the decoder part of this article, we use the standard transformer decoder[26], it consists of N Transformer Decoder Layers, each layer contains three parts: Multi-Head Attention, Masked Multi-Head Attention, Feed-Forward Network.

Multi-Head Attention. Multi-head attention is concatenated from single-head attention. For a given Q, K, V, we compute the head in the projected

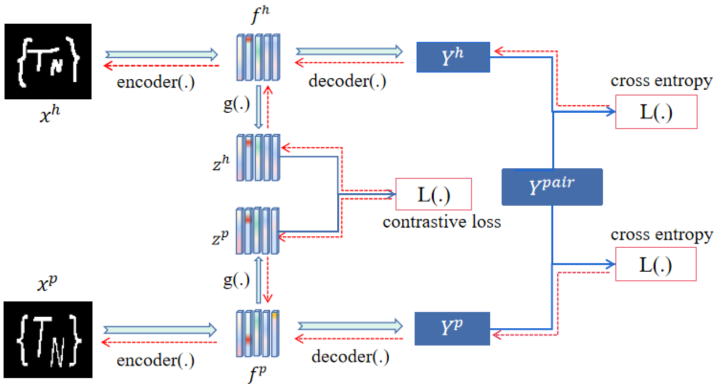

Fig. 3. A hybrid system of self supervised contrastive learning and supervised learning

subspace by utilizing the scaled dot-product attention module.

$$H_i = \frac{(QW_i^Q)(KW_i^K)^T}{\sqrt{d_{model}}}(VW_i^V) \ . \tag{1}$$

where $W_i^Q \in R^{d_{model} \times d_q}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$, represent the projection matrix.

After that, the h heads are concatenated and projected through the projection matrix $W^O \in R^{hd_v \times d_{model}}$ to get the new feature vector:

$$multihead = [H_1; H_2; ...; H_h]W^O \ . \tag{2}$$

Masked Multi-Head Attention. In the process of decoding, the information of future moment cannot be obtained at the current moment, so it is necessary to use the mask technique to cover the information of the future moment during training process.

Feed-Forward Network. FFN is a fully connected network including two linear transformations and a nonlinear function, where the nonlinear function generally adopts the relu activation function.

3.3 Supervised Training

Referring to BTTR[16], we apply a bidirectional training strategy for supervised learning. First, two specific symbols "SOS" and "EOS" were introduced into the dictionary to indicate the beginning and end of a sequence. For a given paired label $Y^{pair} = \{y_1, y_2, ..., y_T\}$ it is represented as $\overrightarrow{Y^{pair}} = \{"SOS", y_1, y_2, ..., y_T, "EOS"\}$ from left to right (L2R) and $\overleftarrow{Y^{pair}} = \{"EOS", y_T, y_{T-1}, ..., y_1, "SOS"\}$ from right to left, where y_i represents mathematical symbols and T is the length of the LaTeX sequence symbols. Considering that the transformer does not actually care about the order of the input symbols, we can use a single transformer decoder for bidirectional language modeling. We use cross-entropy as the objective function, conditioned on the image x^{pair} and the encoder-decoder parameter θ, to maximize the probability of the predicted symbols of the bidirectional target LaTeX sequence.

$$\mathcal{L}_{CE}\left(Y^{pair}|x^{pair}\right) = \frac{1}{2}\Bigg(\mathcal{L}\left(Y^{pair}|x^{p}\right) + \mathcal{L}\left(Y^{pair}|x^{h}\right)\Bigg) \ , \tag{3}$$

$$\mathcal{L}\left(Y^{pair}|x\right) = \frac{1}{2T}\left(\sum_{j=1}^{T} log\, p\left(\overrightarrow{y_j} \mid \overrightarrow{y_{<j}}, x\right) + \sum_{j=1}^{T} log\, p\left(\overleftarrow{y_j} \mid \overleftarrow{y_{<j}}, x\right)\right) \ . \tag{4}$$

3.4 Contrastive Training

In order to enable the encoder to effectively learn the same semantic-invariance features between printed and handwritten MEs and improve the robustness of this model in writing style, self-supervised contrastive learning is introduced to supervised learning. Inspired by the idea of SimCLR[22], we add a projection head after the encoder to map the representation to the space where the contrastive loss is applied. The projection head g(.) will consist of an MLP with one hidden layer:

$$z^{p} = g(f^{p}) = \sigma\left(W^{g}\sigma\left(f^{p}\right)\right) \ . \tag{5}$$

$$z^{h} = g(f^{h}) = \sigma\left(W^{g}\sigma\left(f^{h}\right)\right) \ . \tag{6}$$

where σ stands for ReLU nonlinearity, $\theta_g = W^g \in R^{d\times d}$.

As for $z^p = [z^p_1, z^p_2, ..., z^p_{H'W'}]$ and $z^h = [z^h_1, z^h_2, ..., z^h_{H'W'}]$, A contrastive loss function[31] enables the corresponding positional features of z^p and z^h as close as possible:

$$\mathcal{L}_{CL}\left(x^{pair}\right) = \mathcal{L}_{CL}\left(x^{p}, x^{h}\right) \ , \tag{7}$$

$$\mathcal{L}_{CL}\left(x^{p}, x^{h}\right) = \sum_{i=1}^{H'W'}\left(\mathcal{L}_{NCE}\left(z^{p}_{i}, z^{h}_{i}; z^{p}\bigcup z^{h}\right) + \mathcal{L}_{NCE}\left(z^{h}_{i}, z^{p}_{i}; z^{p}\bigcup z^{h}\right)\right) \ . \tag{8}$$

where $\mathcal{L}_{NCE}(.)$ is a contrastive loss function, called InfoNCE[32]:

$$\mathcal{L}_{NCE}\left(u, v_{+}; U\right) = -log\frac{exp(u^{T}\cdot v_{+}/\tau)}{\sum_{v\in U\setminus u} exp(u^{T}\cdot v/\tau)} \ . \tag{9}$$

where u, v, v_+ are l_2 normalized.

3.5 Combination of Contrastive Learning and Supervised Learning

According to SimCLR, under the condition of not introducing memory bank, the pre-training effect can only be acceptable if the batch size is large enough. However, the available HMEs training data are insufficient to support mass training. Therefore, a new method is proposed in this paper: Skip the pre-training stage of contrastive learning, and directly carry out the combination of self-supervised contrastive learning and supervised learning. The specific operation is to minimize the hybrid loss function:

$$\mathcal{L}_{hybrid} = \mathcal{L}_{CE}\left(Y^{pair}|x^{pair}\right) + \lambda\mathcal{L}_{CL}\left(x^{pair}\right) \ . \tag{10}$$

where λ is a hyperparameter that controls the tradeoff between decoder loss $\mathcal{L}_{CE}$ and contrastive loss $\mathcal{L}_{CL}$.

In the inference phase, after discarding the projection head g(.), our model is capable of recognizing both HMEs and PMEs. Similar to BTTR, the decoder employs approximate joint search [27] to improve decoding performance.

4 Experiments

4.1 Experimental Setup And Results

Table 1. Performance of the BTTR as baseline system on CROHME 2014, CROHME 2016 and CROHME 2019.

Model	2014 ExpRate	2016 ExpRate	2019 ExpRate
BTTR [16]	53.96	52.31	52.96
baseline	55.68	53.44	55.46

Experimental Setup. We evaluated our method on the CROHME Competition dataset[28–30].The training set selected in this paper is CROHME2014 training set, which contains 8836 HMEs pictures in total. We use Matplotlib library to generate corresponding PMEs. CROHME2014 test set containing 986 images, CROHME2016 test set containing 1147 images, and CROHME2019 test set containing 1199 images are employed to test the performance of the model. And we employ Expression Rate (ExpRate) metrics to evaluate HMEs recognition systems. Adadelta algorithm with gradient shear is chosen to learn parameters with batch size set to 15, batch image size set to 440000 and max epochs set to 200. When the expression rate on the validation set does not increase after 30 epochs, the learning rate is set up to decrease. During training $\tau = 2$ is used to soften the output distribution. The model is trained on two NVIDIA 2080Ti GPUs with 11 × 2 GB memory.

Experimental Results. First of all, we rerun related work BTTR[16] as the baseline system. Table 1 shows the results of test on CRHOMR2014, CRHOMR2016 and CRHOMR2019. It is worth mentioning that the open source code reproduction results provided by BTTR are a bit better than their paper results, so we adopt the rerun model as the baseline model.

Table 2. Performance comparison of offline HMER systems on CROHME test sets. * Refers to the ensemble of recognition models utilizing multiple different initializations. In particular, all models listed in the table trained without any augmentation.

Model	2014 ExpRate	2016 ExpRate	2019 ExpRate
Single Model			
WAP [13]	48.38	46.82	–
WS-WAP [15]	53.65	51.96	–
PAL-v2 [20]	48.88	49.61	–
Dual loss attention [21]	51.88	51.53	–
DenseWAP-TD [19]	49.1	48.5	51.4
ABM [17]	56.85	52.92	53.96
SAN [34]	56.2	53.6	53.5
BTTR [16]	53.96	52.31	52.96
baseline	55.68	53.44	55.46
CCLSL(our)	**58.07**	**55.88**	**59.63**
Ensemble Model			
WS-WAP* [15]	55.68	52.57	–
PAL-v2* [20]	54.87	57.89	–
DenseWAP-TD* [19]	54.00	52.10	54.60
BTTR* [16]	57.91	54.49	56.88
CCLSL* (our)	**60.61**	**58.32**	**62.97**

In Table 2, we compare our model with other offline HMER systems on the CROHME 2014/2016/2019 test sets respectively. To ensure fairness of comparison, none of the systems employ integration of multiple models. Results show that our model achieves an expression accuracy of 58.07% on CROHME2014, 55.88% on CROHME2016 and 59.63% on CROHME2019, with an improvement of 2.39%/2.44%/4.17% on CROHME2014/2016/2019 compared to the baseline model, and the recognition performance of our model on the three test sets are obviously better than the most advanced method. In addition, under the condition of $\lambda = 0.02$, two single models are retrained with different initializations, and the six single models trained by λ=0.01, 0.02, 0.03, 0.04, 0.05, and 0.06 are simply integrated together. It can be seen from Table 2 that the recognition effect of the integrated model in CROHME test sets also achieves state-of-the-art at present.

Table 3. Performance of CCLSL under different hyperparameters λ.

Dataset		2014	2016	2019
		ExpRate	ExpRate	ExpRate
baseline		55.68	53.44	55.46
λ	0.01	57.76	55.18	58.21
	0.02	58.07	**55.88**	**59.63**
	0.03	57.76	54.84	58.38
	0.04	57.76	55.01	57.88
	0.05	57.05	55.01	59.13
	0.06	**58.17**	55.10	58.79
	0.07	55.23	54.92	57.46

4.2 Ablation experiments

Ablation: superparameter λ. We evaluate the performance of CCLSL under different superparameter λ shown in Table 3. We set 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07 to λ for the experiment. By comparison, we find that the model achieves the best results in all of the three test sets when $\lambda = 0.02$. ExpRates in CROHME 2014/2016/2019 test sets are 58.07%, 55.88%, 59.63%, significantly improved compared with the baseline model, indicating that Combination of contrastive learning and supervised learning(CCLSL) can effectively improve model recognition performance.

Ablation: Different Training Methods. In order to further illustrate the effectiveness of our method, some ablation experiments are performed in table 4. It can be seen from Table 4 that Pre and Mixed have limited performance improvement of the model. It is likely that the two data distributions are quite different, resulting in a far difference in the feature maps extracted by the encoder. The method proposed in this paper can enable the model, especially the encoder, to learn the semantic invariant features of PMEs and HMEs images.

4.3 Encoder Migration

At present, the classic model WAP [12] in the offline handwritten mathematical formula also uses the DenseNet as the encoder and the double-layers GRU network with the attention mechanism as the decoder. In this paper, the encoder trained by the CCLSL method will be migrated to the WAP model for retraining, while using WAP [12], WAP variant WS-WAP [15], ABM [17] as the baseline model.

Table 5 shows that, by transferring the encoder DenseNet trained by CCLSL to the WAP model for fine-tuning, the recognition accuracy of the model is greatly improved, and the performance is also significantly better than other WAP model variants. This further verifies that the encoder trained with CCLSL

Table 4. The performance of BTTR under different training methods. Pre refers to pre-training the BTTR model using PMEs images, and then fine-tuning them in HMEs images. Mixed refers to training models using both PMEs and HMEs images. CCLSL is the method proposed in this paper.

Dataset	Pre	Mixed	CCLSL	ExpRate
CROHME 2014	×	×	×	55.68
	✓	×	×	56.14
	×	✓	×	56.04
	×	×	✓	**58.07**
CROHME 2016	×	×	×	53.44
	✓	×	×	53.87
	×	✓	×	54.34
	×	×	✓	**55.88**
CROHME 2019	×	×	×	55.46
	✓	×	×	55.96
	×	✓	×	55.54
	×	×	✓	**59.63**

Table 5. Performance of encoder migrated to WAP. The V1 model freezes the encoder parameters of WAP and only trains the decoder. V2 uses the cross-entropy function as the loss function on the basis of V1 to fine-tune the encoder and decoder to make them more adaptable, and V3 uses label smoothing[33] as the loss function on the basis of V1 to fine-tune the encoder and decoder to improve the generalization ability of the model.

Model	2014 ExpRate	2016 ExpRate	2019 ExpRate
WAP [13]	48.38	46.82	–
WS-WAP [15]	53.65	51.96	–
ABM [17]	56.85	52.92	53.96
V1(our)	51.47	48.64	49.54
V2(our)	56.54	53.55	55.57
V3(our)	**59.69**	**54.92**	**57.88**

can indeed learn semantically invariant features between print and handwriting, significantly improving the robustness of the model in terms of writing style.

4.4 Visualization

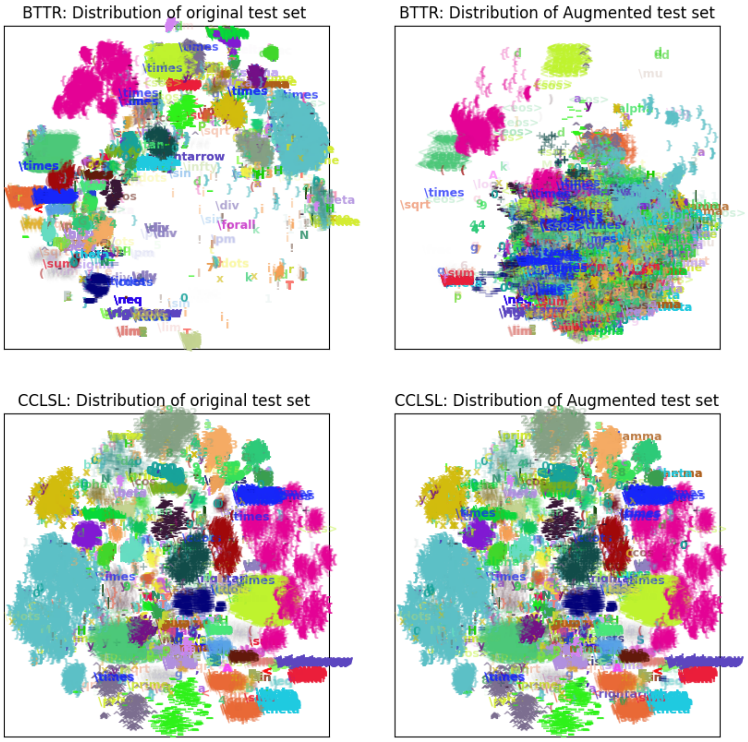

Fig. 4. Visualize the context vectors extracted by the BTTR and CCLSL on the test set and augmented test set.

To visualize the semantically invariant features learned by the proposed system, we show the context vectors for each symbol class in BTTR and CCLSL on the CROHME 2014 test set and the corresponding data augmentation set. The augmented dataset is generated using A general geometric augmentation tool [35] for text images. For visualization, we utilize t-SNE to map the data from high-dimensional to 2-dimensional. It can be observed from Fig. 4 that the character shape changes in the enhanced image, which leads to a large deviation of the context vector extracted by BTTR, and the system proposed in this paper can capture similar context vector representations and learn semantic invariance.

Finally, Fig. 5 illustrates the attention-based decoding process, where mintcream is the background, black is the font, and other color areas are the focus areas of attention. The darker the color, the higher the attention weight. It can be seen that the attention can not only capture the spatial position of each character, but also use the spatial structure information to assist the decoder in parsing symbols '{' and '}'.

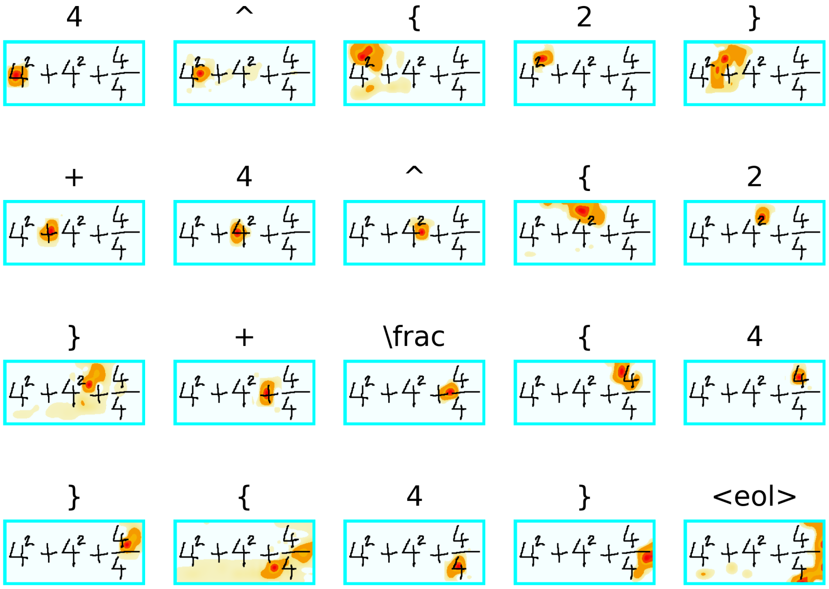

Fig. 5. Visualization of the attention process.

5 Conclusion

In this paper, we have proposed a new method (CCLSL) to effectively recognize offline HMEs: a combination of self-supervised contrastive learning and supervised learning to enable the encoder to learn the semantic-invariant features between PMEs and HMEs, improving the model's robustness in writing style.

Extensive experiments on various CROHME datasets show that our method on both single and integrated models achieved state-of-the-art performance.

Based on the research results of this paper, the future research direction is proposed: mining more latex expressions of mathematical formulas, generating PMEs pictures to participate in the training of the model. Using the method proposed in this paper to ensures that the encoder can learn the semantically invariant features of PMEs and HMEs, the decoder can learn more latex syntax and further improve the performance of the decoder.

Acknowledgments. This work was supported by the Guangdong Provincial Government of China through the Computational Science Innovative Research Team" program and Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University, and the National Science Foundation of China (11971491, 11471012).

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. R. H. Anderson, "Syntax-directed recognition of hand-printed two-dimensional mathematics," in *Symposium on Interactive Systems for Experimental Applied Mathematics: Proceedings of the Association for Computing Machinery Inc. Symposium*, 1967, pp. 436–459.
3. A. Belaid and J.-P. Haton, "A syntactic approach for handwritten mathematical formula recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 105–111, 1984.
4. J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "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.
5. 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.
6. F. Álvaro, J.-A. Sánchez, and J.-M. Benedí, "An integrated grammar-based approach for mathematical expression recognition," *Pattern Recognition*, vol. 51, pp. 135–147, 2016.
7. S. MacLean and G. Labahn, "A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 2, pp. 139–163, 2013.
8. J. Ha, R. M. Haralick, and I. T. Phillips, "Understanding mathematical expressions from document images," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 2. IEEE, 1995, pp. 956–959.
9. I. Ramadhan, B. Purnama, and S. Al Faraby, "Convolutional neural networks applied to handwritten mathematical symbols classification," in *2016 4th International Conference on Information and Communication Technology (ICoICT)*. IEEE, 2016, pp. 1–4.
10. H. Dai, A. Le Duc, and M. NAKAGAWA, "Combination of lstm and cnn for recognizing mathematical symbols," in *Proceedings of the 17th information-based induction sciences workshop*, 2014.

11. D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
12. J. Zhang, J. Du, and L. Dai, “Multi-scale attention with dense encoder for handwritten mathematical expression recognition,” in *2018 24th international conference on pattern recognition (ICPR)*. IEEE, 2018, pp. 2245–2250.
13. J. Wang, J. Du, J. Zhang, and Z.-R. Wang, “Multi-modal attention network for handwritten mathematical expression recognition,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 1181–1186.
14. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
15. T.-N. Truong, C. T. Nguyen, K. M. Phan, and M. Nakagawa, “Improvement of end-to-end offline handwritten mathematical expression recognition by weakly supervised learning,” in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2020, pp. 181–186.
16. W. Zhao, L. Gao, Z. Yan, S. Peng, L. Du, and Z. Zhang, “Handwritten mathematical expression recognition with bidirectionally trained transformer,” in *Document Analysis and Recognition – ICDAR 2021*, J. Lladós, D. Lopresti, and S. Uchida, Eds. Cham: Springer International Publishing, 2021, pp. 570–584.
17. X. Bian, B. Qin, X. Xin, J. Li, X. Su, and Y. Wang, “Handwritten mathematical expression recognition via attention aggregation based bi-directional mutual learning,” *arXiv preprint arXiv:2112.03603*, 2021.
18. T.-N. Truong, H. Q. Ung, H. T. Nguyen, C. T. Nguyen, and M. Nakagawa, “Relation-based representation for handwritten mathematical expression recognition,” in *International Conference on Document Analysis and Recognition*. Springer, 2021, pp. 7–19.
19. J. Zhang, J. Du, Y. Yang, Y.-Z. Song, S. Wei, and L. Dai, “A tree-structured decoder for image-to-markup generation,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 076–11 085.
20. J.-W. Wu, F. Yin, Y.-M. Zhang, X.-Y. Zhang, and C.-L. Liu, “Handwritten mathematical expression recognition via paired adversarial learning,” *International Journal of Computer Vision*, pp. 1–16, 2020.
21. A. D. Le, “Recognizing handwritten mathematical expressions via paired dual loss attention network and printed mathematical expressions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 566–567.
22. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
23. K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.
24. X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” *arXiv preprint arXiv:2003.04297*, 2020.
25. X. Chen and K. He, “Exploring simple siamese representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758.
26. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

27. L. Liu, M. Utiyama, A. Finch, and E. Sumita, "Agreement on target-bidirectional neural machine translation," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 411–416.
28. H. Mouchere, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014)," in *2014 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 791–796.
29. H. Mouchère, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "Icfhr2016 crohme: Competition on recognition of online handwritten mathematical expressions," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2016, pp. 607–612.
30. M. Mahdavi, R. Zanibbi, H. Mouchere, C. Viard-Gaudin, and U. Garain, "Icdar 2019 crohme+ tfd: Competition on recognition of handwritten mathematical expressions and typeset formula detection," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1533–1538.
31. R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.
32. A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
33. R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *Advances in neural information processing systems*, vol. 32, 2019.
34. Y. Yuan, X. Liu, W. Dikubab, H. Liu, Z. Ji, Z. Wu, and X. Bai, "Syntax-aware network for handwritten mathematical expression recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4553–4562.
35. C. Luo, Y. Zhu, L. Jin, and Y. Wang, "Learn to augment: Joint data augmentation and network optimization for text recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 746–13 755.