2022 26th International Conference on Pattern Recognition (ICPR)
August 21-25, 2022, Montréal, Québec, Canada

Primitive Contrastive Learning for Handwritten Mathematical Expression Recognition

Hong-Yu Guo , Chuang Wang , Fei Yin , Heng-Ye Liu , Jin-Wen Wu and Cheng-Lin Liu
National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China
School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
Email: guohongyu2019@ia.ac.cn, {chuang.wang, fyin}@nlpr.ia.ac.cn, liuhengye2019@ia.ac.cn, {jinwen.wu, liucl}@nlpr.ia.ac.cn

2022 26th International Conference on Pattern Recognition (ICPR) | 978-1-6654-9062-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPR56361.2022.9956214

***Abstract*—Contrastive learning has gained significant attention recently as it can learn a representation from a large amount of unlabeled training data to improve downstream tasks. While the existing approaches mainly focus on standard tasks of image classification and object detection, they are not easily applied to structured prediction problems. In this paper, we propose an unsupervised pre-trained model, called PrimCLR, for handwritten mathematical expression recognition. For a formula recognition model of encoder-decoder architecture, a pre-trained representation is obtained by PrimCLR, where the contrastive loss is computed from pairs of patches so as to better discriminate primitives. The pre-trained representation is transferred to downstream formula recognition with supervised fine-tuning. Experiments show that pre-training by PrimCLR can significantly improve the formula recognition performance, and PrimCLR shows superiority to conventional contrastive learning methods. Our model achieves state-of-the-art performance on standard datasets CROHME 2016 and CROHME 2019.**

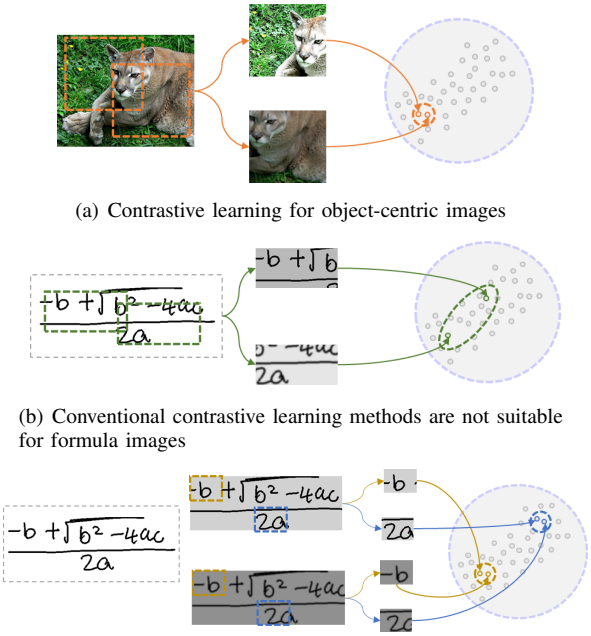
I. Introduction

Self-supervised pre-training model prevails recently in computer vision as it can boost the generalization performance by learning from a large amount of unlabeled data. The paradigm is to first pre-train a visual feature encoder using unlabeled data and then transfer it to downstream tasks by supervised fine-tuning the classifier. The pre-trained representation model can largely improve the generalization ability of multiple downstream tasks such as classification, segmentation and object detection, especially when labeled data is limited.

The major progress of visual pre-training model has been driven by contrastive learning [1]. The aim is to improve the pre-training model to attain better accuracy on linear classification as well as other visual tasks [2], [3], [4], [5].

Successful applications of contrastive learning heavily rely on the datasets with object-centric bias [6], *e.g.*, the curated ImageNet dataset. Little evidence has been reported that contrastive learning is effective for structured prediction problems such as mathematical expression recognition [7], where multiple related objects or primitives in an image need to be classified simultaneously. Learning representation for structured prediction is more challenging as it requires not only discriminative representation but also object localization and contextual reasoning.

Handwritten mathematical expression recognition (HMER) has received considerable attention in intelligent education and office automation. This problem is challenging due to the complex and diverse layout and the writing style variation. It is unsolved despite the numerous progress [8], [9], [10]. Also, for training recognition models, large dataset of labeled mathematical formula images is lacking. Thus, self-supervised pre-training method provides a potential solution to improve the representation model.



(a) Contrastive learning for object-centric images

(b) Conventional contrastive learning methods are not suitable for formula images

(c) Primitive contrastive learning for formula images

Fig. 1. (a) Contrastive learning encodes an object-centric image as a single feature vector to compute the contrastive loss. (b) Conventional contrastive learning methods destroy the structure and neglect the semantic information of formula images. (c) Primitive contrastive learning compares the corresponding primitive patches from different views by a series of mild augmentation.

In this paper, we propose a primitive contrastive learning (PrimCLR) method for representation learning for HMER. The recognition model is an encoder-decoder architecture, following the image-to-markup framework. The encoder consists of a CNN-based image feature encoder and a BiLSTM based contextual encoder. The encoder is pre-trained by PrimCLR on unlabeled images so as to get an initial representation. PrimCLR is different from conventional contrastive learning as the contrastive loss is computed on pairs of patches extracted from the feature map. This effects in improving the discrimination between primitives in images. In addition,

we design the data augmentation strategy containing a set of mild transformations to preserve the structure of formula images, which is destroyed in the default cropping and flipping operations of conventional contrastive learning. See Fig. 1 for details. After self-supervised pre-training, the whole model is fine-tuned in supervised learning to perform the structured recognition task.

In experiments of formula recognition, it is shown that self-supervised pre-training can significantly improve the performance of the baseline image-to-markup model. By pre-training with additional unlabeled data, the performance can be further improved. On the standard benchmark datasets CROHME 2016 and CROHME 2019, our model achieved state-of-the-art performance even equipped with a relatively weak decoder module.

In summary, the main contributions of this work are:

- A structured contrastive learning method, called PrimCLR, is proposed for HMER. It computes contrastive loss on pairs of patches so as to improve the discrimination of primitives.
- We design an encoder-decoder model incorporating self-supervised pre-training and supervised fine-tuning to perform structured formula recognition.
- Our experiments show that the proposed PrimCLR can significantly improve the performance of formula recognition, and our recognition model achieves state-of-the-art performance on standard datasets.

The remaining of this paper is organized as follows. Section 2 reviews related works; Section 3 describes the proposed recognition model with pre-training; Section 4 presents experiment results, and Section 5 draws concluding remarks.

II. Related Works

The related works in visual representation learning and HMER are briefly reviewed in the following.

A. Visual Representation Learning

Early self-supervised visual representation learning methods leverage a variety of pretext tasks to learn useful representation. The pretext tasks explored in previous works include rotation prediction [11], relative location prediction [12], jigsaw puzzle solving [13], colorization [14], [15], etc.

Recently, contrastive learning shows its extraordinary ability to learn discriminative representation without labels. The key ingredient of contrastive learning [1] is to pull together the pairs that belong to the same instance while pushing apart pairs from different instances. By applying aggressive data augmentation to a large amount of images, and comparing them using a contrastive loss, a model can learn feature representation invariant to transformations. MoCo [2], [3] uses asymmetric learning updates in which momentum encoders are updated separately from the main network. BYOL [5] uses a teacher-student distillation framework to improve contrastive learning. Some methods modify contrastive learning to accustom practical problems such as semi-supervised learning [16], keypoint detection [17] and image-to-image translation [18]. Regarding multi-object localization, a few works adapt contrastive learning for object detection and segmentation [19], [20], [21]. More recently, the work [22] considers text sequence parsing with semantic information.

Conventional contrastive learning methods concentrate on object-centric problems with relatively simple structure. For the visual representation of formula images, there are three key ingredients: discriminative feature of symbols (primitives), spatial localization in complex hierarchical structure, and contextual information modeling. Existing methods seldom consider all of the three ingredients. Hence we explore contrastive learning with these ingredients, *i.e.*, contrastive learning for HMER.

B. HMER Methods

The objective of HMER is to convert the input formula images into a structured, machine-readable representation, *e.g.*, LaTeX code or Symbol Layout Tree (SLT) [23]. The structured representation contains both mathematical symbols and their structural relationship. Mainstream solutions to HMER include grammatical methods and image-to-markup methods.

Grammatical methods consider the task of recognizing formula images as a three-step process: symbol segmentation, symbol recognition and structural analysis[24], [25], [26], where the accumulated errors in symbol segmentation and recognition lead to final expression recognition error. Image-to-markup method [27] is a data-driven framework implemented by an encoder-decoder neural network. These methods leverage multiple structures of neural networks, *e.g.*, CNN, RNN, GNN [28] or Transformer based decoder with attention mechanism [29] to generate the target markup languages. The paired adversarial learning (PAL) method [30] uses paired adversarial loss to train the recognizer, and is enhanced by a pre-aware unit in [31]. Several works explore different encoders, such as VGG [32] and multi-branch DenseNet [33], to improve performance. There are also some works utilize data augmentation as means to increase accuracy [34], [35]. Despite the efforts mentioned above, the performance of HMER is still insufficient, and more attention is required to improve the technology.

Our HMER model is built on the original image-to-markup method [27]. We aim to improve the HMER performance by self-supervised pre-training.

III. Recognition Model with Primitive Contrastive Learning

Our formula recognizer is an encoder-decoder architecture, following the image-to-markup framework [27]. The feature representation learning for the encoder is crucial to the final recognition performance. We hence learn an initial representation by self-supervised primitive contrastive learning, and then fine-tune the whole model by supervised learning on labeled data (annotated with LaTeX codes).

A. Encoders and Contrastive Pre-training

The encoder module consists of a CNN-based image feature encoder and a RNN-based contextual encoder. For contrastive

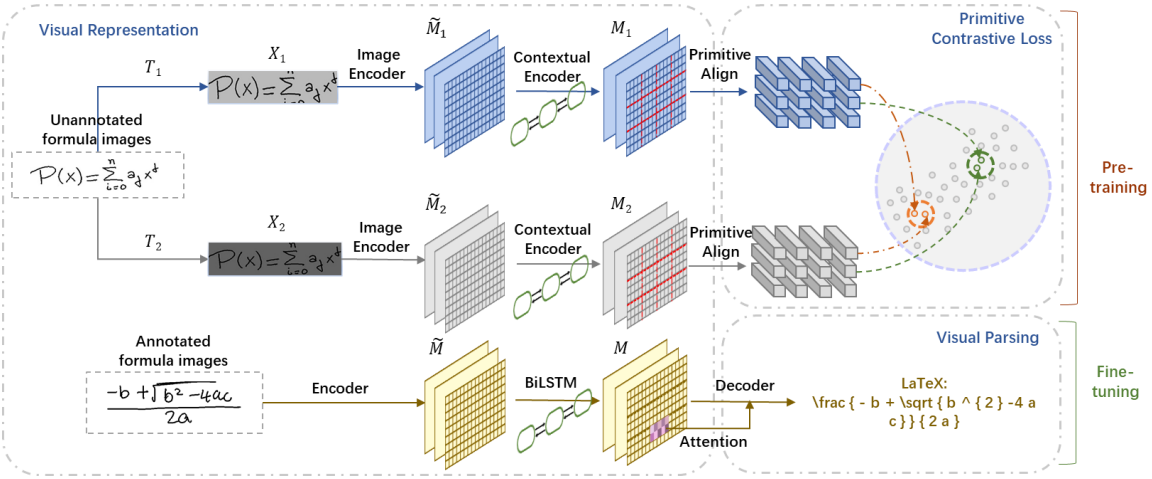


Fig. 2. The overall framework of PrimCLR. For each formula image, two augmented views are generated to go through the base image encoder. Next, the output image features are sent through the BiLSTM contextual encoder to extract contextual information. Finally, the new feature map is split into several primitive patches for calculating the primitive-level contrastive loss. After pre-training by PrimCLR, the encoders and the LaTeX decoder are trained jointly with annotated images.

pre-training, we propose the primitive contrastive loss and design the semantic-invariant data augmentation specialized for formula image encoding.

1) Image Encoder: A multi-layer CNN is used to construct basic visual feature maps of formula images. A batch of input images go through the CNN encoder and output a feature map $\widetilde{M}$ with size $B \times C \times H \times W$, where B, C, H and W represent the batch size, the number of channels, the height and width of feature map, respectively. Input images with different size are scaled to the same height, and their width is aligned using zero-padding.

2) BiLSTM Contextual Encoder: BiLSTM is adopted as a subsequent encoder after the image encoder. The BiLSTM contextual encoder can leverage the contextual information in formula images. The encoded feature map M is computed according to the following rules

$$\begin{aligned} h^{\rightarrow}_{b,(h,w)} &= \mathrm{LSTM}(h^{\rightarrow}_{b,(h,w-1)}, \widetilde{m}_{b,(h,w)}), \\ h^{\leftarrow}_{b,(h,w)} &= \mathrm{LSTM}(h^{\leftarrow}_{b,(h,w+1)}, \widetilde{m}_{b,(h,w)}), \\ m_{b,(h,w)} &= h^{\rightarrow}_{b,(h,w)} + h^{\leftarrow}_{b,(h,w)}, \end{aligned} \tag{1}$$

where $b \in \{1, ..., B\}, h \in \{1, ..., H\}, w \in \{1, ..., W\}$, $h^{\rightarrow}_{b,(h,w)}$ and $h^{\leftarrow}_{b,(h,w)}$ are the hidden states of the BiLSTM, and $\widetilde{m}_{b,(h,w)}, m_{b,(h,w)}$ are pixels in the feature maps of the augmented formula image pair output by image encoder and contextual encoder, respectively.

3) Parallel Contrastive Encoder: The encoder is pre-trained by using contrastive learning following the framework of SimCLR [4]. Specifically, two parallel encoder modules share parameters, and two versions of the formula image with different data augmentation are fed into the modules. In the view of SimCLR, the BiLSTM contextual encoder introduced above can be treated as the projection head of contrastive learning model. Usually, the projection head is discarded in downstream tasks. We choose to reserve its parameters to initialize fine-tuning, which leads to better performance, potentially due to its ability to reserve structured contextual information.

4) Primitive Contrastive Loss: In contrast to a regular classification image usually containing only one object, a formula image contains multiple symbols with hierarchical relations. It will lose the fine-grained information if one use the average pooling of the whole feature map for contrastive learning. To address this problem, we construct the representations on primitive level [36] and propose a primitive contrastive loss that utilizes the structure of formula images. The primitive-align function segments the 2D feature map into several primitive patches and applies contrastive loss to each pair of patches. Specifically, the original feature map with size $C \times H \times W$ is divided into $p_H \times p_W$ primitive patches. The patch size $\mathrm{size}_H \times \mathrm{size}_W$ is computed according to

$$\begin{aligned} \mathrm{stride}_H &= \mathrm{floor}(H/p_H), \\ \mathrm{size}_H &= H - (p_H - 1) \times \mathrm{stride}_H, \\ \mathrm{stride}_W &= \mathrm{floor}(W/p_W), \\ \mathrm{size}_W &= W - (p_W - 1) \times \mathrm{stride}_W. \end{aligned}$$

Accordingly, the patches are generated every $\mathrm{stride}_H \times \mathrm{stride}_W$ step. The feature vector associated with each patch is the average of all feature vectors inside the patch.

For the bth image in a batch, let $f^1_{b,i,j}$ and $f^2_{b,i,j}$ be the C-dimensional feature of a patch (i, j) from two parallel encoder paths. The primitive contrastive loss $\mathcal{L}$ of the inputs with batch size B is

$$\begin{aligned} l(b,i,j) = &\frac{\exp(\mathrm{sim}(f^1_{b,i,j}, f^2_{b,i,j})/\tau)}{\sum_{b,i',j'} \exp(\mathrm{sim}(f^1_{b,i,j}, f^2_{b,i',j'})/\tau)} \\ &+ \frac{\exp(\mathrm{sim}(f^1_{b,i,j}, f^2_{b,i,j})/\tau)}{\sum_{b,i',j'} \exp(\mathrm{sim}(f^1_{b,i',j'}, f^2_{b,i,j})/\tau)}, \\ \mathcal{L} = &\frac{1}{2B \cdot p_H \cdot p_W} \sum_{b=1}^{B} \sum_{i=1}^{p_H} \sum_{j=1}^{p_W} l(b,i,j), \end{aligned} \tag{2}$$

where τ is a hyperparameter and the cosine similarity is adopted as $\mathrm{sim}(\cdot,\cdot)$ function. The primitive contrastive loss considers the two feature vectors $f^1_{b,i,j}$ and $f^2_{b,i,j}$ on the same position of a image pair but from different encoder paths as the positive sample, and treats the others as negative samples, including those from the image pair but associated with different positions and those from different images.

Practically, each primitive patch contains only a few mathematical symbols. Therefore, the model tends to learn discriminative representation at the mathematical symbol level. Besides, the primitive align operation increases the number of negative samples in a batch, which also contributes to better performance [4].

5) Local Semantic-Invariant Data Augmentation: The operations of data augmentation in conventional contrastive learning include cropping, flipping, color distortion and rotation. However, these regular transformations destroy the intrinsic structure of formula images as shown in Fig. 3. We here propose a set of mild operations that preserves local semantic information. The operations are linear contrast, slight random cropping, sharpening, gaussian blur, perspective transformation and affine transformation. Ablation experiments in Section IV-C verify their efficacy.

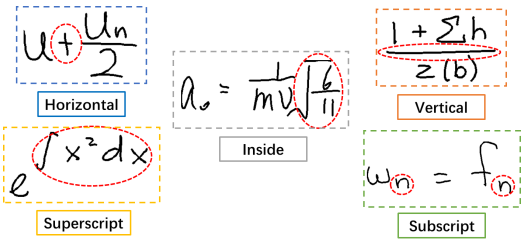


Fig. 3. Examples of intrinsic structure of formula images. The five kinds of common structure are "horizontal", "vertical", "superscript", "subscript" and "inside". The structure can be easily destroyed through aggressive cropping and flipping.

B. Supervised Fine-tuning

After pre-training the encoder, the whole encoder-decoder model is fine-tuned in the supervised learning way. The whole inference path includes the image encoder, the contextual encoder and decoder, which shares the same architecture as the model in the work [27]. The pre-trained parameters are used to initialize the encoders. The decoder is a recurrent neural network with attention mechanism. It takes the feature map M of the contextual encoder as input and generates a LaTeX format markup sequence.

Fig. 4 depicts its basic recurrent structure. In particular, the decoder computes the attention weight $\alpha_{t,(i,j)}$ by

$$\begin{aligned} \alpha_{t,(i,j)} &= \exp(e_{t,(i,j)}) / \sum_{h,w} \exp(e_{t,(h,w)}), \\ e_{t,(h,w)} &= \beta(W_Q h_t \cdot W_K m_{(h,w)}), \end{aligned} \tag{3}$$

where β is the temperature coefficient, and W_Q, W_K are trainable weights, which project the hidden state h_t and visual representation $m_{(h,w)}$ to queries and keys respectively.

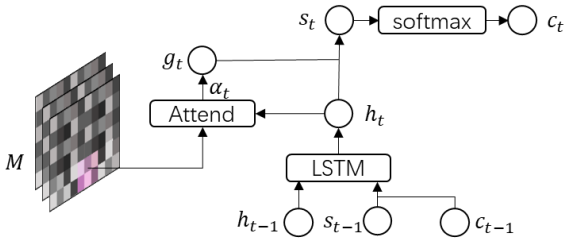


Fig. 4. The structure of the markup language decoder

Then, the glimpse vector is computed by

$$g_t = \sum_{h,w} \alpha_{t,(h,w)} W_V m_{(h,w)},$$

which encodes the visual information with attention mechanism.

Next, an LSTM computes the hidden state h_t by

$$\begin{aligned} h_t &= \mathrm{LSTM}(h_{t-1}, \mathrm{concat}(s_{t-1}, \Phi(c_{t-1}))), \\ s_{t-1} &= \tanh(\mathrm{W_c concat(h_t - 1, g_t - 1)}), \end{aligned}$$

based on the previous state h_{t-1} and a concatenation of previous output s_{t-1}, as well as the target embedding $\Phi(c_{t-1})$.

Finally, the conditional probability of a LaTeX symbol c_t is decoded by

$$p(c_t | c_{<t}; M) = \mathrm{softmax}(W_{out} s_t),$$

where W_c are trainable weights.

IV. Experiments

We conducted a series of experiments to evaluate the performance of PrimCLR on the benchmark datasets.

A. Experimental Setup

The implementation details of pre-training and fine-tuning are as follows.

1) Pre-training Datasets: At pre-training stage, we apply our PrimCLR on two datasets.

CROHME-hybrid: 1.2×10^5 unlabeled math formula images are generated according to the hybrid strategy in the work [37]. The example images are shown in Fig. 5.

TAL-OCR-MATH[1]: Images in this dataset are taken from real scenarios with distortions such as blurring, noises, lacking of strokes, etc. Fig. 6 demonstrates a few examples of TAL-OCR-MATH. The data distribution is largely distinct from commonly used HMER datasets, which only contain binary images.

2) Downstream Datasets: We evaluate our approach on the public datasets from the Competition on Recognition of Online Handwritten Mathematical Expressions (CROHME) [38]. The CROHME train set contains 8,836 formula images with both symbol-level and expression-level annotations. Only expression level annotations are used in our experiments. The test sets of CROHME contain 671, 986, 1,147 and 1,199 formula images in the year 2013, 2014, 2016 and 2019, respectively. All images are normalized with the height of 160

[1]https://ai.100tal.com/dataset

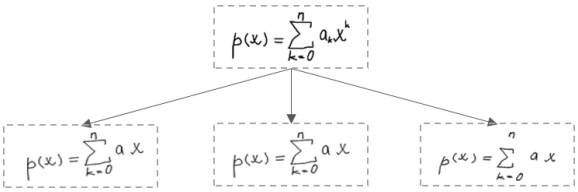


Fig. 5. Examples of CROHME-hybrid images. The top image is the original formula image from CROHME 2019, and below are the images generated using the hybrid strategy.

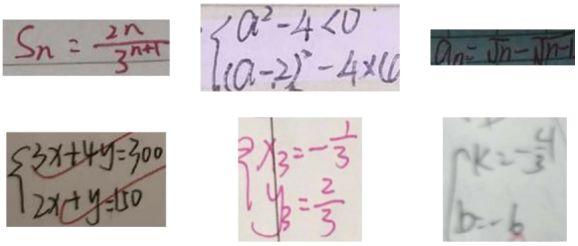


Fig. 6. Examples of TAL-OCR-MATH images. These images are from real scenarios, which suffers from blurring, noises, dullness, lacking of strokes, etc, making them difficult to be identified.

pixels following the protocol in [31], [39]. In each batch, the formula images with shorter lengths are padded with zeros at the end.

3) Pre-training Setup: ResNest-101 is taken as our CNN feature extractor. The stride of the convolution layers is modified from default 32 to 16 in order to increase the size of feature map. A two-layer BiLSTM is adapted as the contextual encoder outputting feature map with dimension $500 \times 10 \times W$, where W is a flexible variable depending on the length of input formula images. The parameters p_h and p_w are set to 2 and 5 respectively. The temperature coefficient τ in (2) is set to 0.5.

The contrastive learning process is optimized via stochastic gradient decent with initial learning rate 10^{-2}. A multi-step learning rate decay schedule is adopted after the first 5×10^4 iterations, multiplying 0.5 every 10^4 steps. The minibatch size is 8. The default maximum iterations are 10^5. Our experiments were implemented in PyTorch on four Nvidia GeForce RTX3090 GPUs.

4) Fine-tuning Setup: Consistent with the previous HMER works [27], [30], [31], [32], CROHME train set is used to fine-tune the downstream image-to-markup model. We consider CROHME 2013 test set as the validation set, and CROHME 2014, 2016 and 2019 as test sets. The performance is measured by ExpRate, which is defined as the percentage of formulas recognized correctly. The temperature coefficient β in (3) is set to 1. Beam search [40] is used during the testing process with beam size 5.

B. Results

1) Visual Representation Learning for HMER: To compare the proposed PrimCLR with other sate-of-the-art contrastive learning methods, we implement these methods with the image-to-markup recognizer on CROHME 2014 and CROHME 2016. The results are listed in Table I.

When pre-trained on CROHME-hybrid and TAL-OCR-MATH, the ExpRate on CROHME 2014 and 2016 increases from 48.1% and 48.1% to 51.9% and 54.8%, leading to the performance gap of 3.8% and 6.7%, respectively. The SimCLR [4] experiments are conducted using our local semantic-invariant data augmentation strategy with an MLP as projection head after the global average pooling and without primitive align mechanism. Our PrimCLR outperforms SimCLR by 2.1% and 3.1% on CROHME 2014 and 2016, respectively.

TABLE I
EXPRATE ON CROHME. ALL THE METHODS ARE PRE-TRAINED FOR 10^5 STEPS ON CROHME-HYBRID AND TAL-OCR-MATH.

Dataset	Pre-training	ExpRate
CROHME 2014	No pre-train	48.1
	SimCLR	49.8
	SeqCLR	50.6
	Ours	**51.9**
CROHME 2016	No pre-train	48.1
	SimCLR	51.7
	SeqCLR	52.1
	Ours	**54.8**

TABLE II
COMPARISON WITH STATE-OF-THE-ART HMER METHODS.

Method	CROHME 2014	CROHME 2016	CROHME 2019
PAL [30]	39.66	-	-
WAP [32]	40.4	37.1	-
DenseWAP [33]	43.0	40.1	41.7
PAL-v2 [31]	48.88	49.61	-
DenseWAP-TD [41]	49.1	48.5	51.4
WS-WAP [42]	53.65	51.96	-
BTTR-Uni [29]	48.17	44.55	44.95
BTTR-Bi [29]	53.96	52.31	52.96
Our baseline	48.1	48.1	48.7
Ours	**51.9**$\pm$0.16	**54.8**$\pm$0.26	**54.9**$\pm$0.24

2) Comparison with other Image-to-Markup Methods: For the public datasets CROHME 2014, 2016 and 2019, we provide the comparison with previous state-of-the-art methods. We only consider the methods trained with CROHME train set and without data augmentation at the supervised training stage.

As presented in Table II, PrimCLR outperforms all other HMER methods on CROHME 2016 and 2019, and is comparable to the state-of-the-art methods on CROHME 2014. We note that at the inference stage, our model only employs the original image-to-markup recognizer as the base structure, which is much weaker than the other models, but the ExpRate is improved dramatically by the contrastive pre-training with extra unlabeled data.

C. Ablation Study

Extensive ablation study is conducted to show how each component contributes to PrimCLR. Models are pre-trained on CROHME-hybrid and TAL-OCR-MATH, and fine-tuned on CROHME train set.

1) Local Semantic Invariant Data Augmentation: Data augmentation is of great importance in self-supervised learning. Six types of augmentation are used during the contrastive learning stage, including contrast changing, gaussian blur, slightly random cropping, sharpening effect, perspective transformation and affine transformation. We systematically study the impact of different kinds of data augmentation.

TABLE III
EFFECT OF THE AUGMENTATION STRATEGY.

Augmentation	CROHME 2014	CROHME 2016
No pre-train	48.1	48.1
Contrast	49.0	51.1
Gaussian Blur	49.3	50.7
Crop	50.7	53.7
Sharpen	50.9	53.9
Perspective	50.7	50.5
Affine	49.4	50.6
All	**51.9**	**54.8**

The results in Table III show that all these six kinds of augmentation contribute to better representation. However, none of them is sufficient to get the best result obtained by using them all ("All" in Table III). When using multiple augmentation types, the difficulty of contrastive prediction task increases so that the model can learn more discriminative representation.

2) Image Encoder: We take ResNest-101 as our CNN feature extractor instead of frequently-used DenseNet-100. To better understand the effect of image encoder, we establish the ablation study in Table IV.

TABLE IV
ABLATION STUDY OF IMAGE ENCODER.

Pre-training	Image Encoder	CROHME 2014	CROHME 2016
No pre-train	DenseNet-100	42.9	45.6
	ResNest-101	48.1	48.1
Ours	DenseNet-100	49.7	50.4
	ResNest-101	51.9	54.8

The results show that our self-supervised pre-training method can improve the ExpRate with both DenseNet-100 image encoder and ResNest-101 image encoder. Replacing DenseNet-100 with ResNest-101 increases the performance by 2.2% and 4.4% on CROHME 2014 and 2016, respectively.

3) Contextual Encoder: In object-centric contrastive learning, MLP projection head improves image classification performance for a large margin. We compare MLP-based projection head and BiLSTM-based projection head in Table V. To test the MLP-based projection head, we replace the BiLSTM in our framework by a MLP, and discard it at fine-tuning stage.

TABLE V
ABLATION STUDY OF PROJECTION HEAD.

Projection Head	CROHME 2014	CROHME 2016
MLP	51.0	53.5
BiLSTM(not reserved)	51.3	52.8
BiLSTM(reserved)	51.9	54.8

The results show that our BiLSTM projection head is better than MLP. Different from the common result in image classification, we discover that it is better to reserve pre-trained BiLSTM parameters instead of random initialize the BiLSTM at the beginning of the fine-tuning step. This phenomenon is probably caused by the complicated structure of formula images. In the image-to-markup structure, the images during fine-tuning stage is not enough to train the BiLSTM adequately, making the pre-training of BiLSTM necessary.

4) Primitive Hyperparameters: We test the sensitivity of the hyperparameters p_H and p_W in the primitive-align function, where we computes the $p_H \times p_W$ primitive patches from the feature map. Larger number of primitive patches results in smaller patch size and vice versa. When $p_H = p_W = 1$, the primitive contrastive loss reduces to the conventional contrastive loss.

TABLE VI
ABLATION STUDY OF PRIMITIVE ALIGN.

Primitives	CROHME 2014	CROHME 2016
5×1	50.6	52.1
5×2	51.9	54.8
10×4	51.5	55.2

Table VI shows that the best choice is 10×4. We choose 5×2 in our experiments as the best tradeoff between accuracy and computational cost.

5) Pre-training Datasets: To verify the robustness of our method, we conduct extra experiments on datasets including: binary HMER datasets containing only black and white pixels, *e.g.*, CROHME train set and CROHME-hybrid, and real scenario HMER datasets, *e.g.*, TAL-OCR-MATH.

TABLE VII
ABLATION STUDY OF PRE-TRAINING DATASET.

Pre-training Dataset	CROHME 2014	CROHME 2016
No pre-train	48.1	48.1
CROHME train set	51.5	52.9
CROHME-hybrid	51.0	54.3
TAL-OCR-MATH	50.2	53.7

As shown in Table VII, when pre-training is done only on CROHME train set, the ExpRate is also boosted, which indicates that our PrimCLR is effective even without extra unlabeled data. Noticeably, self-supervised pre-training on real scenario dataset, *i.e.*, TAL-OCR-MATH, whose distribution is different from CROHME, brings noticeable performance improvement as well, implying that our PrimCLR is robust to dataset distribution. Besides, the CROHME-hybrid dataset which contains much more images than CROHME train set and has similar distribution to CROHME benefits the ExpRate most. In the original work of CROHME-hybrid [37], the model PGS trained by the supervised method achieves 48.78% and 45.60% ExpRate on CROHME 2014 and 2016. Through unsupervised pre-training, our method significantly outperforms PGS.

V. CONCLUSION

We proposed a simple and effective self-supervised learning framework, namely PrimCLR, for handwritten mathematical expression recognition. Our method captures structural and contextual information in formula images, outperforming the mainstream contrastive learning methods. In addition to improving formula recognition significantly, PrimCLR is robust to various pre-training datasets. This work extends the application of visual representation learning to images with complex structure. Our future work aims to explore more structural and contextual information to improve the representation learning for structured prediction problems.

VI. Acknowledgement

This work has been supported by the National Key Research and Development Program under Grant No. 2020AAA0109702, the National Natural Science Foundation of China (NSFC) grants U20A20223 and 61721004, and the Pioneer Hundred Talents Program of CAS under Grant Y9S9MS08.

References

[1] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, Salt Lake City, UT, USA, June 18-22, 2018, pp. 3733–3742.

[2] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, Seattle, WA, USA, June 13-19, 2020, pp. 9726–9735.

[3] X. Chen, H. Fan, R. B. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *CoRR*, vol. abs/2003.04297, 2020.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, July 13-18, 2020, pp. 1597–1607.

[5] J. B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," in *Advances in Neural Information Processing Systems 33: NeurIPS 2020*, December 6-12, 2020.

[6] S. Purushwalkam and A. Gupta, "Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases," in *Advances in Neural Information Processing Systems 33: NeurIPS 2020*, December 6-12, 2020.

[7] R. H. Anderson, "Syntax-directed recognition of hand-printed two-dimensional mathematics," in *Symposium on Interactive Systems for Experimental Applied Mathematics: ACM*, 1967, pp. 436–459.

[8] X. D. Zhou, D. H. Wang, F. Tian, C. L. Liu, and M. Nakagawa, "Handwritten chinese/japanese text recognition using semi-markov conditional random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2413–2426, 2013.

[9] V. Ordonez, X. Han, P. Kuznetsova, G. Kulkarni, M. Mitchell, K. Yamaguchi, K. Stratos, A. Goyal, J. Dodge, A. C. Mensch, H. D. III, A. C. Berg, Y. Choi, and T. L. Berg, "Large scale retrieval and generation of image descriptions," *Int. J. Comput. Vis.*, vol. 119, no. 1, pp. 46–59, 2016.

[10] J. Aneja, A. Deshpande, and A. G. Schwing, "Convolutional image captioning," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, Salt Lake City, UT, USA, June 18-22, 2018, pp. 5561–5570.

[11] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada, April 30 - May 3, 2018.

[12] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *IEEE International Conference on Computer Vision, ICCV 2015*, Santiago, Chile, December 7-13, 2015, pp. 1422–1430.

[13] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Computer Vision - ECCV 2016 - 14th European Conference*, Amsterdam, The Netherlands, October 11-14, 2016, pp. 69–84.

[14] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision - ECCV 2016 - 14th European Conference*, Amsterdam, The Netherlands, October 11-14, 2016, pp. 649–666.

[15] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, USA, July 21-26, 2017, pp. 840–849.

[16] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Advances in Neural Information Processing Systems 33: NeurIPS 2020*, December 6-12, 2020.

[17] Y. Bai, A. Wang, A. Kortylewski, and A. L. Yuille, "Coke: Localized contrastive learning for robust keypoint detection," *CoRR*, vol. abs/2009.14115, 2020.

[18] J. Han, M. Shoeiby, L. Petersson, and M. A. Armin, "Dual contrastive learning for unsupervised image-to-image translation," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021*, June 19-25, 2021, pp. 746–755.

[19] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense contrastive learning for self-supervised visual pre-training," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*, June 19-25, 2021, pp. 3024–3033.

[20] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, Z. Li, and P. Luo, "Detco: Unsupervised contrastive learning for object detection," *CoRR*, vol. abs/2102.04803, 2021.

[21] S. Liu, Z. Li, and J. Sun, "Self-emd: Self-supervised object detection without imagenet," *CoRR*, vol. abs/2011.13677, 2020.

[22] A. Aberdam, R. Litman, S. Tsiper, O. Anschel, R. Slossberg, S. Mazor, R. Manmatha, and P. Perona, "Sequence-to-sequence contrastive learning for text recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*, June 19-25, 2021, pp. 15 302–15 312.

[23] R. Zanibbi and D. Blostein, "Recognition and retrieval of mathematical expressions," *Int. J. Document Anal. Recognit.*, vol. 15, no. 4, pp. 331–357, 2012.

[24] F. Alvaro, J. A. Sánchez, and J. M. Benedí, "Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models," *Pattern Recognit. Lett.*, vol. 35, pp. 58–67, 2014.

[25] F. Alvaro, J. A. Sánchez, and J. M. Benedí, "An integrated grammar-based approach for mathematical expression recognition," *Pattern Recognit.*, vol. 51, pp. 135–147, 2016.

[26] S. MacLean and G. Labahn, "A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets," *Int. J. Document Anal. Recognit.*, vol. 16, no. 2, pp. 139–163, 2013.

[27] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, "Image-to-markup generation with coarse-to-fine attention," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, Sydney, NSW, Australia, 6-11 August 2017, pp. 980–989.

[28] J. W. Wu, F. Yin, Y. M. Zhang, X. Y. Zhang, and C. L. Liu, "Graph-to-graph: Towards accurate and interpretable online handwritten mathematical expression recognition," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, February 2-9, 2021, pp. 2925–2933.

[29] W. Zhao, L. Gao, Z. Yan, S. Peng, L. Du, and Z. Zhang, "Handwritten mathematical expression recognition with bidirectionally trained transformer," *CoRR*, vol. abs/2105.02412, 2021.

[30] J. W. Wu, F. Yin, Y. M. Zhang, X. Y. Zhang, and C. L. Liu, "Image-to-markup generation via paired adversarial learning," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018*, Dublin, Ireland, September 10-14, 2018, pp. 18–34.

[31] J. W. Wu, F. Yin, Y. M. Zhang, X. Y. Zhang, and C. L. Liu, "Handwritten mathematical expression recognition via paired adversarial learning," *Int. J. Comput. Vis.*, vol. 128, no. 10, pp. 2386–2401, 2020.

[32] 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 Recognit.*, vol. 71, pp. 196–206, 2017.

[33] J. Zhang, J. Du, and L. Dai, "Multi-scale attention with dense encoder for handwritten mathematical expression recognition," in *24th International Conference on Pattern Recognition, ICPR 2018*, Beijing, China, August 20-24, 2018, pp. 2245–2250.

[34] H. Ding, K. Chen, and Q. Huo, "An encoder-decoder approach to handwritten mathematical expression recognition with multi-head attention and stacked decoder," in *16th International Conference on Document Analysis and Recognition, ICDAR 2021*, Lausanne, Switzerland, September 5-10, 2021, pp. 602–616.

[35] Z. Li, L. Jin, S. Lai, and Y. Zhu, "Improving attention-based handwritten mathematical expression recognition with scale augmentation and drop attention," in *17th International Conference on Frontiers in Handwriting Recognition, ICFHR 2020*, Dortmund, Germany, September 8-10, 2020, pp. 175–180.

[36] R. Yan, L. Peng, S. Xiao, and G. Yao, "Primitive representation learning for scene text recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*, June 19-25, 2021, pp. 284–293.
[37] A. D. Le, B. Indurkhya, and M. Nakagawa, "Pattern generation strategies for improving recognition of handwritten mathematical expressions," *Pattern Recognit. Lett.*, vol. 128, pp. 255–262, 2019.
[38] H. Mouchère, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "ICFHR2016 CROHME: competition on recognition of online handwritten mathematical expressions," in *15th International Conference on Frontiers in Handwriting Recognition, ICFHR 2016*, Shenzhen, China, October 23-26, 2016, pp. 607–612.
[39] Z. Li, L. Jin, S. Lai, and Y. Zhu, "Improving attention-based handwritten mathematical expression recognition with scale augmentation and drop attention," in *17th International Conference on Frontiers in Handwriting Recognition, ICFHR 2020*, Dortmund, Germany, September 8-10, 2020, pp. 175–180.
[40] K. Cho, "Natural language understanding with distributed representation," *CoRR*, vol. abs/1511.07916, 2015.
[41] J. Zhang, J. Du, Y. Yang, Y. Song, S. Wei, and L. Dai, "A tree-structured decoder for image-to-markup generation," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, July 13-18, 2020, pp. 11 076–11 085.
[42] T. 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 *17th International Conference on Frontiers in Handwriting Recognition, ICFHR 2020*, Dortmund, Germany, September 8-10, 2020, pp. 181–186.