Handwritten Mathematical Expression Recognition via Paired Adversarial Learning

Jin-Wen Wu, Fei Yin, Yan-Ming Zhang, Xu-Yao Zhang & Cheng-Lin Liu

International Journal of Computer Vision

ISSN 0920-5691
Volume 128
Combined 10-11

Int J Comput Vis (2020) 128:2386-2401
DOI 10.1007/s11263-020-01291-5

Springer

International Journal of Computer Vision (2020) 128:2386–2401
https://doi.org/10.1007/s11263-020-01291-5

Handwritten Mathematical Expression Recognition via Paired Adversarial Learning

Jin-Wen Wu[1,2] · Fei Yin[1] · Yan-Ming Zhang[1] · Xu-Yao Zhang[1,2] · Cheng-Lin Liu[1,2,3]

Received: 29 March 2019 / Accepted: 2 January 2020 / Published online: 21 January 2020

Abstract

Recognition of handwritten mathematical expressions (MEs) is an important problem that has wide applications in practice. Handwritten ME recognition is challenging due to the variety of writing styles and ME formats. As a result, recognizers trained by optimizing the traditional supervision loss do not perform satisfactorily. To improve the robustness of the recognizer with respect to writing styles, in this work, we propose a novel paired adversarial learning method to learn semantic-invariant features. Specifically, our proposed model, named PAL-v2, consists of an attention-based recognizer and a discriminator. During training, handwritten MEs and their printed templates are fed into PAL-v2 simultaneously. The attention-based recognizer is trained to learn semantic-invariant features with the guide of the discriminator. Moreover, we adopt a convolutional decoder to alleviate the vanishing and exploding gradient problems of RNN-based decoder, and further, improve the coverage of decoding with a novel attention method. We conducted extensive experiments on the CROHME dataset to demonstrate the effectiveness of each part of the method and achieved state-of-the-art performance.

Keywords Handwritten ME recognition · Paired adversarial learning · Semantic-invariant features · Convolutional decoder · Coverage of decoding

1 Introduction

Handwritten mathematical expression recognition (HMER) has received considerable attention for its potential applications in many areas such as education, office automation and conference systems. This problem still faces a mountain of technical challenges since the images of handwritten MEs contain much more complicated two-dimensional (2D) structures and spatial relations than general images in computer vision (Aneja et al. 2018; Jaderberg et al. 2016; Krishna et al. 2017; Ordonez et al. 2016; Zhou et al. 2013). Furthermore, HMER also suffers from the writing-style variations (see an example in Fig. 1) and the scarcity of annotated data.

HMER has been studied since the 1960s (Anderson 1967). Traditional approaches (Chan and Yeung 2000; Zanibbi and Blostein 2012) use predefined grammars to handle symbol segmentation, symbol recognition, and structural analysis sequentially or simultaneously. Although grammar-driven approaches (Alvaro et al. 2014, 2016; Chan and Yeung 2001; MacLean and Labahn 2013) perform fairly well in several CROHME competitions, they require a large amount of manual work to design grammars. Recently, methods based on deep neural networks (DNNs) have been proposed (Deng

Communicated by Jun-Yan Zhu, Hongsheng Li, Eli Shechtman, Ming-Yu Liu, Jan Kautz, Antonio Torralba.

✉ Jin-Wen Wu
jinwen.wu@nlpr.ia.ac.cn

Fei Yin
fyin@nlpr.ia.ac.cn

Yan-Ming Zhang
ymzhang@nlpr.ia.ac.cn

Xu-Yao Zhang
xyz@nlpr.ia.ac.cn

Cheng-Lin Liu
liucl@nlpr.ia.ac.cn

[1] National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences, Beijing 100190, People's Republic of China

[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China

[3] CAS Center for Excellence of Brain Science and Intelligence Technology, Beijing 100190, People's Republic of China

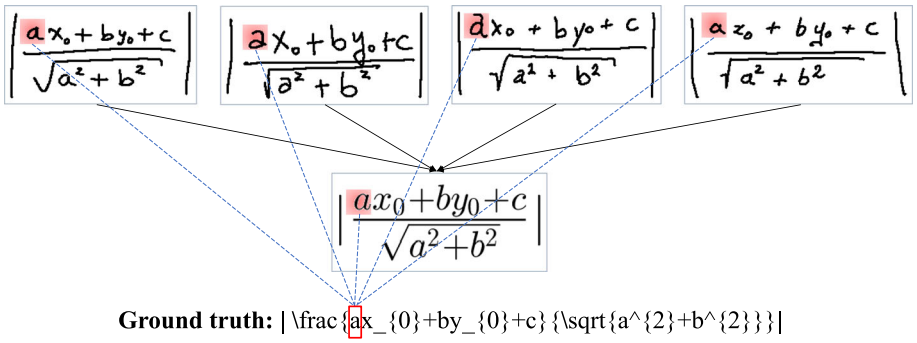


Fig. 1 MEs written by different people (top) and the standard printed template (center) with same ground-truth sequence (bottom). Red cells indicate attention regions with high probabilities. Symbols could be written in very different styles while share invariant features that represent the same semantic meaning (Color figure online)

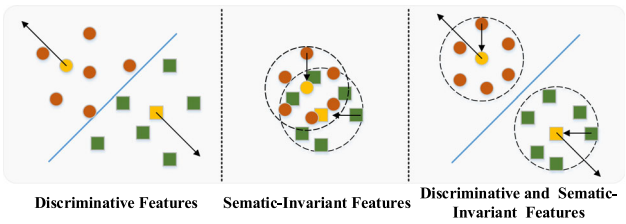


Fig. 2 Illustration of discriminative and semantic-invariant features learning. Circle points in the figure represent features of class i, square points indicate features of class j, where $i \neq j$, red and green represent features of handwritten symbols, and yellow indicates features come from printed templates (Color figure online)

et al. 2017; Le and Nakagawa 2017; Zhang et al. 2017a, b, 2018, 2019). Essentially, they treat HMER as a special case of the general image-to-sequence problem and solve it under the encoder-decoder framework. Compared to traditional approaches, DNNs have promoted the recognition performance significantly.

Despite these efforts, the accuracy of HMER methods is still limited due to the enormous challenge of the problem. Previously, to overcome the writing-style variation, we proposed the paired adversarial learning (PAL) method based on DNNs and adversarial learning (Wu et al. 2018b). The core idea of PAL is to make the recognizer learn semantic-invariant features by mapping handwritten MEs to their printed templates in the feature space (see Fig. 2). Another contribution of PAL is the adoption of the convolutional decoder, which is easier and faster to optimize compared with the more commonly used RNN decoder.

In this paper, we propose a new HMER method, named PAL-v2, which extends PAL in the following three aspects:

1. For the encoder, since MEs are featured by the complex 2D structure and long-range dependence, we aim to incorporate more contextual information by replacing the VGG-based CNN feature extractor with a DenseNet and designing a novel densely connected multi-directional RNN block on top of the CNN feature extractor.
2. For the decoder, it is extremely important to pay close attention to all the symbols in the ME images. In this work, we improve the convolutional decoder with a novel attention method named Pre-aware Coverage Attention (PCA) to enhance the coverage of decoding while allowing parallel computing. In addition, we also utilize an N-gram statistical language model to assist the decoding.
3. For the adversarial learning pipeline, it is similar to our previous work. However, in this study, we demonstrate that the proposed adversarial training method is flexible to the recognizer with different structures. Furthermore, we employ the discriminators at different stages of the recognizer and explore different capacities of the discriminators. We also visualize the feature distributions learned by the recognizer to explain the inner working of our proposed method.

The main contributions of our work are highlighted as follows:

1. To improve the robustness of the recognizer with respect to writing styles, we introduce paired adversarial learning to learn semantic-invariant features from handwritten ME images and their printed templates.
2. We adopt a convolutional decoder to alleviate the vanishing and exploding gradient issues of RNN based decoder and propose a novel attention method to improve the coverage of decoding.
3. To capture 2D long-range contextual dependencies, we design a new densely connected multi-directional RNN block for the encoder.
4. The proposed PAL-v2 model boosts the expression recognition rate (ExpRate) of PAL from 39.66 to 48.88% and achieves state-of-the-art performance on the benchmark datasets CROHME 2014 and 2016.

The rest of this paper is organized as follows. Section 2 briefly reviews existing HMER approaches and other related works. Section 3 details the proposed model and algorithms. Section 4 presents extensive experimental results. Finally, we conclude our work in Sect. 5.

2 Related Works

2.1 A Brief Review of HMER Methods

HMER has received intensive attention, and many previous works have been surveyed in Chan and Yeung (2000) and Zanibbi and Blostein (2012). These approaches usually involve three main parts: symbol segmentation, symbol recognition, and structural analysis. A variety of predefined grammars have been used to solve these tasks, such as the stochastic context-free grammar for HMER in Alvaro et al. (2014, 2016). Such grammar-based approaches have achieved outstanding performance in several CROHME competitions. Other manually designed math grammars, such as definite clause grammars (Chan and Yeung 2001) and relational grammars (MacLean and Labahn 2013), have also been used for HMER.

Depending on the organization of the three steps, the previous approaches can be categorized into sequential solutions and global ones. Sequential solutions (Alvaro et al. 2016; Mouchère et al. 2016b) implement these three parts in turn which is in line with human reading. The disadvantage is that errors in sequential steps will accumulate. Different from sequential solutions, a global solution (Awal et al. 2014) segments symbols implicitly while recognizing them and analyzing structures. However, the computation time of global optimization tends to increase exponentially with the number of symbols.

Compared with approaches based on human-designed grammars, the recently presented attentional framework (Deng et al. 2017; Le and Nakagawa 2017; Zhang et al. 2017a, b, 2018, 2019) shows superior learning power for HMER and has significantly advanced the recognition performance on HMER. These attentional recognizers learn math grammars from training data via embedded language models and can segment symbols in MEs automatically with a data-driven attention mechanism. However, these attentional recognizers are only trained to learn discriminative features and the writing-style variation is not treated adequately.

2.2 Domain-Invariant Features Learning

The main difference between domains of handwritten and printed MEs lies in the writing style. We reduce the intra-class variance of handwritten symbols by guiding the recognizer to learn semantic-invariant features. In a related work, invariant representation of different domains was learned with the generative adversarial network (GAN) (Goodfellow et al. 2014) for domain adaptation.

GAN is a well-known adversarial learning method originally presented for generative learning by Goodfellow et al. It generally consists of a generator G and a discriminator D, which are trained with conflicting objectives:

$$\begin{aligned}&\min_G \max_D V(G, D)\\ =\ & E_{x\sim p_{data}(\mathbf{x})} \log D(x) + E_{z\sim p_z(\mathbf{z})} \log(1 - D(G(z))),\end{aligned} \tag{1}$$

where x denotes the target real sample, z is the input noise and $D(x)$ is the probability that the sample is real. G tries to forge real samples to confuse D while D tries to distinguish fake samples from real ones. Extensions of GANs have been proposed to find domain-invariant representations of different domains (Bousmalis et al. 2017; Radford et al. 2015). Recently, an adversarial-learning-based method was proposed for improving the generation performance of offline handwritten character recognizers (Zhang et al. 2018). Specifically, it incorporates prior knowledge of printed templates, and utilize a discriminator for guiding traditional feature extractor to learn writer-independent features of characters. A parallel work (Liu et al. 2018) for text recognition shares similar idea in learning invariant features.

2.3 Encoder-Decoder with Attention

RNN-based encoder-decoder with attention was widely used for dealing with image-to-sequence and sequence-to-sequence problems, such as image captioning (Cho et al. 2015; Li et al. 2017; Xu et al. 2015), scene text recognition (Shi et al. 2018), machine translation (Bahdanau et al. 2014) and speech recognition (Chorowski et al. 2015). However, in contrast to CNNs, RNNs often suffer from gradient vanishing and exploding problems. Furthermore, the inherently sequential processing of RNNs makes it hard to be implemented in parallel.

Recently, a series of entirely convolutional neural networks with attention have been proposed to address the aforementioned problems faced by the recurrent encoder-decoder and have shown efficacy in both accuracy and training time (Aneja et al. 2018; Bai 2018; Gehring et al. 2017; Wu et al. 2018a, b). Despite the benefits, convolutional decoders face a serious problem, namely, lack of coverage (Tu et al. 2016), which means some regions in the images are over-attended or under-attended in the decoding process. Recurrent decoders (Zhang et al. 2017a, b, 2018, 2019) manage to overcome this problem by utilizing all of historical attention maps as additional information for guiding the attention of the current step. Parallel computing of convolutional decoders becomes a disadvantage in this case since attention maps at every step are calculated independently and simultaneously. In this work, in order to retain the advantages of both convolutional and recurrent decoders, we introduce a novel attention method in Sect. 3.2.2.

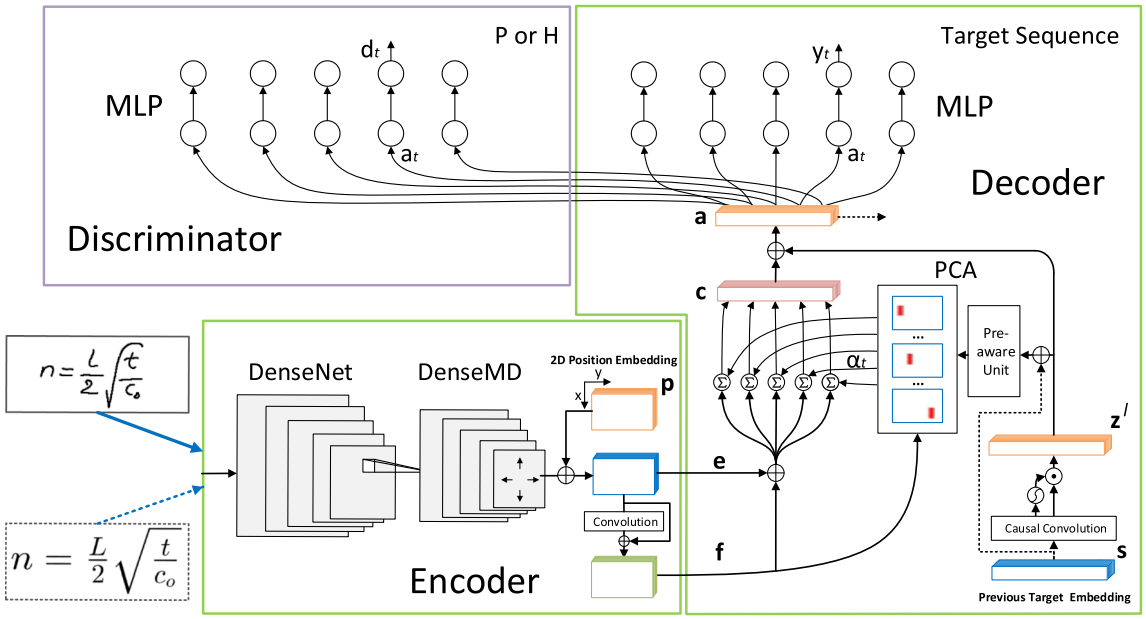


Fig. 3 Architecture of our proposed model. During training procedure, each handwritten ME image is input with its printed template (bottom left). The encoder-decoder recognizer and the discriminator are trained alternately. We equip the convolutional decoder with a novel pre-aware coverage attention for improving the coverage of decoding

3 Proposed Model

In this paper, we treat HMER as a special case of the general image-to-sequence problem. Specifically, given an ME image, our method outputs the LaTeX code of the ME.

The proposed PAL-2 model is comprised of an attentional encoder-decoder recognizer R and a discriminator D.

The encoder in R encodes input ME images into feature maps and then the decoder parses these feature maps into LaTeX. When R attends to the related regions of symbols in the feature map, D guides R to learn semantic-invariant features for making R more robust to writing-style variations.

PAL-v2 is learned in an adversarial learning manner. Specifically, handwritten MEs paired with their printed templates are fed into the model together. R is trained with two aims: 1. Correctly recognize both handwritten MEs and their printed templates. 2. Fool D by making the features of the paired images indistinguishable. D is optimized to judge whether the features are from handwritten or printed MEs. R and D are updated alternately.

See Fig. 3 for the overall architecture of PAL-v2.

In the following sections, we first introduce each part of PAL-v2 in Sects. 3.1–3.3. Then, we describe the training procedure of our proposed model in Sect. 3.4 and the combining of the recognizer R and a statistical language model for decoding in Sect. 3.5.

3.1 Dense Encoder

Our encoder consists of a CNN-based feature extractor and a RNN-based feature extractor. In this work, we utilize DenseNet (Huang et al. 2017) as the CNN feature extractor. The main idea of DenseNet is adopting concatenated output feature maps of all previous layers as input of current layer.

We use a RNN-based feature extractor after the CNN-based extractor to perceive more contextual information (Deng et al. 2016, 2017; Le and Nakagawa 2017). Inspired by the success of DenseNet, we design a novel densely connected MDLSTM, named DenseMD, to mitigate the vanishing gradient problem of deep RNN. Specifically, we replaced the 3×3 convolution in the DenseNet block with a MDLSTM. Each MDLSTM layer employs four LSTM layers in up, down, left and right directions in parallel. The feature maps in different directions are summed up to get output. We combined the final output $x^{last} \in \mathbb{R}^{H \times W \times C}$ of the DenseMD with the absolute position embedding (Gehring et al. 2017) to enhance the model's sense of order:

$$e_{i,j} = x_{i,j}^{last} + \phi(p_{x,i}) + \varphi(p_{y,j}), \tag{2}$$

where $\phi(p_{x,i}) = W_\phi p_{x,i}$ and $\varphi(p_{y,j}) = W_\varphi p_{y,j}$ denote the position embeddings in horizontal and vertical direction, respectively (see details in Fig. 3). $p_{x,i}$ and $p_{y,j}$ are one-hot vectors of the input coordinates i and j. The weight parameters W_φ and W_ϕ are learned by back-propagation. Later, a shallow convolution with residual connection is applied to

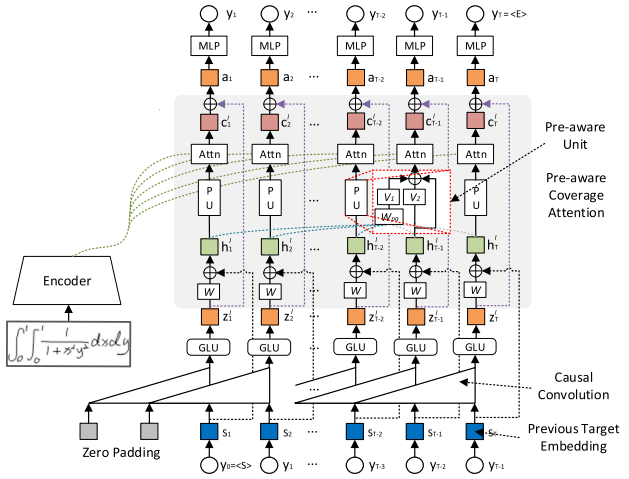


Fig. 4 Architecture of the convolutional decoder. $y_0 = \langle \mathrm{S} \rangle$ and $y_T = \langle \mathrm{E} \rangle$ indicates the beginning and the end of the target sequence, respectively. $a = (a_1, \ldots, a_T) \in \mathbb{R}^{T \times C}$ denote the output feature sequence of the last decoder block. $a_t, t = 1, \ldots, T$ is then classified with an MLP based classifier

e for better fusing semantic and positional information. The final output feature map is denoted as $f \in \mathbb{R}^{H \times W \times C}$.

3.2 Attentional Decoder

The complex structure of ME images makes it hard for the recognizer to accurately locate all the symbols of an ME in parallel. We solve this problem by applying a convolutional decoder based on the improved attention mechanism (see Fig. 4), named pre-aware coverage attention (PCA). We introduce the convolutional decoder in Sect. 3.2.1 and the novel attention mechanism PCA in Sect. 3.2.2.

3.2.1 Convolutional Decoder with Attention

The recurrent decoder sequentially attends to related regions at each decoding step and suffers from vanishing and exploding gradient problems. Different from the recurrent decoder, our attention-based decoder is a multi-block convolutional neural network (Gehring et al. 2017). Each decoder block comprises a one-dimensional causal convolution and a subsequent separate attention mechanism (see Fig. 4). The causal convolution is equipped with a gated linear unit (GLU) (Dauphin et al. 2017) for nonlinear activation. Output of the lth decoder block is predicted based on both the previous block output and related regions in the encoder output feature map. Related regions are chosen via an attention map $\alpha_t^l \in \mathbb{R}^{H \times W}$. With this notation at hand, the weight $\alpha_{t,(i,j)}^l$ of the attention map is calculated as:

$$h_t^l = W^l z_t^l + s_t + b^l, \tag{3}$$

$$\alpha_{t,(i,j)}^l = \frac{\exp(h_t^l \cdot f_{i,j})}{\sum_{n=1}^{H} \sum_{m=1}^{W} \exp(h_t^l \cdot f_{n,m})}, \tag{4}$$

where z_t^l is the output of the lth causal convolution, s_t is the embedding of previous target symbols, and W^l are trainable weights. The attended context vector of the image is obtained by:

$$c_t^l = \sum_{i=1}^{H} \sum_{j=1}^{W} \alpha_{t,(i,j)}^l (f_{i,j} + e_{i,j}), \tag{5}$$

where $e_{i,j}$ is the embedded feature in Eq. (2). According to the key-value memory network (Su et al. 2016), we run $f_{i,j}$ and $f_{i,j} + e_{i,j}$ as the keys and values, respectively. Then, c_t^l and z_t^l are combined to get the output of the lth decoder block by $a_t^l = c_t^l + z_t^l$.

The output of the last block a_t^{last} is taken for predicting the current symbol via:

$$\hat{p}_t = \mathrm{softmax}(W_o a_t^{last} + b_o) \in \mathbb{R}^K, \tag{6}$$

$$\hat{y}_t = \arg\max(\hat{p}_t), \tag{7}$$

where W_o and b_o are the parameters of the fully connected layer, K is the size of the LaTeX symbol set.

3.2.2 Pre-aware Coverage Attention

Classic attention described above tends to ignore past alignment information (Tu et al. 2016), which can result in over-attention and under-attention. Humans have a priori knowledge of reading from left to right. When we pay attention to the current word, we are aware of areas that have been read on the left side. Recurrent decoders could mimic the human reading process by utilizing the coverage of all historical attention locations.

Since the convolutional decoder predicts all symbols in the image simultaneously during training, in order to enable the current step to be pre-aware of the coverage of attention locations in previous steps, we redefine the attention map in Eq. (4) as:

$$\alpha_{t,(i,j)}^l = \frac{\exp(P(h_t^l) \cdot f_{i,j})}{\sum_{m=1}^{H} \sum_{n=1}^{W} \exp(P(h^l, t) \cdot f_{m,n})}, \tag{8}$$

where $P(*)$ denotes the mapping function of the pre-aware unit and is defined as:

$$P(h_t^l) = V_1^l W_{pa,t} h^l + V_2^l h_t^l, \tag{9}$$

where V_1^l and V_2^l are trained weights, while $W_{pa,t}$ is the tth row of the pre-aware matrix $W_{pa} \in \mathbb{R}^{T\times T}$:

$$W_{pa} = \begin{bmatrix} 0 & & & \\ 1 & 0 & 0 & \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \cdots & 0 \end{bmatrix}. \tag{10}$$

The ith element of $W_{pa,t}$ is zero when $i \geq t$. Thus, it is easy to get:

$$P(h_t^l) \cdot f_{i,j} = \left(V_1^l \sum_{k=1}^{t-1} h_k^l \right) \cdot f_{i,j} + (V_2^l h_t^l) \cdot f_{i,j}, \tag{11}$$

where the first item contains forecast information of previous $t-1$ steps attention weights, and the second item contains information about the original weight of the current step. The combination of information is learned via the trained weights V_1^l and V_2^l. In this way, the original parallel structure of the convolutional decoder can be maintained. Moreover, the semantic and positional information of previous symbols can be utilized more effectively in the current step.

In order to improve the back propagation of the gradient, we apply residual connections (He et al. 2016) to the pre-aware unit defined in Eq. (9):

$$P(h_t^l) = (V_1^l W_{pa,t} h^l + V_2^l h_t^l) + h_t^l. \tag{12}$$

If the learned weights V_1^l and V_2^l are zeros, $P(h_t^l)$ will be the original form of attention mechanism.

3.3 Adversarial Semantic-Invariant Features Learning

Traditional recognizers are usually trained to learn only discriminative features for differentiating between symbol classes. So, they cannot handle the writing-style variation very well. Our method matches the handwritten symbols in an HME image with printed symbols in the corresponding printed ME image by using the attention mechanism. Then our recognizer learns semantic-invariant features under the guidance of a discriminator.

Concretely, let $a(x, y, \theta_R) = (a_1, \ldots, a_T) \in \mathbb{R}^{T\times C}$ denote the output feature sequence of the last decoder block. Here, x is the input image of a handwritten ME x_h or its printed template x_p, $y = (y_0, \ldots, y_{T-1})$ denote the previous LaTeX targets and θ_R are the parameters of the recognizer R. The discriminator D guides the recognizer to learn semantic-invariant features by judging whether the feature vector a_t comes from the tth symbol in the handwritten ME image or its printed template. The probability that a_t comes from a printed image is calculated by $D(a_t(x_p, y_{0:t-1}, \theta_R), \theta_D)$, where $y_{0:t-1}$ is the abbreviation for $y_0, \ldots, y_{t-1}$ and θ_D are the parameters of D. The objective function is defined as:

$$\begin{aligned} \mathcal{L}_D = E_{(X,Y)} E_t (&\log D(a_t(x_p, y_{0:t-1}, \theta_R), \theta_D) \\ &+ \log(1 - D(a_t(x_h, y_{0:t-1}, \theta_R), \theta_D)), \end{aligned} \tag{13}$$

where $t \in \{1, \ldots, T\}$, and $(X, Y) = \{((x_h, x_p), y)\}$ indicates the training set of paired ME images. D is optimized to maximize $\mathcal{L}_D$, that is, maximize the probability of correctly classifying handwritten/printed image sources. On the contrary, R is trained to learn semantic-invariant features to confuse D. This optimization problem can be regarded as minimizing the loss that a_t is classified from a printed image:

$$\mathcal{L}_{D_{adv}} = -E_{(X,Y)} E_t (\log D(a_t(x_h, y_{0:t-1}, \theta_R, \theta_D)), \tag{14}$$

Moreover, the primary goal of the recognizer is to correctly identify each symbol in the input image. Thus, the output feature a_t at each decoding step should be classified as y_t with a high probability. The objective function of classifying features learned from the handwritten ME image is defined as:

$$\mathcal{L}_{C_h} = -E_{(X_h,Y)} \sum_{t=1}^{T} \log p(y_t | x_h; y_{0:t-1}; \theta_R), \tag{15}$$

where $(X_h, Y) = \{(x_h, y)\}$ is the training set of handwritten images and $p(y_t|x_h; y_{0:t-1}; \theta_R)$ is given by the y_t-th entry of $\hat{p}_t$ defined in Eq. (6).

Similarly, the loss function of classifying features learned from printed templates is defined as:

$$\mathcal{L}_{C_p} = -E_{(X_p,Y)} \sum_{t=1}^{T} \log p(y_t | x_p; y_{0:t-1}; \theta_R), \tag{16}$$

where $(X_p, Y) = \{(x_p, y)\}$ is the training set of printed templates.

In summary, we train the attentional recognizer by minimizing the loss function of:

$$\mathcal{L}_R = \mathcal{L}_{C_h} + \mathcal{L}_{C_p} + \lambda \mathcal{L}_{D_{adv}}, \tag{17}$$

where λ is a hyperparameter that controls the trade-off between semantic-invariant features and discriminative features. When $\lambda = 0$, the method is a general recognizer trained on paired samples. When λ increases, the method will focus more on learning semantic-invariant features and extract less discriminative features for the classification layer to generate prediction results.

Algorithm 1: Paired adversarial learning algorithm.

Get printed template x_p for x_h by compiling its label y to obtain the training set $((x_h, x_p), y) \in (X, Y)$;
Initialize the recognizer model and the discriminator randomly with parameters θ_R and θ_D;
repeat
  Sample bsz pairs of samples $\{(x_h, x_p)^{(1)}, \ldots, (x_h, x_p)^{(bsz)}\}$ from the training set;
  //Update the recognizer
  Update the recognizer by:
  $\theta_R \leftarrow \theta_R + optim(-\frac{\partial(\mathcal{L}_{C_h}+\mathcal{L}_{C_p}+\lambda\mathcal{L}_{D_{adv}})}{\partial\theta_R}, \eta_R)$;
  //Update the discriminator for m steps
  for *m steps* **do**
    Update the discriminator by:
    $\theta_D \leftarrow \theta_D + optim(\frac{\partial\mathcal{L}_D}{\partial\theta_D}, \eta_D)$;
  end
until $\mathcal{L}_{C_h} + \mathcal{L}_{C_p} + \lambda\mathcal{L}_{D_{adv}}$ *converged*;
//Get the final model for HMER
Parameterize the recognizer by: θ_R;
return *The recognizer R*;

3.4 Training Strategy

During training time, the recognizer R and the discriminator D are optimized jointly under the adversarial learning mechanism. D is trained to distinguish sequences of features extracted from images of handwritten MEs or their printed templates. On the contrary, the recognizer is trained to extract sophisticated semantic-invariant features for fooling D. Meanwhile, R is also optimized to maximize the probability of getting the right recognition results for the input ME images. The importance of these two objective functions is balanced via the hyperparameter λ.

See details in Algorithm 1. We sample minibatch of paired samples to train the recognizer and D for every training cycle. The recognizer model is trained one time first, and D is trained m times then. Specifically, we update parameters for the recognizer as:

$$\theta_R \leftarrow \theta_R + optim\left(-\frac{\partial\left(\mathcal{L}_{C_h} + \mathcal{L}_{C_p} + \lambda\mathcal{L}_{D_{adv}}\right)}{\partial\theta_R}, \eta_R\right). \tag{18}$$

And for the discriminator by:

$$\theta_D \leftarrow \theta_D + optim\left(\frac{\partial\mathcal{L}_D}{\partial\theta_D}, \eta_D\right), \tag{19}$$

where $optim(*)$ is the optimization function of the adaptive moment estimation (Adam) with the input gradient and learning rate and output the updated value, η_R denotes the learning rate for the recognizer and η_D denotes the learning rate of the discriminator, respectively.

3.5 Decoding with Statistical Language Model

Assume that the target sequence includes T symbols $(y_1, \ldots, y_T)$. A forward language model can measure the probability of the sequence by modeling the probability of symbol y_t for a given history $(y_1, \ldots, y_{t-1})$:

$$p(y_1, \ldots, y_T) = \prod_{t=1}^{T} p_{LM}(y_t|y_{1:t-1}). \tag{20}$$

For instance, an N-gram language model considers the previous $N-1$ symbols:

$$p(y_1, \ldots, y_T) = \prod_{t=1}^{T} p_{LM}(y_t|y_{t-N+1:t-1}). \tag{21}$$

When there is no external language model, the recognizer first embeds the target symbols and the input image to the same semantic space. Then, at each decoding step, the decoder of the recognizer makes a prediction of the current word through the recognized words and the attention mechanism:

$$p(y_1, \ldots, y_T|x; \theta_R) = \prod_{t=1}^{T} p_R(y_t|x; y_{0:t-1}; \theta_R). \tag{22}$$

In this work, we utilize an extra 4-gram statistical language model to assist decoding with the beam search algorithm (Cho 2015):

$$\begin{aligned}&\log\tilde{p}(y_1, \ldots, y_T|x; \theta_R)\\ &= \sum_{t=1}^{T} \log((1-\gamma)p_R(y_t|x; y_{0:t-1}; \theta_R)\\ &\quad + \gamma\, p_{LM}(y_t|y_{t-3:t-1})),\end{aligned} \tag{23}$$

where the value of γ is set by experiments and the output probabilities of the 4-gram statistical language model $p_{LM}(y_t|y_{t-3:t-1})$ are approximated by statistical data from the corpus.

4 Experiments

The proposed method is validated on the large public dataset available from the Competition on Recognition of Online Handwritten Mathematical Expressions (CROHME) (Mouchère et al. 2016a). We conduct extensive experiments to analyze the effectiveness of each part of our proposed model and compare the performance with the state-of-the-art approach.

4.1 Experimental Settings

4.1.1 Datasets

The CROHME dataset, which was collected for competition, is currently the largest public dataset for HMER. There are 101 math symbol classes in this dataset. The recent competitions include CROHME 2013, 2014 and 2016, which used different test sets while sharing the same training set. The shared training set contains 8835 handwritten MEs and the test sets for CROHME 2013, 2014 and 2016 contain 671, 986 and 1147 handwritten MEs, respectively. Consistent with the participants in CROHME, we use the test set of CROHME 2013 as a validation set during training and use the test sets of CROHME 2014 and 2016 to evaluate our proposed model.

In the CROHME dataset, each handwritten ME is stored in InkML format, which records the trajectory coordinates of handwritten strokes as well as the LaTeX and MathML format ground truth. In this study, we use the LaTeX format ground truth as other works presented recently (Deng et al. 2016; Le and Nakagawa 2017; Zhang et al. 2017a, 2018, 2017b). Since our model is proposed for offline ME recognition, we did not use the online trajectory information of strokes (Zhang et al. 2019), but instead, we transformed the MEs to offline images by connecting adjacent coordinate points of the same strokes.

As the training of PAL-v2 needs the printed templates of the handwritten ME images, we generated the printed template of each handwritten ME simply by compiling the LaTeX format label with a general LaTeX editor. All the ME images were normalized to the height of 64 pixels. Our models were implemented in Torch and optimized on 4 Nvidia TITAN X GPUs. The batch size was set as 5 for each GPU. In each batch, after images are centered, short images are zero-padded to the length of the longest image. Target sequences whose lengths are short in the batch are padded with constant ⟨P⟩ at the end. The category ⟨P⟩ does not participate in the calculation of cross-entropy loss.

4.1.2 Model Configurations

The configurations of our proposed model are listed in Table 1. We use DenseNet (Huang et al. 2017) as the CNN feature extractor of the encoder. In Table 1, "G" denotes the growth rate and "s" denotes the stride. Then a DenseMD block is added after DenseNet to extract more context information. Different from the original work (Huang et al. 2017), we do not adopt the down sampling method with a 2×2-stride convolution and a subsequent 2×2-stride max pooling for input images. We use three 1×1-stride convolution and two 2×2-stride max pooling before the first dense block, so that the information in the input image is not lost too much. Furthermore, the convolutional layer of the encoder is equipped with a batch normalization layer (Ioffe and Szegedy 2015) and a rectified linear unit (ReLU) (Krizhevsky et al. 2012).

Table 1 Configurations of the proposed model

Input: $H(64) \times W \times D(1)$ binary image	
Encoder	
Convolution	3×3 conv, 32, s 1×1
Convolution	3×3 conv, 32, s 1×1
Transition layer	2×2 max pooling, s 2×2
Convolution	3×3 conv, 48, s 1×1
Transition layer	2×2 max pooling, s 2×2
Dense block (1)	$\begin{bmatrix} 1 \times 1, \text{conv} \\ 3 \times 3, \text{conv} \end{bmatrix} \times 14$, G 24, s 1×1
Transition layer	1×1 conv, 128, s 1×1
	2×2 average pooling, s 2×2
Dense block (2)	$\begin{bmatrix} 1 \times 1, \text{conv} \\ 3 \times 3, \text{conv} \end{bmatrix} \times 16$, G 24, s 1×1
Transition layer	1×1 conv, 256, s 1×1
Dense block (3)	$\begin{bmatrix} 1 \times 1, \text{conv} \\ 3 \times 3, \text{conv} \end{bmatrix} \times 8$, G 24, s 1×1
Transition layer	1×1 conv, 224, s 1×1
DenseMD block	$\begin{bmatrix} 1 \times 1, \text{conv} \\ \text{MDLSTM} \end{bmatrix} \times 6$, G 8, s 1×1
Transition layer	MDLSTM, 400, s 1×1
Position ResBlock	2D position embedding
	[3×3 conv] $\times 4$, 400, s 1×1
Decoder	
Decoder block	[3, causal conv] $\times$ 3, 256, s 1
FC layer	256 units
FC layer	K units, softmax
Discriminator	
FC layer	512 units
FC layer	1 unit, sigmoid

Consistent with the method in Gehring et al. (2017), each causal convolution in the decoder block is equipped with a gated linear unit (GLU) (Dauphin et al. 2017) for nonlinear activation, and a weight normalization layer (Salimans and Kingma 2016) is implemented as a regularization measure. Channel dimensions of the feature map output by the encoder are mapped to the same as the hidden state before calculating attention. The discriminator D is an MLP with two fully connected (FC) layers.

We employ dropout (Srivastava et al. 2014) to prevent overfitting. Specially, for the 3×3 convolutional layer, DropBlock (Ghiasi et al. 2018) is used instead. Weight noise (Graves 2011) and weight decay (Krogh and Hertz 1991) are also implemented as regularization during training procedure.

Table 2 Performance of the recognizer on handwritten datasets and generated printed datasets

ExpRate (%)	CROHME 2014	CRHOME 2016
Handwritten	43.81	43.77
Printed	83.87	81.60

4.2 Exploratory Experiments of Paired Adversarial Learning

4.2.1 Discriminative Feature Learning

We first train the recognizer with handwritten MEs only and evaluate the performance of the recognizer on the CHROME 2014 and 2016 test sets. Then for comparing handwritten ME recognition and printed ME recognition, we train a new recognizer on the generated printed training set and evaluate the performance on the printed test sets. The recognition results are shown in Table 2. The performance is measured by ExpRate, defined as the percentage of correctly recognized expressions. ExpRate is the index that ranks the participating systems in the CROHME competitions.

It is shown in Table 2 that although the handwritten MEs and printed templates have the same contents, there is a big difference in recognition accuracy when two recognizer are trained separately to learn discriminative features. In this case, the ExpRates on the handwritten CROHME 2014 and 2016 datasets are nearly 40% lower than those on printed ones.

4.2.2 Influence of Hyperparameter λ

We run experiments with different values of the hyperparameter λ in Eq. (17) to examine its effect on our proposed PAL-v2 model. Models are all optimized with paired images. We explore different λ for the model while keeping other configurations fixed. The recognition results on CROHME 2014 and 2016 test sets are shown in Fig. 5.

The hyperparameter λ controls the trade-off between semantic-invariant features and discriminative features learned by the recognizer. When λ is small, discriminative features comprise the majority loss of the recognizer and dominate gradients back propagated. With increasing λ, the recognizer masters more semantic-invariant features of same symbols in handwritten ME images and their printed templates. However, with too large λ, the model will focus too much on semantic-invariant features learning and even try to generate same feature sequences for both printed and handwritten ME images to confuse the discriminator. This will lead to less discriminative features for different symbol categories and deteriorate the recognition accuracy. For an extreme case, the recognizer may only pay attention to regions of background or other irrelevant regions at each step to fool the discriminator D. Therefore, an appropriate λ plays an important role in the PAL-v2 model.

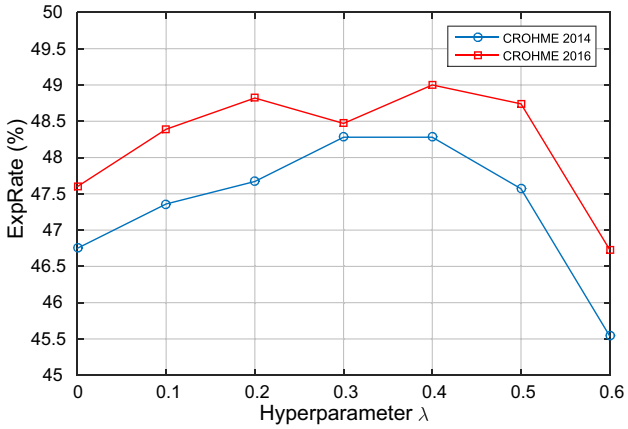


Fig. 5 Comparison of different λ on CROHME 2014 and 2016 test sets

4.2.3 Analysis of Loss Function

In this section, we remove different components in the loss function to examine the effectiveness of printed templates and style discrimination in the learning procedure. The recognition results using difference combinations of loss functions are is listed in Table 3, where ticks indicate that the corresponding loss function terms are selected when training the model. Model configurations in Table 3 are consistent with those in the Table 1 and λ is fixed as 0.4. In fact, when only the supervision loss $\mathcal{L}_{C_h}$ is used, it is equivalent to training the traditional recognizer on handwritten MEs only as in Table 2. When $\mathcal{L}_{C_p}$ is not used, it is hard to ensure that features of printed ME images at each decoding step are matched with handwritten ones. One can observe that the training with $\mathcal{L}_{C_h} + \mathcal{L}_{D_{adv}}$ leads to a 1% accuracy improvement over the original recognizer.

Interestingly, when we optimize the recognizer on paired printed and handwritten images with $\mathcal{L}_{C_h}+\mathcal{L}_{C_p}$, the accuracy is increased by 2% more on CROHME 2014 test set and 3% more on CROHME 2016 test set. Traditional recognizers with $\mathcal{L}_{C_h}$ only are trained on weakly labeled handwritten ME images. It is difficult for the recognizers to learn the 2D structural relations due to the variations of writing style and layout. In contrast, the printed templates have more stable shape appearance and can help the recognizer to learn the 2D structure of the MEs and therefore improve the recognition accuracy.

Actually, *mismatch* between two sequences of feature vectors can easily cause the discriminator converging to irrelevant features, and thus losing the ability to guide the recognizer for learning semantic-invariant features. Therefore, we utilize the supervision loss $\mathcal{L}_{C_p}$ to enable the recognizer to

Table 3 Analysis of the loss function for PAL-v2

$\mathcal{L}_{C_h}$	$\mathcal{L}_{C_p}$	$\mathcal{L}_{D_{adv}}$	CROHME 2014	CROHME 2016
√	-	-	43.81	43.77
√	-	√	44.83	45.60
√	√	-	46.75	47.60
√	√	√	48.20	49.00

The accuracy is measured by ExpRate (%)

Table 4 Configurations of different discriminators

Discriminator 1	
Convolution	3 ×3 conv, 256, s 1 × 1
Pooling layer	2 ×2 max pooling, s 2 × 2
Convolution	3 ×3 conv, 256, s 1 × 1
Convolution	3 ×3 conv, 1, s 1 × 1
Pooling layer	Global average pooling, sigmoid
Discriminator 2	
Pooling layer	Global average pooling
FC layer	512 units
FC layer	1 unit, sigmoid
Discriminator 3	
FC layer	512 units
FC layer	1 unit, sigmoid

Table 5 Performance of recognizers trained with different discriminators

ExpRate(%)	CROHME 2014	CRHOME 2016
Without discriminator	46.75	47.60
$D1$ after encoder	47.36	48.65
$D2$ after encoder	47.16	48.56
$D3$ after attention	48.20	49.00

extract precise features by matching with printed templates. At the same time, we use $\mathcal{L}_{D_{adv}}$ to guide it to learn semantic invariant features. The performance of the recognizer model has been further improved by about 1.5% on both two test sets. Compared with training the recognizer with $\mathcal{L}_{C_h}$ only, PAL boosts the performance of the recognizer more than 5% on the CROHME 2016 test set.

4.2.4 Comparison of Different Discriminators

We employ the discriminators at different stages of the recognizer and consider different capacities of the discriminators (See Table 4). In Table 5 we report the performance of recognizers trained with these discriminators.

Recognizers in Table 5 are all trained with paired images. $D1$ and $D2$ are applied to the same stage of the recognizer (the feature map $e + f$), but $D1$ owns a higher capacity than $D2$. The difference between $D2$ and $D3$ is that $D2$ gets the image vector through a global average pooling layer instead of the attention mechanism. Since the ME images have variable scales, we use global average pooling to get the image vector for discriminators applied on earlier feature maps.

From Table 5, we can observe the following results: (1) adding discriminator for guiding the recognizer to learn semantic invariant features benefits the recognition. (2) Discriminator with higher capacity leads to better performance. (3) Attending to specific symbols of ME images with complicated 2D structures works better than applying global average pooling to get the image vector.

Although a discriminator with higher capacity gives improvement on the performance, we noticed that strong discriminators tend to suffer from the problem of gradient disappearance during training. To ensure the stability of the training process, we set the discriminator and classifier to the same number of layers.

4.2.5 Visualization of Features

To show the inner working of our proposed paired adversarial learning method, we visualize the features of different symbol classes learned by the recognizer in Fig. 6. Specifically, we select 9 common symbols from both CROHME 2014 test set and the generated printed set, including Arabic numerals, English and Greek letters. We input all the previous target symbols to the decoder to predict the current symbols and then visualize the features before the first FC layer of the classifier. The 256D features are reduced to 2D by t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton 2008).

Figure 6 shows that by training the traditional recognizer, there is significant overlap between the features of different handwritten symbol classes in (a). In contrast, features of different printed symbol classes learned by the recognizer have better separability between symbol classes. By training the recognizer with PAL, handwritten features are drawn closer to their corresponding printed templates in the feature space (see details in (b)) and become more separable.

4.3 Analysis of Pre-aware Coverage Attention

4.3.1 Effect of Pre-aware Coverage Attention

To verify the effect of Pre-aware Coverage Attention (PCA) in our proposed model, we compare the recognition results with different decoders in Table 6. All the models are trained to learn semantic-invariant features on paired ME images via PAL.

The recognizer model with classic attention based convolutional decoder (see "CA" in Table 6) achieves ExpRate

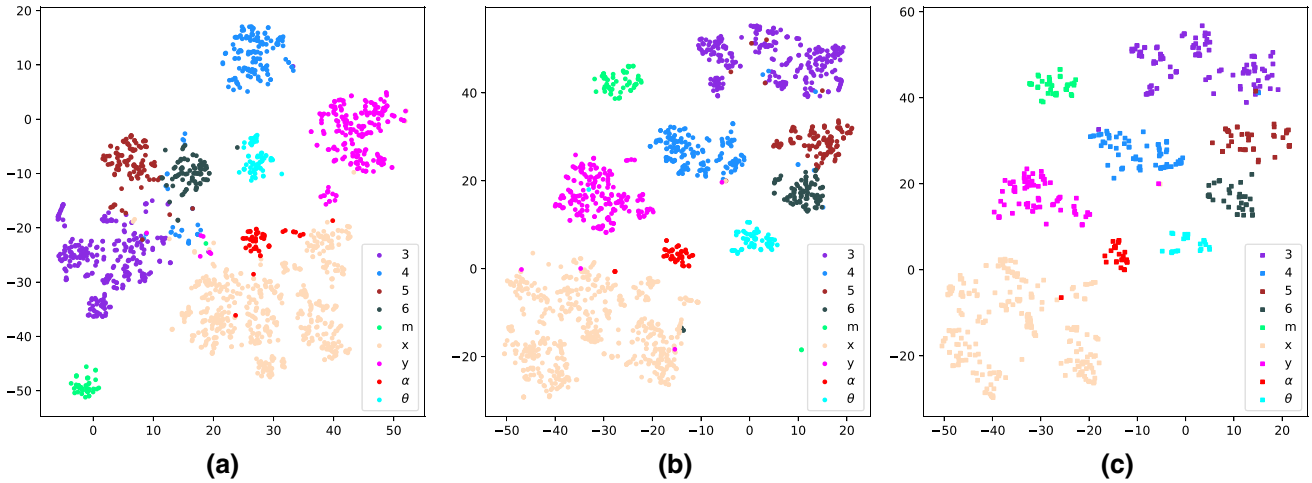


Fig. 6 Visualization of features learned with PAL. From (**a**) to (**c**) are the handwritten features learned by the traditional recognizer, the handwritten features learned by the recognizer trained with PAL, and the printed features learned by the recognizer trained with PAL, respectively. Feature vectors are reduced to 2D for visualization

Table 6 Recognition results of recognizer models with different decoders

ExpRate (%)	CROHME 2014	CRHOME 2016
CA	45.13	46.47
PCA	48.20	49.00
PCA Norm	46.96	47.70
PCA noRes	47.77	49.00

CA: classic attention based convolutional decoder; PCA: pre-aware coverage attention based decoder; PCA Norm: decoder based on PCA with normalized forecast information; PCA noRes: decoder based on PCA without the residual connection.

45.13% and 46.47% on CROHME 2014 and 2016 test sets, respectively. After we equip the recognizer model with "PCA", the performance has been significantly improved with about 3% on both two test sets.

Table 6 also gives the results of two variants of "PCA", namely "PCA Norm" and "PCA noRes". "PCA Norm" differs from "PCA" in the forecast information of attention weights for all previous steps. Specifically, for "PCA Norm", the forecast information is normalized by multiplying the tth row of W_{pa} in Eq. (10) with $1/(t-1), t = 2, \ldots, T$. Although "PCA Norm" has outperformed "CA" on both two test sets, its perception of previous steps is significantly worse than that of "PCA". "PCA noRes" is obtained by removing the residual connection in the pre-aware unit of "PCA" in Eq. (12). The recognition performance with this decoder turns out to be affected only slightly. Nevertheless, we still retain the residual connection, because the recognizer model sometimes has a gradient explosion problem after removing the residual connection.

It is also interesting to investigate the performance of recognizers with "CA" and "PCA" on MEs of different target lengths. The experimental results are shown in Fig. 7. From the histogram, we can see that majority of the MEs in the CHROME dataset do not exceed 30 symbols. Generally, MEs with longer target sequence are more likely to own larger widths and more complex structures. These MEs make the recognizer model tend to over-attention and under-attention. With the appending of PCA, it is observed that for most target length intervals the recognition accuracy is increased on both two test sets. More interestingly, on CROHME 2014 test set, after equipping PCA, some handwritten MEs with more than 40 symbols are correctly identified. In other words, by equipping PCA, the model can more accurately attend to the relevant areas in decoding.

4.3.2 Attention Visualization

Attending to right areas in ME images is essential for learning semantic-invariant features and discriminative features. We visualize PCA during test procedure to illustrate the recognition process of PAL-v2 model in Fig. 8. Since there are three decoder blocks in the decoder, we average the three attention maps at each decoding step and visualize the averaged weights. Attention weights are visualized in red and darker red denotes a higher weight in the attention map.

To analyze the 2D structure of a handwritten ME, target symbols and input images are mapped to the same semantic space. At each decoding step, the decoder has made a rough prediction of the current symbol through previous recognized ones. Then, the decoder utilizes the predicted state to attend to relevant regions for fine prediction with attention mechanism. As shown in Fig. 8, entity symbols such as Arabic

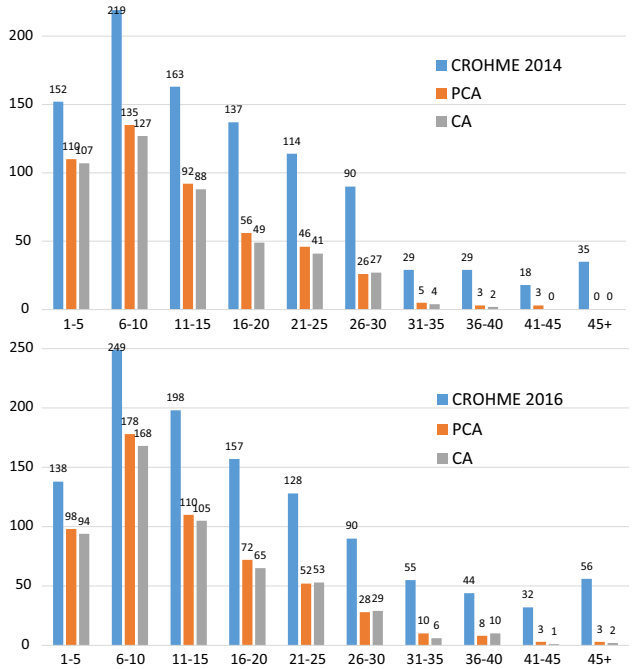


Fig. 7 Performance on handwritten MEs of different target lengths. The horizontal axis is the length of the LaTeX target and the vertical axis is the number of MEs. Blue rectangle represents total MEs of the test set for the length interval. Orange rectangle and gray rectangle indicate MEs correctly recognized by recognizer models with PCA and CA, respectively (Color figure online)

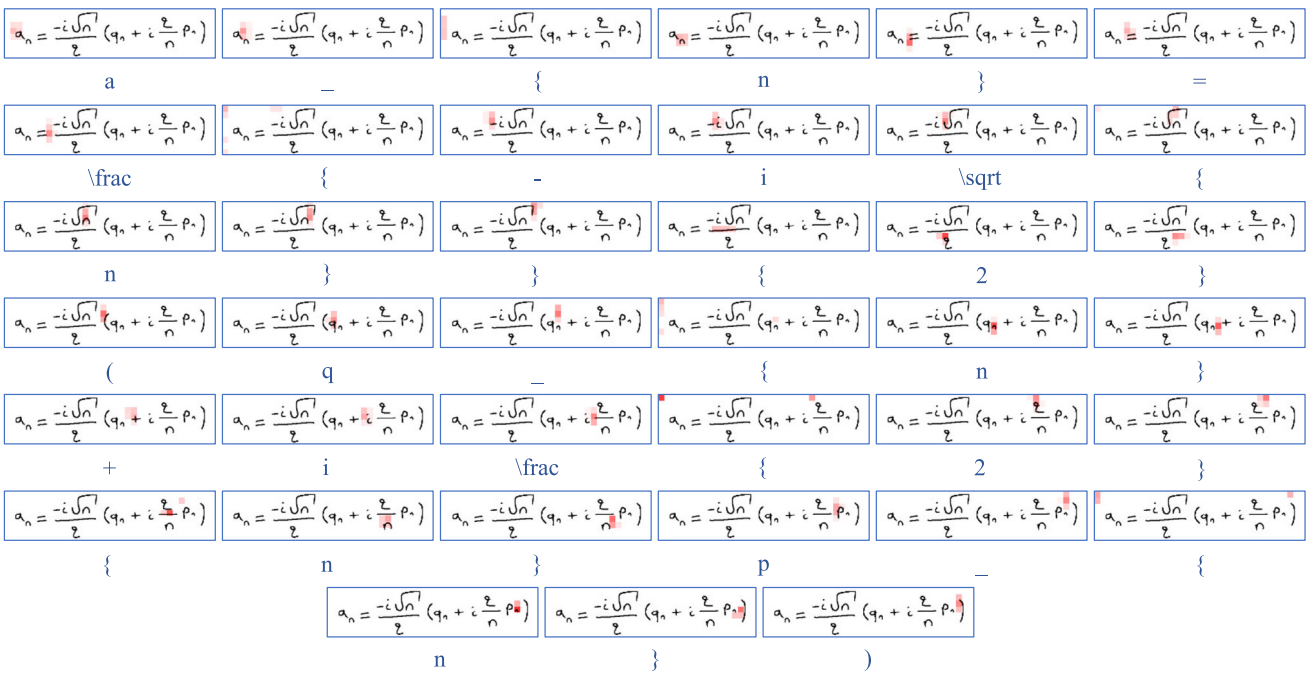
numerals and English letters, like "2" and "a", are easy to learn by focusing attention to symbols written in the image. Moreover, by paying attention to some special locations, spatial relation operators are successfully parsed by PAL-v2. For example, when decoding out subscripts in Fig. 8, PAL-v2 has attended to bottom-right directions of the base. For internal relationships such as root number, PAL-v2 has attended to the upper right border of the inner symbol "n".

In fact, there are many "{" and "}" in LaTeX format labels which do not have corresponding entities in the ME images. From Fig. 8, we can see that when predicting symbols without entities, PAL-v2 pays attention to some background areas of the image and generates them with the embedded language model in the decoder. Our experimental results show that PAL-v2 can indeed learn the LaTeX grammar through data and parse the 2D structure of the MEs like humans do.

Through visualization, we can see that our proposed model can accurately focus on the symbols in the ME image. Even for a ME with as many as 39 target symbols, it still shows good performance.

4.4 Ablation Experiments

Table 7 shows the results of ablation experiments. Specifically, we append each part to the previous system sequentially for verifying the effectiveness of each part of our proposed model. "DenseNet" denotes the benchmark recognizer model



Ground truth: a _ { n } = \frac { - i \sqrt { n } } { 2 } (q _ { n } + i \frac { 2 } { n } p _ { n })

Fig. 8 Visualization of attention. Attention weights at each step are visualized in red and predicted symbols of the corresponding step the predicted symbol of the corresponding step is shown under the handwritten ME image (Color figure online)

Table 7 Ablation experiments on CROHME 2014 on 2016

ExpRate (%)	CROHME 2014	CROHME 2016
DenseNet	43.10	44.64
+ DenseMD	44.83	45.51
+ Position Embedding	44.22	45.86
+ PCA	46.75	47.60
+ Discriminator	48.28	49.00
+ Language Model	48.78	49.35

We conduct experiments by appending each portion of PAL-v2 to previous systems sequentially

with a DenseNet and a subsequent single layer MDLSTM as encoder. The decoder is the classic attention based convolutional decoder with 3 decoder blocks. Sign "+" in Table 7 indicates adding the part to previous systems. All systems are trained on paired ME images.

First, "+ DenseMD" applies a densely connected MDLSTM block before the single layer MDLSTM of the encoder. The recognition accuracy is increased by 1% on both CROHME 2014 and 2016 test sets. However, unlike the original work (Gehring et al. 2017), there is no significant change in performance after "+ Position Embedding". This might be caused by the factor that MDLSTM already enables the model to learn where the feature vectors are in the image through the recurrent hidden state computation. Then "+ PCA" appends the pre-aware coverage attention to the convolutional decoder, and brings an accuracy improvement by more than 2% on both two test sets. Next, "+ Discriminator" indicates adding the discriminator to guide the recognizer for learning semantic-invariant features. By adding this, the performance of the model has been further improved with by 1.5% ExpRate. Finally, the added extra statistical language model uses only LaTeX format targets of the training set as corpus and γ in Eq. (23) is set as 0.1. It is observed that the statistical language model is still helpful for the recognition, although there is already a neural network language model embedded in the decoder.

4.5 Comparison with the State-of-the-Art

Table 8 shows the results of our proposed model with comparison with the submitted systems from CROHME 2014 and attention-based models presented recently. Systems I to VII are participating systems in the competition and the next few systems are attention-based models proposed for HMER recently. To make fair comparison, system III is excluded from Table 8 because it used unofficial extra training data. A recently proposed encoder-decoder model named "TAP" (Zhang et al. 2019) is not included in Tables 8 and 9 since it used an extra math corpus to train the RNN based language model and utilized symbol-level annotations as strong supervision. The attentional recognizer models listed in Table 8 are all trained on offline ME images with only LaTeX level labels. It is worth noting that the ExpRate reported by the participating systems in the CROHME competitions is calculated with a hierarchical graph, named label graph, which also considers the alignment accuracy. Therefore, in Tables 8 and 9, we convert the output LaTeX strings to label graphs and evaluate the performance with official tools provided by the CROHME 2019 (Mahdavi et al. 2019) organizers. ExpRate $\leq 1(\%)$, $\leq 2(\%)$ and $\leq 3(\%)$ denote the expression recognition rates when one, two or three symbol-level errors are tolerable. They show the room for the models to be further improved.

Table 8 ExpRate (%) of different systems on CROHME 2014 test set

System	ExpRate (%)	$\leq 1(\%)$	$\leq 2(\%)$	$\leq 3(\%)$
I	37.22	44.22	47.26	50.20
II	15.01	22.31	26.57	27.69
IV	18.97	28.19	32.35	33.37
V	18.97	26.37	30.83	32.96
VI	25.66	33.16	35.90	37.32
VII	26.06	33.87	38.54	39.96
WYGIWYS* (Deng et al. 2016)	28.70	–	–	–
End-to-end (Le and Nakagawa 2017)	35.19	–	–	–
WAP* (Zhang et al. 2017b)	44.40	58.40	62.20	63.10
PAL (Wu et al. 2018b)	39.66	–	–	–
PAL* (Wu et al. 2018b)	47.06	–	–	–
DenseMSA* (Zhang et al. 2018)	52.80	68.10	72.00	72.70
PAL-v2	48.88	64.50	69.78	73.83
PAL-v2*	54.87	70.69	75.76	79.01

The sign "$*$" in Table 8 denotes utilizing an ensemble of 5 differently initialized recognizer models to improve the generation performance (Zhang et al. 2017b). The state-of-the-art model "DenseMSA" (Zhang et al. 2018) uses DenseNet as the encoder and adds an extra DenseNet branch to deal with different sizes of symbols. Then the output feature maps of the encoder are decoded with a recurrent decoder. Our proposed PAL-v2 model have not used an extra branch for the encoder and outperforms DenseMSA (Zhang et al. 2018) by about 2% ExpRate.

Table 9 compares our proposed model with the participating systems in CROHME 2016 and other models proposed for HMER recently. System "Wiris" won the first place in CROHME 2016 using only the official handwritten MEs training data. However, it used a extra Wikipedia formula corpus to train the language model for assisting recognition. The state-of-the-art model "DenseMSA" is the same as that

Table 9 ExpRate (%) of different systems on CROHME 2016 test set

System	ExpRate (%)	≤ 1 (%)	≤ 2 (%)	≤ 3 (%)
Wiris	49.61	60.42	64.69	–
Tokyo	43.94	50.91	53.70	–
Sao Paolo	33.39	43.50	49.17	–
Nantes	13.34	21.02	28.33	–
WAP* (Zhang et al. 2017b)	44.55	57.10	61.55	62.34
DenseMSA* (Zhang et al. 2018)	50.10	63.80	67.40	68.50
PAL-v2	49.61	64.08	70.27	73.50
PAL-v2*	57.89	70.44	76.29	79.16

in Table 8. It does not use extra language model and shows a slight advantage over "Wiris". To make fair comparison, in both Tables 8 and 9, we get the statistical language model by using only the training set ground truths. Despite this, our proposed model still yields excellent recognition performance and outperforms "Wiris" and "DenseMSR" with a large margin.

Overall, our proposed PAL-v2 model achieves state-of-the-art performance on both CROHME 2014 and 2016 test sets, and still shows a huge room for further improvement.

4.6 Recognition Examples

Though PAL-v2 achieved state-of-the-art performance on CROHME 2014 and 2016 test sets, the results on the handwritten MEs are still far behind the performance on the printed MEs in Table 2. We show some correctly and incorrectly recognized handwritten MEs by the PAL-v2 in Fig. 9 to further analyze the reasons.

In the figure, red symbols of the "reco" are wrongly predicted symbols and blue symbols indicate the corresponding right one in the LaTeX format ground truth. The results show our proposed model is effective in dealing with the complex 2D structures and symbols with various writing styles. For some symbols overlapping or touching with others, wrong symbols may be predicted by the recognizer. For example, the "gamma" connected to the division line in the third ME image is incorrectly identified as a root number. Adhesion of the x to its subscript 1 in the fourth image results in the missing of the subscript.

In addition to the overlapping problem, symbols of the same glyph are often misidentified. These symbols include letters with similar uppercase and lowercase, such as S and C, and some similarly shaped characters such as 9, q and g. This problem is clearly reflected in the fifth image. Besides, excessively skewed symbols may also lead to incorrect recognitions. Symbol 9 in the root number of the last ME image

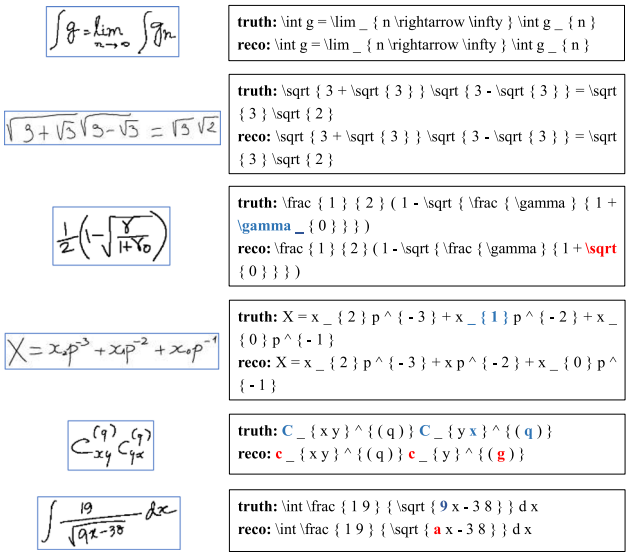


Fig. 9 Examples of handwritten MEs correctly recognized and wrongly recognized. Errors in the generated LaTeX codes are marked in red (Color figure online)

is written with large counterclockwise rotation and is incorrectly identified as an a.

As mentioned in Sect. 4.2.3, compared with the printed MEs, handwritten MEs suffer from serious distortion of symbol shapes and layout. This makes it difficult to train the recognizer with weakly labeled images. Since the misrecognition of any symbol or structural relation results in misrecognition of the whole expression, the recognition accuracy of the handwritten MEs is much lower than that of printed ones. Nevertheless, considering the symbol error tolerable rate, the ExpRate ≤ 3(%) can still be close to 80% in Tables 8 and 9.

5 Conclusion

This paper addresses the problem of handwritten mathematical expression recognition with a novel paired adversarial learning method. The proposed model, called PAL-v2, shows superior performance in dealing with the writing-style variation by learning both semantic-invariant features and discriminative features. Besides, owing to the pre-aware coverage attention mechanism, PAL-2 can effectively parse the 2D spatial structures, although the training ME images have only weak labels in LaTeX format.

Through extensive experiments and ablation study, we demonstrate that the proposed PAL-v2 outperforms the state-of-the-art methods on the public datasets CROHME 2014 and 2016, and justify that the proposed paired adversarial learning method and pre-aware coverage attention are effective to improve the performance. In our future work, we plan to

achieve accurate positioning of symbols of handwritten MEs under weak supervision conditions, and further improve the accuracy and interpretability of the model.

Acknowledgements This work has been supported by the National Key Research and Development Program Grant 2018YFB1005000, the National Natural Science Foundation of China (NSFC) Grants 61721004, 61733007, 61773376, 61633021, 61836014, and the Beijing Science and Technology Program Grant Z181100008918010.

References

Alvaro, F., Sánchez, J., & Benedí, J. (2014). Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden Markov models. *Pattern Recognition Letters*, *35*, 58–67.

Alvaro, F., Sánchez, J., & Benedí, J. (2016). An integrated grammar-based approach for mathematical expression recognition. *Pattern Recognition*, *51*, 135–147.

Anderson, R. H. (1967). 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* (pp. 436–459). ACM.

Aneja, J., Deshpande, A., & Schwing, A. G. (2018). Convolutional image captioning. In *2018 IEEE conference on computer vision and pattern recognition, CVPR 2018*, Salt Lake City, UT, USA, 18–22 June 2018, pp. 5561–5570.

Awal, A., Mouchère, H., & Viard-Gaudin, C. (2014). A global learning approach for an online handwritten mathematical expression recognition system. *Pattern Recognition Letters*, *35*, 68–77.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. CoRR arXiv:1409.0473.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. CoRR arXiv:1803.01271.

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., & Krishnan, D. (2017). Unsupervised pixel-level domain adaptation with generative adversarial networks. In *2017 IEEE conference on computer vision and pattern recognition, CVPR 2017*, Honolulu, HI, USA, 21–26 July 2017, pp. 95–104.

Chan, K., & Yeung, D. (2000). Mathematical expression recognition: A survey. *IJDAR*, *3*(1), 3–15.

Chan, K., & Yeung, D. (2001). Error detection, error correction and performance evaluation in on-line mathematical expression recognition. *Pattern Recognition*, *34*(8), 1671–1684.

Cho, K. (2015). Natural language understanding with distributed representation. CoRR arXiv:1511.07916.

Cho, K., Courville, A. C., & Bengio, Y. (2015). Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, *17*(11), 1875–1886.

Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in neural information processing systems 28: Annual conference on neural information processing systems 2015*, 7–12 December 2015, Montreal, Quebec, Canada, pp. 577–585.

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). Language modeling with gated convolutional networks. In *Proceedings of the 34th international conference on machine learning, ICML 2017*, Sydney, NSW, Australia, 6–11 August 2017, pp. 933–941.

Deng, Y., Kanervisto, A., Ling, J., & Rush, A. M. (2017). 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.

Deng, Y., Kanervisto, A., & Rush, A. M. (2016). What you get is what you see: A visual markup decompiler. CoRR arXiv:1609.04938.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *Proceedings of the 34th international conference on machine learning, ICML 2017*, Sydney, NSW, Australia, 6–11 August 2017, pp. 1243–1252.

Ghiasi, G., Lin, T., & Le, Q. V. (2018). Dropblock: A regularization method for convolutional networks. In *Advances in neural information processing systems 31: Annual conference on neural information processing systems 2018, NeurIPS 2018*, 3–8 December 2018, Montréal, Canada, pp. 10750–10760.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *Advances in neural information processing systems 27: Annual conference on neural information processing systems 2014*, 8–13 December 2014, Montreal, Quebec, Canada, pp. 2672–2680.

Graves, A. (2011). Practical variational inference for neural networks. In *Advances in neural information processing systems 24: 25th annual conference on neural information processing systems 2011*. Proceedings of a meeting held 12–14 December 2011, Granada, Spain, pp. 2348–2356.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition, CVPR 2016*, Las Vegas, NV, USA, 27–30 June 2016, pp. 770–778.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition, CVPR 2017*, Honolulu, HI, USA, 21–26 July 2017, pp. 2261–2269.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd international conference on machine learning, ICML 2015*, Lille, France, 6–11 July 2015, pp. 448–456.

Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2016). Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, *116*(1), 1–20.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., et al. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, *123*(1), 32–73.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems 25: 26th annual conference on neural information processing systems 2012*. Proceedings of a meeting held 3–6 December 2012, Lake Tahoe, Nevada, USA, pp. 1106–1114.

Krogh, A., & Hertz, J. A. (1991) A simple weight decay can improve generalization. In *Advances in neural information processing systems 4*, [NIPS Conference, Denver, Colorado, USA, 2–5 December 1991], pp. 950–957.

Le, A. D., & Nakagawa, M. (2017) Training an end-to-end system for handwritten mathematical expression recognition by generated patterns. In *14th IAPR international conference on document analysis and recognition, ICDAR 2017*, Kyoto, Japan, 9–15 November 2017, pp. 1056–1061.

Li, L., Tang, S., Deng, L., Zhang, Y., & Tian, Q. (2017). Image caption with global-local attention. In *Proceedings of the thirty-first AAAI conference on artificial intelligence*, 4–9 February 2017, San Francisco, California, USA, pp. 4133–4139.

Liu, Y., Wang, Z., Jin, H., & Wassell, I. J. (2018). Synthetically supervised feature learning for scene text recognition. In *Computer vision—ECCV 2018—15th European Conference*, Munich, Germany, 8–14 September 2018, Proceedings, Part V, pp. 449–465.

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*, 2579–2605.

MacLean, S., & Labahn, G. (2013). A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets. *IJDAR*, *16*(2), 139–163.

Mahdavi, M., Zanibbi, R., Mouchere, H., & Garain, U. (2019). ICDAR 2019 CROHME+ TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection. *ICDAR: In Proc*.

Mouchère, H., Viard-Gaudin, C., Zanibbi, R., & Garain, U. (2016a). ICFHR2016 CROHME: Competition on recognition of online handwritten mathematical expressions. In *15th international conference on frontiers in handwriting recognition, ICFHR 2016*, Shenzhen, China, 23–26 October 2016, pp. 607–612.

Mouchère, H., Zanibbi, R., Garain, U., & Viard-Gaudin, C. (2016b). Advancing the state of the art for handwritten math recognition: The CROHME competitions, 2011–2014. *IJDAR*, *19*(2), 173–189.

Ordonez, V., Han, X., Kuznetsova, P., Kulkarni, G., Mitchell, M., Yamaguchi, K., et al. (2016). Large scale retrieval and generation of image descriptions. *International Journal of Computer Vision*, *119*(1), 46–59.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR arXiv:1511.06434.

Salimans, T., & Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in neural information processing systems 29: Annual conference on neural information processing systems 2016*, 5–10 December 2016, Barcelona, Spain, p. 901.

Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., & Bai, X. (2018). Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *1*(99), 1.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Su, J., Carreras, X., & Duh, K. (Eds.). (2016). *Proceedings of the 2016 conference on empirical methods in natural language processing, EMNLP 2016*, Austin, Texas, USA, 1–4 November 2016. The Association for Computational Linguistics.

Tu, Z., Lu, Z., Liu, Y., Liu, X., & Li, H. (2016). Modeling coverage for neural machine translation. In *Proceedings of the 54th annual meeting of the association for computational linguistics, ACL 2016*, 7–12 August 2016, Berlin, Germany, Volume 1: Long Papers.

Wu, Y., Yin, F., Zhang, X., Liu, L., & Liu, C. (2018a). SCAN: Sliding convolutional attention network for scene text recognition. CoRR arXiv:1806.00578.

Wu, J., Yin, F., Zhang, Y., Zhang, X., & Liu, C. (2018b). Image-to-markup generation via paired adversarial learning. In *Machine learning and knowledge discovery in databases—European Conference, ECML PKDD 2018*, Dublin, Ireland, 10–14 September 2018, Proceedings, Part I, pp. 18–34.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., et al.: Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd international conference on machine learning, ICML 2015*, Lille, France, 6–11 July 2015, pp. 2048–2057.

Zanibbi, R., & Blostein, D. (2012). Recognition and retrieval of mathematical expressions. *IJDAR*, *15*(4), 331–357.

Zhang, J., Du, J., & Dai, L. (2017a) A gru-based encoder-decoder approach with attention for online handwritten mathematical expression recognition. In *14th IAPR international conference on document analysis and recognition, ICDAR 2017*, Kyoto, Japan, 9–15 November 2017, pp. 902–907.

Zhang, J., Du, J., & Dai, L. (2018). Multi-scale attention with dense encoder for handwritten mathematical expression recognition. In *24th international conference on pattern recognition, ICPR 2018*, Beijing, China, 20–24 August 2018, pp. 2245–2250.

Zhang, J., Du, J., & Dai, L. (2019). Track, attend, and parse (TAP): An end-to-end framework for online handwritten mathematical expression recognition. *IEEE Transactions on Multimedia*, *21*(1), 221–233.

Zhang, J., Du, J., Zhang, S., Liu, D., Hu, Y., Hu, J., et al. (2017b). Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition. *Pattern Recognition*, *71*, 196–206.

Zhang, Y., Liang, S., Nie, S., Liu, W., & Peng, S. (2018). Robust offline handwritten character recognition through exploring writer-independent features under the guidance of printed data. *Pattern Recognition Letters*, *106*, 20–26.

Zhou, X., Wang, D., Tian, F., Liu, C., & Nakagawa, M. (2013). Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(10), 2413–2426.