Pattern Recognition 119 (2021) 108047

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Stroke constrained attention network for online handwritten mathematical expression recognition

Jiaming Wang[a], Jun Du[a,*], Jianshu Zhang[a], Bin Wang[b], Bo Ren[b]

[a] National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei, Anhui, China
[b] Youtu Lab, Tencent

ARTICLE INFO

Article history:
Received 19 February 2020
Revised 22 March 2021
Accepted 10 May 2021
Available online 25 May 2021

Keywords:
Stroke-level information
Multi-modal fusion
Encoder-decoder
Attention mechanism
Handwritten mathematical expression recognition

ABSTRACT

In this paper, we propose a novel stroke constrained attention network (SCAN) which treats stroke as the basic unit for encoder-decoder based online handwritten mathematical expression recognition (HMER). Unlike previous methods which use trace points or image pixels as basic units, SCAN makes full use of stroke-level information for better alignment and representation. The proposed SCAN can be adopted in both single-modal (online or offline) and multi-modal HMER. For single-modal HMER, SCAN first employs a CNN-GRU encoder to extract point-level features from input traces in online mode and employs a CNN encoder to extract pixel-level features from input images in offline mode, then use stroke constrained information to convert them into online and offline stroke-level features. Using stroke-level features can explicitly group points or pixels belonging to the same stroke, therefore reduces the difficulty of symbol segmentation and recognition via the decoder with attention mechanism. For multi-modal HMER, other than fusing multi-modal information in decoder, SCAN can also fuse multi-modal information in encoder by utilizing the stroke based alignments between online and offline modalities. The encoder fusion is a better way for combining multi-modal information as it implements the information interaction one step before the decoder fusion so that the advantages of multiple modalities can be exploited earlier and more adequately. Besides, we propose an approach combining the encoder fusion and decoder fusion, namely encoder-decoder fusion, which can further improve the performance. Evaluated on a benchmark published by CROHME competition, the proposed SCAN achieves the state-of-the-art performance. Furthermore, by conducting experiments on an additional task: online handwritten Chinese character recognition (HCCR), we demonstrate the generality of our proposed method.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

Handwritten mathematical expression recognition (HMER) is one of the primary branches of document analysis and recognition, which is widely used for the electronization of various scientific literatures. Different from online/offline character or text line recognition [1], HMER is much more challenging as it meets with the complicated two-dimensional structural analysis [2–4].

Generally, HMER consists of two major problems [5], which are symbol recognition and structural analysis. Traditional methods solve these problems sequentially or globally. Concretely, sequential methods [6,7] first segment input expression into mathematical symbols and identify them separately. Then the structural analysis finds out the structure of the expression according to the symbol recognition results. While global methods [8,9] deal with HMER as a global optimization of symbol recognition and structural analysis and the symbol segmentation is performed implicitly.

As deep learning came to prominence, attention based encoder-decoder approaches are extensively adopted for HMER, which can be divided into online and offline cases. For online HMER, [10,11] treat the handwritten mathematical expression (HME) as a point sequence and extract point-level features from input traces. While for offline HMER, [12,13] take the HME as a static image and extract pixel-level features from the input image. Benefiting from rich dynamic (spatial and temporal) information which is extremely helpful for handwritten recognition, online HMER tends to meet fewer difficulties caused by ambiguous handwriting. However, the lack of global information in online HMER may lead to incorrect recognition coming from delayed strokes or inserted strokes [11,12]. On the contrary, offline HMER can easily handle

* Corresponding author.
E-mail addresses: jmwang66@mail.ustc.edu.cn (J. Wang), jundu@ustc.edu.cn (J. Du), xysszjs@mail.ustc.edu.cn (J. Zhang), bingolwang@tencent.com (B. Wang), timren@tencent.com (B. Ren).

https://doi.org/10.1016/j.patcog.2021.108047
0031-3203/© 2021 Elsevier Ltd. All rights reserved.

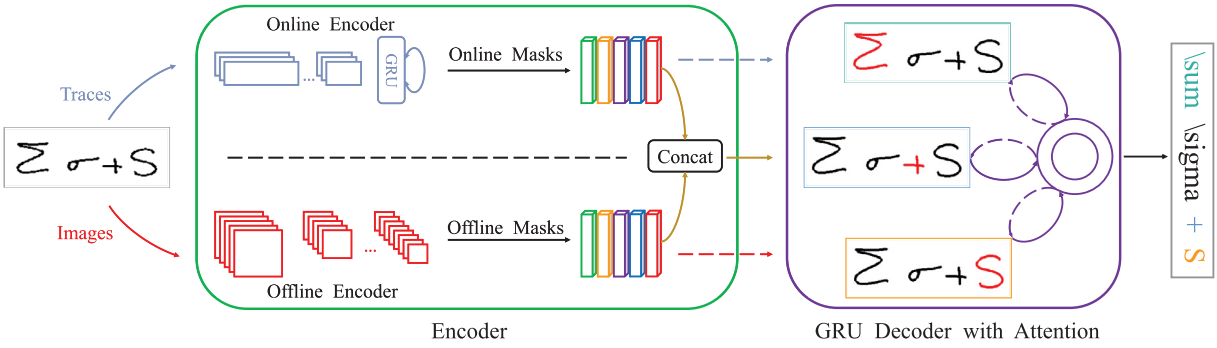

Fig. 1. The overall architecture of stroke constrained attention network (SCAN).

these situations as its input is a static image which contains global information robust to stroke orders. Consequently, it is intuitive to utilize both dynamic traces and static images to build a more powerful recognition system, which is referred as multi-modal HMER [14].

Although encoder-decoder approaches have greatly improved the performance of HMER, they still suffer from the difficulty of symbol segmentation. Because an attention mechanism is utilized to implement symbol segmentation implicitly, the inaccurate attention will lead to the mis-recognition of the input expression. Previous approaches always compute the attention coefficients on low-level features, such as trace points for online modality and image pixels for offline modality. However, for handwriting recognition problems, handwritten input has a distinctive property that points or pixels can be naturally grouped into higher-level basic units, called strokes, formed by a pen-down and pen-up action. Therefore, fully utilizing strokes as the basic units for attention based encoder-decoder models will potentially improve the attention based alignment and even enhance the representation ability of input features for online HMER.

In this study, we propose a novel stroke constrained attention network (SCAN) for online HMER, which treats stroke as the basic unit for encoder-decoder models. It can be adopted in both single-modal and multi-modal HMER. Compared with previous encoder-decoder based approaches [11,12], SCAN has four major striking properties: (i) It greatly improves the alignment generated by attention; (ii) The number of strokes is much smaller than the number of points or pixels, which helps accelerate the decoding process; (iii) For multi-modal recognition, SCAN provides oracle alignments between online traces and offline images, which enables to fuse features from different modalities in encoder and significantly improves the performance. (iiii) An approach combining the encoder fusion and decoder fusion is proposed, which can utilize stroke-level, point-level and pixel-level features at the same time.

As shown in Fig. 1, for online modality, we employ a convolutional neural network with gated recurrent units (CNN-GRU) based encoder to extract point-level features from the input trace sequence. Then the stroke constrained information, i.e., the correspondence between points and strokes, is utilized to convert point-level features into online stroke-level features. Similarly, for offline modality, we adopt a CNN-based encoder to extract pixel-level features from the input image and then convert it into offline stroke-level features. A decoder with attention is introduced to generate the recognition result, where attention actually achieves symbol segmentation implicitly. The stroke-level features as a higher-level and more accurate representation extracted from the low-level point/pixel features can potentially reduce the difficulty of symbol segmentation and recognition.

For multi-modal HMER, SCAN can play a more essential role as it not only groups points and pixels into strokes to generate more efficient symbol segmentation, but also makes fusing features from different modalities in encoder become possible. On top of the stroke-level features from both online and offline modalities, we design two multi-modal fusion strategies, namely encoder fusion and decoder fusion. The decoder fusion is similar to our recent work [14], where a multi-modal attention is equipped with re-attention mechanism to guide the decoding procedure by generating a multi-modal stroke-level context vector with the information of both online and offline modalities. The proposed encoder fusion has one advantage that it takes the information interaction one step before the decoder fusion so that the advantages of multiple modalities can be exploited earlier and more adequately. As [14] treats point and pixel as the basic unit, it is difficult to implement the encoder fusion with no explicit alignments between online point-level features and offline pixel-level features. However, as shown in Fig. 1, SCAN treats stroke as the basic unit and therefore there are oracle alignments between online and offline stroke-level features. Accordingly, we can fuse them in encoder to obtain multi-modal stroke-level features, which are then fed to the decoder and an attention mechanism is adopted to guide the decoding procedure and generate recognition result step by step. It can utilize both online and offline information to acquire more accurate attention results and significantly improve the performance of online HMER. Finally, we combine the encoder fusion and decoder fusion to utilize stroke-level, point-level and pixel-level features with different lengths to further improve the performance. Besides, we also evaluate all above methods on online handwritten Chinese character recognition (HCCR) and prove that our methods can also acquire improvement in this application.

The main contributions of this study are summarized as follows:

- A novel SCAN framework is proposed by fully utilizing the stroke information for encoder-decoder based online HMER and HCCR.
- A single-modal SCAN approach is presented via the novel design of online/offline stroke-level features.
- A multi-modal SCAN approach is introduced with two fusion strategies, namely encoder fusion and decoder fusion. Furthermore, these two strategies can be combined to achieve encoder-decoder fusion.
- We demonstrate the effectiveness and efficiency of SCAN through complete experimental analysis and attention visualization.

This work is an extension of our previous conference paper [14] in six ways: 1) The stroke as a high-level representation is adopted rather than the point/pixel as a low-level representation in both single-modal and multi-modal HMER; 2) The stroke-level features are used in multi-modal attention equipped with re-attention to show the strength of the stroke constrained informa-

tion; 3) The online and offline stroke-level features are fully exploited in the encoder fusion strategy and this strategy can be combined with the decoder fusion to achieve the encoder-decoder fusion; 4) A stroke-level attention guider is proposed to help attention learn better; 5) A comprehensive set of experiments are designed on the published benchmark of CROHME 2014/CROHME 2016/CROHME 2019; 6) The proposed method is also adopted in online HCCR and can also achieve improvement.

2. Related work

In this section, we first describe traditional approaches and then discuss neural network based approaches for HMER. Finally we elaborate multi-modal machine learning approaches.

2.1. Traditional approaches for HMER

One key property of online HMER is that the pen-tip movements (xy-coordinates) and pen states (pen-down and pen-up) can be acquired during the writing process. Traditional approaches for HMER [15–17] usually utilize the pen states to group trajectory points belonging to the same stroke in advance and treat stroke as the basic unit, i.e. representing mathematical expression as a set of strokes. The process of HMER can be divided into two steps: symbol recognition and structural analysis. Symbol recognition involves symbol segmentation and classification. Symbol segmentation is actually grouping the strokes belonging to the same symbol. These two steps can be implemented separately or jointly, referring to sequential and global methods, respectively. Sequential methods [6,7] first achieve symbol recognition by finding the best possible groups of strokes and identifying the symbol corresponding to each stroke group. Then structural analysis is performed using syntactic models for representing spatial relations among symbols, such as tree structure models [18]. In sequential methods, the contextual information is not fully exploited and the symbol segmentation/recognition errors will be subsequently propagated to structural analysis. On the contrary, global methods [8,9] optimize symbol recognition and structural analysis using the complete expression simultaneously. However, global methods are computationally more expensive as all the lower level hypotheses should be kept until the highest-level decision is made to achieve global decision. So efficient search strategies must be defined, e.g., [19] expands interpretations layer by layer so that global interpretations are systematically formed and evaluated, which can help accelerate the search process. [20] introduces a data-driven organization of the dynamic programming beam search to avoid a full search. Besides, [21] proposes a posterior probability-based confidence measure to guide the search.

2.2. Attention based encoder-decoder approaches for HMER

Encoder-decoder framework has been extensively applied to many applications including machine translation [22–24], speech recognition [25,26], image caption [27–29] and handwritten trajectory recovery [30]. Typically, an encoder is first employed to extract high-level representations from input. Then, a decoder is applied to generate a variable-length sequence as the output step by step. To address the issue that both input and output are of variable length, attention mechanism [31–35] is usually incorporated into decoder, which can generate a fixed-length context vector by weighted averaging the variable-length high-level representations to guide the decoding procedure. With the development of deep learning, encoder-decoder based approaches with an attention mechanism are also widely used for HMER, which convert the output format from tree structure into LaTeX string and significantly outperform the traditional methods. According to the different input modalities of HMER, these approaches can be divided into online and offline ones. Online approach treats the HMEs as dynamic traces while offline approach treats the HMEs as static images. The online approach [10] employed GRU-based encoder and GRU-based decoder with a spatial attention, which achieved significant improvements compared with traditional methods for HMER. [11] introduced a TAP model with additional temporal attention and an attention guider to further improve the performance. Besides, [36] adopted residual connection in encoder and a transition probability matrix in decoder. As for the offline approach, [12] utilized a WAP model, which adopted CNN-based encoder to extract features from static images. [13] proposed a coarse-to-fine attention to improve efficiency. In addition, [37] introduced a PAL model and employed an adversarial learning strategy during training.

2.3. Multi-modal machine learning

Recently, an increasing number of studies focus on multi-modal machine learning, which aims to utilize advantages and complementarities from multiple modalities [38–40]. Similiar with HMER, [41] proposed a method to utilize both handwritten and audio modalities for improving the recognition performance. An essential topic of multi-modal is how to fuse the information from different modalities [42,43]. Specific to features with varying length such as sentences, videos and audio streams, one difficulty to make a multi-modal fusion is the unaligned nature of different modalities. As encoder-decoder based framework is widely used for sequence machine learning, here we focus on the discussion of multi-modal fusion in the encoder stage or the decoder stage. For decoder fusion, [44] proposed a co-attention model to jointly reason about image and question attention for visual question answering. [45] developed a generalized multi-modal factorized high-order pooling approach (MFH) to achieve more effective fusion of multi-modal features by exploiting their correlations sufficiently. However, encoder fusion can usually acquire better performance than decoder fusion, as information from different modalities can interact earlier. Unfortunately, we usually lack of optimal mapping between different modalities, which makes the encoder fusion challenging. Although [46,47] utilized cross-modal self-attention to achieve the encoder fusion, the unaligned issue was still existed as the alignments acquired by cross-modal self-attention could not be guaranteed to be exactly accurate.

Differently, for online HMER, we can obtain oracle alignments between online and offline modalities by making full use of stroke constrained information. Therefore, in this study, we propose SCAN to achieve the fusion of online and offline modalities in encoder, which significantly improves the recognition performance. Besides, we combine the encoder fusion with the decoder fusion to achieve the encoder-decoder fusion, which can further improve the performance.

3. Single-modal SCAN

In this section, we introduce the proposed SCAN for single-modal HMER, including online SCAN (OnSCAN) and offline SCAN (OffSCAN). Different from previous single-modal approaches [11–13], we explicitly utilize the stroke constrained information in encoder-decoder based HMER. Specifically, stroke-level features are adopted in SCAN rather than point-level and pixel-level features. Furthermore, the attention mechanism in the decoder is to discover the alignments between the predicted mathematical symbol and input features. Therefore, the attention in SCAN actually groups strokes belonging to the same symbol, which is obviously much easier and more efficient than grouping points or pixels in previous approaches as stroke-level features are a higher-level

representation to reduce the difficulty of attention than the local point-level or pixel-level features.

3.1. Data preparation

For online HMER, the raw input data is the handwritten traces, which can be represented as a variable-length sequence:

$$[(x_1, y_1, s_1), (x_2, y_2, s_2), \cdots, (x_N, y_N, s_N)] \tag{1}$$

where x_i and y_i are the xy-coordinates of the pen movements and s_i indicates which stroke the i^{th} point belongs to. Please note that in this study the stroke constrained information $\{s_i\}$ is always used for both online and offline modalities.

For the online modality, we normalize the traces and extract an 8-dimensional feature vector for each point i:

$$\mathbf{x}_i^{\text{on}} = \left[x_i, y_i, \Delta x_i, \Delta y_i, \Delta' x_i, \Delta' y_i, \text{strokeFlag1}, \text{strokeFlag2}\right] \tag{2}$$

where $\Delta x_i = x_{i+1} - x_i$, $\Delta y_i = y_{i+1} - y_i$, $\Delta' x_i = x_{i+2} - x_i$, $\Delta' y_i = y_{i+2} - y_i$. The last two terms are flags indicating the status of the pen, i.e., $[1, 0]$ means pen-down while $[0, 1]$ means pen-up. We refer to the trace point sequence after processing as $\mathbf{X}^{\text{on}} = \left\{\mathbf{x}_1^{\text{on}}, \mathbf{x}_2^{\text{on}}, \cdots, \mathbf{x}_N^{\text{on}}\right\}$, where N denotes the number of trace points. For the offline modality, we first calculate the heights of all strokes. Then we compute the average height of strokes with the height greater than one tenth of the maximum height. Furthermore, we normalize xy-coordinates of all points in accordance with the average height and simply line trace points of each stroke to convert traces into static images, $\mathbf{X}^{\text{off}}$ of size $H_{\text{in}} \times W_{\text{in}}$.

3.2. Encoder with stroke masks

We believe that the stroke constrained information plays an essential role in online HMER. In [11,14], the stroke information is only used as additional two dimensions of the input 8-dimensional feature vector. So we aim at fully utilizing the stroke information by defining the online and offline stroke masks, which is adopted to convert online point-level features and offline pixel-level features to the corresponding stroke-level features in the encoder stage.

3.2.1. Stroke masks

Here we introduce how to generate online and offline stroke masks. Specifically, for one HME sequence, suppose it consists of M strokes and N points. We define online stroke masks as $\mathbf{Mask}^{\text{on}} = \{\mathbf{mask}_1^{\text{on}}, \mathbf{mask}_2^{\text{on}}, \cdots, \mathbf{mask}_M^{\text{on}}\}$ and offline stroke masks as $\mathbf{Mask}^{\text{off}} = \{\mathbf{mask}_1^{\text{off}}, \mathbf{mask}_2^{\text{off}}, \cdots, \mathbf{mask}_M^{\text{off}}\}$. Each online stroke mask $\mathbf{mask}_j^{\text{on}}$ is a N-dimensional vector and the value of each element i is 1 or 0, indicating whether the i^{th} point belongs to the j^{th} stroke or not by using the original stroke information $\{s_i\}$. Each offline stroke mask $\mathbf{mask}_j^{\text{off}}$ is a matrix of size $H_{\text{in}} \times W_{\text{in}}$ and each element (h, w) is 1 or 0, indicating whether the pixel (h, w) belongs to the j^{th} stroke or not by using the original stroke information $\{s_i\}$.

3.3. Online encoder

The online encoder is designed to extract the online stroke-level features based on $\mathbf{X}^{\text{on}}$ and $\mathbf{Mask}^{\text{on}}$. As shown in the left part of Fig. 2, different from [10,11], we employ 1-D DenseNet-20 following a fewer stack of GRUs, which can acquire better local information and improve the recognition performance. The convolutional layers of CNN are configured as densely connected layers in DenseNet [48]. The output of CNN is a tensor of size $1 \times L \times D'$, which is then transformed into a D'-dimensional vector sequence of length L, $\mathbf{A}' = \left\{\mathbf{a}_1', \mathbf{a}_2', \cdots, \mathbf{a}_L'\right\}$. To capture the context information from input traces, a stack of GRUs are built on top of CNN. The hidden state of GRU can be calculated as:

$$\mathbf{h}_t' = \text{GRU}\left(\mathbf{a}_t', \mathbf{h}_{t-1}'\right) \tag{3}$$

Furthermore, as unidirectional GRU cannot exploit the future context information, we actually adopt bidirectional GRU which can utilize both past and future context information. The detailed implementation of GRU can be found in [11].

The output of CNN-GRU encoder is a variable-length vector sequence, namely point-level features, which can be represented as $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_L\}$ and each element is a D-dimensional vector. Note that N is a multiple of L based on the number of pooling layers in CNN part. With the point-level features, we utilize online stroke masks to convert point-level features into online stroke-level features, which is illustrated in the right part of Fig. 2. First, the same number of downsampling as that in CNN part of online encoder is used to process online stroke masks, which converts each online mask from a N-dimensional vector $\mathbf{mask}_j^{\text{on}}$ to a L-dimensional vector $\mathbf{pmask}_j^{\text{on}}$. Then, the j^{th} online stroke-level feature can be calculated as:

$$\mathbf{s}_j^{\text{on}} = \mathbf{A}^{\text{T}} \frac{\mathbf{pmask}_j^{\text{on}}}{||\mathbf{pmask}_j^{\text{on}}||_1} \qquad \mathbf{S}^{\text{on}} = \{\mathbf{s}_1^{\text{on}}, \mathbf{s}_2^{\text{on}}, \cdots, \mathbf{s}_M^{\text{on}}\} \tag{4}$$

where $||\cdot||_1$ is the vector 1-norm, $\mathbf{s}_j^{\text{on}}$ is a D-dimensional vector and $\mathbf{S}^{\text{on}}$ is the final output of online encoder.

3.4. Offline encoder

The offline encoder is designed to extract the offline stroke-level features based on $\mathbf{X}^{\text{off}}$ and $\mathbf{Mask}^{\text{off}}$. We first introduce a DenseNet-99 to extract pixel-level features, which is illustrated in the left part of Fig. 3. The output of CNN encoder is a tensor of size $H \times W \times D$. Note that H_{in} and W_{in} are multiples of H and W based on the number of downsampling in CNN encoder, respectively. We transform this tensor into a variable-length vector sequence $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{H \times W}\}$ as the pixel-level features and each element is a D-dimensional vector.

Similar to the online encoder, we utilize offline stroke masks to convert pixel-level features into offline stroke-level features, which is illustrated in the right part of Fig. 3. First, the same number of downsampling as that in CNN encoder is used to process offline stroke masks, which converts each offline stroke mask from a matrix $\mathbf{mask}_j^{\text{off}}$ of size $H_{\text{in}} \times W_{\text{in}}$ to a matrix $\mathbf{pmask}_j^{\text{off}}$ of size $H \times W$. Then we transform each offline mask into a $(H \times W)$-dimensional vector and offline stroke-level features are extracted from pixel-level features as:

$$\mathbf{s}_j^{\text{off}} = \mathbf{B}^{\text{T}} \frac{\mathbf{pmask}_j^{\text{off}}}{||\mathbf{pmask}_j^{\text{off}}||_1} \qquad \mathbf{S}^{\text{off}} = \left\{\mathbf{s}_1^{\text{off}}, \mathbf{s}_2^{\text{off}}, \cdots, \mathbf{s}_M^{\text{off}}\right\} \tag{5}$$

where $\mathbf{s}_j^{\text{off}}$ is a D-dimensional vector and $\mathbf{S}^{\text{off}}$ is the final output of offline encoder.

3.5. Decoder with attention

As online and offline stroke-level features ($\mathbf{S}^{\text{on}}$ and $\mathbf{S}^{\text{off}}$) are both vector sequences, we employ the same decoder architecture with a coverage-based attention for both online and offline SCAN. But the parameters contained in decoder and attention are not shared. As shown in Fig. 4, the decoder accepts online or offline stroke-level features and generates a LaTeX sequence for recognition:

$$\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_C\}, \mathbf{y}_i \in \mathbb{R}^K \tag{6}$$

where K is the number of total math symbols in the vocabulary and C is the length of LaTeX sequence. To address the problem

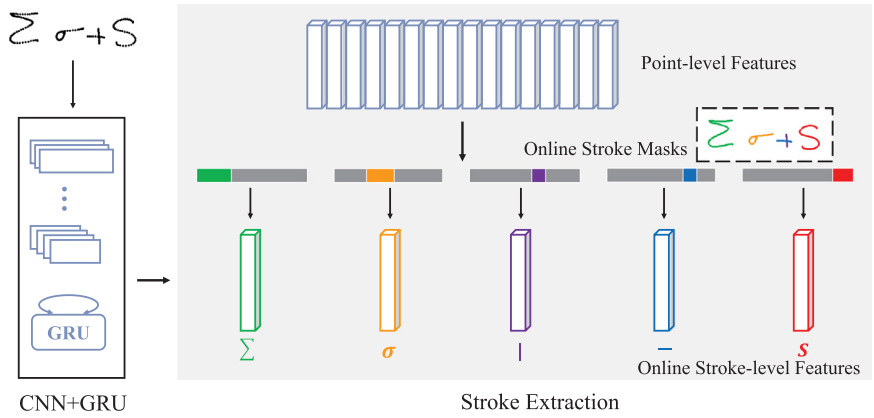

Fig. 2. The architecture of online encoder. The left part is point-level feature extraction from input traces using CNN-GRU. The right part is online stroke-level feature extraction from point-level features.

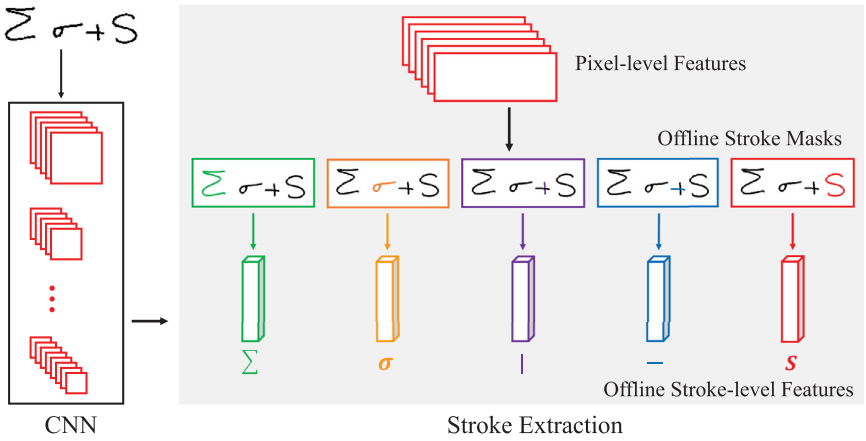

Fig. 3. The architecture of offline encoder. The left part is pixel-level feature extraction from input images using a deep CNN. The right part is offline stroke-level feature extraction from pixel-level features.

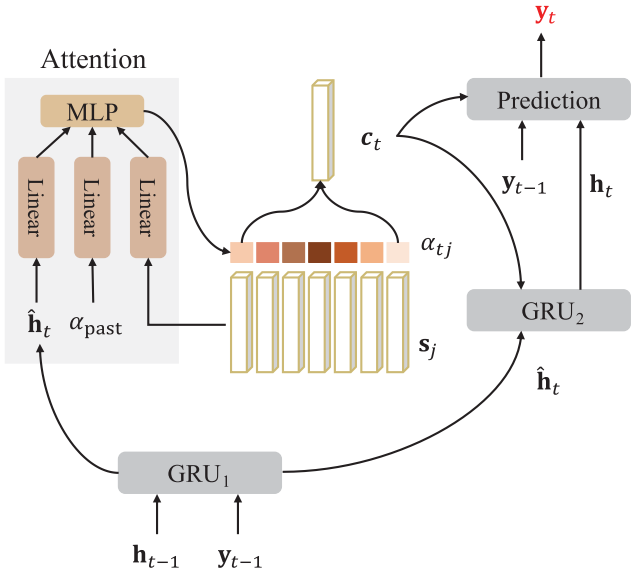

Fig. 4. The decoder architecture with two GRU layers and a coverage-based attention. α_{past} denotes $\sum_{\tau=1}^{t-1} \boldsymbol{\alpha}_\tau$.

that the stroke-level features have a variable length and the length of LaTeX string is not fixed, we employ an intermediate fixed-size vector $\mathbf{c}_t$, namely context vector generated by a unidirectional GRU with a coverage-based attention, which will be described later. Then another unidirectional GRU is adopted to produce the LaTeX sequence symbol by symbol. The decoder structure can be denoted as:

$$\hat{\mathbf{h}}_t = \text{GRU}_1(\mathbf{y}_{t-1}, \mathbf{h}_{t-1}) \tag{7}$$

$$\mathbf{c}_t = f_{\text{att}}\left(\hat{\mathbf{h}}_t, \mathbf{S}\right) \tag{8}$$

$$\mathbf{h}_t = \text{GRU}_2\left(\mathbf{c}_t, \hat{\mathbf{h}}_t\right) \tag{9}$$

where GRU_1, GRU_2 indicate two GRU layers, f_{att} denotes the coverage-based attention, $\hat{\mathbf{h}}_t$ and $\mathbf{h}_t$ represent the hidden states of the first and the second GRU layers, $\mathbf{S}$ denotes online or offline stroke-level features. Besides, we utilize $\hat{\mathbf{h}}_t$ instead of $\mathbf{h}_{t-1}$ to calculate attention coefficients as we believe that $\hat{\mathbf{h}}_t$ is a more accurate representation of the current alignment information than $\mathbf{h}_{t-1}$.

The probability of each predicted symbol is then computed by the context vector $\mathbf{c}_t$, the hidden state of the second GRU layer $\mathbf{h}_t$ and one-hot vector of previous output symbol $\mathbf{y}_{t-1}$ using the following equation:

$$p(\mathbf{y}_t) = g(\mathbf{W}_o\phi(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{W}_h\mathbf{h}_t + \mathbf{W}_c\mathbf{c}_t)) \tag{10}$$

where g represents the softmax activation function and ϕ represents the maxout activation function. $\mathbf{W}_o \in \mathbb{R}^{K\times\frac{m}{2}}$, $\mathbf{W}_h \in \mathbb{R}^{m\times n}$, $\mathbf{W}_c \in \mathbb{R}^{m\times D}$, and $\mathbf{E} \in \mathbb{R}^{m\times K}$ denotes the embedding matrix. m and n are dimensions of embedding and GRU decoder.

Attention mechanism is widely adopted in sequence learning [31–33]. It is intuitive that for each predicted symbol, only parts of the input rather than the entire input is necessary to provide the useful information, which means only a subset of feature vectors mainly contribute to the recognition. As shown in Fig. 4, we introduce a coverage-based attention f_{att}, which can be represented as:

$$\mathbf{F} = \mathbf{Q} * \sum_{\tau=1}^{t-1} \boldsymbol{\alpha}_{\tau} \tag{11}$$

$$e_{tj} = \boldsymbol{\nu}_{\text{att}}^{\text{T}} \tanh\left(\mathbf{W}_{\text{att}}\hat{\mathbf{h}}_t + \mathbf{U}_{\text{att}}\mathbf{s}_j + \mathbf{U}_f\mathbf{f}_j\right) \tag{12}$$

where e_{tj} denotes the energy of stroke-level feature vector $\mathbf{s}_j$ in decoding step t. $\mathbf{F}$ with its element $\mathbf{f}_j$ as the coverage vector is computed by feeding the past attention into a convolution layer $\mathbf{Q}$ with q output channels, which can help alleviate the problem of standard attention mechanism, namely lack of coverage [49]. Let n' denotes the dimension of the attention, then $\boldsymbol{\nu}_{\text{att}} \in \mathbb{R}^{n'}$, $\mathbf{W}_{\text{att}} \in \mathbb{R}^{n' \times n}$, $\mathbf{U}_{\text{att}} \in \mathbb{R}^{n' \times D}$, $\mathbf{U}_f \in \mathbb{R}^{n' \times q}$.

The attention coefficients α_{tj} can be obtained by feeding e_{tj} into a softmax function, which is utilized to calculate the context vector as:

$$\alpha_{tj} = \frac{\exp\left(e_{tj}\right)}{\sum_{k=1}^{M} \exp\left(e_{tk}\right)} \qquad \mathbf{c}_t = \sum_{j=1}^{M} \alpha_{tj}\mathbf{s}_j \tag{13}$$

3.6. Stroke-level attention guider

For online HMER, the correspondence information between strokes and symbols is provided in the training stage. For example, there is an expression "$s+2$" which consists of four strokes: the first stroke for "s", the second and the third strokes for " + " and the last stroke for "2". Obviously, when we predict the symbol " + ", the coverage-based attention should be supposed to only attend the second and the third strokes. Generally for the symbol w_t in time step t, we first introduce an oracle attention map, $\boldsymbol{\gamma}_t = \left\{\gamma_{tj} | j = 1, 2, \ldots, M\right\}$ with $\gamma_{tj} = \frac{1}{M'}$ if the j^{th} stroke belongs to the symbol w_t, otherwise 0, where M denotes the number of all strokes and M' denotes the number of strokes belonging to the symbol w_t. We can regard γ_{tj} and α_{tj} as two probability distributions because $\sum_{j=1}^{M} \gamma_{tj} = \sum_{j=1}^{M} \alpha_{tj} = 1$ and it is intuitive to employ the cross entropy function as the stroke-level attention guider:

$$G_t = -\sum_{j=1}^{M} \gamma_{tj} \log \alpha_{tj} \tag{14}$$

Note that for spatial structure, such as "$\wedge$", "{" and "}", which are used to meet the requirement of LaTeX grammar, we simply remove the guider as they are lack of explicit alignments to strokes. This stroke-level attention guider is adopted as a regularization item for parameter learning as elaborated in Section 5.1.

4. Multi-modal SCAN

In this section, we discuss multi-modal SCAN, which can take both advantages of online and offline modalities for online HMER. First, we employ a multi-modal encoder with both online and offline encoder to extract online stroke-level features $\mathbf{S}^{\text{on}}$ and offline stroke-level features $\mathbf{S}^{\text{off}}$, as shown in Section 3.3 and Section 3.4. Then two fusion strategies are proposed for multi-modal SCAN, namely the decoder fusion (denoted as MMSCAN-D) and the encoder fusion (denoted as MMSCAN-E). In the decoder fusion, similar to our previous work [14], a multi-modal attention equipped with re-attention mechanism to fuse online and offline stroke-level features is introduced. More importantly, SCAN makes the fusion of online and offline stroke-level features in encoder become possible as it provides oracle alignments between online and offline modalities. Finally, we combine the encoder fusion and the decoder fusion to achieve the encoder-decoder fusion (denoted as MMSCAN-ED), which can further improve the performance.

4.1. Decoder fusion

To fully utilize the complementarities between online and offline modalities, a two-stage re-attention mechanism is designed with pre-attention and fine-attention models, which is illustrated in Fig. 5. Actually the decoder structure here is similar to the single-modal case as described in Eq. (7), (8), (9). The main difference is that f_{att} in Eq. (8) is replaced by the re-attention mechanism, which accepts online and offline stroke-level features and generates a multi-modal stroke-level context vector $\mathbf{c}_t^{\text{mm}}$. In the first stage, the pre-attention model can be represented as:

$$\hat{\mathbf{c}}_t^{\text{on}} = f_{\text{att}}^{\text{on}}\left(\hat{\mathbf{h}}_t, \mathbf{S}^{\text{on}}\right) \qquad \hat{\mathbf{c}}_t^{\text{off}} = f_{\text{att}}^{\text{off}}\left(\hat{\mathbf{h}}_t, \mathbf{S}^{\text{off}}\right) \tag{15}$$

where $\hat{\mathbf{c}}_t^{\text{on}}$ and $\hat{\mathbf{c}}_t^{\text{off}}$ denote two single-modal stroke-level context vectors. Note that the superscripts "on" and "off" in Eq. (15) are only used to distinguish coverage-based attention f_{att} over online and offline stroke-level features as the attention parameters are not shared.

Based on the results of the pre-attention model, the fine-attention model is employed to generate multi-modal stroke-level context vector $\mathbf{c}_t^{\text{mm}}$ in the second stage. Compared with the pre-attention model, the fine-attention model adds the context vector of one modality from the pre-attention model as the auxiliary information to improve the attention of another modality, which is implemented as:

$$\alpha_{tj}^{\text{on}} = g\left(\boldsymbol{\nu}_{\text{att}}^{\text{T}} \tanh\left(\mathbf{W}_{\text{att}}\hat{\mathbf{h}}_t + \mathbf{U}_{\text{att}}^{\text{on}}\mathbf{s}_j^{\text{on}} + \mathbf{U}_f^{\text{on}}\mathbf{f}_j^{\text{on}} + \mathbf{U}_p^{\text{off}}\hat{\mathbf{c}}_t^{\text{off}}\right)\right) \tag{16}$$

$$\alpha_{tj}^{\text{off}} = g\left(\boldsymbol{\nu}_{\text{att}}^{\text{T}} \tanh\left(\mathbf{W}_{\text{att}}\hat{\mathbf{h}}_t + \mathbf{U}_{\text{att}}^{\text{off}}\mathbf{s}_j^{\text{off}} + \mathbf{U}_f^{\text{off}}\mathbf{f}_j^{\text{off}} + \mathbf{U}_p^{\text{on}}\hat{\mathbf{c}}_t^{\text{on}}\right)\right) \tag{17}$$

where $\mathbf{U}_p^{\text{on}} \in \mathbb{R}^{n' \times D}$, $\mathbf{U}_p^{\text{off}} \in \mathbb{R}^{n' \times D}$. Then the stroke-level context vectors of fine-attention model are calculated as:

$$\mathbf{c}_t^{\text{on}} = \sum_{j=1}^{M} \alpha_{tj}^{\text{on}}\mathbf{s}_j^{\text{on}} \qquad \mathbf{c}_t^{\text{off}} = \sum_{j=1}^{M} \alpha_{tj}^{\text{off}}\mathbf{s}_j^{\text{off}} \tag{18}$$

Finally, the multi-modal stroke-level context vector $\mathbf{c}_t^{\text{mm}}$ can be obtained as:

$$\mathbf{c}_t^{\text{mm}} = \tanh\left(\mathbf{W}_{\text{FC}}\begin{bmatrix}\mathbf{c}_t^{\text{on}} \\ \mathbf{c}_t^{\text{off}}\end{bmatrix}\right) \tag{19}$$

where $\mathbf{W}_{\text{FC}} \in \mathbb{R}^{D \times 2D}$.

The re-attention can be also equipped with stroke-level attention guider as described in Section 3.6 and the main difference is that here we utilize oracle attention map $\boldsymbol{\gamma}_t$ to supervise the learning of both online and offline attention coefficients as:

$$G_t = -\left(\sum_{j=1}^{M} \gamma_{tj} \log \alpha_{tj}^{\text{on}} + \sum_{j=1}^{M} \gamma_{tj} \log \alpha_{tj}^{\text{off}}\right) \tag{20}$$

4.2. Encoder fusion

A key component of multi-modal learning is to fuse features from different modalities. In our previous work [14], point-level and pixel-level features are extracted from the inputs of online and offline modalities. On account of the problem that these two types of features are unaligned, we can only fuse online and offline modalities in decoder.

However, as illustrated in Fig. 6, SCAN converts point-level and pixel-level features into online and offline stroke-level features. Inherently, there are oracle alignments between online and offline modalities in terms of stroke-level features. Specifically, online

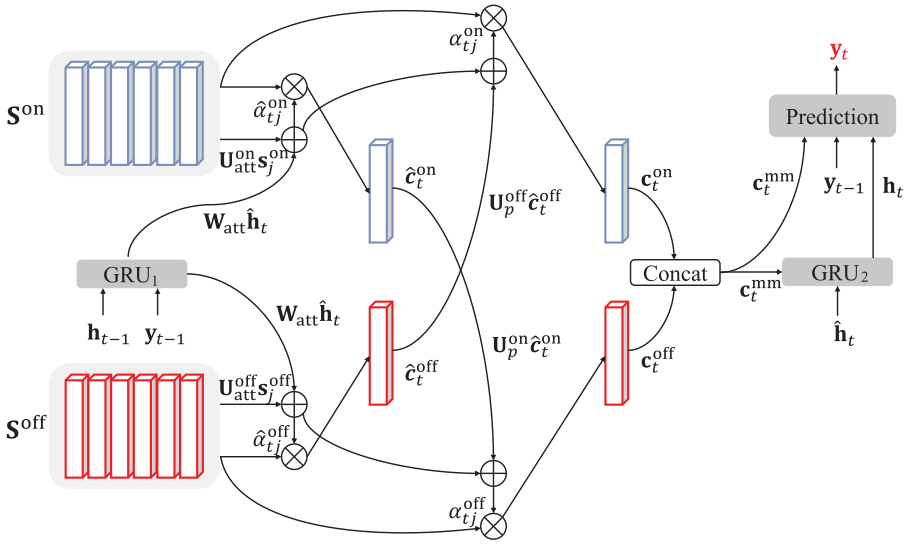

Fig. 5. The two-stage re-attention mechanism with pre-attention and fine-attention models. To simplify the illustration, we have omitted the coverage vectors and activation functions.

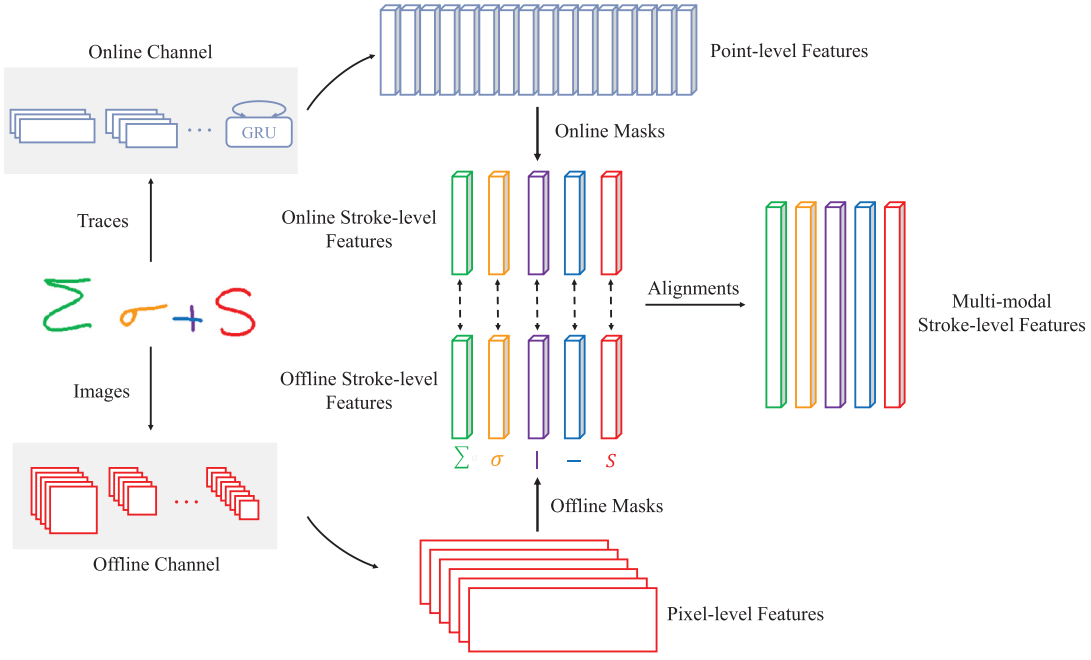

Fig. 6. Encoder fusion to generate multi-modal stroke-level features with the oracle alignments between online and offline stroke-level features.

and offline stroke-level features are one-to-one correspondence, both indicating the high-level representations of a certain stroke. Therefore, we can fuse online and offline stroke-level features into multi-modal stroke-level features as:

$$\mathbf{S}^{\text{mm}} = \{\mathbf{s}_1^{\text{mm}}, \mathbf{s}_2^{\text{mm}}, \ldots, \mathbf{s}_M^{\text{mm}}\} \qquad \mathbf{s}_j^{\text{mm}} = \begin{bmatrix} \mathbf{s}_j^{\text{on}} \\ \mathbf{s}_j^{\text{off}} \end{bmatrix} \tag{21}$$

With the multi-modal stroke-level features, we employ a decoder with coverage-based attention and the stroke-level attention guider described in Section 3.6 to generate the recognition result. The structure is similar to that illustrated in Section 3.5 by replacing $\mathbf{S}^{\text{on}}/\mathbf{S}^{\text{off}}$ with $\mathbf{S}^{\text{mm}}$, which can be denoted as:

$$\hat{\mathbf{h}}_t = \text{GRU}_1(\mathbf{y}_{t-1}, \mathbf{h}_{t-1}) \tag{22}$$

$$\mathbf{c}_t^{\text{mm}} = f_{\text{att}}^{\text{mm}}\left(\hat{\mathbf{h}}_t, \mathbf{S}^{\text{mm}}\right) \tag{23}$$

$$\mathbf{h}_t = \text{GRU}_2\left(\mathbf{c}_t^{\text{mm}}, \hat{\mathbf{h}}_t\right) \tag{24}$$

By comparison of MMSCAN-E and MMSCAN-D, although MMSCAN-D introduces re-attention to help information interaction between two modalities, it still only considers information from one single modality in the pre-attention model. Moreover, the errors in the pre-attention model will be inherited by the fine-attention model which might degrade the performance. Nevertheless, MMSCAN-E makes full use of stroke constrained information to obtain oracle alignments between online and offline stroke-level features and fuse them in encoder. Consequently, MMSCAN-E takes the fusion one step before MMSCAN-D, which can potentially improve the recognition performance of HMER.

4.3. Encoder-decoder fusion

In this section, we propose an encoder-decoder fusion approach, which combines the encoder fusion and decoder fusion. Specifically, we first employ a multi-modal encoder to extract point-level features **A** and pixel-level features **B** and convert them into online stroke-level features $\mathbf{S}^{\text{on}}$ and offline stroke-level features $\mathbf{S}^{\text{off}}$. Based on online and offline stroke-level features, multi-modal stroke-level features $\mathbf{S}^{\text{mm}}$ can be acquired using the encoder fusion. Other than only feeding stroke-level features to the decoder like the encoder fusion and decoder fusion in Section 4.1 and Section 4.2, here we feed multi-modal stroke-level, point-level and pixel-level features to the decoder at the same time. Then the decoder fusion is adopted to fuse these features with different lengths and generate a multi-modal multi-level context vector $\mathbf{c}_t^{\text{mmml}}$ at each decoding step. As there are three features to be processed, we modify the calculation of context vector in Section 4.1, which can be regarded as a more general and extended version of the decoder fusion.

In the first stage of the decoder fusion, the pre-attention model is employed to compute multi-modal stroke-level, point-level and pixel-level context vectors similar to Eq. (15):

$$\hat{\mathbf{c}}_t^{\text{mm}} = f_{\text{att}}^{\text{mm}}\left(\hat{\mathbf{h}}_t, \mathbf{S}^{\text{mm}}\right) \quad \hat{\mathbf{c}}_t^{\text{point}} = f_{\text{att}}^{\text{point}}\left(\hat{\mathbf{h}}_t, \mathbf{A}\right) \quad \hat{\mathbf{c}}_t^{\text{pixel}} = f_{\text{att}}^{\text{pixel}}\left(\hat{\mathbf{h}}_t, \mathbf{B}\right) \tag{25}$$

In the second stage, the fine-attention model is employed to generate the multi-modal multi-level context vector $\mathbf{c}_t^{\text{mmml}}$. For the original fine-attention model, the context vector of another modality computed in the pre-attention model will be considered. However, as there are three context vectors now, we additionally concatenate every two context vectors first:

$$\hat{\mathbf{c}}_t^{\text{pmm}} = \begin{bmatrix} \hat{\mathbf{c}}_t^{\text{point}} \\ \hat{\mathbf{c}}_t^{\text{pixel}} \end{bmatrix} \quad \hat{\mathbf{c}}_t^{\text{ppoint}} = \begin{bmatrix} \hat{\mathbf{c}}_t^{\text{mm}} \\ \hat{\mathbf{c}}_t^{\text{pixel}} \end{bmatrix} \quad \hat{\mathbf{c}}_t^{\text{ppixel}} = \begin{bmatrix} \hat{\mathbf{c}}_t^{\text{mm}} \\ \hat{\mathbf{c}}_t^{\text{point}} \end{bmatrix} \tag{26}$$

Then the fine-attention model can be represented as:

$$\alpha_{tj}^{\text{mm}} = g\left(\mathbf{v}_{\text{att}}^{\text{T}} \tanh\left(\mathbf{W}_{\text{att}}\hat{\mathbf{h}}_t + \mathbf{U}_{\text{att}}^{\text{mm}}\mathbf{s}_j^{\text{mm}} + \mathbf{U}_f^{\text{mm}}\mathbf{f}_j^{\text{mm}} + \mathbf{U}_p^{\text{mm}}\hat{\mathbf{c}}_t^{\text{pmm}}\right)\right) \tag{27}$$

$$\alpha_{tj}^{\text{point}} = g\left(\mathbf{v}_{\text{att}}^{\text{T}} \tanh\left(\mathbf{W}_{\text{att}}\hat{\mathbf{h}}_t + \mathbf{U}_{\text{att}}^{\text{point}}\mathbf{a}_j + \mathbf{U}_f^{\text{point}}\mathbf{f}_j^{\text{point}} + \mathbf{U}_p^{\text{point}}\hat{\mathbf{c}}_t^{\text{ppoint}}\right)\right) \tag{28}$$

$$\alpha_{tj}^{\text{pixel}} = g\left(\mathbf{v}_{\text{att}}^{\text{T}} \tanh\left(\mathbf{W}_{\text{att}}\hat{\mathbf{h}}_t + \mathbf{U}_{\text{att}}^{\text{pixel}}\mathbf{b}_j + \mathbf{U}_f^{\text{pixel}}\mathbf{f}_j^{\text{pixel}} + \mathbf{U}_p^{\text{pixel}}\hat{\mathbf{c}}_t^{\text{ppixel}}\right)\right) \tag{29}$$

Finally, the multi-modal multi-level context vector $\mathbf{c}_t^{\text{mmml}}$ can be computed as:

$$\mathbf{c}_t^{\text{mm}} = \sum_{j=1}^{M} \alpha_{tj}^{\text{mm}}\mathbf{s}_j^{\text{mm}} \quad \mathbf{c}_t^{\text{point}} = \sum_{j=1}^{L} \alpha_{tj}^{\text{point}}\mathbf{a}_j \quad \mathbf{c}_t^{\text{pixel}} = \sum_{j=1}^{H\times W} \alpha_{tj}^{\text{pixel}}\mathbf{b}_j \tag{30}$$

$$\mathbf{c}_t^{\text{mmml}} = \tanh\left(\mathbf{W}_{\text{FC}}\begin{bmatrix} \mathbf{c}_t^{\text{mm}} \\ \mathbf{c}_t^{\text{point}} \\ \mathbf{c}_t^{\text{pixel}} \end{bmatrix}\right) \tag{31}$$

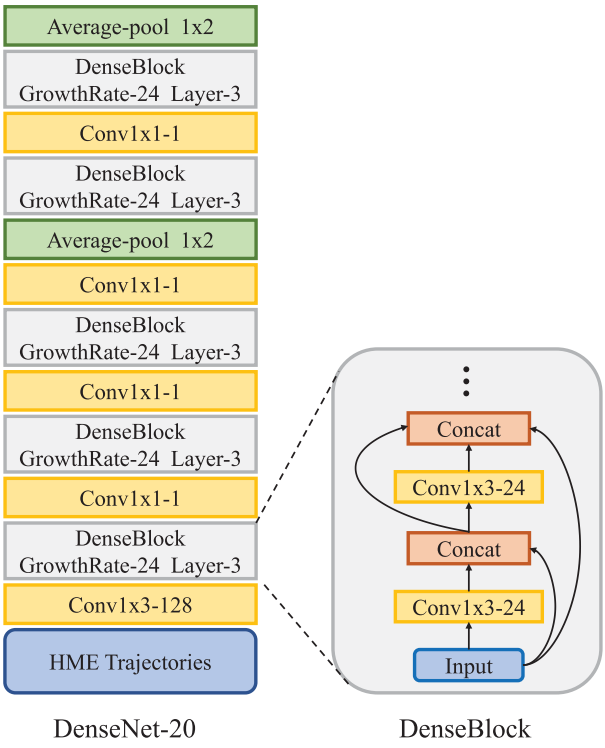

Fig. 7. The architecture of DenseNet-20.

5. Training and testing procedures

5.1. Training

Our models aim to maximize the predicted symbol probability as shown in Eq. (10) and employ cross entropy (CE) as the criterion. The objective function for optimization, which consists of CE criterion and the stroke-level attention guider, is shown as follows:

$$O = -\sum_{t=1}^{C} \log p(w_t|\mathbf{y}_{t-1}, \mathbf{X}^{\text{on}}, \mathbf{X}^{\text{off}}) + \lambda \sum_{t=1}^{C} G_t \tag{32}$$

where w_t represents the ground truth word at time step t, C is the length of output string in LaTeX format, G_t is the stroke-level attention guider, and λ is set to 0.2. Note that for single-modal HMER, only one of $\mathbf{X}^{\text{on}}$ and $\mathbf{X}^{\text{off}}$ is used. Besides, we set weight decay to 10^{-5} for online modality and multi-modal, 10^{-4} for offline modality to reduce overfitting.

There are three kinds of encoders in this study, namely online encoder, offline encoder and multi-modal encoder while multi-modal encoder is the combination of online encoder and offline encoder with the parameters pretrained from single-modal cases. The online encoder is a CNN-GRU architecture. The CNN part is a DenseNet-20 as illustrated in Fig. 7, with 5 dense blocks in the main branch. 1×2 average pooling is applied after the third and fifth dense blocks, which reduces the length of input point sequence by a factor of 4. The growth rate is set to 24 and the compression factor in transition layer is set to 1. As shown in the right part of Fig. 7, each dense block without bottleneck structure has 3 convolutional layers with kernel size 1×3 and 24 output channels. The GRU part is two layers of bidirectional GRU and each GRU layer has 250 forward and 250 backward units.

The offline encoder is a DenseNet-99 as illustrated in Fig. 8, with 3 dense blocks in the main branch. 1×1 convolution followed by 2×2 average pooling between every two contiguous dense blocks is used. The growth rate is set to 24 and the compression factor in transition layer is set to 0.5. As shown in the

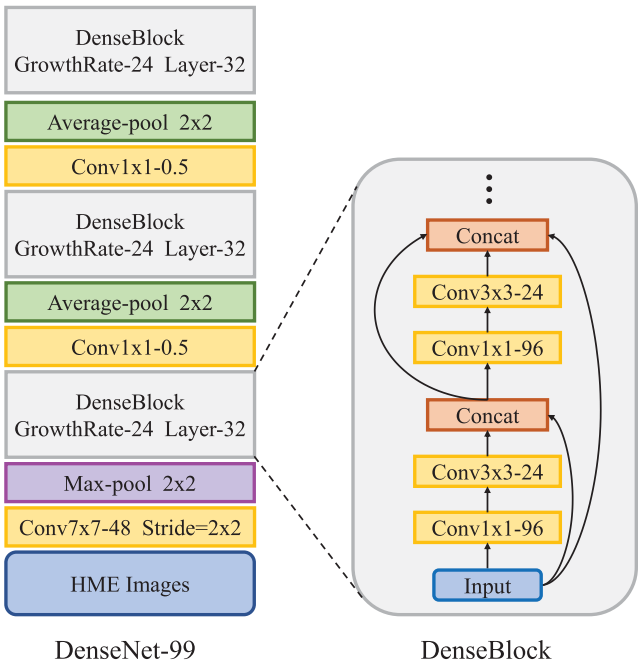

Fig. 8. The architecture of DenseNet-99.

right part of Fig. 8, each dense block adopts the bottleneck structure, i.e., a 1×1 convolution is introduced before each 3×3 convolution to reduce the input to 96 feature maps and the total number of convolutional layers in each block is 32. Note that there are additional fully connected layers on top of online and offline channels of multi-modal encoder to convert the output dimensions of these two channels to be the same, namely $D = 500$.

The decoder adopts 2 unidirectional GRU layers and each layer has 256 forward GRU units. The embedding dimension m and GRU decoder dimension n are both set to 256 while the attention dimension n' is 500. The kernel sizes of convolution layers $\mathbf{Q}$ are set to 1×7 for online modality and 11×11 for offline modality. We train our model by the adadelta algorithm [50] for optimization and the corresponding hyperparameters are set as $\rho = 0.95$, $\varepsilon = 10^{-8}$ for online modality, $\rho = 0.9$, $\varepsilon = 10^{-6}$ for offline modality and $\rho = 0.9$, $\varepsilon = 10^{-8}$ for multi-modal.

5.2. Testing

In the recognition stage, we expect to obtain the most likely LaTeX string as:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \log P(\mathbf{y}|\mathbf{x}) \tag{33}$$

Different from the training stage, we do not have the ground truth of the previous predicted symbol. Consequently, we employ a simple left-to-right beam search algorithm [51] to implement the decoding procedure, beginning with the start-of-sentence token $<sos>$. At each time step, we maintain a set of 10 partial hypotheses. Each hypothesis is expanded with every possible symbol and only the hypotheses with 10 minimal scores are kept:

$$\mathbf{S}_t = \mathbf{S}_{t-1} - \log p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{x}) \tag{34}$$

where $\mathbf{S}_{t-1}$ and $\mathbf{S}_t$ represent the scores at time steps $t-1$ and t, respectively. $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{x})$ denotes the probability of all predicted symbols in the dictionary. The prediction procedure for each hypothesis ends when the output symbol meets the end-of-sentence token $<eos>$.

Table 1
Performance comparison of different encoder-decoder approaches for the online modality using point-level features (TAP), online stroke-level features (OnSCAN), and feature fusion in decoder (OnSCAN+TAP) on CROHME 2014 and CROHME 2016 testing sets.

System	CROHME 2014		CROHME 2016	
	ExpRate	StruRate	ExpRate	StruRate
TAP [14]	48.47%	67.24%	44.81%	63.12%
OnSCAN	51.22%	70.49%	46.12%	65.30%
OnSCAN+TAP	52.64%	70.89%	47.17%	66.78%

6. Experiments on online HMER

In this section, we design a set of experiments to validate the effectiveness of the proposed SCAN on online HMER by answering the following questions:

- Q1 Is the proposed single-modal SCAN effective for online HMER?
- Q2 Is the proposed multi-modal SCAN using the encoder/decoder/encoder-decoder fusion effective?
- Q3 How does SCAN improve the performance by attention visualization?
- Q4 Can SCAN help accelerate the recognition speed?

The experiments are all implemented with Pytorch 0.4.1 [52] and an NVIDIA GeForce GTX 1080Ti 11G GPU.

6.1. Dataset and metric

Our experiments are conducted on CROHME competition database [53,54], which is currently the most widely used dataset for HMER. The CROHME 2014 competition dataset consists of a training set of 8836 HMEs and a testing set of 986 HMEs. The CROHME 2016 competition dataset only includes a testing set of 1147 HMEs. There are totally 101 math symbol classes and none of the handwritten expressions in the testing set appears in the training set. We apply CROHME 2014 training set as our training set and evaluate the performance of our models on CROHME 2014 testing set and CROHME 2016 testing set. Besides, we also evaluate our models on the latest CROHME 2019 competition dataset [55] of 1199 HMEs.

The main metric in this study is expression recognition rate (ExpRate) [56], i.e., the percentage of predicted mathematical expressions matching the ground truth. Besides, we list the structure recognition rate (StruRate) [56], which only focuses on whether the structure is correctly recognized and ignores symbol recognition errors.

6.2. Evaluation of single-modal SCAN (Q1)

In this section, we examine the effectiveness of single-modal SCAN. First, we investigate the performance of different encoder-decoder approaches for the online modality as shown in Table 1. TAP refers to the improved version of encoder-decoder approach using point-level features as in [14]. OnSCAN+TAP denotes the decoder fusion of TAP using point-level features and OnSCAN using online stroke-level features via the multi-modal attention in [14]. The ExpRate is increased from 48.47% to 51.22% on CROHME 2014 testing set and from 44.81% to 46.12% on CROHME 2016 testing set after replacing point-level features (TAP) with online stroke-level features (OnSCAN). By comparing OnSCAN with OnSCAN+TAP, the ExpRate is increased from 51.22% to 52.64% on CROHME 2014 testing set and from 46.12% to 47.17% on CROHME 2016 testing set. Similar observations could be made for StruRate. All these results demonstrate the superiority of online stroke-level features as

Table 2
Performance comparison of different encoder-decoder approaches for the offline modality using pixel-level features (WAP), offline stroke-level features (OffSCAN), and feature fusion in decoder (OffSCAN+WAP) on CROHME 2014 and CROHME 2016 testing sets.

System	CROHME 2014		CROHME 2016	
	ExpRate	StruRate	ExpRate	StruRate
WAP [14]	48.38%	70.08%	46.82%	66.17%
OffSCAN	47.67%	68.56%	46.64%	65.65%
OffSCAN+WAP	49.39%	71.81%	49.60%	68.18%

Table 3
Performance comparison of different encoder-decoder approaches for both online and offline modalities on CROHME 2019 testing set. The expression recognition accuracies with one, two and three errors per expression are represented by "≤ 1", "≤ 2" and "≤ 3".

System	ExpRate	≤ 1	≤ 2	≤ 3	StruRate
TAP [14]	44.20%	58.80%	62.72%	63.55%	63.64%
OnSCAN	46.46%	62.47%	66.14%	67.14%	66.31%
OnSCAN+TAP	47.62%	62.64%	67.06%	67.72%	67.22%
WAP [14]	48.12%	63.47%	67.22%	67.97%	67.97%
OffSCAN	47.62%	63.14%	67.06%	67.56%	67.81%
OffSCAN+WAP	49.62%	66.89%	69.97%	70.73%	70.56%

Table 4
Performance comparison of different multi-modal approaches on CROHME 2014 and CROHME 2016 testing sets.

System	CROHME 2014		CROHME 2016	
	ExpRate	StruRate	ExpRate	StruRate
MAN [14]	52.43%	71.60%	49.87%	68.18%
E-MAN [14]	54.05%	72.11%	50.56%	67.39%
MMSCAN-D	55.38%	71.30%	52.22%	68.35%
MMSCAN-E	57.20%	73.94%	53.97%	70.62%
MMSCAN-ED	58.11%	74.24%	54.29%	69.89%

Table 5
Performance comparison of different multi-modal approaches on CROHME 2019 testing set. The expression recognition accuracies with one, two and three errors per expression are represented by "≤ 1", "≤ 2" and "≤ 3".

System	ExpRate	≤ 1	≤ 2	≤ 3	StruRate
MAN	52.21%	66.64%	69.97%	70.39%	70.56%
E-MAN	52.88%	67.64%	70.81%	71.06%	71.06%
MMSCAN-D	53.88%	68.31%	70.56%	71.14%	70.98%
MMSCAN-E	56.21%	69.47%	71.64%	72.06%	71.73%
MMSCAN-ED	57.38%	71.61%	73.98%	74.48%	74.14%
USTC-iFLYTEK	80.73%	88.99%	90.74%	-	91.49%
Samsung R&D 1	79.82%	87.82%	89.15%	-	89.32%
MyScript	79.15%	86.82%	89.82%	-	90.66%

a higher-level representation over the point-level features and the complementarity between them.

Then we compare the performance of different encoder-decoder approaches for the offline modality as shown in Table 2. WAP refers to the improved version of encoder-decoder approach using pixel-level features as in [14]. OffSCAN+WAP denotes the decoder fusion of WAP using pixel-level features and OffSCAN using offline stroke-level features via the multi-modal attention in [14]. Compared with WAP, the ExpRate of OffSCAN is slightly decreased from 48.38% to 47.67% on CROHME 2014 testing set and from 46.82% to 46.64% on CROHME 2016 testing set. This observation is different from that in online modality by the comparison between TAP and OnSCAN. The reason might be that the pooling operation of 2D images in offline modality leads to higher misalignment between each stroke and the corresponding pixels (or points) than that of 1D sequence in online modality. However, performance improvements could be achieved by OffSCAN+WAP over both WAP and OffSCAN, e.g., with ExpRate increasing from 48.38%/47.67% to 49.39% on CROHME 2014 testing set and from 46.82%/46.64% to 49.60% on CROHME 2016 testing set, which indicates the strong complementarity between the offline stroke-level features and pixel-level features.

To further confirm the generalization of single-modality SCAN, we also evaluate on the latest CROHME 2019 competition database as shown in Table 3. For online modality, OnSCAN can achieve better performance than TAP while OnSCAN+TAP can achieve the best performance. As for offline modality, OffSCAN slightly underperforms WAP but OffSCAN+WAP still yields the best performance. All these variation trends on CROHME 2019 testing set are the same as those on CROHME 2014 and 2016 testing sets, which verify the effectiveness of single-modal SCAN for online HMER in both online modality and offline modality.

6.3. Evaluation of multi-modal SCAN (Q2)

In Table 4, we show the performance comparison of different multi-modal approaches on CROHME 2014 and CROHME 2016 testing sets. Please note that MAN and E-MAN are our previously proposed work [14] using the decoder fusion of point-level features and pixel-level features. And E-MAN is an enhanced version of MAN by adopting the re-attention mechanism. E-MAN can be considered as the decoder fusion of TAP and WAP while MMSCAN-D is the decoder fusion of OnSCAN and OffSCAN. So from the point-level/pixel-level feature fusion to online/offline stroke-level feature fusion (E-MAN vs. MMSCAN-D), the ExpRate is increased from 54.05% to 55.38% on CROHME 2014 testing set and from 50.56% to 52.22% on CROHME 2016 testing set, still demonstrating the superiority of treating stroke as the basic modeling unit rather than point or pixel in the multi-modal case. Besides, as SCAN provides oracle alignments between online and offline modalities, MMSCAN-E using the encoder fusion outperforms MMSCAN-D using the decoder fusion, i.e., with the ExpRate increasing from 55.38% to 57.20% on CROHME 2014 testing set and from 52.22% to 53.97% on CROHME 2016 testing set, which confirms that the early-stage encoder fusion is better than the late-stage decoder fusion in the SCAN framework. Finally, MMSCAN-ED can achieve the best ExpRate results (58.11% on CROHME 2014 testing set and 54.29% on CROHME 2016 testing set), which demonstrates that combining the encoder fusion and decoder fusion is useful.

Furthermore, we evaluate the above approaches on CROHME 2019 testing set in Table 5 to show that the improvements are significant and stable. Note that the best system MMSCAN-ED can achieve more significant gains over MMSCAN-E and MMSCAN-D compared with CROHME 2014 and 2016 datasets. Overall, in comparison to single-modal approaches OnSCAN and OffSCAN, the best performing multi-modal approach MMSCAN-ED yields large performance gains, e.g., with an absolute ExpRate gain of 9.76% and an absolute StruRate gain of 6.33% on CROHME 2019 testing set (OffSCAN in Table 3 vs. MMSCAN-ED in Table 5). Besides, we list the top three systems in CROHME 2019 competition [55] and all these systems can achieve very high performance but they use data augmentation (or external data) and other strategies.

Finally, we make a comparison of our best approach MMSCAN-ED and other state-of-the-art approaches on both CROHME 2014 and CROHME 2016 testing sets, as shown in Table 6. The system Wiris, Tokyo and São Paulo denote the top three systems in CROHME 2016 competition using only official dataset (note that Wiris actually used an additional large corpus to train a strong language model) and the details can be seen in [54]. WYGIWYS [13] is an encoder-decoder model with a coarse-to-fine attention to improve efficiency. PAL [37] introduced a paired adversarial learning method to help solve difficulties in HMER due to writing styles. Please note that the results of the end-to-end approaches are not

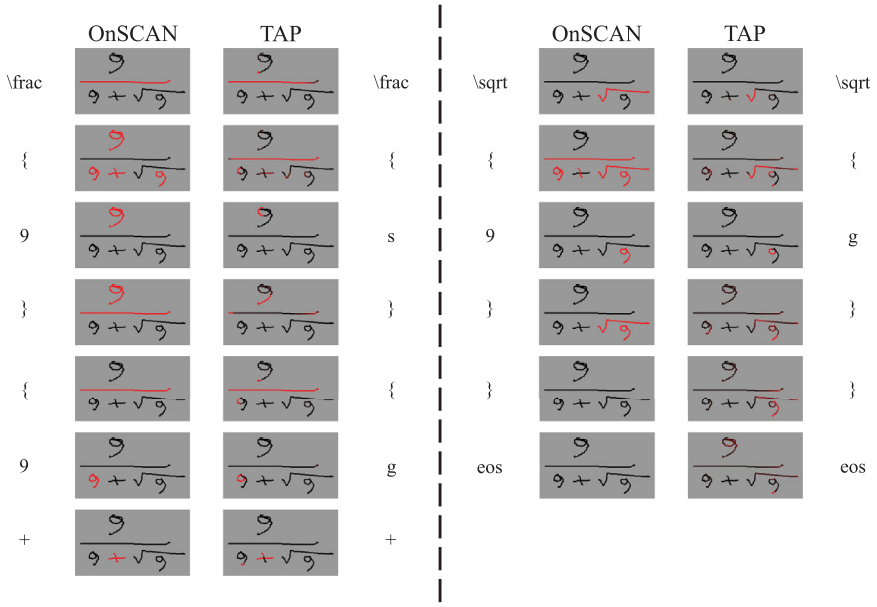

Fig. 9. The attention visualization and recognition result comparison between OnSCAN and TAP for one handwritten mathematical expression with the LaTeX ground truth " \frac { 9 } { 9 + \sqrt { 9 } } ".

Table 6
Overall performance comparison on CROHME 2014 and CROHME 2016 testing sets.

System	CROHME 2014		CROHME 2016	
	ExpRate	StruRate	ExpRate	StruRate
Wiris	-	-	49.61%	74.28%
Tokyo	-	-	43.94%	61.55%
São Paulo	-	-	33.39%	57.02%
WYGIWYS [13]	35.90%	-	-	-
PAL [37]	39.66%	-	-	-
TAP	48.47%	67.24%	44.81%	63.12%
WAP	48.38%	70.08%	46.82%	66.17%
PGS [57]	48.78%	-	45.60%	-
Res-BiRNN [58]	53.35%	-	47.95%	-
MAN	52.43%	71.60%	49.87%	68.18%
E-MAN	54.05%	72.11%	50.56%	67.39%
MMSCAN-ED	**58.11%**	**74.24%**	**54.29%**	69.89%

exactly comparable with traditional approaches in the submitted systems to CROHME competitions as the segmentation error is not explicitly considered. Obviously, the proposed MMSCAN-ED significantly outperforms other end-to-end approaches with an ExpRate of 58.11% on CROHME 2014 testing set and an ExpRate of 54.29% on CROHME 2016 testing set.

6.4. Attention visualization (Q3)

In Section 6.2 and Section 6.3, we have demonstrated that SCAN can improve the performance of online HMER in both single-modal and multi-modal cases. In this section, we further show that SCAN can acquire more accurate symbol segmentation which is performed by attention. Moreover, the advantage of encoder fusion over the decoder fusion is explained by attention visualization.

We first compare the attention and recognition results of OnSCAN and TAP of one handwritten mathematical expression with the LaTeX ground truth " \frac { 9 } { 9 + \sqrt { 9 } } " in Fig. 9. It is obvious that OnSCAN correctly recognizes the example expression while TAP fails. Specifically, OnSCAN can focus on the exact points of the current predicted symbol at each time step, which in fact achieves accurate symbol segmentation and thus generates correct recognition result. As for TAP, it can only focus on parts of the points belonging to the current symbol. Besides, it will improperly focus on some redundant points belonging to other symbols. Therefore, TAP mistakenly recognizes the first "9" as "s" and the second/third "9" as "g". It is reasonable as the attended parts can be regraded as a part of "9", "s" or "g".

The attention and recognition results of MMSCAN-E and MMSCAN-D for one handwritten mathematical expression are shown in Fig. 10. As the decoder of MMSCAN-D accepts both online and offline stroke-level features, accordingly attention results for both online and offline modalities are given. Ideally, the attention results of MMSCAN-D online and MMSCAN-D offline should be the same, namely the same attended strokes belonging to the symbol at each decoding step. However, as MMSCAN-D generates attention results over online and offline stroke-level features separately in the decoder, the attention results for online and offline modalities might be different leading to incorrect recognition. For example, at the first three steps of MMSCAN-D, the different attention results of online and offline modalities lead to a deletion error, namely incorrectly recognizing " (- \infty " as " (\infty " with the symbol "-" missing. On the contrary, MMSCAN-E generates exactly accurate attention results at each step and correctly recognizes the example expression. This indicates the superiority of early-stage fusion by better utilizing the alignments between online and offline modalities.

One main motivation of multi-modal fusion for online HMER is to overcome problems in single modality by using information from both online and offline modalities. For example, a very common problem is caused by delayed strokes. As shown in Fig. 11, we take one expression with the LaTeX ground truth " B + B = B " as an example. The difficulty of recognizing this sample is that the writing order of this expression is different from the normal writing order. This expression consists of ten strokes with the corresponding writing order in Fig. 11. In general, after writing the second "B", we will write the "=" by two strokes. However, in this example, one stroke (marked red) of the symbol "=" is delayed to be written as the final stroke, which makes online modality difficult to correctly recognize. Although MMSCAN-D has information of two modalities and adopts re-attention to implement information interaction, it only has single modality information in pre-

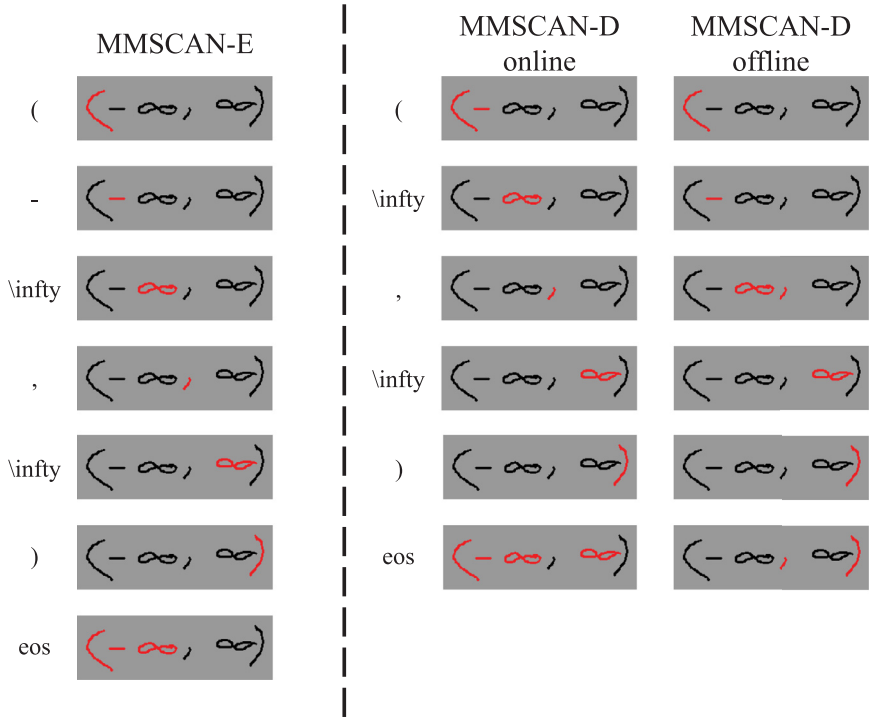

Fig. 10. Attention and recognition results of MMSCAN-E and MMSCAN-D for one handwritten mathematical expression with the LaTeX ground truth " (- \infty, \infty) ".

Table 7
Comparison of time efficiency between SCAN and local (point-level or pixel-level) feature based approaches in both single-modal and multi-modal cases.

Modality	System	ExpRate	StruRate	Time Cost
Online	TAP	48.47%	67.24%	1
	OnSCAN	51.22%	70.49%	0.91
Offline	WAP	48.38%	70.08%	1.51
	OffSCAN	47.67%	68.56%	1.28
Multi-modal	E-MAN	54.05%	72.11%	2.66
	MMSCAN-D	55.38%	71.30%	1.94
	MMSCAN-E	57.20%	73.94%	1.32
	MMSCAN-ED	58.11%	74.24%	3.34

attention and still meets problems as the errors caused by pre-attention will be inherited in fine-attention. Therefore, MMSCAN-D incorrectly recognizes this expression as " B + B - B". Nevertheless, MMSCAN-E fuses two modalities in encoder, which authentically has both online and offline information when performing attention in the decoder. As a result, the global information in offline modality can help solve the delayed stroke problem and MMSCAN-E correctly recognizes this expression.

6.5. Comparison of recognition speed (Q4)

We compare the computational costs of whether employing SCAN in single-modal and multi-modal cases by investigating the test speed in this section. We present the total time cost (normalized by the time cost of TAP system) for recognizing the CROHME 2014 testing set in Table 7. For the single modality, it is obvious that converting point-level/pixel-level features (TAP/WAP) into online/offline stroke-level features (OnSCAN/OffSCAN) can accelerate the testing procedure as the number of strokes is much smaller than the number of points/pixels, which reduces the computation cost of the decoder part. Similarly, in the multi-modal case, MMSCAN-D is faster than E-MAN as MMSCAN-D replaces both point-level and pixel-level features with online and offline stroke-level features at the same time. MMSCAN-E with the better ExpRate and StruRate can also achieve the better efficiency compared with MMSCAN-D and E-MAN due to the early-stage fusion. Besides, MMSCAN-E system only uses a half of time cost of E-MAN system and is even faster than offline WAP system. The system MMSCAN-ED achieves the best recognition performance but demands the highest computational cost.

7. Experiments on online HCCR

In this section, we employ the proposed SCAN in online HCCR to evaluate the generalization and robustness. We use CASIA dataset [59], including OLHWDB1.0 and OLHWDB1.1 as our training set and ICDAR 2013 Chinese handwritten recognition competition dataset [60] as our testing set. The raw data is handwritten traces with stroke constrained information and we employ the same data preparation as HMER to acquire online input, online stroke masks, offline input and offline stroke masks. Note the difference here is we resize each offline input (a static image) as 64×64.

Instead of treating a Chinese character as a single character category, we treat each character as a composition of radicals as RAN [61]. Then, the recognition of handwritten Chinese character becomes a sequential problem rather than a classification problem and can be solved by encoder-decoder frameworks. We use the same training criterion as in HMER except stroke-level attention guider as there is no such information in online HCCR. The experimental results are shown in Table 8.

For single-modal part, OnSCAN can achieve better performance than TAP while OffSCAN is slightly worse than WAP, which has the same trendency as in HMER. Note that WAP here is called RAN in [61] for unified description. As for multi-modal part, MMSCAN-E can achieve better performance while MMSCAN-D outperforms E-MAN, which proves the effectiveness of encoder fusion and stroke-level representation. Besides, MMSCAN-ED can still achieve the best performance, which further proves the effectiveness of the encoder-decoder fusion. The results of these experiments demon-

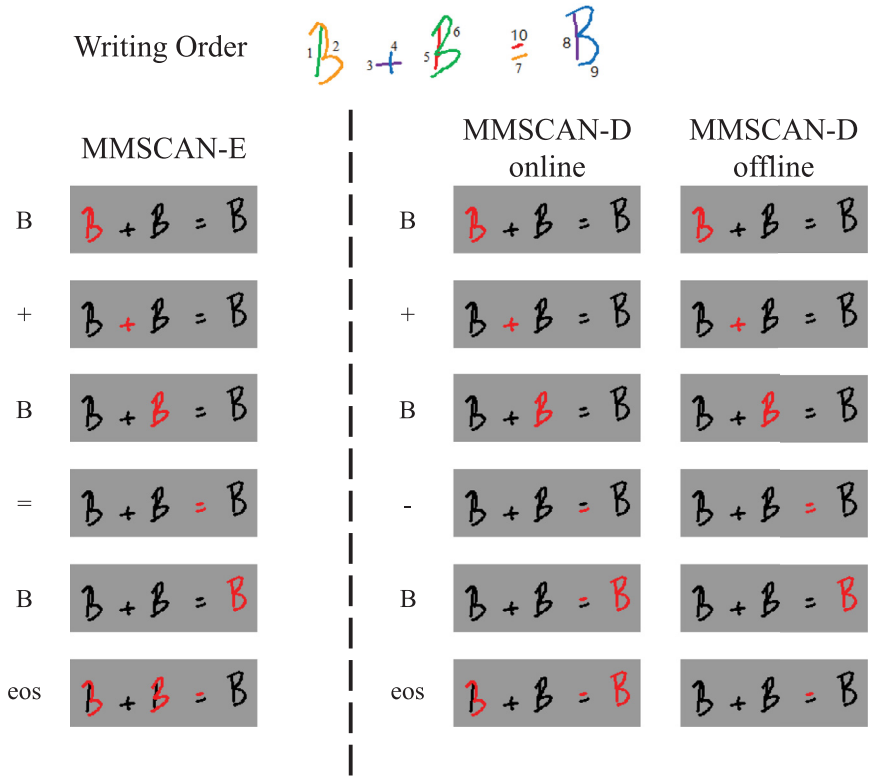

Fig. 11. Attention visualization and recognition results of MMSCAN-E and MMSCAN-D for one handwritten mathematical expression with the LaTeX ground truth " B + B = B ". The problem of delayed strokes exists in this example.

Table 8
The comparison of different models on online handwritten Chinese character recognition.

Single-modal		Multi-modal	
System	Accuracy	System	Accuracy
TAP	96.55%	E-MAN	96.91%
OnSCAN	96.67%	MMSCAN-D	97.04%
WAP	94.89%	MMSCAN-E	97.11%
OffSCAN	94.57%	MMSCAN-ED	97.16%

strate that SCAN can also be adopted in online HCCR and achieve better performance.

8. Conclusion and future work

In this study, we introduce a novel stroke constrained attention network (SCAN) for online handwritten mathematical expression recognition and online handwritten Chinese character recognition. The proposed model can be applied in both single-modal and multi-modal cases. For single modal case, SCAN can help attention learn easily and better than TAP or WAP, as SCAN adopts stroke as the basic unit, which is a high-level representation than point (TAP) or pixel (WAP). Specifically, by using this high-level representation, attention becomes to achieve the new alignment, i.e., which strokes belonging to each symbol, rather than which points or pixels belonging to each symbol. Besides, as discussed in Section 4, SCAN achieves the feasibility of encoder fusion as SCAN can provide oracle alignment between online and offline modalities, which is very important in multi-modal machine learning. By combining the encoder fusion and decoder fusion, we achieve the encoder-decoder fusion with the best performance. We demonstrate through experimental results that SCAN can significantly improve the recognition performance and accelerate the testing procedure. Moreover, we verify that SCAN greatly improves the alignment via the attention visualization. In the future, we aim to investigate a better approach for fusing features from different modalities to acquire a more reasonable representation and we will investigate the explicit symbol segmentation by using attention results, which can be utilized as stroke-level attention guider. Furthermore, we will consider more complex situations such as repeating strokes or cross-out while writing [62].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by the MOE-Microsoft Key Laboratory of USTC, and Youtu Lab of Tencent.

References

[1] N. Bhattacharya, P.P. Roy, U. Pal, Sub-stroke-wise relative feature for online indic handwriting recognition, ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 18 (2) (2018) 1–16.
[2] E.G. Miller, P.A. Viola, Ambiguity and constraint in mathematical expression recognition, in: AAAI, 1998, pp. 784–791.
[3] R.H. Anderson, Syntax-directed recognition of hand-printed two-dimensional mathematics, in: Symposium on Interactive Systems for Experimental Applied Mathematics: Proceedings of the Association for Computing Machinery Inc. Symposium, 1967, pp. 436–459.
[4] A. Belaid, J.-P. Haton, A syntactic approach for handwritten mathematical formula recognition, IEEE Trans Pattern Anal Mach Intell (1) (1984) 105–111.
[5] K.-F. Chan, D.-Y. Yeung, Mathematical expression recognition: a survey, Int. J. Doc. Anal. Recogn. 3 (1) (2000) 3–15.

[6] R. Zanibbi, D. Blostein, J.R. Cordy, Recognizing mathematical expressions using tree transformation, IEEE Trans Pattern Anal Mach Intell 24 (11) (2002) 1455–1467.
[7] F. Álvaro, J.-A. Sánchez, J.-M. Benedí, Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models, Pattern Recognit Lett 35 (2014) 58–67.
[8] A.-M. Awal, H. Mouchère, C. Viard-Gaudin, A global learning approach for an online handwritten mathematical expression recognition system, Pattern Recognit Lett 35 (2014) 68–77.
[9] F. Alvaro, J.-A. Sánchez, J.-M. Benedí, An integrated grammar-based approach for mathematical expression recognition, Pattern Recognit 51 (2016) 135–147.
[10] J. Zhang, J. Du, L. Dai, A gru-based encoder-decoder approach with attention for online handwritten mathematical expression recognition, in: International Conference on Document Analysis and Recognition, 1, 2017, pp. 902–907.
[11] J. Zhang, J. Du, L. Dai, Track, attend and parse (TAP): an end-to-end framework for online handwritten mathematical expression recognition, IEEE Trans Multimedia 21 (1) (2019) 221–233.
[12] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, L. Dai, Watch, attend and parse: an end-to-end neural network based approach to handwritten mathematical expression recognition, Pattern Recognit 71 (2017) 196–206.
[13] Y. Deng, A. Kanervisto, J. Ling, A.M. Rush, Image-to-markup generation with coarse-to-fine attention, in: International Conference on Machine Learning, 2017, pp. 980–989.
[14] J. Wang, J. Du, J. Zhang, Z.-R. Wang, Multi-modal attention network for handwritten mathematical expression recognition, in: International Conference on Document Analysis and Recognition, 2019, pp. 1181–1186.
[15] S. Lehmberg, H.-J. Winkler, M. Lang, A soft-decision approach for symbol segmentation within handwritten mathematical expressions, in: International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, 6, 1996, pp. 3434–3437.
[16] K.-F. Chan, D.-Y. Yeung, Pencalc: A novel application of on-line mathematical expression recognition technology, in: International Conference on Document Analysis and Recognition, 2001, pp. 774–778.
[17] E. Tapia, R. Rojas, Recognition of on-line handwritten mathematical formulas in the e-chalk system, in: International Conference on Document Analysis and Recognition, 3, 2003, pp. 980–984.
[18] D. Pru_a, V. Hlavác, Mathematical formulae recognition using 2d grammars, in: International Conference on Document Analysis and Recognition, 2, 2007, pp. 849–853.
[19] T.H. Rhee, J.H. Kim, Efficient search strategy in structural analysis for handwritten mathematical expression recognition, Pattern Recognit 42 (12) (2009) 3192–3201.
[20] H. Ney, D. Mergel, A. Noll, A. Paeseler, A data-driven organization of the dynamic programming beam search for continuous speech recognition, in: ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing, 12, IEEE, 1987, pp. 833–836.
[21] S. Abdou, M.S. Scordilis, Beam search pruning in speech recognition using a posterior probability-based confidence measure, Speech Commun 42 (3–4) (2004) 409–428.
[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[23] T. He, X. Tan, Y. Xia, D. He, T. Qin, Z. Chen, T.-Y. Liu, Layer-wise coordination between encoder and decoder for neural machine translation, in: Advances in Neural Information Processing Systems, 2018, pp. 7944–7954.
[24] P.-Y. Huang, F. Liu, S.-R. Shiang, J. Oh, C. Dyer, Attention-based multimodal neural machine translation, in: Conference on Machine Translation, 2, 2016, pp. 639–645.
[25] W. Chan, N. Jaitly, Q. Le, O. Vinyals, Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, in: International Conference on Acoustics, Speech and Signal Processing, 2016, pp. 4960–4964.
[26] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio, End-to-end attention-based large vocabulary speech recognition, in: International Conference on Acoustics, Speech and Signal Processing, 2016, pp. 4945–4949.
[27] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: International Conference on Machine Learning, 2015, pp. 2048–2057.
[28] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, T.-S. Chua, Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5659–5667.
[29] J. Lu, C. Xiong, D. Parikh, R. Socher, Knowing when to look: Adaptive attention via a visual sentinel for image captioning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 375–383.
[30] A.K. Bhunia, A. Bhowmick, A.K. Bhunia, A. Konwer, P. Banerjee, P.P. Roy, U. Pal, Handwriting trajectory recovery using end-to-end deep encoder-decoder network, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 3639–3644.
[31] K. Cho, A. Courville, Y. Bengio, Describing multimedia content using attention-based encoder-decoder networks, IEEE Trans Multimedia 17 (11) (2015) 1875–1886.
[32] J. Xu, R. Zhao, F. Zhu, H. Wang, W. Ouyang, Attention-aware compositional network for person re-identification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2119–2128.
[33] T. Bluche, J. Louradour, R. Messina, Scan, attend and read: end-to-end handwritten paragraph recognition with MDLSTM attention, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 1, IEEE, 2017, pp. 1050–1055.
[34] T. Bluche, Joint line segmentation and transcription for end-to-end handwritten paragraph recognition, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 838–846.
[35] A.K. Bhunia, A. Konwer, A.K. Bhunia, A. Bhowmick, P.P. Roy, U. Pal, Script identification in natural scene image and video frames using an attention based convolutional-lSTM network, Pattern Recognit 85 (2019) 172–184.
[36] Z. Hong, N. You, J. Tan, N. Bi, Residual BiRNN based Seq2Seq model with transition probability matrix for online handwritten mathematical expression recognition, in: International Conference on Document Analysis and Recognition, 2019, pp. 635–640.
[37] J.-W. Wu, F. Yin, Y.-M. Zhang, X.-Y. Zhang, C.-L. Liu, Image-to-markup generation via paired adversarial learning, in: European Conference on Machine Learning and Knowledge Discovery in Databases, 2018, pp. 18–34.
[38] T. Baltrušaitis, C. Ahuja, L.-P. Morency, Multimodal machine learning: a survey and taxonomy, IEEE Trans Pattern Anal Mach Intell 41 (2) (2018) 423–443.
[39] A.K. Bhunia, S. Mukherjee, A. Sain, A.K. Bhunia, P.P. Roy, U. Pal, Indic handwritten script identification using offline-online multi-modal deep network, Information Fusion 57 (2020) 1–14.
[40] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1933–1941.
[41] S. Medjkoune, H. Mouchère, S. Petitrenaud, C. Viard-Gaudin, Handwritten and audio information fusion for mathematical symbol recognition, in: 2011 International Conference on Document Analysis and Recognition, IEEE, 2011, pp. 379–383.
[42] S. Chen, Q. Jin, Multi-modal dimensional emotion recognition using recurrent neural networks, in: International Workshop on Audio/Visual Emotion Challenge, 2015, pp. 49–56.
[43] S.E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, et al., Emonets: multimodal deep learning approaches for emotion recognition in video, Journal on Multimodal User Interfaces 10 (2) (2016) 99–111.
[44] J. Lu, J. Yang, D. Batra, D. Parikh, Hierarchical question-image co-attention for visual question answering, in: Advances in neural information processing systems, 2016, pp. 289–297.
[45] Z. Yu, J. Yu, C. Xiang, J. Fan, D. Tao, Beyond bilinear: generalized multimodal factorized high-order pooling for visual question answering, IEEE Trans Neural Netw Learn Syst 29 (12) (2018) 5947–5959.
[46] Y.-H.H. Tsai, S. Bai, P.P. Liang, J.Z. Kolter, L.-P. Morency, R. Salakhutdinov, Multimodal transformer for unaligned multimodal language sequences, arXiv preprint arXiv:1906.00295 (2019).
[47] L. Ye, M. Rochan, Z. Liu, Y. Wang, Cross-modal self-attention network for referring image segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10502–10511.
[48] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[49] Z. Tu, Z. Lu, Y. Liu, X. Liu, H. Li, Modeling coverage for neural machine translation, arXiv preprint arXiv:1601.04811 (2016).
[50] M.D. Zeiler, Adadelta: an adaptive learning rate method, arXiv preprint arXiv:1212.5701 (2012).
[51] K. Cho, Natural language understanding with distributed representation, arXiv preprint arXiv:1511.07916 (2015).
[52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019, pp. 8024–8035.
[53] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, U. Garain, Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014), in: International Conference on Frontiers in Handwriting Recognition, 2014, pp. 791–796.
[54] H. Mouchère, C. Viard-Gaudin, R. Zanibbi, U. Garain, ICFHR2016 crohme: Competition on recognition of online handwritten mathematical expressions, in: International Conference on Frontiers in Handwriting Recognition, 2016, pp. 607–612.
[55] M. Mahdavi, R. Zanibbi, H. Mouchere, C. Viard-Gaudin, U. Garain, Icdar 2019 crohme+ tfd: Competition on recognition of handwritten mathematical expressions and typeset formula detection, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2019, pp. 1533–1538.
[56] H. Mouchere, R. Zanibbi, U. Garain, C. Viard-Gaudin, Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011–2014, Int. J. Doc. Anal. Recogn. 19 (2) (2016) 173–189.
[57] A.D. Le, B. Indurkhya, M. Nakagawa, Pattern generation strategies for improving recognition of handwritten mathematical expressions, Pattern Recognit Lett 128 (2019) 255–262.
[58] Z. Hong, N. You, J. Tan, N. Bi, Residual BiRNN based Seq2Sq model with transition probability matrix for online handwritten mathematical expression recognition, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2019, pp. 635–640.
[59] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, Casia online and offline chinese handwriting databases, in: 2011 International Conference on Document Analysis and Recognition, IEEE, 2011, pp. 37–41.
[60] F. Yin, Q.-F. Wang, X.-Y. Zhang, C.-L. Liu, Icdar 2013 chinese handwriting recog-

nition competition, in: 2013 12th International Conference on Document Analysis and Recognition, IEEE, 2013, pp. 1464–1470.
[61] J. Zhang, J. Du, L. Dai, Radical analysis network for learning hierarchies of chinese characters, Pattern Recognit (2020) 107305.
[62] N. Bhattacharya, V. Frinken, U. Pal, P.P. Roy, Overwriting repetition and crossing-out detection in online handwritten text, in: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, 2015, pp. 680–684.

Jiaming Wang received a B.Eng. degree from Anhui University in 2018. He is currently a Master degree candidate of University of Science and Technology of China (USTC). His current research area is handwritten mathematical expression recognition and Chinese character recognition.

Jun Du received his B.Eng. and Ph.D. degrees from the Department of Electronic Engineering and Infor-mation Science, University of Science and Technology of China (USTC) in 2004 and 2009, respectively. From 2004 to 2009, he was with iFlytek Speech Lab of USTC. During the above years, he worked as an Intern for two 9-month periods at Microsoft Research Asia (MSRA), Beijing. In 2007, he worked as a Research Assistant for 6 months in the Department of Computer Science, University of Hong Kong. From July 2009 to June 2010, he worked at iFLYTEK Research on speech recognition. From July 2010 to January 2013, he joined MSRA as an Associate Researcher, working on handwriting recognition, OCR, and speech recognition. Since February 2013, he has been with the National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP) of USTC.

Jianshu Zhang received his B.Eng. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC) in 2015. He is currently a Ph.D. candidate of USTC. In 2018, he worked as a visiting student for 6 months in the Queen Mary University of London. His current research areas include deep learning, handwriting mathematical expression recognition, Chinese document analysis and speech analysis. February 19, 2020 DRAFT