A Sequence-to-Sequence Framework Based on Transformer With Masked Language Model for Optical Music Recognition

CUIHONG WEN$^{1,2}$ AND LONGJIAO ZHU$^1$

$^1$College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China
$^2$Department of Advanced Manufacturing and Robotics, College of Engineering, Peking University, Beijing 100871, China

Corresponding author: Cuihong Wen (cuihongwen@hunnu.edu.cn)

This work was supported in part by the Natural Science Foundation of Hunan Province, China, under Grant 2020JJ5372 and Grant 2019JJ50363; and in part by the Scientific Research Projects of Department of Education of Hunan Province under Grant 18C0005.

ABSTRACT Optical music recognition technology is of great significance in the development of the digital music. In recent years, the convolutional recurrent neural network framework with connectionist temporal classification has been used in music recognition. However, its loss function is calculated in serial mode, which leads to low efficiency in training and difficulty in convergence. Additionally, because of the gradient disappearance of excessive long music sequences, the existing music recognition models are hard to learn the relationships between musical symbols, resulting in high sequence error rate. Therefore, we propose a sequence-to-sequence framework based on transformer with masked language model to deal with these problems. The context representation between musical symbols can be captured further by the self-attention module in the transformer, which will reduce the sequence error rate. In addition, we refer to the masked language model and design a mask matrix to predict each musical symbol in a parallel way, so as to speed up the training process. Our experiments are carried out on the printed images of music stave dataset, and the results show that our proposed method is training-efficient and has great improvement in sequence accuracy rate.

INDEX TERMS Masked language model, neural networks, optical music recognition, sequence-to-sequence, transformer.

I. INTRODUCTION

In the early days, paper was the main means for composers to record their music, namely handwritten music scores. Through handwritten music scores, music can be preserved and transmitted around the world, but it was delivered slowly and worn out easily as time goes by. With the development of computer and storage technologies, some researchers began to turn their eyes to the application of computer in music [1], [2], [3], [4], [5]. For this reason, it is necessary to talk about Optical Music Recognition (OMR) technology. OMR technology is designed to teach computers to read and transcribe music symbols into a machine-readable digital format (such as MIDI or MusicXML). On the one hand, digital music saved in computer can be preserved for a long time with less space and spread more widely and faster. On the other hand, digital music makes the traditional manual way of music creation, modification, performance, transmission become intelligent, which has brought a fundamental change in production mode for human music activities. In addition, it makes the music retrieval based on the music contents more easily. Therefore, the OMR technology is important to the development of digital music library, music teaching with computer and music information retrieval.

Traditionally, OMR pipelines consist of four stages: preprocessing, music primitives recognition, symbols assembly and encoding [3]. And there are a large number of related works [4], [5], [6]. However, each stage of the pipelines is hard to get a high accuracy, so that a certain number of errors will be caused and could be exponentially enlarged in the...
successive stages. With the evolution of machine learning, Support Vector Machine (SVM) [7], k-Nearest Neighbor (kNN) [8], Hidden Markov Model (HMM) [9] are applied in OMR, but it is difficult that they usually need to extract features manually to train a classifier. In recent years, deep learning has become a trend of application in various artificial intelligence tasks due to its ability in automatic feature extraction. Among them, both Convolutional Neural Network (CNN) which is good at processing visual images, and Recurrent Neural Network (RNN) which does well in handling sequential issues, have got remarkable results in music recognition. Besides, the Connectionist Temporal Classification (CTC) is proposed firstly for training RNNs to label unsegmented sequence directly [15]. Then it was used to train deep learning models for OMR with an end-to-end manner in [16] and [17].

However, the Convolutional Recurrent Neural Network (CRNN) with CTC for OMR still has some limitations. Firstly, the serial training of CTC is time-consuming, and it is difficult to converge when one of the predictions is error, because the next prediction in it is based on the previous predictions. Secondly, the input image is usually regarded as a sequence whose length is the width of the image. For CRNN, this will cause the gradient disappearance because the width is always large. And it is hard to capture the context features between symbols while these features are very important for prediction. Thirdly, it is noted that most of music scores in the real world are chord music scores, which means the output of OMR model is not only one at each time step. But, the output in CTC can only be single, and the length of output sequence must be less than or equal to the length of input sequence (i.e., the width of input image).

Given the above reasons, we design a Sequence-to-Sequence (Seq2Seq) framework based on Transformer with Masked Language Model, namely ST-MLM. Firstly, we refer to the Transformer with Masked Language Model (T-MLM) and manually define a mask matrix in Transformer to mask some future tokens that should not be seen. In contrast to traditional Transformer, the mask matrix can realize that the attention of the entire input sequence is bidirectional, while the attention of the output sequence is unidirectional. And the model also can predict all tokens at a time with parallel mode, so as to achieve efficient training and prevent overfitting. Secondly, with the help of self-attention in the Transformer, the model can better extract context relationships to reduce the sequence error rate. Thirdly, ST-MLM is based on the Seq2Seq framework whose feature dimension of output at each time step is unrestricted, and thus gets multiple outputs. And it can output a sequence of arbitrary length in recursive way.

In overall, the main contributions of this paper are given as follows:

1) We propose a ST-MLM architecture whose self-attention in Transformer can better extract the context relationships between musical symbols, so as to improve the musical sequence accuracy in OMR.

2) In order to train the model efficiently and faster, we design a mask matrix in Transformer. Compared with the recursive calculation of CTC, it masks some future tokens to predict the musical symbols in parallel way. Thus, the calculation of model speeds up.

3) We also analyse the effects of our proposed ST-MLM model on multi-output problem at one time-step in music recognition.

The structure of this paper is as follows. In Section 2, some related works are reviewed. Section 3 details our proposed ST-MLM structure. Section 4 presents some experimental settings and comparative results of our model. Finally, the entire contents are summarized in Section 5.

II. RELATED WORKS

A. MUSIC SCORE RECOGNITION WITH NEURAL NETWORK METHODS

With artificial neural networks, it is crucial to capture feature information of objectives as much as possible. Thus, a large number of high-quality images such as DeepScores, PrIMus, MuseScore, and sufficient training are the promise to get better performance. For decades, researchers have been done abundant works in OMR field with neural networks [11], [12], [13], [16], [18], [19], [20], [21], [22]. In [13], CNN and LSTM were applied to extract musical semantics including pitch and rhythm, then HMMs classify and interpret pitch and rhythm respectively. Moreover, Calvo-Zaragoza and Rizo [16] proposed a structure of CRNN/CTC to recognize monophonic scores. The model reaches a very low symbol error rate (0.8%), but the sequence error rate is still very high (12.5%). Calvo-Zaragoza et al. [19] hybridized HMM with deep multilayer perceptrons for handwritten music recognition. Baró et al. [20] explored the application of LSTM which helps in keeping the context dependence of symbols in OMR. In [21], a recurrent residual convolutional neural network was developed, and its experiments proved that this model has significant improvement than the up-to-date CRNN model.

Additionally, there are some music recognition tasks which rely on objects detection method, such as YOLO, which outputs both the symbol category and coordinates in the image [23]. In [24] and [25], two-dimensional natures of musical symbols were explored with an end-to-end approach. And some researchers treated input music score as a sentence, so they borrowed from text learning and machine translation [26]. In [27], authors followed the architecture, while the difference is that they add another RNN layer to model data linguistics characteristics implicitly. In order to take the performance of the agnostic case for OMR into account, Thomae et al. [28] implemented a similar machine translator that taking the agnostic representation of music symbols sequence in staff-level into corresponding semantic representation. For handwritten historical music recognition, Baró et al. [29] built a seq2seq model with attention to compare to CRNN/CTC model, and results showed that this model obtained promising performance.
**B. Seq2Seq FRAMEWORK**

Seq2Seq is an encoder-decoder framework whose input and output are all sequences [30]. The encoder encodes the input sequence of arbitrary length into a context vector. The decoder decodes the context vector into a prediction sequence with arbitrary length. Due to the arbitrary length of sequences, RNN is always employed as the encoder and the decoder [31]. However, the parallel efficiency of RNN is poor because of its iterative calculation. Therefore, Vaswani et al. [32] proposed Transformer, a powerful and parallel neural network module, to replace the RNN in Seq2Seq framework. The Transformer dispenses with the iterative calculation in encoder and based solely on attention mechanism. The Transformer encoder can achieve computational parallelism, but the Transformer decoder still needs to compute iteratively. After that, Dong et al. [33] used a designed mask matrix in Transformer to implement a parallel unidirectional language model, which makes it possible to decode parallely in training process. Therefore, we refer to UNILM and design a new mask matrix to train model parallely in OMR task.

**C. CTC**

CTC is proposed to label unsegmented acoustic signal in speech recognition because of its automatic alignment between input sequence and output sequence [15]. In speech recognition [34], the input sequence is an acoustic signal, and the output sequence is a sentence. In generally, we need to pre-segment training data frame by frame to generate the sentence, which is an extremely difficult operation. However, by using CTC, this problem can be solved well. Since OMR task is similar to the speech recognition, CTC can also be applied in OMR task [16].

**III. PROPOSED METHODS**

The proposed ST-MLM in our work is detailed in Figure 1. There are five modules: pre-processing, encoder layer, decoder layer, T-MLM layer and output layer. Among them, the encoder layer consists of a multi-scale CNN and a two-layer Bi-LSTM, and they aim to collect information of the music symbols. The decoder layer just includes a two-layer LSTM to embed the prediction information. In the T-MLM layer, a masked multi-head self-attention is used to further obtain the context relationship between musical symbols and decode the all collected information, then a feed-forward network is applied to get predictions representation.

**A. PRE-PROCESSING**

The symbols in Figure 2 are the main objectives to be recognized in the OMR task. However, there are many blanks...
which indicate no actual music symbol between every two separate musical symbols in the music image, which leads to some redundant computation, as illustrated in Figure 3(a). Therefore, we consider removing these blanks from the raw image. We provide a pixel-based approach to drop the blanks. First, we calculate pixel sum of each column of input image. If the pixel sum is close to zero, the column will be blank (in an image, pixel 0 is white, pixel 1 is black). Then, we retain these columns containing musical symbols and delete blank columns. Next, all the retained columns are concatenated together according to their original order, as shown in Figure 3(b). After that, images are rescaled at a fixed height of 128 pixels, without modifying their aspect ratio. We get a two-dimension vector with 128 height and w width. The vector can be viewed as an embedded sequence whose length is the width and the embedding size is the height. In addition, the predictive target is also a sequence containing digital musical symbols. Therefore, we will use a Seq2Seq framework to model their relationships.

FIGURE 2. The main music symbols to be recognized in single-staff image.

FIGURE 3. Single-staff music score (a) A raw input image (b) An image with height of 128 pixels after preprocessing.

B. ST-MLM FRAMEWORK FOR OPTICAL MUSIC RECOGNITION

Our ST-MLM framework contains an encoder part and a decoder part. In the encoder part, feature extraction is carried out on the input embedded sequence, and in the decoder part, it generates a predicted sequence based on the output of the encoder layer and the decoder layer. The Transformer with Masked Language Model (T-MLM) is not only as an encoder to increase context relationship, but also as a decoder to predict with parallel mode.

1) ENCODER LAYER

In the encoder layer, it is important to learn as abundant as possible features of musical symbols. We choose CNN and RNN to deal with the pre-processed image. According to different pixel width of different symbols, we use a multi-scale CNN to extract low-level features from musical symbols by applying different scale of one-dimension convolution kernels, and obtain multiple feature maps. Then, these feature maps are concatenated into a multi-dimension feature sequence. For the feature sequence, in order to capture the context relationship between musical symbols, we realize it through a two-layer Bi-LSMT neural network. At the same time, it is able to reduce the error of prediction by taking the forward and backward directions of symbols into account. The pre-processed image \(x'\) filled into the encoder layer can be calculated as:

\[
E = \text{Encoder}(x')
\]

where \(\text{Encoder}(\cdot)\) means the whole calculation of the multi-scale CNN and the Bi-LSTM neural network. And \(E \in \mathbb{R}^{l_1 \times f_1}\) is the output sequence of the encoder layer. \(l_1\) and \(f_1\) denote the length of the output sequence and the feature dimension of each column, respectively.

2) DECODER LAYER

At the decoder layer, we need to extract features for initial input labels. In traditional Seq2Seq framework, the main object of the decoder part is to generate target sequence corresponding to the input image. The decoder part generates next prediction by using previous predictions which could be incorrect. And the incorrect predictions will obtain incorrect target sequence, resulting in slow and hard convergence in training. Therefore, we apply the Teacher Forcing [35] technique in the decoder layer of ST-MLM, Teacher Forcing technique has a certain probability to replace the previous predictions with actual label, which can alleviate the problem of convergence. With Teacher Forcing technique, we need to extract features from the actual labels during the training of the model. Firstly, each actual label is embedded into a numeric vector. Then, a two-layer LSTM is applied to extract features of the numeric vectors. Thus, we can get the output sequence of the decoder layer. Assuming that the input of decoder layer is \(y\), and the calculation of decoder layer is shown as:

\[
D = \text{Decoder}(y)
\]

where \(\text{Decoder}(\cdot)\) means the whole calculation of the decoder layer. And \(D \in \mathbb{R}^{l_2 \times f_2}\) is the output sequence of the decoder layer, \(l_2\) and \(f_2 = f_1\) denote the length of the output sequence and the feature dimension of each column, respectively.

3) PREDICTION GENERATION IN T-MLM LAYER

The main purpose of the T-MLM layer is to generate prediction. We consider employing a parallel and powerful unidirectional Transformer to implement the target prediction generation.

Firstly, we obtain the input vector of the T-MLM layer. After extracting features in encoder and decoder layers,
we can get $E$ as the input image, and $D$ as the actual labels (i.e., the sequence of digital musical symbols) corresponding to the image. We first concatenate $E$ with $D$ at the first dimension, expressed as $C = [E : D] \in \mathbb{R}^{(l_1+l_2) \times f_1}$. To model the order information of $C$, we use sine and cosine function to obtain its position embedding:

$$PE_{\text{pos},2i} = \sin(pos/10000^{2i/f_1})$$
$$PE_{\text{pos},2i+1} = \cos(pos/10000^{2i/f_1})$$

(3)

where $pos \in [1, l_1 + l_2]$ is the index of position, $i$ is the index of dimension. Then, we sum $C$ and $PE$, and obtain $H^0$ as the input of Transformer.

Secondly, we calculate the output vector of each Transformer module in T-MLM layer. We stack four Transformer modules in T-MLM layer. In order to give the Transformer the ability to generate sequence of digital musical symbols, we mask some future musical symbols in $H^0$ for each token prediction. As illustrated in Figure 1, a mask matrix $M \in \mathbb{R}^{(l_1+l_2) \times (l_1+l_2)}$ is added to the multi-head self-attention, which can control what context a token can attend to when computing its contextualized representation. That is to say, during training process, all future tokens after current token to predict are masked by $M$ matrix.

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

(4)

where $M_{11} = \begin{pmatrix} 0_{l_1 \times l_1} & \end{pmatrix}, M_{12} = \begin{pmatrix} -\infty_{l_1 \times l_2} \\ \end{pmatrix}, M_{21} = \begin{pmatrix} 0_{l_2 \times l_1} \\ \end{pmatrix}, M_{22} = \begin{bmatrix} -\infty & \cdots & -\infty & -\infty \\ 0 & \cdots & -\infty & -\infty \\ \vdots & 0 & \cdots & \vdots \\ 0 & 0 & \cdots & -\infty \end{bmatrix}$ is a triangular matrix:

$$M_{22} = \begin{bmatrix} -\infty & \cdots & -\infty & -\infty \\ 0 & \cdots & -\infty & -\infty \\ \vdots & 0 & \cdots & \vdots \\ 0 & 0 & \cdots & -\infty \end{bmatrix},$$

(5)

where element 0 indicates that the token at current horizontal position can be attended, and element $-\infty$ means that the token at current horizontal position is masked.

In each Transformer module, a multi-head self-attention is applied to enhance the context representation. Take the computation in $l$-th Transformer module as an example:

$$Q^l = H^{l-1}W^l_Q, K^l = H^{l-1}W^l_K, V^l = H^{l-1}W^l_V$$

(6)

$$A^l = \text{Softmax} \left( \frac{Q^l K^l T}{\sqrt{d_k}} + M^l V^l \right)$$

(7)

where $H^{l-1} \in \mathbb{R}^{(l_1+l_2) \times f_1}$ means the output of $(l-1)$-th Transformer module, and it is linearly projected to a triple of queries $Q^l$, keys $K^l$, and values $V^l$ using parameter matrices $W^l_Q, W^l_K, W^l_V \in \mathbb{R}^{f_1 \times d_k}$, respectively. $\text{Softmax}(\cdot)$ is the softmax activation function, and $A^l \in \mathbb{R}^{(l_1+l_2) \times d_k}$ denotes the output of self-attention in a single head. We have multiple such head, i.e., the multi-head self-attention. The output of each head is concatenated together, and a linear projection is used:

$$Z^l = A^l W^l_O$$

(8)

where $A^l = \{A^l_1 : A^l_2 : \cdots : A^l_{d_h}\} \in \mathbb{R}^{(l_1+l_2) \times (d_h \times d_k)}$ is the concatenated output of each head, $h_d$ is the number of head, $W^l_O \in \mathbb{R}^{(d_h \times d_k) \times f_1}$ is a learnable parameter, $Z^l$ is the output of multi-head self-attention of $l$-th layer, and it can represent the context relationship. After that, a two-layer feed-forward network is used to nonlinearly transform $Z^l$:

$$H^l = W^l_{ffn} \text{ReLU} (W^l_{ffn_1} Z^l + b^l_{ffn_1}) + b^l_{ffn_2}$$

(9)

where $W^l_{ffn}, W^l_{ffn_1}, b^l_{ffn}, b^l_{ffn_1}$ are learnable parameters and biases of the feed-forward network. $\text{ReLU}(\cdot)$ is the Rectified Linear Unit (ReLU) activation function.

As we know, the final target prediction is based on the output of the last layer of Transformer. Hypothetically, the output of last layer of Transformer is $H^L \in \mathbb{R}^{(l_1+l_2) \times f_1}$, where the first $l_1$ lines contain the information of the input image, and the last $l_2$ lines are the information of actual tokens used to generate target predictions. While generating $i$-th predicted token, the model can only leverage the information of previous tokens before $i$-th position, but cannot use the information of future tokens after $i$-th position. Therefore, we apply the mask matrix $M \in \mathbb{R}^{(l_1+l_2) \times (l_1+l_2)}$ to mask some tokens in the attention computation, as shown in Equal (7). The mask matrices are detailed in Equals (4) and (5). Each row of $M$ indicates whether the current position can attend other positions.

In sequence generation, the most important part of $M$ is its last $l_2$ lines, because these lines are corresponding to the generated target predictions. At first line of the $l_2$ lines, the attention of image part is allowed (elements in first row of $M_{21}$ are all 0), while the actual token part is masked (elements in first row of $M_{22}$ are all $-\infty$). It means that the model should generate the first token from the image. At latter line of the $l_2$ lines, in the actual token part, only the front tokens can be attended (elements in latter row of $M_{22}$ are some 0 followed by many $-\infty$). This means that the next token should be generated from the image and the previous tokens. In other words, the mask matrix $M$ motivates the ST-MLM to learn the distribution of $P(y_l | y_1, y_2, \ldots, y_{l-1})$, i.e., the n-gram language mode. Therefore, the trained model is able to predict next token based on all known tokens, and with parallel operation, our model can generate actual target predictions corresponding to input image one by one.

4) FINAL OUTPUT SEQUENCE IN OUTPUT LAYER

When training the model, we calculate the loss in output layer according to the generated prediction and the actual labels. After the T-MLM layer, we can obtain the output vector $H^L$, which is transformed into a vector linearly with the dimension size of category number ($= \sum$). Next, the cross entropy loss function is applied to calculate the loss. And with multiple gradient descent, every learnable parameter is optimized until the model converges. At last, the final ST-MLM model is obtained.

During the testing stage, the initial input of the decoder layer is ‘SOS’ which stands for the start-of-sequence. The
generation of the next prediction label depends on the previous generated labels. Through iterative calculation until the ‘EOS’ which stands for the end-of-sequence is predicted, the final output of the musical symbols sequence is generated. After T-MLM layer, a linear projection followed by softmax activation function is used to obtain the probability distribution of each musical symbol. It shows the final prediction sequence \( \hat{y} \) of testing stage.

\[
\hat{y} = \text{argmax}(\text{Softmax}(\text{Linear}(H^L)))
\]

where \( \text{Linear}() \) is linear projection operation, \( \text{Softmax}() \) is softmax activation function, and \( \text{argmax}() \) calculates the index of maximum value in probability distribution.

### IV. EXPERIMENTS AND RESULTS

#### A. DATASET

In our experiment, we use the dataset named “Printed Images of Music Staves” (PrIMuS) [16], which is available at https://grfia.dlsi.ua.es/primus/. PrIMuS contains 87,678 real-music images and their two types of corresponding musical symbol representations. In each music image, it includes a series of musical symbols standing on the single staff-lines, as shown in Figure 4(a). The main musical symbols are notes, clefs and rests, etc., and some of them are difficult to recognize, such as time-signature, sharp and barline. And for the two types of representations, they are semantic representation (Figure 4(b)) and agnostic representation (Figure 4(c)). The former is a simple format containing the sequence of symbols in the score with their musical meaning, while the latter is composed of a list of graphic symbols in the score without predefined musical meaning, and each symbol locates in a position in the staff. Additionally, the third type of representation is splitting-agnostic representation (Figure 4(d)), which is a new way to describe each digital musical symbol according to the graphic and position of symbol in agnostic representation. As shown in Figure 4, the multirest-10 in (b) is the content meaning of red box. Whereas, the multirest-10 is expressed as digit.1-S5, digit.0-S5, multirest-L3 in (c), which are framed in blue boxes. And the contents in (d) is similar to (c) but combined as a tuple, which are framed in green boxes.

#### B. PARAMETER SETTINGS

To train and test the ST-MLM, we split the PrIMuS dataset into training set and testing set, with 90% for training and 10% for testing. For the model training, we set the maximum epoch to 128. And we use early stop strategy to stop the training when loss value has not reduced within 20 epochs. The batch size is set to 16. The learning rate is equal to 0.0001. And Adam is applied to optimize the learning process. Table 1 shows the parameter settings of ST-MLM model when training. The code developed to reproduce this experimentation is available at https://github.com/zljallen/crnn_transformer_mlm.

#### C. EVALUATION METRICS

When concerning evaluation metrics, two different metrics are calculated, similar to [16]:

- **Sequence Error Rate (SeER, %):** the ratio of incorrectly predicted sequences (at least one error).
- **Symbol Error Rate (SyER, %):** computed as the average number of elementary editing operations (insertions, modifications, or deletions) needed to produce the reference sequence from the sequence predicted by our model.

The semantic and agnostic representation are encoding different aspects of the same source PrIMuS. It is noted that the length of them is usually different. Therefore, although there are a few flaws in the comparison of the SyER and the SeER considering perfectly predicted sequence, the two different representations are equivalent.
D. RESULTS

The subsection presents the comparisons of our ST-MLM with CRNN/CTC [16] and S2S-Attention [25] on PrIMuS dataset. Due to that the CRNN/CTC and S2S-Attention perform experiments on raw images without blanks removed, we provide the results of ST-MLM that ignore the blank removing operation (i.e., the no-rm-blank ST-MLM in Table 2). Looking at Table 2, it shows that the no-rm-blank ST-MLM obtains better performance in SeER, with 8.1% in semantic representation and 14.7% in agnostic representation, which are 35.2%, 17.9% lower than CRNN/CTC, respectively. And no-rm-blank ST-MLM achieves more than one half improvement than the S2S-Attention in semantic representation, but almost keep the similar result in agnostic representation. This indicates that no-rm-blank ST-MLM can better extract the contextual semantic information of musical symbols due to CNN, LSTM and transformer. As for SyER, our model reaches 0.6% in semantic representation, and 1.0% in agnostic representation. In contrast to S2S-Attention, no-rm-blank ST-MLM decreases much, while compare with CRNN/CTC, the decrease of SyER is not as huge as SeER because the SyER is already very small, and a tiny decrease is a big improvement.

Furthermore, after removing blanks between musical symbols on raw images of the PrIMus dataset, ST-MLM can obtain a better performance. As shown in Table 3, the SeER and SyER are further decreased. For semantic representation, the SeER reduces from 8.1% to 6.7%, which is 46.4% lower than the CRNN/CTC model, and 17.3% lower than the no-rm-blank ST-MLM. While for agnostic representation, the SeER reaches to 10.6%, which is 40.8% lower than the CRNN/CTC, and 27.9% lower than the no-rm-blank ST-MLM. Of course, the ST-MLM shows more low number than the S2S-Attention method. Since the blanks are used only to separate musical symbols, they may interfere to the recognition of symbols. Therefore, the removal of blanks can effectively improve the recognition performance. Besides, the calculated quantity will be reduced because of the removal of blanks, which can shorten the training time.

There are some charts to have a comparison of our ST-MLM method with CRNN/CTC and S2S-Attention, in order to show the effectiveness of ST-MLM. As shown in Figure 5 to Figure 8, the CRNN/CTC and S2S-Attention can get fitting with less epoch, about 30 epochs in semantic representation. However, the training time of CRNN/CTC takes longer time per epoch, and S2S-Attention have a higher SeER and SyER. Compared with these two models, while our ST-MLM has a large number of training epochs, the training time of each epoch is short, so the overall training time is shorter (see in Table 5), and the SeER and SyER can reach a minimum of about 6.0% and 0.6%, respectively. Besides, it is obvious that the CRNN/CTC will have large fluctuations after fitting, which may be caused by the instability of the model itself. However, the SeER of our ST-MLM still has the trend of continuing to decrease, which should be due to the better robustness of the Transformer with MLM.

| Model            | Semantic Representation | Agnostic Representation |
|------------------|-------------------------|-------------------------|
|                  | SeER(%) | SyER(%) | SeER(%) | SyER(%)  |
| S2S-Attention    | 24.3    | 2.1     | 14.3    | 1.5      |
| CRNN/CTC         | 12.5    | 0.8     | 17.9    | 1.0      |
| no-rm-blank ST-MLM | 8.1    | 0.6     | 14.7    | 1.0      |

TABLE 2. Comparison of results between no-rm-blank ST-MLM and CRNN/CTC and S2S-Attention on two encoding representations are presented.

| Representations     | SeER(%) | SyER(%) |
|---------------------|---------|---------|
| Agnostic Representation | 10.6    | 0.9     |
| Splitting-agnostic Representation | 7.2      | 0.6     |

TABLE 3. Self-comparison results of ST-MLM with or without blank-removal operation.

| Representations     | SeER(%) | SyER(%) |
|---------------------|---------|---------|
| CRNN/CTC            | 10.6    | 0.9     |
| ST-MLM              | 7.2     | 0.6     |

TABLE 4. The results comparison of agnostic representation with splitting-agnostic representation in ST-MLM.
In addition, we explore the performance of ST-MLM on multiple outputs. We choose the agnostic representation to simulate the multiple outputs because the representation can be split into the graphic part and the position part for each musical symbol. We extract the two parts for each musical symbol and thus we can get two target sequence: graphic part sequence and position part sequence. The ST-MLM is trained to generate the two sequences in this experiment. The results are shown in Table 4, the Splitting-agnostic Representation row. The SeER and SyER are 7.2% and 0.6%, respectively, which is lower over 30% than the raw agnostic representation. We speculate the reason is the decrease of the number of classification categories. In the raw agnostic representation, the number of classification categories is 758, while in the splitting-agnostic representation, the number of graphic and position classification categories are only 73 and 24, respectively. As a result, it is more easily to predict accurately in splitting-agnostic representation.

E. DISCUSSION

1) DISCUSSION OF TRAINING TIME

As mentioned in the methodology section, ST-MLM implements parallel decoding by a mask matrix to improve computational efficiency. Thus, in the same training environment, we also conduct a comparison experiment about training time in semantic representation. As we can see in Table 5, the training and fitting time of S2S-Attention are relatively shorter than the CRNN/CTC and ST-MLM model. This is probably because of the simple structure of the model, learning music score features difficultly, leading to poor results. As for CRNN/CTC, its training time is about 110 minutes per epoch, and its convergence time is up to 55 hours, which is caused by the low efficiency of the GPU usage of the model itself, and the redundant computation of blanks. However, when using the ST-MLM model for training, the training time per epoch is only about 32 minutes, and the model convergence time is close to 35 hours, especially that the SeER and SyER is more lower. This indicates that our ST-MLM model takes less training time, and it is more friendly to do semantic encoding for music scores.

2) VISUALIZATION ANALYSIS OF RESULTS

To explore the working principle of ST-MLM, we randomly choose a music score image to conduct a visualization analysis of the self-attention mechanism of transformer. After music score images are pre-processed, they are fed into ST-MLM and then the self-attention matrix is obtained. The matrix is visualized using a heat map, and the visualization result is shown in Figure 9. In this heat map, the brighter the color is, the more attention our model pays to the location of image. As you can see, at each line, only one position is concerned, which means only one little part of the image is concerned when each symbol is predicted. This makes perfect sense because each symbol corresponds to only one position in the image. In addition, as the model predicts, the highlights move to the right, meaning the model predicts symbols one by one from the front to the back. That is, our model recognizes each musical symbol in order. Until the ‘EOS’ token is predicted, our model stops predicting.

| Time                  | S2S-Attention | CRNN/CTC | ST-MLM |
|----------------------|---------------|----------|--------|
| Training Time Per Epoch (M%) | 8.4           | 110.9    | 32.6   |
| Training Convergence Time (H%)  | 12.3          | 55.4     | 35.3   |

FIGURE 9. The visualization of self-attention weight coefficient (color map).
from the expression of the self-attention matrix, it shows that our model correctly finds the position of each symbol in the image and provides reasonable inference, which confirms the effectiveness and reliability of our model.

V. CONCLUSION AND EXPECTATION

In this work, we propose a ST-MLM model for the recognition of monophonic music score. On the one hand, a mask matrix in self-attention mechanism is applied to predict digital musical symbol sequence in parallel mode, improving training efficiency and making the model converge better when comparing to CRNN/CTC model. On the other hand, in ST-MLM, we use self-attention in Transformer to further learn the contextual relationship between musical symbols. By this way, we can solve the problem of the gradient disappearance of long music scores, so as to obtain lower sequence error rate. Moreover, the multiple outputs can be achieved with our ST-MLM due to the encoder-decoder framework has no limit on the feature dimensions. And the length between input image and output sequence has no limit in ST-MLM. Through experiments, when identifying the entire symbols sequence, our model has progressed much both in semantic representation, agnostic representation and splitting-agnostic representation. At the same time, the training time is also reduced largely. Chord music is more common than monophonic music score in the real world. In the future, we could further consider the recognition of a line or a page chord music whose musical symbols occur more than one at each column. Although there are some works about recognizing the chord music score or even the whole music document, their results are not ideal. Therefore, improving their recognition accuracy is still an interesting trial. Of course, it is more critical to speed up the progress of handwriting music score recognition which is beneficial to the creation of digital music library.

REFERENCES

[1] D. H. Pruslin, “Automatic recognition of sheet music,” Ph.D. dissertation, Dept. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 1966.

[2] I. Fujinaga, “Staff detection and removal,” in Visual Perception of Music Notation: On-Line and Off Line Recognition. Harrisburg, PA, USA: IGI Global, 2004, pp. 1–39.

[3] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes, and J. S. Cardoso, “Optical music recognition: State-of-the-art and open issues,” Int. J. Multimedia Inf. Retr., vol. 1, no. 3, pp. 173–190, Oct. 2012.

[4] D. Bainbridge and T. Bell, “The challenge of optical music recognition,” Comput. Humanities, vol. 35, no. 2, pp. 95–121, 2001.

[5] J. D. S. Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. P. D. Costa, “Staff detection with stable paths,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 6, pp. 1134–1139, Jun. 2009.

[6] C. Raphael and J. Wang, “New approaches to optical music recognition,” in Proc. ISMIR, 2011, pp. 305–310.

[7] M. Akbari, A. T. Targhi, and M. M. Dehshibi, “TeMu-App: Music characters recognition using HOG and SVM,” in Proc. 9th Iranian Conf. Mach. Vis. Image Process. (MVIP), Nov. 2015, pp. 128–131.

[8] A.-J. Gallego, J. Calvo-Zaragoza, J. J. Valero-Mas, and J. R. Rico-Juan, “Clustering-based k-nearest neighbor classification for large-scale data with neural codes representation,” Pattern Recognit., vol. 74, pp. 531–543, Feb. 2018.

[9] Y. Mitobe, H. Miyao, and M. Maruyama, “A fast HMM algorithm based on stroke lengths for on-line recognition of handwritten music scores,” in Proc. 9th Int. Workshop Frontiers Handwriting Recognit., 2004, pp. 521–526.

[10] T. Tuna, A. Beke, and T. Kumbasar, “Correction to: Deep learning frameworks to learn prediction and simulation focused control system models,” Applied Intelligence, vol. 52, no. 1, p. 680, 2022.

[11] J. Calvo-Zaragoza, A.-J. Gallego, and A. Pertusa, “Recognition of handwritten music symbols with convolutional neural codes,” in Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR), Nov. 2017, pp. 691–696.

[12] A. Baró, P. Riba, J. Calvo-Zaragoza, and A. Fornés, “From optical music recognition to handwritten music recognition: A baseline,” Pattern Recognit. Lett., vol. 123, pp. 1–8, May 2019.

[13] L. Chen, R. Jin, S. Zhang, S. Lee, Z. Chen, and D. Crandall, “A hybrid hmm-rnn model for optical music recognition,” in Proc. 17th Int. Soc. Music Inf. Retr. Conf. Extended Abstr. Late-Breaking Demo Session, New York, NY, USA, Aug. 2016, pp. 1–3.

[14] A. Ríos-Vila, J. M. Inesta, and J. Calvo-Zaragoza, “On the use of transformers for end-to-end optical music recognition,” in Proc. Iberian Conf. Pattern Recognit. Image Anal. Cham, Switzerland: Springer, 2022, pp. 470–481.

[15] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in Proc. 23rd Int. Conf. Mach. Learn. (ICML), 2006, pp. 369–376.

[16] J. Calvo-Zaragoza and D. Rizo, “End-to-end neural optical music recognition of monophonic scores,” Appl. Sci., vol. 8, no. 4, p. 606, Apr. 2018.

[17] M. Alfaoro-Conterras, A. Ríos-Vila, J. J. Valero-Mas, J. M. Inesta, and J. Calvo-Zaragoza, “Decoupling music notation to improve end-to-end optical music recognition,” Pattern Recognit. Lett., vol. 158, pp. 157–163, Jun. 2022.

[18] S. Ulich, F. Giron, and Y. Mitsufuji, “Deep neural network based instrument extraction from music,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2015, pp. 2135–2139.

[19] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, “Hybrid hidden Markov models and artificial neural networks for handwritten music recognition in mensural notation,” Pattern Anal. Appl., vol. 22, no. 4, pp. 1573–1584, Nov. 2019.

[20] A. Baro, P. Riba, J. Calvo-Zaragoza, and A. Fornés, “Optical music recognition by recurrent neural networks,” in Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR), Nov. 2017, pp. 25–26.

[21] A. Liu, L. Zhang, Y. Mei, B. Han, Z. Cai, Z. Zhu, and J. Xiao, “Residual recurrent CRNN for End-To-End optical music recognition on monophonic scores,” in Proc. Workshop Multi-Modal Pre-Training Multimedia Understanding, Aug. 2021, pp. 23–27.

[22] A. Hartelt and F. Puppe, “Optical medieval music recognition using background knowledge,” Algorithms, vol. 15, no. 7, p. 221, Jun. 2022.

[23] Z. Huang, X. Jia, and Y. Guo, “State-of-the-art model for music object recognition with deep learning,” Appl. Sci., vol. 9, no. 13, p. 2645, Jun. 2019.

[24] M. Alfaoro-Conterras and J. J. Valero-Mas, “Exploiting the two-dimensional nature of agnostic music notation for neural optical music recognition,” Appl. Sci., vol. 11, no. 8, p. 3621, Apr. 2021.

[25] A. Ríos-Vila, J. Calvo-Zaragoza, and J. M. Inesta, “Exploring the two-dimensional nature of music notation for score recognition with end-to-end approaches,” in Proc. 17th Int. Conf. Frontiers Handwriting Recognit. (ICFHR), Sep. 2020, pp. 193–198.

[26] Y. Ding, Y. Liu, H. Luan, and M. Sun, “Visualizing and understanding neural machine translation,” in Proc. 55th Annua. Meeting Assoc. Comput. Linguistics, vol. 1, 2017, pp. 1150–1159.

[27] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, “Handwritten music recognition for mensural notation with convolutional recurrent neural networks,” Pattern Recognit. Lett., vol. 128, pp. 115–121, Dec. 2019.

[28] M. E. Thomae, A. Ríos-Vila, J. Calvo-Zaragoza, D. Rizo, and J. M. Inesta, “Retrieving music semantics from optical music recognition by machine translation,” in Proc. Music Encoding Conf. Medford, MA, USA: Tisch Library, Tufts Univ., May 2020, pp. 1–6.

[29] A. Baro, C. Badal, and A. Fornés, “Handwritten historical music recognition by Sequence-to-Sequence with attention mechanism,” in Proc. 17th Int. Conf. Frontiers Handwriting Recognit. (ICFHR), Sep. 2020, pp. 205–210.
[30] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 27, 2014, pp. 3104–3112.

[31] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3156–3164.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 5998–6008.

[33] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, “Unified language model pre-training for natural language understanding and generation,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, Curran, 2019, pp. 1–13.

[34] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, “Speech recognition using deep neural networks: A systematic review,” IEEE Access, vol. 7, pp. 19143–19165, 2019.

[35] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, “Scheduled sampling for sequence prediction with recurrent neural networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 28. Curran, 2015, pp. 1–9.

CUIHONG WEN received the B.S. degree from the College of Physical and Electronic Sciences, Hunan Normal University, in 2009, and the Ph.D. degree in control science and engineering from Hunan University, in 2016. She has been a Lecturer and a Master Tutor with Hunan Normal University, since 2017. Her research interests include computer vision and quantum machine learning.

LONGJIAO ZHU received the B.S. degree from Jishou University, in 2019. She is currently pursuing the M.S. degree with Hunan Normal University. Her research interest includes digital image processing.

CUIHONG WEN

LONGJIAO ZHU

VOLUME 10, 2022

IEEE Access