Optical Music Recognition System Based on CNN and RNN

Lai Xiangwei , Huang Yanjie ,
He Ruichen , Li Sijie

[1]Xiamen University, China

Abstract

Optical Music Recognition is a field of research that investigates how to computationally decode music notation from images. A solution to this problem was published back in 2018. Our work is more or less a repetition of that. In this work, we are going to study the use of neural networks that work in an end-to-end manner. It is achieved by using a neural network model that combines the capabilities of convolutional neural networks, which work on the input image, and recurrent neural networks, which deal with the sequential nature of the problem. Thanks to the use of Connectionist Temporal Classification loss function, models can be directly trained with input images accompanied by their corresponding transcripts to music symbol sequences. We use the Printed Images of Music Staves (PrIMuS) dataset, presented by the same authors of that 2018 paper, to train and evaluate this neural network approach. We will be testing the capability of this neural network in our experiments.

Introduction

The availability of huge collections of digital scores has facilitated both the music professional practice and the amateur access to printed sources that were difficult to obtain in the past. In addition to instant availability, the advantages of having the digitized image of a work over its printed material are restricted to the ease to copy and distribute, and the lack of wear that digital media intrinsically offers over any physical resource. The great possibilities that current music-based applications can offer are restricted to scores symbolically encoded. Notation software such as Finale, Sibelius, MuseScore, or Dorico, computer-assisted composition applications such as OpenMusic, digital musicology systems such as Music21, or Humdrum, or content-based search tools (Casey et al. 2008), cannot deal with pixels contained in digitized images but with computationally-encoded symbols such as notes, bar-lines or key signatures.

Different initiatives have been proposed to manually fill this gap between digitized music images and digitally encoded music content. However, the manual transcription of music scores does not represent a scalable process, given that its cost is prohibitive both in time and resources. Therefore, it is necessary to resort to assisted or automatic transcription systems. The Optical Music Recognition (OMR) is defined as the research about teaching computers how to read musical notation , with the ultimate goal of exporting their content to a desired format.

Despite the great advantages of its development, OMR is far from being reliable as a black box, as current optical character recognition (Liwicki et al. 2007) (Calvo-Zaragoza et al. 2018) or speech recognition technologies (Graves, Mohamed, and Hinton 2013) do. In the scientific community, there are hardly any complete approach for its solution. Traditionally, this has been motivated because of the small subtasks in which the workflow can be divided. Simpler tasks such as staff-line removal, symbol localization and classification, or music notation assembly, have so far represented major obstacles. Nonetheless, recent advances in machine learning, and specifically in Deep Learning (DL) (Graves, Mohamed, and Hinton 2013), not only allow solving these tasks with some ease, but also to propose new schemes with which to face the whole process in a more elegant and compact way, avoiding heuristics that make systems limited to the kind of input they are designed for. In fact, this new sort of approaches has broken most of the glass-ceiling problems in text and speech recognition systems.

Considering this as a starting point, this work was restricted to the consideration of monodic short scores taken from real music works in Common Western Modern Notation (CWMN). Then, one can use the Connectionist Temporal Classification (CTC) loss function (Voigtlaender, Doetsch, and Ney 2016),which means that it is not necessary to provide information about the composition or location of the symbols in the image, but only pairs of input scores and their corresponding transcripts into music symbol sequences. According to previous experimental results, this approach proves to successfully solve the end-to-end task.

Related Work

Music score is the main form of non-voice communication of music. With the development of technology, the way to preserve and disseminate music scores is changing. Computer music scoring software in the 1970s and 1980s, although can enter music score, but difficult to use. The subsequent addition of keyboard and mouse typing, input efficiency remains low. Therefore, a large number of music works (such as films, dramas, concerts, etc.) exist in paper

form. In order to realize the automatic conversion from paper music to symbolic music, Optical Music Recognition (OMR) technology came into being (Ng, McLean, and Marsden 2014). OMR is a technique that converts paper music images into symbolic forms (MIDI or XML) that can be directly recognized and used by computers. However, it is very time-consuming to convert paper music score into computer-readable semantic symbol form, and retrieve, analyze and other operations. The development of target detection and recognition algorithms (Calvo-Zaragoza, Gallego, and Pertusa 2017) in image processing promotes the development of OMR system and related algorithms.

Music image preprocessing.

Image preprocessing is a basic step in many computer vision tasks. The main purpose of this stage is to make the adjusted image easier to operate. The most common image processing includes enhancement, de-tilt, blur, denoising and binarization. To tilt is to adjust the image tilt to get a better perspective. Most digital images are affected by noise in the process of acquisition (Keil and Ward 2019), transmission or processing, and their color and brightness have random noise signals. Generally, the global threshold is determined by the whole image, and for the adaptive threshold, the local information in the image should be considered (Glorot, Bordes, and Bengio 2011). Ng et al modified the global threshold proposed by Ridler and Calvard. Some recent OMR studies also used adaptive thresholds.n the process of binarization, the image is analyzed to determine what is noise and what is useful for the task. The technologies for selecting binary threshold include global method and adaptive method.

Music symbol recognition

After basic preprocessing, music image enters the process of music symbol recognition. This process mainly includes spectral line processing and music symbol processing.

Music symbol recognition includes three main steps : spectral line processing, music symbol separation and classification. Usually, the spectral line is first detected and removed from the image (Byrd and Simonsen 2015). Then, the symbols of the model are separated into basic elements, and these basic elements are then used to extract features, and these features are fed back to the classifier.

Construction of Music Model

The construction of music score model is to embed all the retrieved information into the appropriate output file, and construct the semantic model or data model by processing the output of the previous steps. This model should represent the recoding of the scores in the input. The output model should be expressed in machine-readable format. The usual OMR output formats include MIDI, MusicXML, MEI, NIFF, Finale, WAVE (Rebelo et al. 2012), etc. To decide which code to use, we must consider what the application might need. Using the knowledge obtained in the previous steps and different studies will help to standardize this stage. At present, there are few studies on OMR processing coding, but many works in other fields focus on the coding format that better represents music and its structure.

Proposed Solution

We describe in this section the neural models for the OMR task in an end-to-end manner. In this case, a monodic staff section will be processed at each moment.

Let X = $\{(x_1, y_1), (x_2, y_2), ...\}$ be our end-to-end application domain, where x_i represents a single staff image and y_i is its corresponding sequence of music symbols. An image x is considered to be a sequence of variable length, given by the number of columns. Given an input image x, the problem can be solved by retrieving its most likely sequence of music symbols $\hat{y}$:

$$\hat{y} = \underset{\hat{y} \in \sum^*}{\arg\max} P(y|x) \tag{1}$$

In this work, the Recurrent Neural Network will be producing the sequence of musical symbols that fulfills Equation (1). However, we first add a Convolutional Neural Network for learning how to process the input image . This is implemented by concatenating all output channels into an image. Then, columns of resulting images are treated as individual frames for the recurrent block. The restriction above is that, for each staff, the training set only provides its corresponding sequence of symbols, without any explicit information about the location of the semantic or agnostic symbols. This scenario can be solved by the CTC loss function . A graphical scheme of the framework is given in Figure 1.

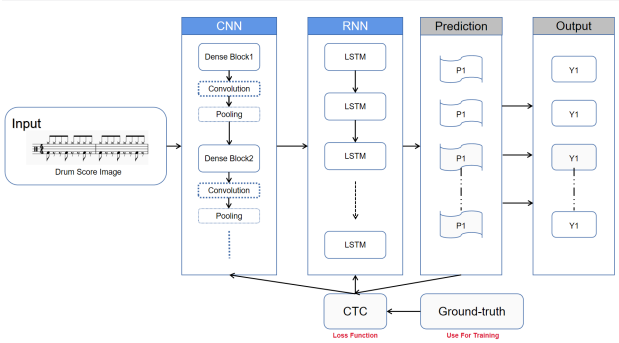


Figure 1: Graphical scheme of the end-to-end neural approach considered.

Implementation Details

The details of the neural model are given in Table 3. Input variable-width grayscale images are rescaled at the height of 128 pixels, without modifying aspect ratio. Two recurrent layers try to convert the filtered image into a discrete sequence of musical symbols. Each frame performs a classification, with a fully-connected layer of alphabet plus one (the blank symbol) numbers of neurons.

A mini-batch has 16 samples. Adadelta algorithm for adaptively updating the learning rate. Predictions must be post-processed to emit the actual sequence. Thanks to the

Table 1: Notation: Input(h × w × c) is an image of size h × w and c channels; Conv(n, h × w) denotes a convolution operator of n filters and kernel size of h × w; MaxPooling(h × w) is a down-sampling operation of size h × w; BLSTM(n) is a bi-directional Long Short Term Memory unit of n neurons; Dense(n) denotes a dense layer of n neurons; and Softmax() is the softmax activation function. Σ denotes the alphabet of musical symbols.

Input (128 × W × 1)
Convolutional Block
Conv(32, 3 × 3), MaxPooling(2 × 2)
Conv(64, 3 × 3), MaxPooling(2 × 2)
Conv(128, 3 × 3), MaxPooling(2 × 2)
Conv(256, 3 × 3), MaxPooling(2 × 2)
Recurrent block
BLSTM(256)
BLSTM(256)
Dense($\|\Sigma\|$ + 1)
Softmax()

CTC loss function, the decoding can be performed greedily: when the symbol predicted by the network in a frame is the same as the previous one, it is assumed that they are the same frame and only one symbol is concatenated to the sequence. There are two indications for a new symbol: either the predicted symbol is different from the previous one or the predicted symbol is the blank symbol.

Note that the limitation is that the output cannot contain more musical symbols than the number of frames of the input image, which is unlikely to happen.

Experimental Overview And Results

Experimental Setup

Concerning evaluation metrics, there is an open debate on which metrics should be used in OMR . This is especially arguable because of the different points of view that the use of its output has: it is not the same if the intention of the OMR is to reproduce the content or to archive it in order to build a digital library. Here we are only interested in the computational aspect itself, in which OMR is understood as a pattern recognition task. So, we shall consider metrics that, even assuming that they might not be optimal for the purpose of OMR, allow us to draw reasonable conclusions from the experimental results. Therefore, let us consider the following evaluation metrics:

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

Note that the length of the agnostic and semantic sequences are usually different because they are encoding different aspects of the same source. Therefore, the comparison in terms of Symbol Error Rate, in spite of being normalized (%), may not be totally fair. Furthermore, the Sequence Error Rate allows a more reliable comparison because it only takes into account the perfectly predicted sequences (in which case, the outputs in different representations are equivalent).

Below we present the results achieved with respect to these metrics. In the first series of experiments we measure the performance that neural models can achieve as regards the representation used. First, they will be evaluated in an ideal scenario, in which a huge amount of data is available. Therefore, the idea is to measure the glass ceiling that each representation may reach. Next, the other issue to be analyzed is the complexity of the learning process as regards the convergence of the training process and the amount of data that is necessary to learn the task. Finally, we analyze the ability of the neural models to locate the musical symbols within the input staff, task for which it is not initially designed. For the sake of reproducible research, source code and trained models are freely available.

Performance

We show in this section the results obtained when the networks are trained with all available data. This means that about 80,000 training samples are available, 10% of which are used for deciding when to stop training and prevent overfitting. The evaluation after a 10-fold cross validation scheme is reported in Figure 2.

	Representation	
	Agnostic	**Semantic**
Sequence Error Rate (%)	17.9	12.5
Symbol Error Rate (%)	1.0	0.8

Figure 2: Evaluation metrics with respect to the representation considered. Results reported represent averages from a 10-cross validation methodology.

Interestingly, the semantic representation leads to a higher performance than the agnostic representation. This is clearly observed in the sequence-level error (12.5% versus 17.9%), and somewhat to a lesser extent in the symbol-level error (0.8% versus 1.0%). It is difficult to demonstrate why this might happen because of the way these neural models operate. However, it is intuitive to think that the difference lies in the ability to model the underlying musical language. At the image level, both representations are equivalent (and, in principle,

the agnostic representation should have some advantage). On the contrary, the recurrent neural networks may find it easier to model the linguistic information of the musical notation from its semantic representation, which leads—when there is enough data, as in this experiment—to produce sequences that better fit the underlying language model.

In any case, regardless of the selected representation it is observed that the differences between the actual sequences

and those predicted by the networks are minimal. While it cannot be guaranteed that the sequences are recognized with no error (only 12.5% at best), the results can be interpreted as that only around 1% of the symbols predicted need correction to get the correct transcriptions of the images. Therefore, the goodness of this complete approach is demonstrated, in which the task is formulated in an elegant way in terms of input and desired output.

Concerning computational cost we would like to emphasize that although the training of these models is expensive—in the order of several hours over high-performance Graphical Processing Units (GPUs)— the prediction stage allows fast processing. It takes around 1 second per score in a general-purpose computer like an Intel Core i5-2400 CPU at 3.10 GHz with 4 GB of RAM, and without speeding-up the computation with GPUs. We believe that this time is appropriate for allowing a friendly usability in an interactive application.

Error Analysis

In order to dig deeper into the previously presented results, we conducted an analysis of the typology of the errors produced. The most repeated errors for each representation are reported in Figure 3.

Rank	Representation			
	Agnostic		Semantic	
	Symbol	Percentage	Symbol	Percentage
# 1	*barline-L1*	45.5%	*barline*	38.6%
# 2	*gracenote.sixteenth-L4*	1.8%	*tie*	9.4%
# 3	*accidental.natural-S3*	1.4%	*gracenote.C5-sixteenth*	1.5%

Figure 3: List of the 3 most common errors with respect to the representation considered. Percentages are relative to the total error rates from Figure 2.

In both cases, the most common error is the barline, with a notable difference with respect to the others. Although this may seem surprising at first, it has a simple explanation: the incipits often end without completing the bar. This, at the graphic level, hardly has visible consequences because the renderer almost always places the last barline at the end of the staff (most of the incipits contain complete measures). Thus, the responsibility of discerning whether there should be a barline or not lies almost exclusively in the capacity of the network to take into account "linguistic" information. The musical notation is a language that, in spite of being highly complex to model in its entirety, has certain regularities with which to exploit the performance of the system, as for instance the elements that lead to a complete measure. According to the results presented in the previous section, we can conclude that a semantic representation, in comparison with the agnostic one, makes it easier for the network to estimate such regularities. This phenomenon is quite intuitive, and may be the main cause of the differences between the representations' performance.

As an additional remark, note that both representations miss on grace notes, which clearly represent a greater complexity in the graphic aspect, and are worse estimated by the language model because of being less regular than conventional notes.

In the case of the semantic representation, another common mistake is the tie. Although we cannot demonstrate the reason behind these errors, it is interesting to note that the musical content generated without that symbol is still musically correct. Therefore, given the low number of tie symbols in the training set (less than 1%), the model may tend to push the recognition towards the most likely situation, in which the tie does not appear.

Learning Complexity

The vast amount of available data in the previous experiment prevents a more in-depth comparison of the representations considered. In most real cases, the amount of available data (or the complexity of it) is not so ideal. That is why in this section we want to analyze more thoroughly both representations in terms of the learning process of the neural model.

First, we want to see the convergence of the models learned in the previous section. That is, how many training epochs the models need for tuning their parameters appropriately. The curves obtained by each type of model are shown in Figure 4.

From these curves we can observe two interesting phenomena. On the one hand, both models converge relatively quick, as after 20 epochs the elbow point has already been produced. In fact, the convergence is so fast that the agnostic representation begins to overfit around the 40th epoch. On the other hand, analyzing the values in further detail, it can be seen that the convergence of the model that trains with the agnostic representation is more pronounced. This could indicate a greater facility to learn the task.

To confirm this phenomenon, the results obtained in an experiment in which the training set is incrementally increased are shown below. In particular, the performance of the models will be evaluated according to the size of training set sizes of 100, 1000, 10,000, and 20,000 samples. In addition,in order to favor the comparison, the results obtained in Section 5.2 will be drawn in the plots (around 80,000 training samples).

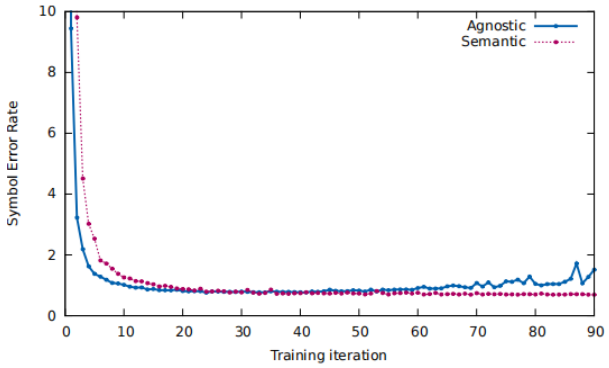


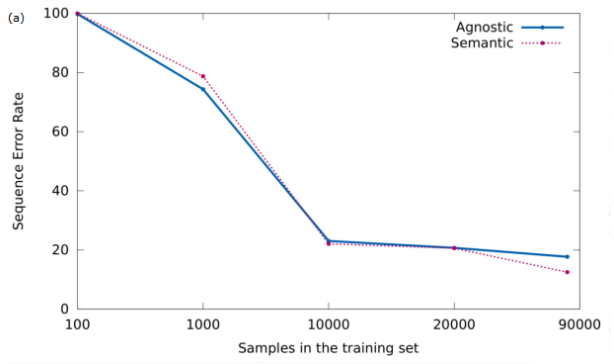
Figure 4: Symbol Error Rate over validation partition with respect to the training epoch.

The evolution of both Sequence and Symbol Error Rate

are given in Figure 6a,b, respectively, for the agnostic and semantic representations.

These curves certify that learning with the agnostic model is simpler, because when the number of training samples is small, this representation achieves better results.

We have already shown that, in the long run, the semantic representation slightly outperforms its performance. However, these results may give a clue as to which representation to use when the scenario is not so ideal like the one presented here. For example, when either there is not so much training data available or the input documents depict a greater difficulty (document degradation, handwritten notation, etc.).



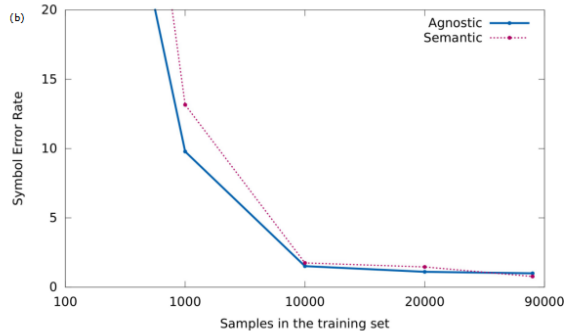


Figure 5: Comparison between agnostic and semantic representations in the evolution of the evaluation metrics with respect to the size of the training set. Note that the x-axis does not present a linear scale.(a)Sequence Error Rate; (b) Symbol Error Rate.

The examples given above show some of the most representative errors found. During the search of these examples, however, it was difficult to find samples where both system failed. In turn, it was easy to find examples where Photoscore failed and our system did not. Obviously, we do not mean that our system behaves better than Photoscore, but rather that our approach is competitive with respect to it.

Conclusion

In this work, we have studied the suitability of the use of the neural network approach to solve the OMR task in an end-to-end fashion through a controlled scenario of printed single-staff monodic scores from a real world dataset.

The main contribution of the present work consisted of analyzing the possible codifications that can be considered for representing the expected output. In this paper we have proposed and studied two options: an agnostic representation, in which only the graphical point of view is taken into account, and a semantic representation, which codifies the symbols according to their musical meaning.

Our experiments have reported several interesting conclusions:

- The task can be successfully solved using the considered neural end-to-end approach.
- The semantic representation that includes musical meaning symbols has a superior glass ceiling of performance, visibly improving the results obtained using the agnostic representation.
- The learning process with the agnostic representation made up of just graphic symbols is simpler, since the neural model converges faster and the learning curve is more pronounced than those with the semantic representation.
- Regardless of the representation, the neural model is not able to locate the symbols in the image—which could be expected because of the way the CTC loss function operates.

As future work, this work opens many possibilities for further research. it is undoubted that the most promising avenue is to extend the neural approach so that it is capable of dealing with a comprehensive set of notation symbols, including articulation and dynamic marks, as well as with multiple-voice polyphonic staves.

References

Amodei, D.; Ananthanarayanan, S.; Anubhai, R.; Bai, J.; Battenberg, E.; Case, C.; Casper, J.; Catanzaro, B.; Cheng, Q.; Chen, G.; et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, 173–182. PMLR.

Byrd, D.; and Simonsen, J. G. 2015. Towards a standard testbed for optical music recognition: Definitions, metrics, and page images. *Journal of New Music Research*, 44(3): 169–195.

Calvo-Zaragoza, J.; Castellanos, F. J.; Vigliensoni, G.; and Fujinaga, I. 2018. Deep neural networks for document processing of music score images. *Applied Sciences*, 8(5): 654.

Calvo-Zaragoza, J.; Gallego, A.-J.; and Pertusa, A. 2017. Recognition of handwritten music symbols with convolutional neural codes. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, 691–696. IEEE.

Casey, M. A.; Veltkamp, R.; Goto, M.; Leman, M.; Rhodes, C.; and Slaney, M. 2008. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4): 668–696.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323. JMLR Workshop and Conference Proceedings.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, 369–376.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, 6645–6649. Ieee.

Keil, K.; and Ward, J. A. 2019. Applications of RISM data in digital libraries and digital musicology. *International Journal on Digital Libraries*, 20(1): 3–12.

Liwicki, M.; Graves, A.; Fernàndez, S.; Bunke, H.; and Schmidhuber, J. 2007. A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*.

Ng, K.; McLean, A.; and Marsden, A. 2014. Big data optical music recognition with multi images and multi recognisers. BCS.

Rebelo, A.; Fujinaga, I.; Paszkiewicz, F.; Marcal, A. R.; Guedes, C.; and Cardoso, J. S. 2012. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3): 173–190.

Voigtlaender, P.; Doetsch, P.; and Ney, H. 2016. Handwriting recognition with large multidimensional long short-term memory recurrent neural networks. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 228–233. IEEE.