Pattern Recognition Letters 123 (2019) 1–8

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

From Optical Music Recognition to Handwritten Music Recognition: A baseline☆,☆☆

Arnau Baró[a,*], Pau Riba[a], Jorge Calvo-Zaragoza[b], Alicia Fornés[a]

[a] Computer Vision Center, Computer Science Department, Universitat Autònoma de Barcelona, Edifici O, Bellaterra 08193, Spain
[b] Department of Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain

ARTICLE INFO

Article history:
Received 27 July 2018
Available online 26 February 2019

MSC:
68U10
68T10
68T05

Keywords:
Optical music recognition
Handwritten music recognition
Document image analysis and recognition
Deep neural networks
LSTM

ABSTRACT

Optical Music Recognition (OMR) is the branch of document image analysis that aims to convert images of musical scores into a computer-readable format. Despite decades of research, the recognition of handwritten music scores, concretely the Western notation, is still an open problem, and the few existing works only focus on a specific stage of OMR. In this work, we propose a full Handwritten Music Recognition (HMR) system based on Convolutional Recurrent Neural Networks, data augmentation and transfer learning, that can serve as a baseline for the research community.

1. Introduction

For centuries, music has been written and transmitted among generations through sheet music. Not surprisingly, the digitization and transcription of music scores existing in archives and museums is of paramount importance to preserve and disseminate our musical heritage. Given that there are still thousands of music scores waiting to be transcribed, a manual transcription becomes unfeasible, and therefore, the research on methods for automatically transcribing music becomes necessary.

Optical Music Recognition (OMR) can be defined as the conversion of music score images into a machine-readable format (e.g. MusicXML, MEI, MIDI, etc.). It has been an active research field for more than five decades [1,2], and there are many commercial OMR software such as PhotoScore[1] or SharpEye[2] with good performance under relatively good conditions. However, their accuracy dramatically decreases when dealing with handwritten scores, mainly because of the high variability in the handwriting style. Unfortunately, most of the still unknown music compositions existing in archives are indeed handwritten music scores. For this reason, more research effort must be devoted to overcoming this limitation.

Although the interest in OMR has reawakened with the appearance of deep learning, as far as we know, the few existing methods that attempt to recognize handwritten scores are mostly focused on solving a particular stage of OMR, such as layout analysis [3] or detection and classification of graphic primitives [4] or music symbols [5,6]. However, in the particular case of Western classical music, music scores are complex documents composed of staves (five horizontal lines), music symbols (e.g. notes, rests, accidentals), slurs, ornaments, dynamic and tempo markings, lyrics, etc. Therefore, we believe that it is time to focus on the full recognition.

With this aim, in this paper we propose a full staff-wise Handwritten Music Recognition (HMR) system, which can serve as a baseline for future improvements in this research field. Our architecture is based on Convolutional and Recurrent Neural Networks. This work is based on our previous work [7], where we addressed OMR for printed scores as a sequential recognition task, disentangling the output of the network in the two main components of music notation: rhythm and pitch. In the present work, we improve this architecture to deal with handwritten scores, and we show its viability both in printed and handwritten scenarios.

☆ This paper is handled by Associate Editor Prof. S. Sarkar.

☆☆ We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

* Corresponding author.
E-mail address: abaro@cvc.uab.cat (A. Baró).

[1] http://www.neuratron.com/photoscore.htm.

[2] http://www.visiv.co.uk/.

https://doi.org/10.1016/j.patrec.2019.02.029
0167-8655/© 2019 Elsevier B.V. All rights reserved.

Concretely, the improvements are the following: First, we add Convolutional Neural Networks as feature extractor. Secondly, since the existing amount of annotated handwritten music scores is scarce, we propose a novel data augmentation technique, and incorporate transfer learning from printed scores. Finally, we also share the handwritten data[3] that has been manually labeled for the experimental evaluation.

The rest of the paper is organized as follows. Section 2 overviews the state of the art. Section 3 describes our architecture. Section 4 explains how we deal with few handwritten data. Section 5 discusses the results, and conclusions are drawn in Section 6.

2. Related work

This section describes the key references of Optical Music Recognition that are relevant to the present work.

2.1. Traditional approaches

Traditional OMR methodologies can be divided into four groups: segmentation, grammars/rules, sequences and graphs. The first group segments symbols before their recognition. For example, Fornés et al. propose symbol descriptors [8,9], whereas Rebelo et al. [10] use Neural Networks, Nearest Neighbour, Support Vector machines or Hidden Markov models. The second group defines some grammars or rules to combine graphical primitives (*i.e.* note-heads, steams, beam, etc.) to build music notes and symbols. Baró et al. [11] recognize compound music notes using dendrograms to join graphical primitives with a set of predefined rules. Coüasnon and Rétif [12] use grammars to detect symbols and minimize possible errors. Thanks to the particular properties of monophonic scores, sequential-based approaches attempt the recognition directly as a sequence using Hidden Markov Models [13,14]. Finally, graph-based approaches [15] use a graph to define the relationship of primitives or to codify the symbols' shape.

2.2. Deep learning-based approaches

Since Deep Learning [16] arose, several OMR approaches have been proposed. For example, Van der Wel and Ullrich [17] use Convolutional Neural Networks (CNNs) and sequence-to-sequence (seq2seq) models for recognizing monophonic printed music scores. Calvo-Zaragoza et al. [18,19] also use a CNN to extract features from printed music scores and feed a Recurrent Neural Network. To avoid the alignment between the music score and the ground-truth data, they use the Connectionist Temporal Classification (CTC) loss function commonly used in speech and text recognition. Nevertheless, as the authors point out, these methods are only able to recognize monophonic music scores (no chords). In addition, they cannot recognize dense music scores containing many accidentals, dynamics, or expression marks. Contrary, we are able to deal with multiple symbols in the same time step. This is necessary to recognize chords or typical music artifacts such as dynamics. Finally, Wen et al. [20] use connected components to segment symbols, which are later recognized using CNNs. This method is tested on both printed and handwritten scores.

2.3. Approaches for handwritten scores

It is true that there are some complete OMR methods for ancient (mensural) notation [21–23], but in this work we focus on Western music notation. Some researchers have started by classifying isolated music symbols [24] and some of them have even shared their own datasets [9,10,25].

Since the recently creation of the MUSCIMA++ [26] dataset, which consists of 140 handwritten scores labeled at primitive level, the research on OMR has been boosted. For example, Hajič and Pecina [4] propose a method to detect noteheads in music scores. The network first detects which regions are important, and then, it decides if a pixel belongs to a notehead and predicts the bounding box. Finally a filter combines outputs to refuse the mismatches. This approach gives good results but decreases its performance when chords appear.

Other authors detect all primitives, not only noteheads. For example, Tuggener et al. [6] use ResNets to predict dense energy maps that will be used to predict the location, class and bounding box of each symbol. They can detect the symbols without preprocessing the page (e.g. cropping each staff). A similar approach is [5], where Pacha et al. propose an end-to-end trainable object detector for music primitives. The proposed method uses a machine-learning approach considering region-based deep convolutional neural networks. Moreover, authors use transfer learning from general object detection, and obtain very good results.

2.4. Summary

We observe that there are not complete OMR systems for handwritten scores on Western notation yet. There only exist successful approaches for sub-stages of the process. Nevertheless, these methods are based on the detection of music symbols, instead of the full OMR pipeline.

Moreover, the reported results might not be really convincing because the MUSCIMA++ dataset is a subset of the CVC-MUSCIMA dataset [27], which was created for writer identification. Since the above mentioned works randomly split the pages into train, validation and test partitions, using writer-independent partitions only, the same music work could appear in the training and test sets at the same time, with the only difference of being written by different persons. Consequently, the system could be biased towards the recognition of these specific sequences of melodies and rhythms.

For all the above reasons, we believe that a baseline for OMR in handwritten scores is required.

3. Proposed architecture

Many music scores, including polyphonic ones, are written using a single staff. Therefore, we propose to read each staff as a sequence, similar to text recognition [28], by using Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN). Although they can extract information directly from image pixels, we incorporate Convolutional Neural Networks (CNN) as image feature extractor. Fig. 1 shows an schema of our architecture. The different stages are described next.

Input: In this work, we assume that the music scores pages have been previously segmented into staves. The segmented staves correspond to binary images resized to a height of 100 pixels in order to feed pixel columns of the same size to the network. The aspect ratio will be kept, therefore the width will change for each batch. The images of the same batch are padded according to the longest staff in the batch.

Convolutional block: The convolutional block is composed by three convolutional layers increasing the depth and kernel size of 3x3, followed by Batch Normalization [29] and Rectified Linear Unit activation [30]. Finally a max-pool 2×1 operator is used to reduce the vertical dimension while keeping the same image width. In other words, the output of the Convolutional Block will have the same width as the input image.

[3] www.cvc.uab.es/people/abaro/datasets.html.

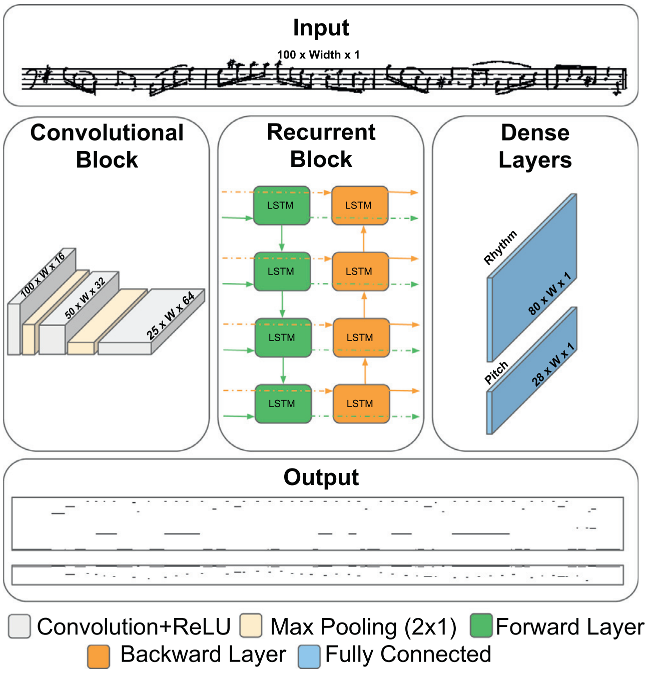

Fig. 1. Architecture of our method. Each staff is the input of the convolutional block to extract features, and then, it passes the recurrent block. Finally, two fully connected layers separate the rhythm and melody.

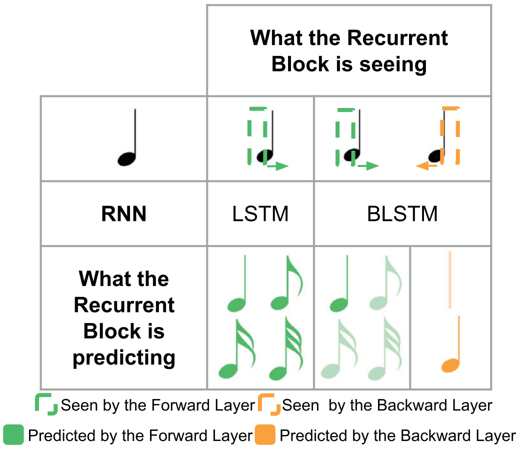
What the Recurrent Block is seeing
RNN
LSTM
BLSTM
What the Recurrent Block is predicting
Seen by the Forward Layer
Seen by the Backward Layer
Predicted by the Forward Layer
Predicted by the Backward Layer

Fig. 2. BLSTM predictions. The backward direction helps to reduce the ambiguities when predicting a symbol.

Recurrent block: This block uses Bi-directional LSTM networks (BLSTM) [31] to benefit from context when recognizing each symbol. Compared to RNNs, LSTMs are able to learn long-term dependencies, avoiding the vanishing gradient problem, and keeping information for longer time. Moreover, the bi-directionallity provides an extra information that reduces ambiguities, because it takes into account the forward and backward directions. For example, if one direction is reading a vertical line and the other direction is seeing a notehead, the network can correctly predict a quarter note. Fig. 2 shows an example of the ambiguity reduction provided by both directions. In our architecture, we use four BLSTM layers of 512 neurons each.

Dense layers: After the recurrent block, we incorporate two fully connected (FC) layers. In this way, we will obtain two outputs: one for the rhythm and one for the pitch. If we had one single output, we should consider any combination of pitch-rhythm as a different class, which would become in a very large number of classes. Another reason to separate pitch-rhythm and consider them independent, is that we can obtain many more examples of each class to train. For instance, the system learns the shape of a 16th note, no matter its pitch. Please note that here we define the pitch as the location of the note in the staff (e.g. the note is located on the third staff line), instead of the real pitch (e.g. C4 note), because it depends on the clef. Also, in this way, we can represent all pitches with few classes.

Output: Finally, the output of each dense layer is a matrix, whose columns are symbol and pitch probabilities per pixel column in the original image. Each matrix has the same width as the original image and has a height of 80 classes for the rhythm and 28 classes for the pitch. By thresholding these matrices, we can decide which symbols appear in the music scores. In our previous work [7] we performed an exhaustive analysis where we evaluated several thresholds. The one which provided the best performance was 0.5. In other words, the network has to be at least 50% confident when recognizing each symbol. Note that more than one symbol may appear at the same time step (column). Two symbols have been manually added to ease the recognition:

- Epsilon (ε) is used to know where a symbol starts and ends. If ε is activated, none of the other symbols can be activated. This symbols works as a separator.
- *No note* is a symbol only found in the pitch matrix. When this symbol is activated it means that the symbol activated in the rhythm matrix (at the same instance of time) has not pitch (e.g. symbols without pitch, such as rests).

Finally, these outputs are converted into an array, combining the rhythm and pitch. These arrays will be used to evaluate the method at rhythm and pitch level and also to evaluate the complete system, where both parts should be predicted correctly.

As it has been stated, in OMR several symbols can appear at the same time stamp (e.g. chords, time signature, etc.). Hence, several labels can be predicted at the same output step. For this reason, we choose the Smooth L_1-loss function. Concretely, our architecture has been trained using the *Stochastic Gradient Descent* (SGD) optimizer with Momentum and weight decay *i.e.* L_2 regularization. The Smooth L_1-loss has been used as objective function defined as

$$\mathcal{L}(x,y) = \frac{1}{n}\sum \begin{cases} 0.5(x_i - y_i)^2, & \text{if } |x_i - y_i| < 1 \\ |x_i - y_i| - 0.5, & \text{otherwise,} \end{cases} \tag{1}$$

where x is the output of the proposed architecture and y is the target we want to achieve. The proposed loss function can deal with multi-label problems being less sensitive than L_2-loss with respect to outliers.

4. Data augmentation and transfer learning

This section describes the training strategies that have been used to exploit our architecture. As stated before, there is very few labeled handwritten data. Since little groundtruth data for training leads to overfitting problems, we propose two different strategies. First, we propose to apply transfer learning by fine-tuning a printed model with handwritten data. Second, we propose a data augmentation technique for music scores.

Transfer learning: Training our system with printed scores give insights of the suitability of the proposed approach for OMR. However, a model for printed scores may fail when recognizing handwritten scores due to the elastic deformations in handwriting styles. To overcome this issue, we propose to pre-train our model with printed scores, and then, fine-tune it with the few available handwritten data.

Data augmentation: To increase the amount and variability of training data, some distortions have been applied to both the

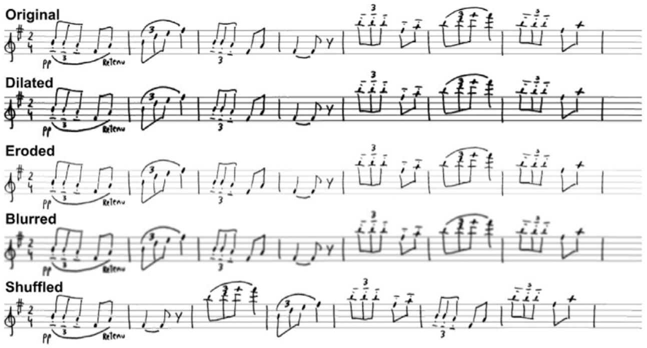

Fig. 3. Different techniques of data augmentation. Dilating, eroding and blurring have been applied to both datasets, printed and handwritten ones. Shuffling is only applied to the handwritten dataset.

printed and handwritten training sets. First, we have applied dilation, erosion and blurring distortions. Note that this data augmentation has been randomly applied for each music score. Beside the morphological operations, the number of handwritten music scores in the training set has been increased by shuffling the bar units. For this purpose, we crop each measure (bar unit) and shuffle among the different measures of the staff, with the exception of the first and the last bar unit. These two measures are fixed because the first one contains the clef and the time signature whereas the last one can contain a final barline. Fig. 3 shows the different data augmentation techniques applied to each dataset. Note that this shuffling also prevents the model to learn a specific melody and rhythm.

5. Experimentation

This section experimentally validates the performance of our architecture. As it has already mentioned, we propose to firstly train a model able to recognize printed musical scores and later transfer this learning to handwritten data. Hence, two datasets have been used.

5.1. Datasets

Printed dataset: we use a subset of PrIMuS dataset [19], which consists of rendered incipts from the RISM[4]. It is annotated at primitive level *i.e.* the symbols are labeled as noteheads, steams and flags, among others instances instead of quarter notes, 8th notes, 16th notes and such on. This dataset is latter converted into symbol level. Our set contains almost 50,000 music scores rendered with 3 different typographies.

Handwritten dataset: The MUSCIMA++ dataset [26] is a selection of 140 pages from the CVC-MUSCIMA dataset [27], annotated at primitive level. Although these primitives are related each other using a graph, they cannot be directly used for OMR evaluation. For this reason, having into account the graph relations and keeping the noteheads as the main node of notes, we have manually labeled 20 music pages at symbol level (including slurs, dynamic marks, etc.) in order to evaluate a full OMR system. In any case, we should take into account that the original CVC-MUSCIMA dataset was created for staff removal and writer identification (for this reason, it contains the same 20 different musical compositions, rewritten by 50 different writers). This fact leads us to some limitations when splitting the sets *i.e.* into train, validation and test. Our method must never see the same musical composition at test and train or it may be biased towards the recognition of a specific melody. For this reason, we have selected these 20 pages (musical compositions), from different writers (see Table 1).

5.2. Evaluation

We use the Symbol Error Rate (SER) [17–19] metric. Similarly to Word Error Rate (WER) [28], commonly used in text recognition community, SER is computed as the Levenshtein distance: the sum of edit operations that are needed to convert the output of our method into the groundtruth in terms of symbol insertions (I), substitutions (S) and deletions (D). Formally,

$$SER = \frac{S + D + I}{N}, \qquad (2)$$

where N is the number of symbols in the ground truth. The lower this value, the better.

To perform the evaluation at different levels, we propose to evaluate Rhythm and Pitch separately. Therefore, we will provide the SER for both outputs of the proposed architecture. Finally, both outputs are merged and the SER for pairs Rhythm and Pitch (considered as one symbol) is provided.

5.3. Results on printed documents

We first evaluate our model in the printed scenario. Thus, we can test the suitability of our architecture in a controlled scenario. An ablation study has been performed to test several architecture details. Table 2 presents this study in order to evaluate the importance of the BLSTM recurrent block, CNN features and Data augmentation. Moreover, we compare the current work with our previous work [7].

As expected, the best configuration uses a CNN to extract image features containing richer information than merely using pixel columns. Moreover, the BLSTM provides more context information and improves the previous approaches. Finally, data augmentation slightly improves the performance whereas making it more robust to the initialization. The first row shows our previous work, while the last row shows the best configuration of the current work. The main difference is that here we propose to incorporate a convolutional block before the recurrent layers, and we have increased the number of neurons from 128 to 512 and layers from 3 to 4. In this way, we obtained a better performance (the SER decreases from 0.028 to 0.003 when we consider the rhythm and pitch together).

5.4. Results on handwritten documents

As stated before, we aim to create a full staff-wise HMR system for handwritten music scores that can serve as starting point for future improvements in this field. Table 3 shows the results of our method using the selected pages of the MUSCIMA++ dataset. Note that each line introduces an improvement to the previous one. In the first row, we do not use any of the proposed improvements (no pre-training, CNNs, etc.). Observe that pre-training with printed data decreases the error (second row). Data augmentation on printed data helps a little bit. However, in the fourth row, we can see that the BLSTM is the key modification to reduce the error rates by 0.2 points. This is because of its ability to use context to minimize ambiguities. Then, the feature extraction based on CNN also helps to recognize the handwritten music scores (fifth row). By shuffling the measures (the sixth row) we obtain the best approach. Finally, in the last row, we observe that morphological operations for data augmentation only introduce noise and increases the error rates. The main reason for this behaviour could be that morphological techniques may make printed scores look closer to handwritten, but when these techniques are used in handwritten scores, the result may look unrealistic.

[4] http://rism.info .

Table 1
Selected Muscima++ pages for train, validation and test sets. We indicate the number of staves per page, and if the page is polyphonic.

	Train												Validation				Test			
Page	2	4	5	6	7	8	9	13	15	16	19	20	11	12	14	17	1	3	10	18
Writer	20	12	21	16	31	35	32	23	43	34	1	18	49	18	29	44	17	13	15	10
Polyphonic					✓		✓	✓			✓	✓			✓	✓			✓	✓
# Staves	6	5	7	6	4	6	4	5	5	8	3	8	6	8	6	6	5	7	6	8

Table 2
Results on printed documents. All results are between [0–1]. The first number is the mean of the five executions and the number between parenthesis is the standard deviation. The first row corresponds to our previous work, the others are results of the current architecture.

RNN	CNN	Data Augm.	Rhythm SER	Pitch SER	Rhy. + Pit. SER
BLSTM [7]	–	✓	0.020 (±0.001)	0.015 (±0.001)	0.028 (±0.002)
LSTM	–	–	0.168 (±0.014)	0.144 (±0.011)	0.174 (±0.012)
LSTM	–	✓	0.163 (±0.009)	0.139 (±0.013)	0.169 (±0.008)
BLSTM	–	–	0.005 (±0.002)	0.003 (±0.001)	0.006 (±0.002)
BLSTM	–	✓	0.005 (±0.002)	0.002 (±0.000)	0.005 (±0.001)
BLSTM	✓	–	0.003 (±0.001)	0.002 (±0.001)	0.003 (±0.001)
BLSTM	✓	✓	0.002 (±0.001)	0.001 (±0.000)	0.003 (±0.001)

Table 3
Results on handwritten documents. All results are between [0–1]. The first number is the mean of the five executions and the number between parenthesis is the standard deviation.

Pre-train Printed	D. Augm. Printed	BLSTM	CNN	D. Augm. Handwritten		Rhythm SER	Pitch SER	Rhythm + Pitch SER
				Shuffle	Morph.			
–	–	–	–	–	–	0.826 (±0.009)	0.709 (±0.012)	0.899 (±0.007)
✓	–	–	–	–	–	0.771 (±0.021)	0.668 (±0.021)	0.872 (±0.016)
✓	✓	–	–	–	–	0.762 (±0.019)	0.690 (±0.004)	0.854 (±0.019)
✓	✓	✓	–	–	–	0.523 (±0.018)	0.464 (±0.020)	0.610 (±0.016)
✓	✓	✓	✓	–	–	0.493 (±0.015)	0.396 (±0.012)	0.559 (±0.015)
✓	✓	✓	✓	✓	–	**0.476** (±0.009)	**0.387** (±0.008)	**0.545** (±0.009)
✓	✓	✓	✓	✓	✓	0.490 (±0.005)	0.393 (±0.004)	0.554 (±0.007)

Using the best configuration in Table 3, we provide the results from each one test page in Table 4 (two of them are polyphonic). Note that each row corresponds to a different writer and different page.

5.5. Comparison with commercial OMR software

Since we could not find any complete OMR for handwritten scores in the literature, we could not make a quantitative comparison. However, we could find a commercial software for qualitative evaluation. Photoscore is a commercial software able to recognize handwritten and printed music scores. It must to be said that we do not know whether Photoscore uses any post-processing or grammar rules (detecting the time signatures might be counting the number of beats in each measure and validating the recognition) in the recognition, so the comparison could not be completely fair.

Figs. 4–8 show some qualitative results comparing the Photoscore results with our method. We have used different colors to highlight the common mistakes of our method. The blue color is used when different symbols appear in the same column, and our method is not capable to relate each symbol with the correspondent pitch. Orange boxes show that some symbols, as accents, could confuse our system. For example, sometimes the method predicts that an accent is a notehead, thus it detects the notehead located higher up (see Fig. 8), whereas other times it can predict

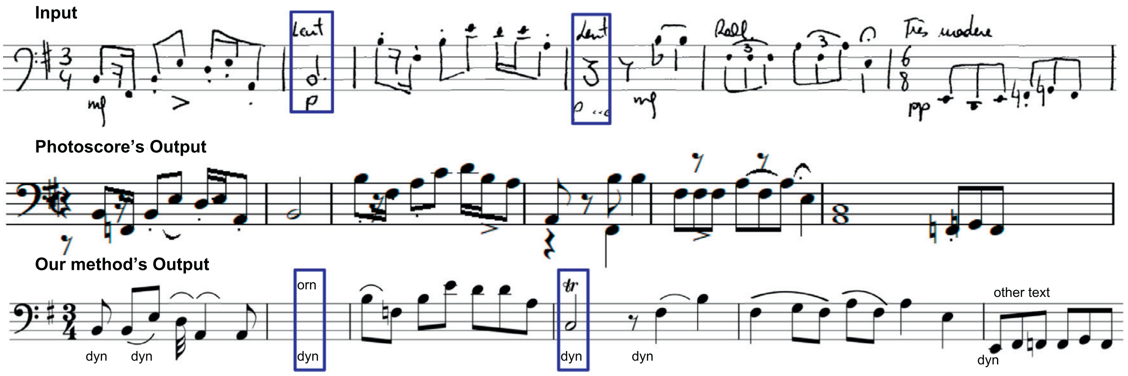

Fig. 4. Qualitative comparison with Photoscore. Example of one staff of page 1. The blue box shows that our method is not able to recognize the symbols when several of them appear in the same column. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Results on the handwritten documents, shown per page. All results are between [0–1]. The first number is the mean of the five executions and the number between parenthesis is the standard deviation.

	Polyph.	Rhythm SER	Pitch SER	Rhy. + Pit. SER
W. 17 - P. 1	–	0.528 (±0.019)	0.349 (±0.019)	0.594 (±0.014)
W. 13 - P. 3	–	0.226 (±0.018)	0.175 (±0.008)	0.270 (±0.016)
W. 15 - P. 10	✓	0.716 (±0.017)	0.620 (±0.010)	0.796 (±0.018)
W. 10 - P. 18	✓	0.483 (±0.018)	0.422 (±0.008)	0.565 (±0.013)

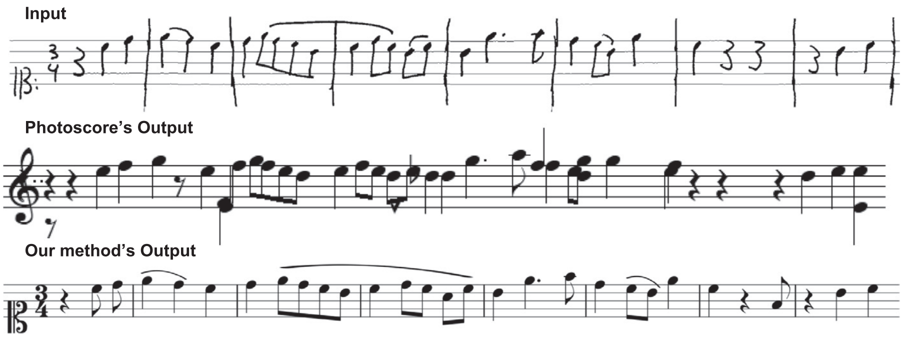

Fig. 5. Qualitative comparison with Photoscore. Example of one staff of page 3. Contrary to Photoscore, note that our method could detect all the slurs.

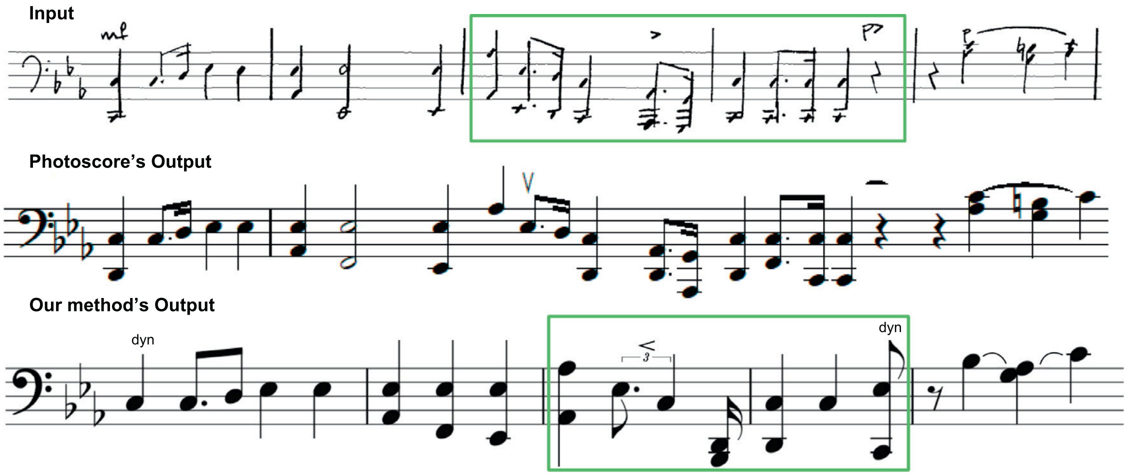

Fig. 6. Qualitative comparison with Photoscore. Example of one staff of page 10. The green box shows that our method is not able to recognize all the noteheads in polyphonic music scores. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

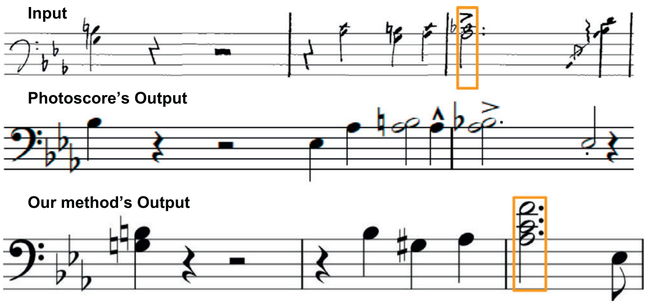

Fig. 7. Qualitative comparison with Photoscore. Example of one staff of page 10. The orange box shows that our method could confuse some symbols by others by the position *i.e.* accents by noteheads. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

that accent is another notehead and detects a chord (see Fig. 7). In red we show when the system confuses some symbols because of shape, for example a text dynamics is confused by a quarter note. Finally, Fig. 6 shows in green the difficulties to detect all noteheads in a chord. In these images, please note that when we draw the output of our network, the compound music symbols have been manually joined for better visualization.

5.6. Discussion

From these results, we could conclude that our methodology is valid and has shown to be able to recognize simple staves. From the qualitative point of view, bearing in mind that the Photoscore software might be using music rules for validation, our method obtains pretty good results. In fact, in many cases, our method outperforms Photoscore.

Concerning the quantitative results, although we are aware that the overall SER is close to 50%, these results are promising. First, we have used very few handwritten data, and secondly, we have not applied any grammar or rule to validate each bar unit.

Nevertheless, there are several limitations, most of them related to the way of labeling the data, which are described next.

Polyphonic music scores: The ground-truth is not able to relate which pitch corresponds to each notehead in the case that the rhythm within a chord (or polyphonic voices) is different (see

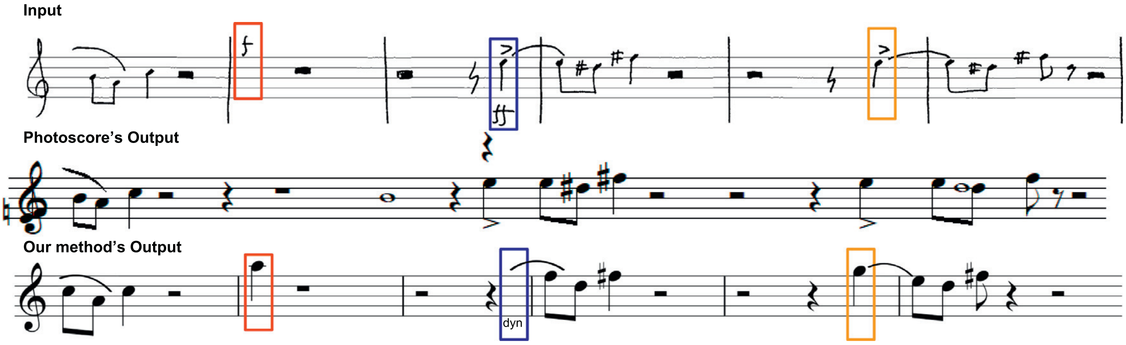

Fig. 8. Qualitative comparison with Photoscore. Example of one staff of page 18. The red box shows that our method could confuse some symbols by others by the shape. The blue box shows that our method is not able to recognize the symbols when there are many in the same column. The orange box shows that our method could confuse some symbols by others by the position. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

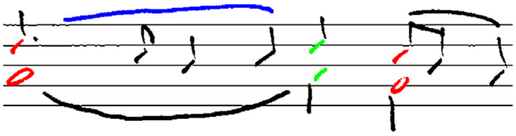

Fig. 9. Method's limitations. In red polyphonic notes that could not be correctly recognized because they have different duration at the same time step. In blue the slur that will not be detected because there is another slur at the same time. In green, symbols that will be correctly detected because they have the same duration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 9 red symbols). However, it is able to recognize polyphony correctly if the rhythm is the same for all the symbols (see Fig. 9 green symbols).

Repeated symbols: If a symbol without pitch appears more than one time at the same time step, the method will only detect one (see Fig. 9 the blue slur will not be recognized).

Compound music symbols: The compound music symbols such as 8th notes, 16th notes, 32th notes and so on, joined by a beam, will be separately recognized because there is no symbol for notating this *i.e.* each notehead will have its steam and its flag, will not be joined by a beam.

Clef position on the stave: The ground truth does not provide the position of the clef on the stave. This means that a bass clef on the third or fourth staff lines are predicted as the same.

6. Conclusions and future work

In this work, we have proposed a complete Handwritten Music Recognition (HMR) system based on CNNs and RNNs, data augmentation and transfer learning from printed scores. The experimental results have demonstrated the viability of this approach, showing that staves can be recognized as a sequence using BLSTMs, and also, that the convolutional block acts as an effective feature extractor. We have first demonstrated that our architecture is valid through the evaluation over printed scores. Secondly, we have showed that our methodology greatly benefits from data augmentation from handwritten scores as well as transfer learning from printed scores.

Taking into account that we have used only 20 pages of the MUSCIMA++ database in the experiments, the results are promising. Of course, the incorporation of more handwritten data labeled at symbol level would help to obtain better results.

We hope that these results, together with our labeled data, can serve as a baseline for the community, fostering the research towards full OMR systems. Future work will be focused on the incorporation of music notation rules to solve ambiguities and improve the performance. Also, we will investigate segmentation-free methods in order to deal with polyphonic music scores that are written in several staves.

Acknowledgments

This work has been partially supported by the Spanish project TIN2015-70924-C2-2-R, the CERCA Program/Generalitat de Catalunya, a FPU fellowship FPU15/06264, the Ramon y Cajal Fellowship RYC-2014-16831 and a FI fellowship AGAUR 2018 FI_B 00546 (with the support of the Secretaria d'Universitats i Recerca of the Generalitat de Catalunya and the Fons Social Europeu). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2019.02.029.

References

[1] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A.R.S. Marçal, C. Guedes, J.S. Cardoso, Optical music recognition: state-of-the-art and open issues, IJMIR 1 (3) (2012) 173–190.
[2] A. Fornés, G. Sánchez, Analysis and recognition of music scores, in: Handbook of Document Image Processing and Recognition, Springer-Verlag London, 2014, pp. 749–774.
[3] J. Calvo-Zaragoza, F.J. Castellanos, G. Vigliensoni, I. Fujinaga, Deep neural networks for document processing of music score images, Appl. Sci. 8 (5) (2018) 654–674.
[4] J. Hajič jr., P. Pecina, Detecting noteheads in handwritten scores with convnets and bounding box regression, CoRR (2017). abs/1708.01806
[5] A. Pacha, K.-Y. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi, H.M. Eidenberger, Handwritten music object detection: open issues and baseline results, in: DAS, 2018, pp. 163–168.
[6] L. Tuggener, I. Elezi, J. Schmidhuber, T. Stadelmann, Deep watershed detector for music object recognition, ISMIR, 2018, pp. 271–278.
[7] A. Baró, P. Riba, J. Calvo-Zaragoza, A. Fornés, Optical music recognition by long short-term memory networks, Graphic Recognition, Current Trends and Challenges. LNCS, 11009, 2018, pp. 81–95.
[8] A. Fornés, J. Lladós, G. Sánchez, D. Karatzas, Rotation invariant hand drawn symbol recognition based on a dynamic time warping model, IJDAR 13 (3) (2010) 229–241.
[9] S. Escalera, A. Fornés, O. Pujol, P. Radeva, G. Sánchez, J. Lladós, Blurred shape model for binary and grey-level symbol recognition, Pattern Recognit. Lett. 30 (15) (2009) 1424–1433.
[10] A. Rebelo, G. Capela, J.S. Cardoso, Optical recognition of music symbols: a comparative study, IJDAR 13 (1) (2010) 19–31.
[11] A. Baró, P. Riba, A. Fornés, Towards the recognition of compound music notes in handwritten music scores, in: ICFHR, 2016, pp. 465–470.
[12] B. Coüasnon, B. Rétif, Using a Grammar for a Reliable Full Score Recognition System, 1995.
[13] L. Pugin, Optical music recognition of early typographic prints using hidden Markov models, in: ISMIR, 2006, pp. 53–56.
[14] L. Pugin, J.A. Burgoyne, I. Fujinaga, Map adaptation to improve optical music recognition of early music documents using hidden Markov models, in: ISMIR, 2007, pp. 513–516.

[15] J.C. Pinto, P. Vieira, J.M. Sousa, A new graph-like classification method applied to ancient handwritten musical symbols, Doc. Anal. Recognit. 6 (1) (2003) 10–22.

[16] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. http://www.deeplearningbook.org

[17] E. van der Wel, K. Ullrich, Optical music recognition with convolutional sequence-to-sequence models, in: ISMIR, 2017, pp. 731–737.

[18] J. Calvo-Zaragoza, J.J. Valero-Mas, A. Pertusa, End-to-end optical music recognition using neural networks, in: ISMIR, 2017, pp. 472–477.

[19] J. Calvo-Zaragoza, D. Rizo, End-to-end neural optical music recognition of monophonic scores, Appl. Sci. 8 (2018) 1–23.

[20] C. Wen, A. Rebelo, J. Zhang, J. Cardoso, A new optical music recognition system based on combined neural network, Pattern Recognit. Lett. 58 (2015) 1–7.

[21] C. Ramirez, J. Ohya, Automatic recognition of square notation symbols in western plainchant manuscripts, J. New Music Res. 43 (4) (2014) 390–399.

[22] J. Calvo-Zaragoza, A.H. Toselli, E. Vidal, Handwritten music recognition for mensural notation: formulation, data and baseline results, in: ICDAR, 2017, pp. 1081–1086.

[23] A. Pacha, J. Calvo-Zaragoza, Optical music recognition in mensural notation with region-based convolutional neural networks, ISMIR, 2018, pp. 240–247.

[24] A. Pacha, H.M. Eidenberger, Towards self-learning optical music recognition, in: ICMLA, 2017, pp. 795–800.

[25] J. Calvo-Zaragoza, J. Oncina, Recognition of pen-based music notation: the homus dataset, in: 2014 22nd International Conference on Pattern Recognition, 2014, pp. 3038–3043.

[26] J. Hajič jr., P. Pecina, The MUSCIMA++ dataset for handwritten optical music recognition, in: ICDAR, 2017, pp. 39–46.

[27] A. Fornés, A. Dutta, A. Gordo, J. Lladós, CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal, IJDAR 15 (3) (2012) 243–251.

[28] V. Frinken, H. Bunke, Continuous handwritten script recognition, in: Handbook of Document Image Processing and Recognition, Springer, 2014, pp. 391–425.

[29] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, Co RR (2015). arXiv: 1502.03167

[30] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, J. Mach. Learn. Res. 15 (2010) 315–323.

[31] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.