International Journal on Document Analysis and Recognition (IJDAR) (2023) 26:347–362
https://doi.org/10.1007/s10032-023-00432-z

SPECIAL ISSUE PAPER

End-to-end optical music recognition for pianoform sheet music

Antonio Ríos-Vila[1] · **David Rizo**[1,2] · **José M. Iñesta**[1] · **Jorge Calvo-Zaragoza**[1]

Received: 14 November 2022 / Revised: 2 April 2023 / Accepted: 8 April 2023 / Published online: 12 May 2023

Abstract
End-to-end solutions have brought about significant advances in the field of Optical Music Recognition. These approaches directly provide the symbolic representation of a given image of a musical score. Despite this, several documents, such as pianoform musical scores, cannot yet benefit from these solutions since their structural complexity does not allow their effective transcription. This paper presents a neural method whose objective is to transcribe these musical scores in an end-to-end fashion. We also introduce the GRANDSTAFF dataset, which contains 53,882 single-system piano scores in common western modern notation. The sources are encoded in both a standard digital music representation and its adaptation for current transcription technologies. The method proposed in this paper is trained and evaluated using this dataset. The results show that the approach presented is, for the first time, able to effectively transcribe pianoform notation in an end-to-end manner.

Keywords Optical music recognition · Polyphonic music scores · GrandStaff · Neural networks

1 Introduction

Transcribing the content of musical documents to structured formats brings benefits to digital humanities and musicology, as it enables the application of algorithms that rely on symbolic music data and makes musical score libraries more browsable. Given the price of manual transcription, it is unaffordable to transcribe large historical archives manually. In this scenario, the reading of music notation invites automation, much in the same way as modern technology in the fields of Optical Character Recognition (OCR) or Handwritten Text Recognition (HTR) has enabled the automatic processing of written texts. The field of Optical Music Recognition (OMR) covers the automation of this computational *reading* in the context of music [1].

Holistic approaches, also referred to as *end-to-end* approaches, have begun to dominate the fields of sequential labeling, with notable examples such as HTR or Automatic Speech Recognition. In OMR, these approaches have proved successful in those contexts in which music notation retrieval can be easily expressed as a sequence. This applies to monophonic scores, or legacy music-notation languages in which different voices were written individually. However, the scores of many compositions are written using *grand staves*, i.e., a combination of two staves put together, such as those used for the piano (see Fig. 1). In the related literature, this kind of scores is also referred to as *pianoform* [1–3]. However, no end-to-end system that has attempted to recognize the content of this type of scores is known to date.

This work proposes the first end-to-end recognition approach for pianoform scores. This constitutes a first step in the application of holistic models to the full spectrum of OMR applications. We consider a neural approach inspired by state-of-the-art full-paragraph HTR research, with which the OMR problem shares some of its challenges. This approach provides a serialization of the scores based on textual encodings of music notation. Likewise, since it is the first attempt to address this problem, this work also introduces the GRANDSTAFF dataset, a large corpus of isolated grand staves

✉ Antonio Ríos-Vila
arios@dlsi.ua.es
URL: https://praig.ua.es/author/rios-vila-antonio/

David Rizo
drizo@dlsi.ua.es
URL: https://praig.ua.es/author/drizo/

José M. Iñesta
inesta@dlsi.ua.es
URL: https://praig.ua.es/author/inesta/

Jorge Calvo-Zaragoza
jcalvo@dlsi.ua.es
URL: https://praig.ua.es/author/jcalvo/

[1] U.I for Computing Research, University of Alicante, Alicante, Spain

[2] Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana (ISEA. CV), Alicante, Spain

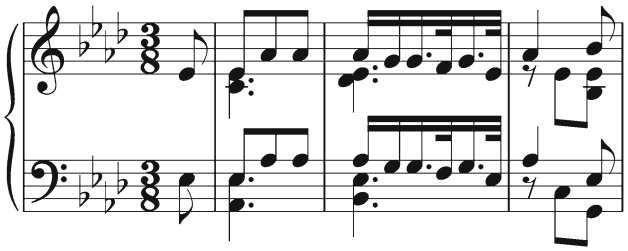


Fig. 1 Example of a *grand staff* for the piano, which consists of the combination of two staves that are played simultaneously when interpreting the music score

rendered from real symbolic data. In order to introduce more variability, the images are provided both in perfect condition and augmented by computer vision techniques so as to resemble the possible distortions of a real optical capturing process.

In our experiments, we consider (1) various neural schemes that differ as regards the way in which they process the sequential character of the input, (2) several means of encoding the output sequence, and (3) different scenarios according to the graphic quality of the samples. All of the above enables our work to establish the first baseline for end-to-end pianoform OMR, along with a solid benchmark for future research.

The remainder of this work is structured as follows: Sect. 2 provides a brief review of how OMR has been addressed in the recent past. The GRANDSTAFF dataset is then presented in 3, in which we define the representation of corpus music notation and detail the process applied in order to generate its samples. The proposed end-to-end OMR approach is then presented in Sect. 4. The experimental setup—in which all the implementations and evaluation metrics are defined—is described in Sect. 5, while the results attained are analyzed in Sect. 6. Finally, the conclusions of this work, along with future research avenues, are discussed in Sect. 7.

2 Background

Given its complexity, the OMR process has traditionally been divided into several stages that are tackled independently [4]. Fundamentally, there is a first set in which the basic symbols such as note heads, beams, or accidentals (usually referred to as "primitives") are detected. This involves processing the input image in order to isolate and categorize these components, which is not straightforward owing to the presence of artifacts such as staff lines and composite symbols [5]. In the second set of stages, the syntactic relationships among different primitives are inferred so as to recover the structure of the music notation. These stages have traditionally been solved by employing a combination of image processing techniques with heuristic strategies based on hand-crafted rules [6]. More recently, these same stages have been approached independently through the use of Deep Learning. This has greatly improved the performance of each of the individual tasks [7, 8], but has not, in turn, contributed equally to the advancement of the field of research itself. Multi-stage solutions have, in general, proved to be insufficient [1, 2].

Deep Learning has also diversified the way in which OMR is approached as a whole: there are now alternative pipelines with their own ongoing research that attempt to confront the whole process in a single step. This holistic paradigm, also referred to as *end-to-end* formulation, has begun to dominate the current state of the art in other applications, such as the recognition of text, speech, or mathematical formulae [9–11]. However, the complexity of inferring music structures from the image currently makes it difficult to formulate OMR as an end-to-end learnable optimization problem. While end-to-end systems for OMR do exist, they are generally limited to monophonic music notation [12–14].

Some approaches have recently managed to extend end-to-end formulations in order to deal with scores of higher complexity, such as homophony [15, 16] and single-staff polyphony [17]. However, having a universal OMR end-to-end transcription system that can deal with all kinds of notations, including pianoform scores, is still a challenge to be met.

3 The GrandStaff dataset

Several efforts have been made to create datasets for OMR. On the one hand, there are corpora, such as DeepScores [18] and the MUSCIMA dataset [19], that contain a wide variety of annotated music documents, including subsets of pianoform scores. Despite providing interesting samples, they have not been conceived to train end-to-end OMR solutions and do not contain ground truths in a standard digital music notation format. On the other hand, there are corpora—such as PrIMuS [13], Il Lauro Secco [20], Capitan [21] or FMT [22]—that are specially labeled for end-to-end OMR transcription. However, practically all of them lack polyphonic and pianoform samples, as they mainly contain monophonic or homophonic music excerpts, which makes them unsuitable for the objective of this study.

Given this gap, we have designed a dataset focused on the task of end-to-end pianoform transcription: the GRANDSTAFF corpus.[1]

The term "grand staff" is used in music notation to represent piano scores [23]. It consists of two staves that are joined by a brace at the beginning, and whose bar lines cross both staves (see Fig. 1).

[1] The dataset will be available after the reviewing process at https://sites.google.com/view/multiscore-project.

The dataset introduced in this work consists of 53,882 synthetic images of single-line (or system) pianoform scores, along with their digital score encoding.

In this section, we introduce the encoding representations of the musical scores in this dataset, as they are key aspects of the approach proposed in this paper, and we detail the way in which the corpus itself was created.

3.1 Ground-truth encoding

Since the goal of this dataset is for it to be a useful resource for the OMR community, we decided to generate a corpus based on standard digital notation documents. These output files can be then applied to other domains, such as graphic visualization software or the indexing of digital libraries.

First, it is necessary to analyze which encoding is most beneficial as regards being the endpoint of an end-to-end OMR system. The first options that can be considered are the most widespread musical encodings in libraries and musicology contexts: MEI [24] and MusicXML [25], which represent the components and metadata of a musical score in an XML-based markup encoding. Despite being extended formats, these music representations have a major drawback when considering their use in OMR systems, as they are too verbose. This is not convenient for OMR systems, since it would be hard to align input images with their correspondent notation representation.

In this paper, we use the text-based **kern encoding format, which is included in the Humdrum tool-set [26] and is hereafter referred to simply as KERN. This music notation format is one of the representations most frequently utilized for computational music analysis. Its features include a simple vocabulary and easy-to-parse file structure, which is very convenient for end-to-end OMR applications. Moreover, KERN files are compatible with dedicated music software [27, 28] and can be automatically converted to other music encodings, such as those mentioned above, by means of straightforward operations.

A KERN file is basically a sequence of lines. Each line is, in turn, another sequence of columns or *spines* that are separated by a *tab* character. Each column contains an instruction, such as the creation or ending of spines, or the encoding of musical symbols such as clefs, key signatures, meter, bar lines, or notes, to name but a few. When interpreting a KERN file, all spines are read simultaneously, thus providing the concept of polyphony to the format. That is, a line in a KERN document should be read from left to right—interpreting all the symbols that appear simultaneously—and then from top to bottom, advancing in time through the score.

In conceptual terms, the design of a KERN file resembles a music score that has been rotated to appear top to bottom rather than left to right (see Fig. 2). A basic example of how the encoding works is presented in Fig. 3, in which the word

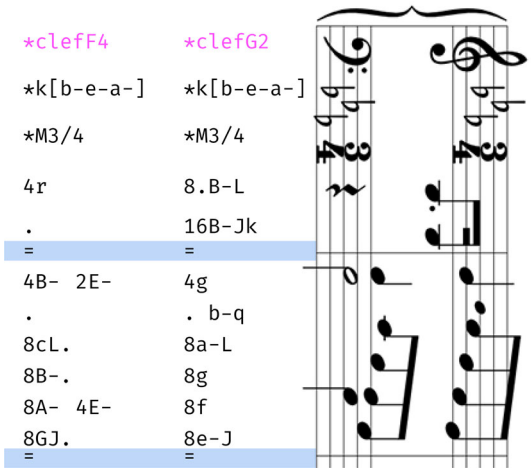


Fig. 2 Example of a KERN score (left) aligned with its rendered music document (right)

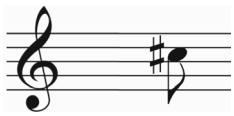


Fig. 3 Example of a simple excerpt of music. The corresponding KERN notation: `clefG2 \n 8cc#`

(`clefG2`) denotes a treble clef in the second line of the music staff and the symbol (`8cc#`) indicates that the note has a duration of an eighth note (`8`), has a pitch of C5 (`cc`), and comes with an accidental sharp (`#`), which alters the pitch of the note one semitone up. Thanks to its compactness—which eases score-representation alignment during transcription—and its compatibility with other music encodings and tools, the KERN format represents an excellent choice for end-to-end OMR approaches.

However, the fact of being such a highly compact format has some drawbacks for machine learning approaches, the most important of which is that the same visual structure can be encoded in different ways depending solely on personal preferences. This is owing to the fact that the token components can be ordered differently and the visual result is the same. For example, as shown in the note token in Fig. 2, the ending of a beam is encoded using the 'J' character. In KERN, it is valid to encode the whole symbol as in the figure, i.e., with `8e-J` denoting an eighth note ('8') of pitch E ('e') altered by a flat accidental ('−'), but also as `J8e-`.

Another problem is that, as observed in Table 1, the KERN music notation produces an extensive vocabulary size (unique symbols). We believe that this may hinder the performance of neural network-based approaches, signifying that a simplification of this music notation base would be convenient.

In this work, we, therefore, introduce an extension of this format that corrects the aforementioned. We have denominated it as **bekern, which is the abbreviation of "basic extended kern" and is referred to simply as BEKERN in the remainder of this paper. We allow just one "canonic" encoding of each feature, which is why we have denominated it as

"basic." In order to avoid different encodings for the same visual result, the ordering of token components has been restricted to a single one. The alphabet has been reduced by decomposing tokens into components delimited by means of a separator '.'. This signifies that the last token in Fig. 2, `8e-J`, despite having originally being encoded as `J8e-`, is encoded only as `8.e.-.J` in BEKERN format.[2]

The grammar and definition of this encoding are detailed in Appendix A.

3.2 Dataset building process

The dataset was constructed using the following steps:

1. All piano scores (those containing the '*Ipiano' signifier) from Kern Scores[3] were downloaded.
2. Those files that contained more than two staves, or that had any KERN parsing errors, were removed. Finally, 474 full-length scores were maintained. These comprised piano sonatas, mazurkas, preludes, and other compositions by Scarlatti, Mozart, Beethoven, Hummel, Chopin, and Joplin.
3. Three different pitch transpositions of the original pieces were used in order to augment data for training: major second, minor third, and major third. Each of these transformations moves the notes vertically, in addition to introducing new accidentals, and in many cases forces note stems and beams to have a different appearance (see Fig. 4).
4. For all the pieces obtained, the whole composition was randomly split into segments of 3 to 6 measures in order to obtain single-system scores (i.e., scores that are composed of just one system of two staves, like those in Fig. 4).
5. All dynamics, expression slurs, lyrics, and nongraphic information tokens were removed from the scores in order to generate what we have denominated as BEKERN.
6. These excerpts were then rechecked so as to retain only those that were valid KERN scores.
7. All the excerpts retained were used as the basis on which to generate new files with file extension.bekrn in the BEKERN format.
8. The music images were obtained by employing the Verovio digital engraver [28], which generates an SVG file from KERN. These input KERN files were obtained from the BEKERN by simply removing the dot separator. JPG images were then obtained from the SVG files through an automatic process. The variability of the engraved scores was increased by using randomly different parameter values of the Verovio tool in the range permitted by its documentation. Namely, we altered the parameters: all-line thickness, maximum beam slopes, the slur curve factor, the grace note factor, repeat bar line dot separation and font family.
9. Two versions of each image file were generated: the JPG file from the previous step, and a distorted version of the image that resembles a low-quality photocopy or print (see Fig. 5). The method used to distort images is described in [29].
10. Finally, all those samples for which Verovio generated an error and that did not generate the image were deleted.

Table 1 Transcription features for both of the proposed encodings

	KERN	BEKERN
Max. sequence length	1276	1716
Min. sequence length	32	34
Avg. sequence length	240 ± 107	367 ± 169
Unique tokens	20,575	188

Note that measures are provided depending on the tokenization method employed. KERN files use complete symbols as a basic token and BEKERN uses characters. However, these features are presented from the perspective of a transcription methodology, which will have to deal with these hyperparameters

Table 2 Summary table of the image features for both the GRANDSTAFF corpus and the camera distorted version

	GRANDSTAFF	Camera GRANDSTAFF
Max. width	3056	4048
Max. height	256	256
Min. width	143	164
Min. height	256	256
Avg. width	783	1047
Avg. height	256	256

All the size measures shown are in pixels at a resolution of 72ppi

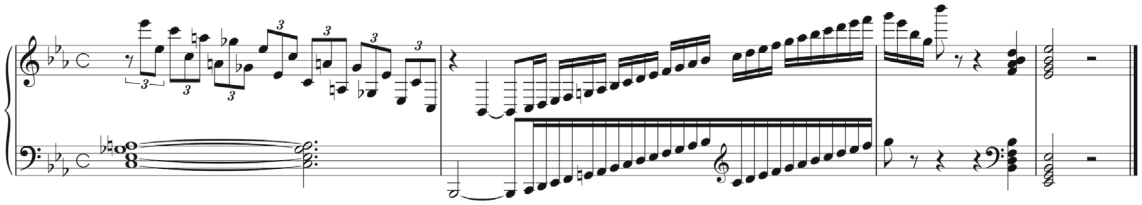
Information regarding image properties is found in Table 2, along with the KERN features depicted in Table 1. These data make it possible to observe that they are large images containing quite varied transcription lengths, thus making it particularly difficult to align information, a challenge that is related not only to OMR, but also to general document analysis.

4 Neural approach

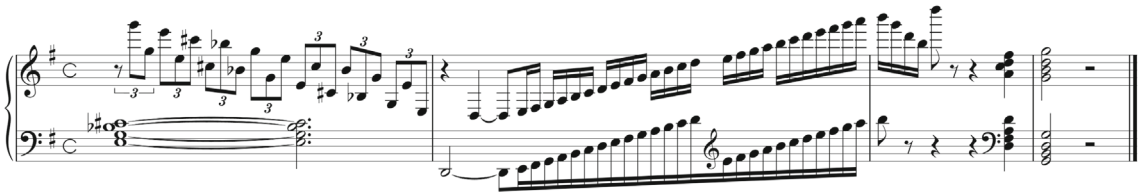
In this section, we briefly describe how end-to-end OMR has traditionally been addressed and why pianoform musical scores cannot follow this formulation. We then describe the proposed solution with which to tackle the pending challenge.

[2] Note that the token '.' is used solely as a separator, and the recognition model is, therefore, not expected to provide it explicitly.

[3] http://kern.ccarh.org/.



(a) Original piano staff



(b) Ascending major third transposition

Fig. 4 Example of transposition. The transposition has not only moved the position of notes but also the accidentals, note stems, and beam positions accordingly

Fig. 5 Example of a distorted image

As in previous works, input images are assumed to have undergone a previous layout analysis stage that leaves single-system sections [30], in the same way that end-to-end HTR works on single-line text sections [31].

4.1 End-to-end OMR

State-of-the-art OMR seeks the most probable symbolic representation $\hat{\mathbf{s}}$—encoded in the Σ_a music notation vocabulary—for each staff-section image x:

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}\in\Sigma_a} P(\mathbf{s} \mid x) \tag{1}$$

Neural networks approximate this probability by training with the Connectionist Temporal Classification (CTC) loss [32]. This alignment-free expectation–maximization method forces the network to maximize the sum of the probability of all the possible alignments between a ground-truth sequence s and the input source x. Since our input is an image, we treat x as a sequence of frames from this source. This is formalized as:

$$P(\mathbf{s} \mid x) = \sum_{\mathbf{a}\in\mathcal{A}_{\mathbf{s},x}} \prod_{t=1}^{T} p_t(\mathbf{a}_t \mid x) \tag{2}$$

where $\mathbf{a}$ is an auxiliary variable that defines a label in the output vocabulary at frame t. This variable belongs to the set $A_{s,x}$, which groups all the possible valid alignments between the image x and sequence s. Since $\mathbf{a}$ is a sequence that has length t, CTC implements a many-to-one map function $\mathcal{B}(\cdot)$ that compresses $\mathbf{a}$ to retrieve the transcription output [32]. To determine if $\mathbf{a}$ is a valid alignment, $\mathcal{B}(\mathbf{a}) = s$. This sum marginalizes our solution for all the valid combinations that are within the space between s and x sequences (defined as $\mathcal{A}_{(s,x)}$), since we understand the probability of a sequence to be the sequential combination of the probability of all its time steps.

The output of the network consists of a posteriorgram, which contains the probabilities of all the tokens within the vocabulary Σ_a. To allow for the possibility of no prediction

at a given timestep, CTC provides an extra blank token (ϵ). Therefore, the output vocabulary of the network becomes $\Sigma'_a = \Sigma_a \cup \{\epsilon\}$.

At inference, OMR methods resort to a *greedy* decoding, from which the most probable sequence is retrieved given an input image x. This can be decomposed as retrieving the most probable token at each timestep and applying $\mathcal{B}$ to retrieve the output sequence

$$\begin{aligned} \hat{a} &= \arg\max_{\hat{a}\in\Sigma'} \prod_{t=1}^{T} P(a_t \mid x). \\ \hat{s} &= \mathcal{B}(\hat{a}) \end{aligned} \tag{3}$$

The formulation presented treats the transcription task as a sequence retrieval problem, and the output of the network is, therefore, always a character sequence. A sequence of this nature is obtained from an image by converting the image domain $\mathcal{R}^{h\times w\times c}$—which is defined by the width w, height h and number of channels c of the image—into a sequence domain $\mathcal{R}^{l,\sum'_a}$, where l stands for the output sequence length and $\sum'_a$ is the aforementioned music notation vocabulary. CTC-based methods specifically define a reshape function $h : \mathcal{R}^{h\times w\times c} \rightarrow \mathcal{R}^{l,\sum'_a}$ based on the vertical collapse of the feature map, as symbols can be read from left to right and frames[4] always contain information about the same symbol in this case.

4.2 The challenge of polyphony

The methodology described above is able to solve single-staff music transcription problems and is currently the basis of the state-of-the-art systems in OMR for both printed and handwritten notation music scores.

Despite this, the end-to-end transcription of polyphonic and piano form scores is still a challenge (see Sect. 2). As stated in Sect. 3.1, pianoform music scores follow a particular reading order during interpretation, since there are staves that are read simultaneously. Rather than performing a line-by-line reading from top to bottom and left to right, interpretation is tied to staff groups, in which all elements are read simultaneously from left to right.

This increase in simultaneous events in a score is challenging, since the principle that a frame contains the information of a single music symbol is not satisfied, as there are multiple vertically aligned notes. When complexity does not grow significantly, as is the case of homophonic scores,[5] some vocabulary-based approaches can be employed. For example, in the work of Alfaro et. al [16] a special token is defined in order to differentiate between whether a note is played along with the previous one or belongs to the next time step. This approach could also be extended to polyphonic transcription at the cost of greatly increasing the ground truth sequence length, as simultaneous events are very frequent in these scores. However, as samples grow in size—e.g., full page-sized polyphonic music scores—this approach is no longer effective, as vertical collapse cannot produce sufficient frames to transcribe the complete music representation of the score.

It would, therefore, appear to be more convenient to search for new approaches or adaptations beyond the state-of-the-art single-staff music transcription formulation, as we require a more robust and scalable approach with which to address this challenge.

4.3 End-to-end polyphony transcription

In this section, we present a reading interpretation that aligns grand staves with their corresponding ground truth representation. We then provide details on a methodological approach with which to perform end-to-end transcription in order to solve its associated challenge.

4.3.1 Aligning polyphonic scores with their music representation

Although the current formulation cannot properly handle piano form notation, these scores can be interpreted in such a way that end-to-end transcription becomes applicable.

Upon closely studying the KERN and BEKERN encoding formats—which is found in Sect. 3.1 and Appendix A—it will be noted that each text line represents a specific *timestep* in the music score. That is, all the symbols in a KERN line are played at the same time, as they belong to different spines. The reading order of these documents is from top to bottom and left to right. This matches the left-to-right reading of the musical score. It is, therefore, possible to obtain a graphic alignment between them by rotating the source image 90° clockwise. When applying this transformation—as exemplified in Fig. 2—it is observed that both image and transcription are read in the same order. This is due to the nature of the KERN spines, which represent single musical staves aligned in the same way as displayed in the image.

By following this interpretation, we obtain both a document and a ground-truth text representation that are read like a text paragraph. This consequently makes it possible to reformulate the solution inspired by segmentation-free multiline transcription approaches.

[4] Column-wise elements of the image.

[5] Homophony occurs in a music score when all the symbols that are aligned vertically start and end at the same time.

4.3.2 Score unfolding approach

Segmentation-free multi-line document transcription is a text methodology whose objective is to transcribe document images that contain more than one line without the need to perform any previous line detection processes. Although it is a recent research topic, several works with which to address this challenge have been proposed. The most relevant approaches found are those based on *attention* [33–36], which perform a line-by-line or token-by-token transcription process by means of an attention matrix or self-attention, and those of a *document unfolding* nature [37, 38], in which the model learns to unfold text lines in order for them to be read sequentially in their corresponding reading order.

The attention-based methods apply backpropagation when all the lines in a sample are processed, which is not, in our case, convenient owing to the large number of lines that KERN files typically contain. In this paper, we have, therefore, employed a document unfolding approach and were specifically inspired to do so by the work of Coquenet et. al. [38], as document unfolding is learned without an input image size constraint.

Here, rather than concatenating frame-wise elements along the height axis (h) during the vertical collapse, we reshape the feature map by concatenating all of its rows (w) to subsequently obtain a $(c, h \times w)$ sequence, in which c is the number of filters used by the convolutional layers of the model. From a high-level perspective, this method can be understood as a pairwise polyphonic region concatenation process—as illustrated in Fig. 2. This operation is performed from top to bottom of the image. Graphic visualization of this method is depicted in Fig. 6.

This means of processing the feature map obtained allows the transcription of the musical score in its original KERN format, as labels are aligned in the same way—from top to bottom and left to right. However, some symbols have to be introduced into the vocabulary in order to produce correct KERN sequences. These are the line breaks, as this is mandatory in order to know where music timesteps are separated, the tab token, as this indicates spine jumps, and the space token, which identifies homophonic symbols.

5 Experiments

In this section, we define the environment designed in order to evaluate the performance of the end-to-end polyphonic music recognition method presented in the corpus of this paper.[6]

[6] Source code for the implementation of the experimental environment is available at https://github.com/multiscore/e2e-pianoform.

5.1 Implementations considered

Three different implementations have been proposed for study purposes. All of them contain a convolutional block, which acts as an image encoder that extracts the most relevant features from the input. The implementation of [38] is followed, which consists of a network of ten stacked convolutional layers with pooling operators, which eventually produce a feature map of size $(b, c, h/8, w/16)$, where h and w are the height and the width of the input image, c are the filters in the last convolutional layer, and b is the batch size. An illustration of this encoder architecture is provided in Fig. 7. It must be mentioned that all the considered implementations have similar parameters, around 23 M, where the majority of the weights are located in the convolutional encoder. The decoding architectures proposed to process the sequence obtained after using the reshaping method are, therefore, the following.

5.1.1 Recurrent neural network

We implemented the decoder from the original Convolutional Recurrent Neural Network (CRNN) single-staff transcription model from the work of Calvo-Zaragoza et al. [21], in which the reshaped feature map is fed into a single Bidirectional LSTM (BLSTM) layer and fed into a fully connected layer that converts from the RNN feature space into the output vocabulary one. We specifically implemented a BLSTM with 256 units. This decoder implementation is depicted in Fig. 8b.

5.1.2 The transformer

The base model of OMR decoders typically implements Recurrent Neural Networks (RNN) in order to process the reshaped feature map as a sequence. However, there is a recurrent-free model that has gained popularity in Natural Language Processing (NLP): the Transformer [39]. This model replaces the RNN architecture by implementing sequence modeling through the use of attention mechanisms and position learning. This model solves some common issues related to RNN—such as processing long sequences—at the cost of requiring more data in order to converge. As noted in the reshaping step and the KERN format for polyphonic music scores, we believe that the model would have to process significantly long output sequences, something that Transformers tend to handle better than RNN. In previous works, the Transformer has been studied as regards its use to perform transcription tasks in both OMR and HTR [40, 41]. This research has shown that the Transformer model is a promising architecture for performing both OMR and HTR tasks. Although it currently does not yield better performance than traditional RNNs —if no support synthetic data or training techniques are provided—in these two areas,

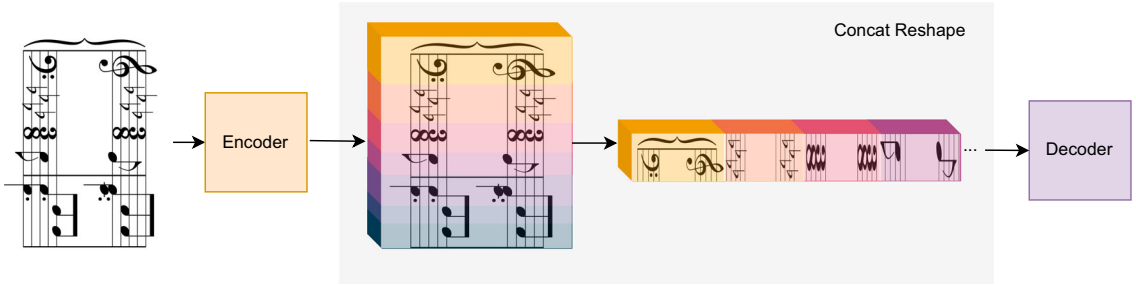


Fig. 6 Graphical scheme of the proposed reshape method used to transcribe polyphonic music scores. It should be noted that this reshaping is performed in a feature space, not on the image itself, and visualization of it has, therefore, been included for the sake of clarity

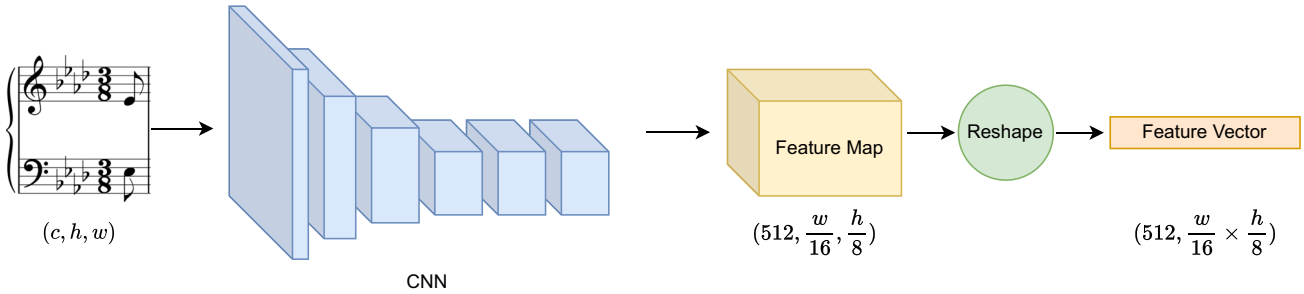


Fig. 7 Scheme of the encoder architecture implemented in all the models evaluated. The model input is a 90°-rotated polyphonic image of height h, width w, and c channels. It outputs a feature vector of $\frac{w}{16} \times \frac{h}{8}$ frames and 512 features

relevant improvements have been found in the OCR field [42]. We, therefore, propose an implementation that replaces the recurrent layer of the CRNN model with a Transformer encoder module in the same way it is done in [40]—shown in Fig. 8c—which is referred to as CNNT in what remains of the paper. We specifically implemented one encoder layer with an embedding size of 512 units, a feed-forward dimension of 1024, and 8 attention heads.

5.1.3 Encoder-only model

As mentioned previously, the proposed methodology with which to transcribe polyphony is based on analogous works for multi-line transcription in the HTR field [37, 38]. These works are based on convolutional-only architectures—in which no sequence processing decoders are implemented, as the solution lies in preserving the prediction space in two dimensions, and applying backpropagation directly to the feature map retrieved before being reshaped. In order to carry out our study on the architecture, we implemented an encoder-only network. As it is based only on fully convolutional layers, it will be referred to as FCN in the results section. This implementation is depicted in Fig. 8a.

5.2 Sequence codification

In this paper, we have used two encodings to represent polyphonic musical scores. These are Humdrum KERN—which is the semantic encoding chosen to represent the digital music documents of the GRANDSTAFF corpus—and its basic encoding (BEKERN)—which is a semantic-based tokenization performed in order to dramatically reduce the number of unique symbols of KERN. The utility of this proposed encoding was assessed by evaluating the performance of the transcription method. This was done by outputting a third KERN vocabulary reduced using a non-semantic-aware tokenization method, which would be the first approach to employ if there were no prior knowledge of music encodings.

We shrank the KERN vocabulary from the GRANDSTAFF dataset by employing the *Sentence Piece* strategy [43], which is a standard utility in the Machine Translation field when performing vocabulary compression. This text tokenizer provides a set of unsupervised methods based on sub-word units, such as the Byte Pair Encoding algorithm [44] and the Unigram Language Model [45]. This tool was chosen not only because it is a standard utility, but also because it allows the specification of a vocabulary size, which is ideal for comparison purposes, since we can create a vocabulary that is equal in size to that obtained with the BEKERN encoding. This new vocabulary is referred to as the KERN- SP encoding in the remainder of this paper.

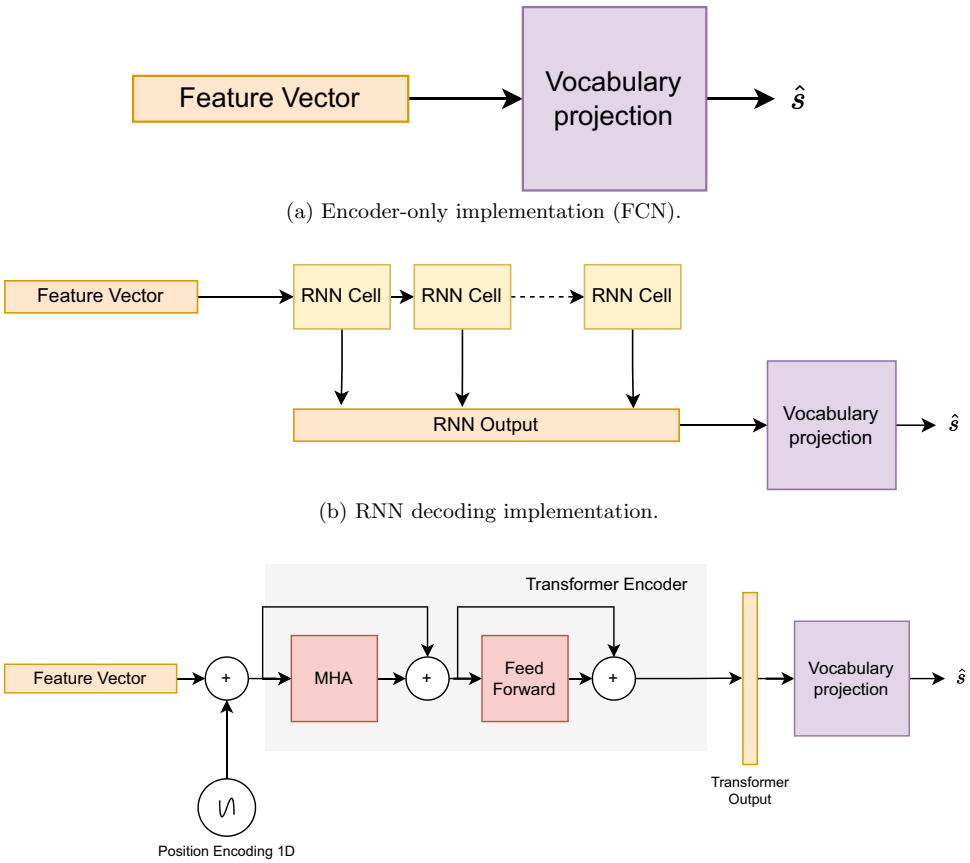


Fig. 8 Architecture schemes of different implementations of the decoder used in this paper

(a) Encoder-only implementation (FCN).

(b) RNN decoding implementation.

(c) Transformer decoding implementation (CNNT).

5.3 Evaluation procedure

The GRANDSTAFF dataset provides two data splits, a training one and a test one. The test split consists of all the real musical scores extracted in order to create the corpus, as we believe that test results should be provided by means of real samples. The training and validation splits consist of all the altered musical scores, the preparation of which is detailed in Sect. 3.2. We specifically train and validate on 46,221 samples and perform tests on 7661 samples.

5.4 Metrics

One issue that may be encountered when evaluating OMR experiments is that of correctly assessing the performance of a transcription model, as certain features must be taken into account in music notation. However, OMR does not have a specific evaluation protocol [1]. In our case, it is convenient to use text-related metrics to evaluate the accuracy of the predictions. Three metrics have been proposed in order to evaluate the performance of the models implemented. All of these measures are based on the normalized mean edit distance between a hypothesis sequence $\hat{\mathbf{s}}$ and a reference sequence $\mathbf{s}$ in the form of:

$$\mathcal{E}(\hat{S}, S) = \frac{\sum_{i=0}^{n} d(\mathbf{s}_i, \hat{\mathbf{s}}_i)}{\sum_{i=0}^{n} |\mathbf{s}_i|} \tag{4}$$

where $\hat{S}$ is the hypotheses set, S is the ground-truth set, $d(\cdot, \cdot)$ is the edit distance between the tokens of each paired hypothesis and ground-truth sequences $(\mathbf{s}_i, \hat{\mathbf{s}}_i)$, and $|\mathbf{s}_i|$ is the length of the reference sequence in tokens.

As will be observed in the operation determined by Eq. 4, the edit distance-based error $\mathcal{E}$ depends on what is defined as a token in the codification. This is used as the basis on which to compute the Character Error Rate (CER), which tokenizes sequences at a BEKERN character level, as detailed in Appendix A. The second metric is the Symbol Error Rate (SER), which computes the edit distance between complete KERN symbols.[7] Finally, as the problem is solved using a multi-line transcription approach that rotates the music score and attempts to align each KERN line with the input image by predicting a line break token flag, we compute the Line Error Rate (LER), which makes it possible to assess the amount of error produced while retrieving lines—by setting them

[7] In this context, SER can be understood as an analogous measure of the Word Error Rate in the HTR field, as the network outputs single characters that have to be joined to complete music symbols.

as complete tokens when calculating $\mathcal{E}(\hat{S}, S)$. We consider that this metric is particularly interesting for both the paper and the polyphonic music transcription problem. KERN files rely heavily on these text structures to represent this music notation, as it easily indicates the notes to be played and the temporal sequentiality of the score. Since correctly predicting and differentiating all the lines of a given document are a key aspect, the overall quality of the outputted KERN files can be assessed using this metric.

6 Results

Table 3 shows the results obtained by the methodology proposed in this paper on the test set for both the perfectly printed and the distorted datasets. Note that no reference/baseline results are shown, as the state-of-the-art end-to-end methods [1, 15] failed to converge during training for this specific dataset. This was caused by the issues described in Sect. 4.2.

From an overall perspective, the results show that the score unfolding method was able to allow the neural network to solve the problem with fair results, with the best SER values being 5.8% and 6.5% with the BEKERN encoding using the Transformer. These error values directly scale depending on the image complexity, as the distorted version of the corpus contains features that make recognition more difficult and, understandably, have an impact on the overall performance of the models. This impact can be clearly seen in the encoder-only implementation of the model, with a drop in performance of approximately 16%. This value shows that sequence processing modules are, indeed, necessary in order to perform polyphonic transcription, as they provide stability against increasing corpora complexity, with a drop in performance of 12% in the best-case scenario.

Upon comparing sequence processing implementations, the results show that, for polyphonic music transcription, the combination of a CNN with a Transformer Encoder—when outputting BEKERN vocabulary—provides the best transcription results. Table 1 supports the idea that, on average, the BEKERN encoding produced longer sequences than the Kern one, in exchange for having a significantly narrower vocabulary. By replacing recurrence with self-attention and position encoding, the transformers improved computation time and accuracy at the cost of requiring more data in order to converge. Indeed, Transformers literature reports relevant improvements in terms of sequence length limitations, being able to process longer sequences than RNNs. In this case, it would appear that the GRANDSTAFF dataset creates a scenario that is ideal for Transformer-based models, as there is a large amount of available data and, on average, long sequences to be transcribed. Indeed, RNN-based decoders provided the best performance results when transcribing raw KERN sequences.

In terms of output sequence tokenization, the results showed that a reduction in vocabulary improves the results of the model, since the number of parameters to be optimized in the last layer is significantly reduced. We observe, depending on the model, some variant gaps between the semantic-based tokenization method—BEKERN—and the unsupervised learned one, in our case, *Sentence Piece*. It seems that vocabulary selection may be an *ad-hoc* decision when implementing a model. However, from the best results obtained in these experiments, it seems that the BEKERN format provides better performance, as it is a semantic encoding based on prior knowledge of music notation.

Finally, we should highlight the LER obtained by the implemented methods. As described in Sect. 5.4, the LER metric indicates how well the model aligns the input image with the output transcription in terms of complete KERN lines. The results show an overall LER performance of 16.26% and 17.53%. This means that the error produced by the model is mostly intra-line and that the proposed methodology was, therefore, able to correctly align the rotated music image with its KERN transcription. This result proves that our results can

Table 3 Average CER, SER, and LER (%) obtained by the studied models on the test set for both the perfectly printed and the distorted versions of the GRANDSTAFF dataset

Encoding	Model	GrandStaff			Camera GrandStaff		
		CER	*SER*	*LER*	CER	*SER*	*LER*
KERN	FCN	14.6	23.9	67.9	20.6	30.2	69.0
	CRNN	5.0	7.3	23.2	7.2	9.9	29.5
	CNNT	7.9	11.1	32.4	9.4	12.3	33.3
KERN- SP	FCN	6.4	11.3	29.8	11.9	22.5	58.3
	CRNN	5.0	9.2	25.9	5.8	10.4	27.9
	CNNT	5.1	7.8	21.4	5.8	10.3	27.1
BEKERN	FCN	8.1	12.1	35.3	23.6	28.3	70.8
	CRNN	6.1	9.1	23.4	9.6	13.0	34.1
	CNNT	3.9	5.8	16.3	4.6	6.5	17.5

Table 4 Encoding of the original piano staff shown in Fig. 9 and its transcription with CNTT. Italics represent elements wrongly predicted by the transcription model (deletions). Bold in the original sequence represents missing tokens in the prediction (insertions)

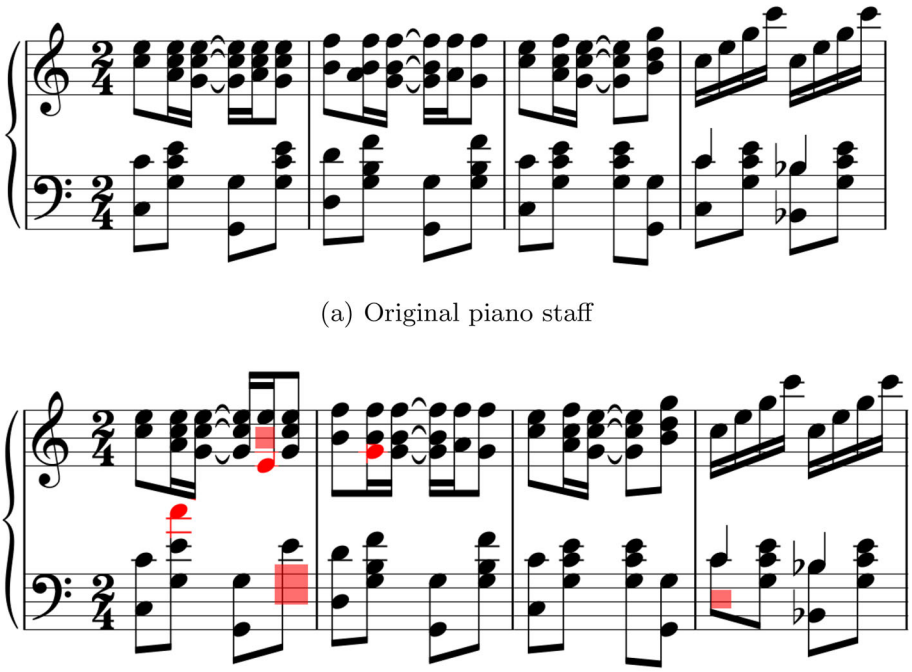
Original			Prediction		
**kern	**kern		**kern	**kern	
*clefF4	*clefG2		*clefF4	*clefG2	
*k[]	*k[]		*k[]	*k[]	
*M2/4	*M2/4		*M2/4	*M2/4	
=-	=-		=-	=-	
8cL 8C	8eeL 8cc		8cL 8C	8eeL 8cc	
8eJ 8c 8 G	16eeL 16cc 16a		8eJ 8c*c* 8 G	16eeL 16cc 16a	
.	16eeJJ[16cc[16 g[		.	16eeJJ[16cc[16 g[	
8GL 8GG	16eeLL] 16cc] 16 g]		8GL 8GG	16eeLL] 16cc] 16 g]	
.	16eeJ **16cc 16a**		.	16eeJ *8 8e*	
8eJ 8c **8 G**	8eeJ 8cc 8 g		8eJ *8*	8eeJ 8cc 8 g	
=	=		=	=	
8dL 8D	8ffL 8b		8dL 8D	8ffL 8b	
8fJ 8B 8 G	16ffL 16b 16a		8fJ 8B 8 G	16ffL 16b 16*g*	
.	16ffJJ[16b[16 g[		.	16ffJJ[16b[16 g[	
8GL 8GG	16ffLL] 16b] 16 g]		8GL 8GG	16ffLL] 16b] 16 g]	
.	16ffJ 16a		.	16ffJ 16a	
8fJ 8B 8 G	8ffJ 8 g		8fJ 8B 8 G	8ffJ 8 g	
=	=		=	=	
8cL 8C	8eeL 8cc		8cL 8C	8eeL 8cc	
8eJ 8c 8 G	16ffL 16cc 16a		8eJ 8c 8 G	16ffL 16cc 16a	
.	16eeJJ[16cc[16 g[		.	16eeJJ[16cc[16 g[	
8eL 8c 8 G	8eeL] 8cc] 8 g]		8eL 8c 8 G	8eeL] 8cc] 8 g]	
8GJ 8GG	8ggJ 8dd 8b		8GJ 8GG	8ggJ 8dd 8b	
=	=		=	=	
*^	*		*^	*	
4c	8cL **8C**	16ccLL	4c	8cL	16ccLL
.	.	16ee	.	.	16ee
.	8eJ 8c 8 G	16gg	.	8eJ 8c 8 G	16gg
.	.	16cccJJ	.	.	16cccJJ
4B-	8B-L 8BB-	16ccLL	4B-	8B-L 8BB-	16ccLL
.	.	16ee	.	.	16ee
.	8eJ 8c 8 G	16gg	.	8eJ 8c 8 G	16gg
.	.	16cccJJ	.	.	16cccJJ
=	=	=	=	=	=
*v	*v	*	*v	*v	*
*-	*-		*-	*-	

be easily exported to practical applications that deal with KERN files and that errors should principally be corrected by reviewing line content, not the overall format of the document (Fig. 9).

6.1 Evaluation on monophonic scores

The method proposed in this paper for music transcription involves aligning input images with their corresponding KERN ground truth notation by approaching it as a multiline endeavor. This method is not limited to polyphonic music—since it relies on visual-text alignment—and can be applied to other KERN-encoded music scores, including monophonic ones, which have been the main target of existing end-to-end OMR techniques.

To complete the analysis of the methodology considered in this work, we conducted additional experiments to evaluate its effectiveness for monophonic music score transcription. We trained our models with the camera version of the "Printed Images of Music Staves" (Camera-PrIMuS) dataset [29], which is a well-known benchmark for end-to-end OMR.



(a) Original piano staff

(b) Transcribed prediction with CNNT, rendered with the Verovio toolkit

Fig. 9 Visualization of the transcription produced by the CNNT model for a test pianoform staff. The corresponding KERN encoding is shown in Table 4. Errors are displayed in the prediction image, where red boxes highlight missing symbols and red notes indicate wrong predictions. In this particular case, the CER obtained is 2.8%, the SER is 4.1% and the LER is 16.2%

Table 5 Average SER (%) obtained by the studied architectures and reshape methods on the test set for the monophonic camera priMuS dataset

Architecture	Reshape method	*SER*
FCN	Vertical collapse	6
	Unfolding	7.8
CRNN	Vertical collapse	**3.3**
	Unfolding	4.8
CNNT	Vertical collapse	9.8
	Unfolding	10.4
State of the art [46]	Vertical collapse	4.7

Best average result is in bold

The results of our experiments, which compared the performance of the implemented models in this work using both the state-of-the-art reshape approach (vertical collapse) and the unfolding method considered in this paper, are presented in Table 5.

Our experimental results indicate that the unfolding method is able to successfully perform end-to-end monophonic transcription. However, this approach reports lower accuracy compared to the vertical collapse approach. This performance is mainly obtained thanks to the convolutional architecture implemented within, where it is able to improve 1% SER the state-of-the-art results.

It is important to note that we also conducted one additional experiment to directly transcribe monophonic scores using the networks trained with GRANDSTAFF. However, the results of this case showed that the models were unable to retrieve barely accurate predictions. All our empirical outcomes, therefore, suggest that our methodology can effectively perform transcription for both monophonic and polyphonic tasks, but it has yet to be performed by training independent task-specific models.

7 Conclusions

This work shows a proposal for the first end-to-end OMR approach with which to solve the transcription of pianoform musical scores. This solution extends state-of-the-art staff-level transcription methods and was inspired by multi-line document transcription. We specifically take advantage of a standard digital music notation system, Humdrum **kern (KERN), and implement a neural network that learns to unfold a rotated pianoform system and align it with its corresponding transcript. This method is trained with weakly annotated data, as it requires only pairs of images and their digital document representation, without any geometric information, such as staff positions or symbol locations in the image.

In addition to this approach, we also present the GRANDSTAFF Dataset for use in experiments. This dataset consists specifically of a collection of 53 882 polyphonic single-line pianoform scores extracted from the KernScores repository and rendered using the Verovio tool. This dataset provides two music encodings for each score: the original kern document and the Basic **kern (BEKERN) notation sequence,

which consists of a simplification of the base encoding that reduces vocabulary size in order to ease the work of the transcription systems.

The evaluation results obtained show that the proposed method successfully transcribes pianoform music systems with fair error rates. This represents a clear advance as regards attaining effective end-to-end OMR systems. Our work also provides baseline results for future work addressing the same challenge.

As future work, this paper opens up several research avenues. In this paper, we propose an output sequence constructed with semantic music grammar. However, most of the results in OMR are framed in graphic-based vocabularies as the output of their systems. A comparative study between using this approach or a joint transcription and machine translation pipeline could be performed—as occurs in [46]. Moreover, the proposed approach is limited to simultaneous-only music staves. That is, this method can be extended only to full pages that contain completely simultaneous music, but not sequentially structured polyphonic staves, as we stick to a specific reading order that is not followed in those cases. Future efforts should, therefore, focus on how to extend transcription systems in order to address the full-page polyphonic music score recognition topic, as is also occurring in the HTR field with full-page documents [36, 47]. Finally, this work demonstrates that the implemented method is able to transcribe both polyphonic—pianoform—and monophonic music images by rotating and aligning them vertically with their digital music representation, thanks to the KERN format. However, given the reported results, networks have to be trained as separate tasks to do so. The general application of this method to other musical score types could also be explored, thus leading to research toward universal OMR solutions.

Acknowledgements This paper is part of the MultiScore project (PID2020-118447RA-I00), funded by MCIN/AEI/10.13039/501100011033. The first author is supported by Grant ACIF/2021/356 from the "Programa I+D+i de la Generalitat Valenciana."

Author Contributions All authors have made equal contributions to the manuscript.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors declare no conflict of interest.

Appendix A **bekern grammar

The grammars describing the **bekern format introduced in Sect. 3 are detailed below in Extended Backus–Naur Form (EBNF) notation. The rules have been given self-explanatory names and should be understood by any person with some music notation knowledge. An ANTLR [48] grammar is provided along with the dataset.

A.1 Lexical specification

TOKEN	DEFINITION
space	`' '`
tab	`'\t'`
comma	`','`
colon	`':'`
semicolon	`';'`
pipe	'|'
dot	`'.'`
sep	`'.'`
plus	`'+'`
minus	`'-'`
underscore	`'_'`
octothorpe	`'#'`
circumflex	`'^'`
slash	`'/'`
equal	`'='`
exclamation	`'!'`
leftParenthesis	`'('`
rightParenthesis	`')'`
leftBracket	`'['`
rightBracket	`']'`
angleBracketOpen	'<'
angleBracketClose	'>'
digit	`'0'.. '9'`
number	digit+
eol	`'\r'?'\n'`
eof	`"end of file character"`
bekern	`'**ekern_1.0'`
tandemStaff	`'*staff'`
tandemKeySignature	`'*k'`
tandemMet	`'*met'`
tandemTimeSignature	`'*M'`
spineTerminator	`'*-'`
spineAdd	`'*+'`
spineSplit	`'*^'`
spineJoin	`'*v'`
asterisk	`'*'`

A.2 Grammar

start	→ header (eol record)* eol* eof
start	→ header (eol record)* eol* eof
header	→ headerField (tab headerField)*
record	→ fields | spineOperations
headerField	→ bekern
fields	→ field (tab field)*
spineOperations	→ spineOperation (tab | spineOperation)*
field	→ graphicalToken | placeHolder
placeHolder	→ dot
graphicalToken	→ (tandemInterpretation | barline | rest | note | chord)?
tandemInterpretation	→ staff | clef | keySignature | timeSignature | meterSymbol | nullInterpretation
lowerCasePitch	→ 'a' .. 'g'
upperCasePitch	→ 'A' .. 'G'
pitchClass	→ lowerCasePitch sep accidental
staff	→ tandemStaff plus? number (slash number)?
clef	→ tandemClef clefValue
clefValue	→ clefSign clefLine? (sep clefOctave)?
clefSign	→ 'C' | 'F' | 'G'
clefLine	→ '1' .. '5'
clefOctave	→ 'v' 'v'? '2' | circumflex circumflex? '2'
keySignature	→ tandemKeySignature leftBracket keySignaturePitchClass* rightBracket keySignatureCancel?
keySignaturePitchClass	→ lowerCasePitch accidental
keySignatureCancel	→ 'X'
timeSignature	→ tandemTimeSignature (numerator slash denominator)
numerator	→ number
denominator	→ number
meterSymbol	→ (tandemTimeSignature | tandemMet leftParenthesis (modernMeterSymbolSign) rightParenthesis
modernMeterSymbolSign	→ ('c' | 'C') pipe?
nullInterpretation	→ asterisk
barline	→ equal barLineType? minus?
barLineType	→ exclamation pipe colon | colon pipe exclamation (pipe colon)? | pipe pipe | pipe exclamation colon? | equal colon pipe exclamation | equal
spineOperation	→ spineTerminator | spineAdd | spineSplit | spineJoin | spinePlaceholder
spinePlaceholder	→ asterisk
rest	→ duration sep 'r' (sep staffChange)? (sep fermata)?
duration	→ modernDuration
fermata	→ semicolon
modernDuration	→ number (sep augmentationDot+)?
augmentationDot	→ dot
pitch	→ diatonicPitchAndOctave (sep alteration)?
alteration	→ accidental (sep alterationDisplay)?
note	→ (duration sep)? pitch (sep staffChange)? afterNote
staffChange	→ angleBracketOpen | angleBracketClose
chord	→ note (chordSpace note)+
chordSpace	→ space
graceNote	→ (duration sep)? 'q'
tie	→ (tieStart | tieEnd | tieContinue) (staffChange)?
afterNote	→ (sep (tie | beam | fermata | glissando | graceNote))*
The elements inside this rule are always ordered alphabetically	
diatonicPitchAndOctave	→ bassNotes | trebleNotes
trebleNotes	→ lowerCasePitch+
bassNotes	→ upperCasePitch+
accidental	→ octothorpe (octothorpe octothorpe?)? | minus minus? | 'n'
alterationDisplay	→ 'x' | 'X' | 'i' | 'I' | 'j' | 'Z' | ('y' 'y'?) | ('Y' 'Y'?)
glissando	→ colon
tieStart	→ angleBracketOpen | leftBracket 'y'?
tieContinue	→ underscore
tieEnd	→ angleBracketClose | rightBracket
beam	→ (('L' | 'J' | 'K' | 'k') staffChange?)+

References

1. Calvo-Zaragoza, J. J. H., Jr., Pacha, A.: Understanding optical music recognition. ACM Comput. Surv **53**(4), 77–17735 (2020)
2. Byrd, D., Simonsen, J.G.: Towards a standard testbed for optical music recognition: definitions, metrics, and page images. J. N. Music Res. **44**(3), 169–195 (2015)
3. Hajič jr., J., Pecina, P.: The MUSCIMA++ dataset for handwritten optical music recognition. In: 14th International Conference on Document Analysis and Recognition, Kyoto, Japan, pp. 39–46 (2017)
4. Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marcal, A.R., Guedes, C., Cardoso, J.S.: Optical music recognition: state-of-the-art and open issues. Int. J. Multim. Inf. Retr. **1**(3), 173–190 (2012)
5. Dalitz, C., Droettboom, M., Pranzas, B., Fujinaga, I.: A comparative study of staff removal algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **30**(5), 753–766 (2008)
6. Rossant, F., Bloch, I.: Robust and adaptive OMR system including fuzzy modeling, fusion of musical rules, and possible error detection. EURASIP J. Adv. Sig. Proc. **2007**, 1–25 (2007)
7. Gallego, A.-J., Calvo-Zaragoza, J.: Staff-line removal with selectional auto-encoders. Exp. Syst. Appl. **89**, 138–148 (2017)
8. Pacha, A., Eidenberger, H.: Towards a universal music symbol classifier. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 2, pp. 35–36 (2017). IEEE
9. Chowdhury, A., Vig, L.: An efficient end-to-end neural model for handwritten text recognition. In: 29th British Machine Vision Conference (2018)
10. Chiu, C.-C., Sainath, T.N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R.J., Rao, K., Gonina, E., Jaitly, N., Li, B., Chorowski, J., Bacchiani, M.: State-of-the-art speech recognition with sequence-to-sequence models. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4774–4778 (2018)
11. Zhang, J., Du, J., Zhang, S., Liu, D., Hu, Y., Hu, J., Wei, S., Dai, L.: Watch, attend and parse: an end-to-end neural network based approach to handwritten mathematical expression recognition. Pattern Recogn. **71**, 196–206 (2017)
12. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2017)
13. Calvo-Zaragoza, J., Rizo, D.: End-to-end neural optical music recognition of monophonic scores. Appl. Sci. **8**(4), 606 (2018)
14. Alfaro-Contreras, M., Ríos-Vila, A., Valero-Mas, J.J., Iñesta, J.M., Calvo-Zaragoza, J.: Decoupling music notation to improve end-to-end optical music recognition. Pattern Recognit. Lett. **158**, 157–163 (2022)
15. Baró, A., Riba, P., Calvo-Zaragoza, J., Fornés, A.: From optical music recognition to handwritten music recognition: a baseline. Pattern Recogn. Lett. **123**, 1–8 (2019)
16. Alfaro-Contreras, M., Calvo-Zaragoza, J., Iñesta, J.M.: Approaching end-to-end optical music recognition for homophonic scores. In: 9th Iberian Conference Pattern Recognition and Image Analysis. Lecture Notes in Computer Science, vol. 11868, pp. 147–158. Springer, Madrid, Spain (2019)
17. Edirisooriya, S., Dong, H., McAuley, J.J., Berg-Kirkpatrick, T.: An empirical evaluation of end-to-end polyphonic optical music recognition. In: Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, pp. 167–173 (2021)
18. Tuggener, L., Elezi, I., Schmidhuber, J., Pelillo, M., Stadelmann, T.: DeepScores-A Dataset for Segmentation, Detection and Classification of Tiny Objects. In: Proceedings of the 24th International Conference on Pattern Recognition, pp. 3704–3709 (2018)
19. Jan Hajič, j., Pecina, P.: The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. In: 14th International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 13 - 15, 2017, pp. 39–46. IEEE Computer Society, New York, USA (2017). Dept. of Computer Science and Intelligent Systems, Graduate School of Engineering, Osaka Prefecture University
20. Parada-Cabaleiro, E., Batliner, A., Schuller, B.W.: A diplomatic edition of Il Lauro Secco: ground truth for OMR of white mensural notation. In: Proceedings of the 20th International Society for Music Information Retrieval Conference, pp. 557–564 (2019)
21. Calvo-Zaragoza, J., Toselli, A.H., Vidal, E.: Handwritten music recognition for mensural notation with convolutional recurrent neural networks. Pattern Recognit. Lett. **128**, 115–121 (2019)
22. Ros-Fábregas, E.: Codified Spanish Music Heritage Through Verovio: The Online Platforms Fondo de Música Tradicional IMF-CSIC and Books of Hispanic Polyphony IMF-CSIC. Alicante, Spain (2021)
23. Gould, E.: Behind Bars: The Definitive Guide to Music Notation, Faber Faber Music, London, United Kingdom (2011)
24. Hankinson, A., Roland, P., Fujinaga, I.: The music encoding initiative as a document-encoding framework. In: International Conference on Music Information Retrieval (2011)
25. Good, M., Actor, G.: Using MusicXML for file interchange. In: Web Delivering of Music, International Conference on 0, 153 (2003)
26. Huron, D.: Humdrum and Kern: Selective feature encoding BT - beyond MIDI: the handbook of musical codes. In: Beyond MIDI: The Handbook of Musical Codes, pp. 375–401. MIT Press, Cambridge, MA, USA (1997)
27. Sapp, C.S.: Verovio humdrum viewer. In: Proceedings of Music Encoding Conference (MEC), Tours, France (2017)
28. Pugin, L., Zitellini, R., Roland, P.: Verovio - a library for engraving MEI music notation into SVG. In: International Society for Music Information Retrieval (2014)
29. Calvo-Zaragoza, J., Rizo, D.: Camera-PrIMuS: Neural end-to-end optical music recognition on realistic monophonic scores. In: Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, pp. 248–255 (2018)
30. Castellanos, F.J., Garrido-Munoz, C., Ríos-Vila, A., Calvo-Zaragoza, J.: Region-based layout analysis of music score images. Exp. Syst. Appl. **209**, 118211 (2022)
31. Sánchez, J., Romero, V., Toselli, A.H., Villegas, M., Vidal, E.: A set of benchmarks for handwritten text recognition on historical documents. Pattern Recognit. **94**, 122–134 (2019)
32. Graves, A., Fernández, S., Gomez, F.J., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the Twenty-Third International Conference on Machine Learning, (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006, pp. 369–376 (2006)
33. Bluche, T.: Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS'16, pp. 838–846. Curran Associates Inc., Red Hook, NY, USA (2016)
34. Bluche, T., Louradour, J., Messina, R.O.: Scan, attend and read: End-to-end handwritten paragraph recognition with MDLSTM attention. In: 14th IAPR International Conference on Document Analysis and Recognition. ICDAR 2017, pp. 1050–1055. IEEE, Kyoto, Japan (2017)
35. Coquenet, D., Chatelain, C., Paquet, T.: End-to-end handwritten paragraph text recognition using a vertical attention network. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)

36. Singh, S.S., Karayev, S.: Full page handwriting recognition via image to sequence extraction. In: Document Analysis and Recognition - ICDAR 2021: 16th International Conference. Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part III, pp. 55–69. Springer, Berlin, Heidelberg (2021)
37. Yousef, M., Bishop, T.E.: Origaminet: Weakly-supervised, segmentation-free, one-step, full page textrecognition by learning to unfold. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
38. Coquenet, D., Chatelain, C., Paquet, T.: Span: A simple predict & align network for handwritten paragraph recognition. In: 16th International Conference on Document Analysis and Recognition, ICDAR. Lecture Notes in Computer Science, vol. 12823, pp. 70–84 (2021)
39. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30**, 5998–6008 (2017)
40. Ríos-Vila, A., Iñesta, J.M., Calvo-Zaragoza, J.: On the use of transformers for end-to-end optical music recognition. In: Pattern Recognition and Image Analysis, pp. 470–481. Springer, Cham (2022)
41. Kang, L., Riba, P., Rusiñol, M., Fornés, A., Villegas, M.: Pay attention to what you read: Non-recurrent handwritten text-line recognition. Pattern Recogn. **129**, 108766 (2022)
42. Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models. arXiv (2021)
43. Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (2018)
44. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1715–1725. Association for Computational Linguistics, Berlin, Germany (2016)
45. Kudo, T.: Subword regularization: Improving neural network translation models with multiple subword candidates. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 66–75. Association for Computational Linguistics, Melbourne, Australia (2018)
46. Ríos-Vila, A., Rizo, D., Calvo-Zaragoza, J.: Complete optical music recognition via agnostic transcription and machine translation. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) 16th International Conference on Document Analysis and Recognition, ICDAR. Lecture Notes in Computer Science, vol. 12823, pp. 661–675 (2021)
47. Coquenet, D., Chatelain, C., Paquet, T.: DAN: A Segmentation-free Document Attention Network for Handwritten Document Recognition. arXiv (2022)
48. Parr, T.: The Definitive ANTLR 4 Reference, 2nd edn. Pragmatic Bookshelf, Raleigh (2013)