International Journal on Document Analysis and Recognition (IJDAR)
https://doi.org/10.1007/s10032-022-00417-4

ORIGINAL PAPER

A holistic approach for image-to-graph: application to optical music recognition

Carlos Garrido-Munoz[1] · Antonio Rios-Vila[2] · Jorge Calvo-Zaragoza[2]

Received: 1 April 2022 / Revised: 19 June 2022 / Accepted: 1 September 2022

Abstract
A number of applications would benefit from neural approaches that are capable of generating graphs from images in an end-to-end fashion. One of these fields is optical music recognition (OMR), which focuses on the computational reading of music notation from document images. Given that music notation can be expressed as a graph, the aforementioned approach represents a promising solution for OMR. In this work, we propose a new neural architecture that retrieves a certain representation of a graph—identified by a specific order of its vertices—in an end-to-end manner. This architecture works by means of a double output: It sequentially predicts the possible categories of the vertices, along with the edges between each of their pairs. The experiments carried out prove the effectiveness of our proposal as regards retrieving graph structures from excerpts of handwritten musical notation. Our results also show that certain design decisions, such as the choice of graph representations, play a fundamental role in the performance of this approach.

Keywords Optical music recognition · Graph representation · Deep learning

1 Introduction

Graphs are mathematical structures that model pairwise relationships between objects in linked data sources. These data representations have, in the last few decades, become increasingly popular in the deep learning research field owing to their expressiveness and ability to represent complex associations among pieces of information—such as social network communities, molecule dynamics, or drug dealing interactions [28].

Many studies dealing with the extension of deep learning approaches for graphs have recently emerged. This growing research area focuses on the processing of graphs as an input structure. The existing literature reports that the main architecture employed is that of graphs neural networks (GNN), with the goal of performing tasks such as node classification, edge prediction, and graph classification [22]. Despite the increasing interest in this area, most of this literature understands graphs as a learnable input data source, but less attention is paid to their use as an output representation of a specific information source. That is, little literature explores the inference of graphs conditioned to a given input.

One interesting subject based on this concept is image-to-graph formulation. In this context, a neural model outputs a graph representing the connected elements of an input image by following a holistic or end-to-end formulation.[1] Several research fields could benefit from this approach, one of which is that of optical music recognition (OMR), which studies how to computationally read music notation from score images [4].

Work produced with the support of a 2021 Leonardo Grant for Researchers and Cultural Creators, BBVA Foundation. The Foundation takes no responsibility for the opinions, statements and contents of this project, which are entirely the responsibility of its authors. The second author is supported by grant ACIF/2021/356 from the "Programa I+D+i de la Generalitat Valenciana".

✉ Jorge Calvo-Zaragoza
jcalvo@dlsi.ua.es

Carlos Garrido-Munoz
carlos.garrido@ua.es

Antonio Rios-Vila
arios@dlsi.ua.es

[1] University Institute for Computing Research, University of Alicante, Alicante, Spain

[2] Department of Software and Computing Systems, University of Alicante, Alicante, Spain

[1] The term "holistic" will be used from here on, since "end-to-end" could be interpreted as those strategies that solve an entire task, regardless of whether or not they are divided into substages.

Published online: 16 September 2022

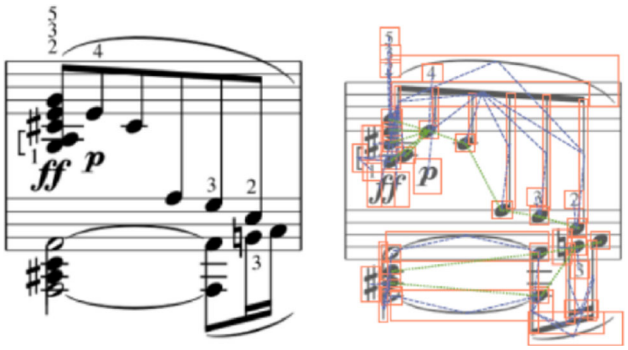


Fig. 1 Excerpt from the "Ludwig van Beethoven, Piano Sonata op. 2 no. 2, Largo apassionato" which illustrates some ubiquitous spatial relationships of music notation. On the left side, there is the original excerpt; on the right side, one possible interpretation of the music-notation structure as a graph: the primitives highlighted in red as nodes and a set of possible relationships, which include both syntactic ones (expressed in blue dashed lines) and reading order (green dashed lines), as edges

Music notation can intuitively be interpreted as a visual language whose primitives—the graphic elements of which a music symbol is composed—can be modeled using pairwise relationships [12]. This intuition motivates the understanding of music notation as a graph structure. One example of how music notation could be interpreted as a graph is found in Fig. 1. In this specific interpretation, music primitives—such as noteheads, flags, or stems—are considered the nodes of the graphs. Two types of relationships are found in this interpretation: syntactic ones—which group the primitives that conform to a music structure—and time-dependency ones, which indicate the order in which these nodes have to be read. Regardless of which relationships are applied, we observe that the graph structure is capable of conveying the structural information. Image-to-graph formulation is, therefore, of particular interest as regards developing holistic approaches for OMR.

In this paper, we propose a holistic image-to-graph neural methodology that is able to retrieve a graph representation from an input image. This method can be trained with a dataset that consists only of pairs of images and their corresponding graphs. These data samples do not need to provide any geometric information about the elements of the image—such as pixel-level regions or bounding box annotations. In other words, our method works with weakly annotated datasets.

Experiments were carried out in which our method was applied to excerpts of handwritten music notation from the MUSCIMA++ dataset. Our method successfully retrieved the graphs that represent the structures of the input images as regards predicting both the categories of the vertices and their pairwise relationships.

The remainder of the paper is organized as follows: Sect. 2 provides a description of the context and background of this work as regards both the OMR and graph retrieval fields, while Sect. 3 shows a formal definition of the problem and presents the proposed approach. Section 4 contains an explanation of the experimental setup of this work for the sake of reproducibility, and the experimental outcomes are reported and discussed in Sect. 5. Finally, the conclusions are shown in Sect. 6.

2 Background

This section shows the fundamental background of the two research fields that converge in this work: OMR and graph retrieval. In order to assess the contributions of this work to each field, both backgrounds are presented individually.

2.1 Optical music recognition

OMR has traditionally been divided into several stages that have been approached independently [17]. Fundamentally, there is a first step in which the music primitives—such as noteheads, beams, or accidentals—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 [11]. In the second stage, syntactic relationships among the primitives retrieved are inferred in order to recover the structure of the music score. These stages are solved by combining image processing techniques with heuristic strategies based on handcrafted rules [18]. Unfortunately, these solutions have proven to be largely insufficient [3].

Some authors have proposed solutions to OMR by following this aforementioned state-of-the-art holistic approach utilizing a serialized (sequential) version of music notation [1,20], in much the same way as has been attempted in other visual domains [27]. In the case of sheet music, this serialization represents a great simplification that still wastes most of the relational information. In the simplest cases, such as monophonic music or certain preceding notational systems—such as mensural or pneumatic notation—sequential holistic methods have shown promising results [5]. This approach, however, has never been extended in order to deal with any kind of sheet music because of current limitations related to the expressiveness of the output structure [4].

One promising holistic formulation for OMR requires the output domain to be more expressive than a sequence, as relevant information may otherwise be wasted. This problem is not specific to OMR, as this same challenge appears in other graphic domains, such as mathematical expression recogni-

tion [8] or road layout generation [10], to name but a few. In the case of music notation, a graph does represent a structure that fits naturally with the type of expressiveness that needs to be captured [12]. However, no attempts have been made to holistically retrieve a graph from an image, or more specifically, from a music score image.

2.2 Graph prediction

As stated above, there is a vast body of research focused on processing input graphs using GNNs. Nevertheless, the objective herein is not to process a graph, but rather to retrieve these structures as a whole conditioned to an input image.

The closest work to our goal is known as deep graph generation. This research field focuses on generative neural models for graphs. Two main approaches for graph generation currently exist: (i) one-shot generating methods [6,19] and (ii) sequential generation [15,25]. The former attempts to generate graphs in a single step, while the latter uses sequential neural networks, such as those with recurrent units, that iteratively generate the sequence of nodes and edges of which the graph is built. Recent work on this specific formulation has also been carried out, including a proposal for a general framework [24] and some practical approaches [13,16]. In all cases, however, the problem is approached by learning a generative model from a given dataset of graphs. However, in this work, we aim to learn a discriminative model that is conditioned to an image input.

Another research area that is worth highlighting is that of scene graphs. This field focuses on the extraction and construction of graphs that model semantic relationships between entities detected in images [9]. However, the whole construction process is based on the premise of having spatial information about the objects for their detection. This assumes that, in order to establish such semantic relationships, precise information about the position of the objects in the scene is available.

To the best of our knowledge, there is only one work that addresses discriminative graph retrieval conditioned by an input image [2]. However, this work focuses solely on the structure of the graph, as the method predicts only edges between unlabeled nodes. This is clearly not applicable to OMR, for which the nodes must have a category that represents a music-notation primitive.

3 Methodology

In this section, we describe the problem of holistic graph retrieval from images and the methodology adopted to deal with these tasks in the context of OMR.

3.1 Formulation

In its simplest definition, a graph is an abstract mathematical structure that represents pairwise relationships between elements—nodes or vertices—through connections—edges. As mentioned previously, the goal of this paper is to directly retrieve graphs from images that contain music-notation structures. The formal definition of the problem is shown as follows.

A graph can be defined as a pair (V, E) in which V represents the set of nodes and E the set of edges. Two nodes, $v_i, v_j \in V$ are connected if there is an edge, $e_{ij} = (v_i, v_j) \in E$. Let us use $\mathcal{G}$ to denote the space of all possible graphs. Given an image $\mathbf{x}$, we seek to retrieve the most probable graph $\hat{g}$:

$$\hat{g} = \arg\max_{g \in \mathcal{G}} P(g \mid \mathbf{x}). \tag{1}$$

A graph can, in computational terms, be modeled by employing a set of representations, each of which is identified by a specific order of its nodes. In such a case, the nodes are denoted by a sequence rather than a set. Let $\mathcal{R}(g)$ be the set of all representations of a graph $g \in \mathcal{G}$. Equation 1 can, therefore, be rewritten as:

$$\hat{g} = \arg\max_{g \in \mathcal{G}} P(g \mid \mathbf{x}) = \arg\max_{g \in \mathcal{G}} \sum_{r \in \mathcal{R}(g)} P(r \mid \mathbf{x}) \tag{2}$$

It is unfeasible to compute Eq. 2 in practice, even for small graphs, given that the possible representations of a graph grow as the factorial of the number of nodes $|V|!$. Instead, we approximate the solution by seeking the graph that indicates the most probable representation $\hat{r}$:

$$\hat{g} = \arg\max_{g \in \mathcal{G}} P(g \mid \mathbf{x}) \approx \arg\max_{g \in \mathcal{G}} \max_{r \in \mathcal{R}(g)} P(r \mid \mathbf{x}) \tag{3}$$

The objective of our approach is to solve Eq. 3 by means of deep neural networks.

3.2 Approach

Figure 2 provides a general overview of the proposed neural network.

In order to retrieve a graph representation, we conceptually divide the problem into two tasks: node and edge prediction. In the first task, the nodes of a specific representation of a graph are retrieved by means of a sequential prediction of node categories.

In the second task, the network predicts whether a pair of nodes are connected, which can be seen as predicting the adjacency matrix of that graph representation.

In this work, we propose a multi-output architecture that performs the two tasks simultaneously. With regard to the

goal of retrieving a sequence of symbols as nodes from an image, the first stage involves the implementation of an image-to-sequence neural model similar to image captioning approaches—in which the goal is to describe an image by means of natural language. This is done using an encoder–decoder architecture that retrieves image features by employing a convolutional neural network (CNN). These images are then fed to a recurrent neural network (RNN), which decodes a primitive and its representation for each timestep[2] until an end-of-graph—$< EOG >$ token—is attained. The embeddings generated at each timestep (RNN activations) are also used for node classification and, by grouping them by pairs, the model predicts whether there is an edge between them. The main challenge the network has to deal with is that of obtaining appropriate embeddings that represent both the music primitives and the features present in the image, as they are key elements for both node and edge prediction. We are, therefore, forcing the network to capture the richness of music-notation structures in a holistic manner.

3.3 Training

In our case, we require the neural network to be trained without any specific geometric information concerning where the vertices are located in the input image. Note that this might represent a competitive advantage when creating training sets, as it is much less expensive to annotate music scores in this way and the process could also be automated from existing encoded sheet music.

Let us assume that our training set consists of pairs $(\mathbf{x}, g)$. As occurs in Eq. 3, it will not be possible to directly train the network in order to optimize g given $\mathbf{x}$ and it is instead necessary to choose a specific representation from $\mathcal{R}(g)$ as a ground truth for the neural network. However, this decision should not be made arbitrarily in each case, and it is, rather, necessary to ensure some consistency in order to facilitate the learning process of the network. Let us denote as Φ a function that consistently maps a graph g onto one of its representations in the form $r = \Phi(\mathbf{x}, g)$. Note that Φ can also be conditioned to the input image x. This mapping transforms a set of nodes $\mathbf{v} = \{v_1, v_2, v_3, \dots\}$ into an ordered sequence $\mathbf{v}' = [v_1, v_2, v_3, \dots]$. The way in which function Φ operates will be described in the implementation details and will be analyzed empirically.

Transforming the graph representation into a sequence of nodes makes it possible to deal with the first task as an image-to-sequence problem. Let Σ be the set of possible music-notation primitives. It is then necessary to seek the sequence of nodes $\hat{\mathbf{v}}$:

$$\hat{\mathbf{v}} = \arg\max_{\mathbf{v} \in \Sigma^*} \mathrm{P}(\mathbf{v} \mid \mathbf{x}) \tag{4}$$

in which the goalin which the goal is to predict the sequence of tokens $\mathbf{v} = [v_1, v_2, \dots v_t]$. This can be trained by minimizing the sum of the negative log likelihood of the probability of the correct symbol v_t for each timestep t in the sequence v:

$$\mathcal{L}_{\text{nodes}} = -\sum_{t=1}^{N} \log \mathrm{P}(\hat{v}_t = v_t | t) \tag{5}$$

With regard to edges, this prediction task is addressed as a binary classification problem in which the binary cross-entropy (BCE) loss for every pair of nodes in the graph is minimized as:

$$\mathcal{L}_{\text{BCE}} = \sum_{e_{ij} \in A} e_{ij} \, \log(\hat{e}_{ij}) + (1 - e_{ij}) \, \log(1 - \hat{e}_{ij}) \tag{6}$$

where e_{ij} is the edge that pairs nodes (v_i, v_j). This corresponds to a specific cell in the adjacency matrix A of the given graph.

The sum of both losses can be seen as minimizing the following multi-loss function $\mathcal{L}_{\text{total}}$, given by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{nodes}} + \mathcal{L}_{\text{BCE}}. \tag{7}$$

This multi-loss is minimized jointly at training time via conventional backpropagation.

3.4 Prediction

The prediction process of this method closely follows the workflow described during training. First, an image $\mathbf{x}$ is fed into the CNN, which outputs a feature map. The RNN block then receives this feature map in order to condition its internal states and outputs one embedding vector for each timestep t. A sequence of vertices $\mathbf{v}$ and their corresponding edges are subsequently obtained.

In order to compute $\mathbf{v}$, we pass the embedding vector of each timestep through a multilayer perceptron (MLP) layer that performs a classification. The class that represents each node v_i is determined by following a *greedy* strategy in which the primitive with the highest activation in the output layer is chosen. The edge set is obtained using the combination of the embeddings retrieved from the RNN to predict the adjacency matrix of the graph in the edge prediction module. For each pair, the module predicts the probability of the existence of an edge between them.

Once the nodes and the edges have been retrieved, we have implicitly retrieved the most probable representation of a graph given the input image, which satisfies Eq. 3.

[2] A timestep can be understood as each instantiation of the RNN cells throughout the input/output sequences.

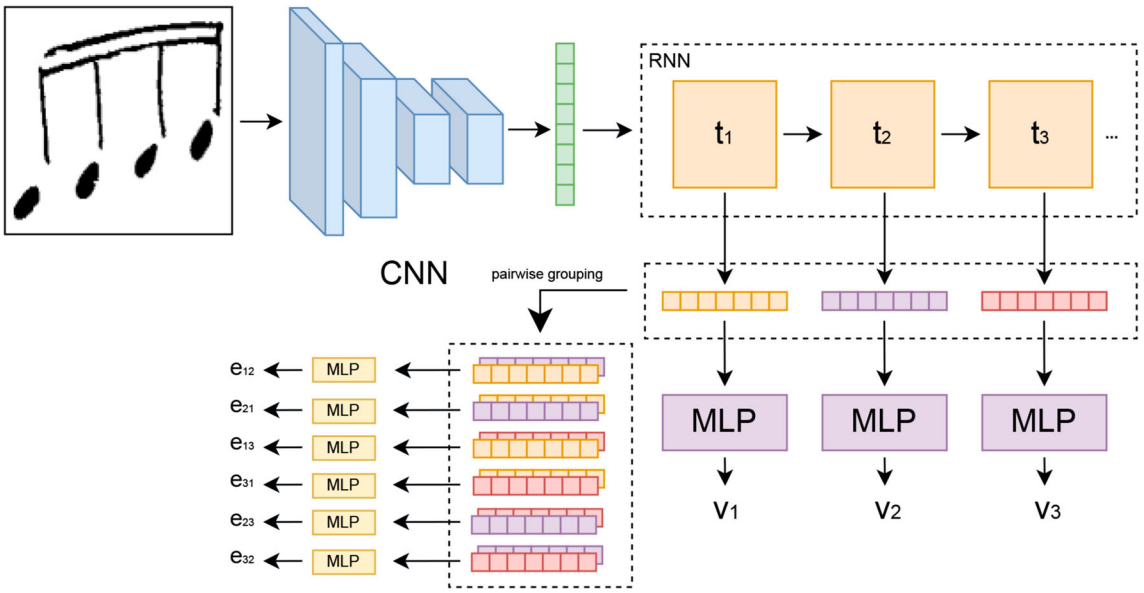


Fig. 2 General schema of the methodology. A CNN extracts the image representation, which is the input to the RNN decoder. This RNN is trained to produce each symbol for each timestep t alongside the node representation. Finally, we predict edges using these node representations per pairs

4 Experimental setup

In this section, we present the experiments carried out in order to evaluate the image-to-graph task in the context of OMR. We describe the data and metrics used, along with the implementation details for the models presented. Finally, we present the evaluation scenarios designed for this work.

4.1 Data

The experiments were carried out using the MUSCIMA++ dataset [12]. This dataset provides a great variety of handwritten music scores, which consist of musical symbols—primitives—and the annotated relationships between them, presented in the form of graphs.

For our task, we extracted the musical structures present in each score. We specifically considered only those examples in which the number of primitives was strictly greater than three. We assumed that musical structures with less than three symbols could be easily recognized and eventually obtained a dataset of independent annotated image-graphs pairs containing the nodes—music-notation primitives—and the relationships between them as an adjacency matrix. Figure 3 shows some examples of the dataset eventually obtained. For further details, Tables 1, 2 and Fig. 4 present some statistics regarding the symbols and the graphs considered.

4.2 Implementation details

We implement our neural approach with two specific neural networks with few differences between them. Both implementations consist of an encoder part that extracts image features and a decoder that retrieves the sequence of symbols and constructs the adjacency matrix. Using the technique employed for the encoder–decoder connection part as a basis, we denote them as (i) the non-attention model and (ii) the visual-attention-based model. The most important difference concerns the encoder–decoder connection part. The first model extracts the image features with 4 layers of alternating convolution and poolings with 64, 128, 256, and 256 filters, while the second outputs 512 filters in the last layer. The poolings are 2×2 in both models for each layer, with the exception of the last layer of the first model, which is 4x4 in order to reduce the dimensionality of the feature vector. All these layers use rectilinear uniform (ReLU) as the activation function.

Another difference between the models concerns how the features map obtained from the CNN is reshaped to be treated as a sequence. For the model without visual attention, the feature map is flattened in order to compute a 1024-dimensional feature vector. The visual-attention model instead uses the visual attention mechanism, as in the work of Xu et al. [23]. This mechanism is applied to the feature map, obtaining 64 vectors with 512 features each that are then fed to the decoder as an initial state.

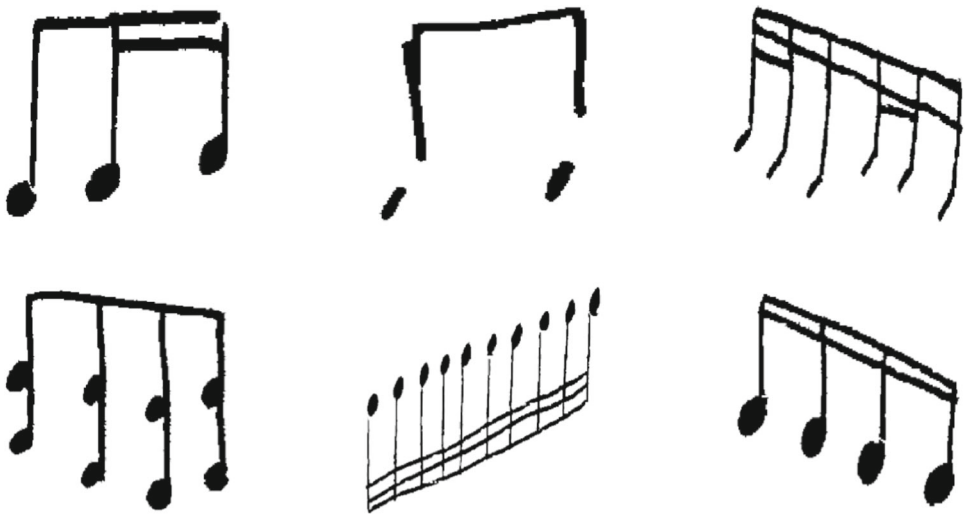


Fig. 3 Music notation structures extracted from the MUSCIMA++ dataset

Table 1 Statistics of the music-notation primitives (Σ) in the structures selected from the MUSCIMA++ dataset

Primitive	
Full notehead	14,868
Empty notehead	213
Stem	13,843
Beam	6575
8th flag	698
16th flag	487
Total	36,684

Table 2 Statistics regarding the ground truth data obtained once the MUSCIMA++ dataset had been filtered

Graphs	
No. of graphs	5047
No. of nodes	36,684
No. of edges	70,558
Avg. no. of nodes	7.3 ± 2.6
Avg. degree	1.8 ± 0.3

Avg. indicates the average $\pm$ the standard deviation

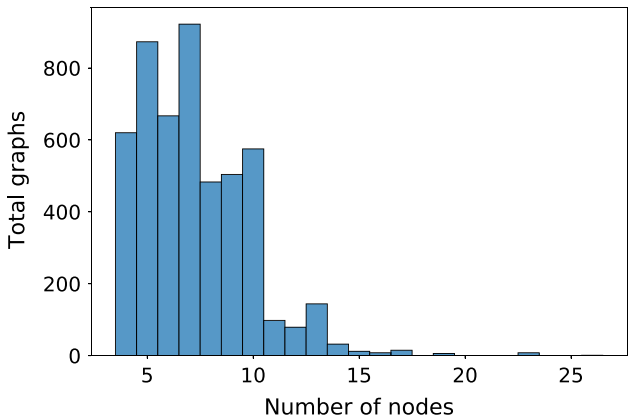


Fig. 4 Distribution of the graph size in the dataset considered. The range of nodes per graph goes from 4 to 26

With regard to the decoder, both models use a long short-term memory (LSTM) layer in the recurrent part to extract each symbol for each timestep, with 1024 hidden units in (i) and 512 in (ii). In the last part of the decoder, both models have a linear layer with as many units as the size of the vocabulary Σ and a softmax activation for the node classification. For the adjacency matrix reconstruction, we concatenate the hidden states by pairs corresponding to the node embeddings generated by the LSTM. For this binary classification task, we use a two-layered MLP with 256 units in the hidden layer plus a sigmoid activation function to predict the probability of a link between two nodes from their embeddings. At prediction time, we establish a decision threshold of 0.5 to determine the existence of an edge. Concerning the size of the models, the neural network without visual attention and the neural network with visual attention comprise 9.5M and 6.5M trainable parameters, respectively.

We also apply basic data augmentation techniques, such as rotations up to 45 degrees, padding, and vertical and horizontal flips, to the input images.

With regard to the learning process, we train both models for a maximum of 200 epochs. A batch size of 64 is used and an Adam optimizer [14] is applied with a learning rate of 0.001. The loss function used in these models is defined in Eq. 7. A doubly stochastic attention term from [23] is also included for the model with attention.

Finally, we use two approaches for the node ordering given by the function Φ: (i) sorting nodes according to the image topology, ordered from left to right and top to bottom (this approach is, hereafter, denoted as "topology"), and (ii) sort-

ing the nodes alphabetically, according to the vocabulary Σ, which we denote as "alphabetical."

4.3 Metrics

Computing a graph edit distance is known to be an NP-hard problem [21]. This fact leads us to discard a single metric that encompasses all the differences between the predicted and ground truth graphs. Alternative metrics that could correlate with the performance are, therefore, required in this task. Here, we measure two factors for the graph prediction task in hand: (i) the accuracy of the node sequences predicted and (ii) how accurately the edges between these nodes are predicted. For the former, we consider the symbol accuracy (Acc), whereas the latter is measured using the F_1.

4.3.1 Symbol accuracy

Symbol accuracy is a metric based on the symbol error rate (SER). This is a common metric in tasks related to sequences, such as handwritten text recognition.[3] This value measures the error of the model in the recognition task and correlates to the effort that a user would have to make in order to manually correct the results. Let H be the predicted sequence and R be the reference sequence. The SER is computed by measuring the edit distance between H and R normalized by the length of R. When measuring accuracy, it is possible to resort to symbol accuracy (Acc), which is computed as $1 - \text{SER}$. This is an analogous metric to the Word Accuracy metric. We resort to this metric for the sake of readability so that both metrics (F1 and Acc) stick to "the higher the better."

4.3.2 F_1 metric

The F_1 metric is defined as the harmonic mean of precision (P) and recall (R):

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}} \tag{9}$$

where TP, FP, and FN are the true positives, false positives, and false negatives, respectively. As we wish to measure the accuracy in the predicted edges, we use actual edge connections as the positive class to compute the F_1.

[3] In this field, the metric is called word error rate or character error rate depending on the unit considered. Here, we employ the generic term "symbol," as there is no clear analogy between music-notation primitives and words/characters from text.

4.4 Evaluation scenarios

In this work, we have considered the following scenarios for evaluation:

1. *Function* **8 choice**. As explained above, we must consider a function Φ that consistently converts a graph into a specific representation in which its nodes are expressed as a sequence. We shall now analyze how the choice of Φ affects the performance of the models.
2. *Visual-attention*. The main differences between the two neural models implemented concern the visual-attention mechanism. In this work, we study the impact of using this mechanism on the final performance.
3. *Scarcity of data*. We study the behavior of this proposed formulation under data limitation conditions. This could be of special relevance in the case of handwritten scores, for which it is not easy to obtain labeled data. In order to simulate different conditions, we restrict the data used for training to the following percentage: 5 %, 25 %, 50 %, and 100 %.

5 Results

We shall now present the results obtained for the different evaluation scenarios described in the previous section. It must be noted that all the reported scores come from strategies and scenarios proposed in this paper, as we only focus on the image-to-graph retrieval formulation. Unfortunately, our work cannot be compared with the state of the art since they have different endpoints: in (some) previous works, a sequence of music-notation tokens representing the music shown in the image is obtained. However, our work—designed precisely for music scores for which a sequence is not enough—ends with a graph on which a final step needs to be performed to recover the semantics of musical notation. There is currently no known strategy to perform this last step "from graph to music," so the results of the state of the art are still incompatible with ours.

Table 3 shows the average results in the test set, following a fivefold cross-validation strategy. For a better global comparison, a graphical representation of these same results is provided in Fig. 5.

First note that the proposed approaches are successful in carrying out their tasks, obtaining satisfactory results in almost any scenario. Note also that the symbol accuracy and F_1 score are closely related. The correlation of these metrics, calculated on the basis of the results obtained, is specifically 0.998, as shown in Fig. 6. This high correlation denotes that, although these tasks are measured independently, their performances are linked. That is, if the model

Table 3 Average test set results for the different evaluation scenarios from the fivefold CV process. Best results for each data availability percentage are highlighted

Model		% of training data							
Φ choice	Attention	5		25		50		100	
		Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1
Topology	✓	59.0	53.7	64.2	55.8	69.0	63.1	77.0	74.1
	✗	53.7	53.2	64.3	64.1	73.6	71.5	77.9	76.4
Alphabetical	✓	**62.3**	**56.1**	**88.7**	**80.8**	**92.0**	**86.2**	**95.4**	**91.6**
	✗	46.7	42.8	82.7	78.4	80.2	77.4	87.1	85.8

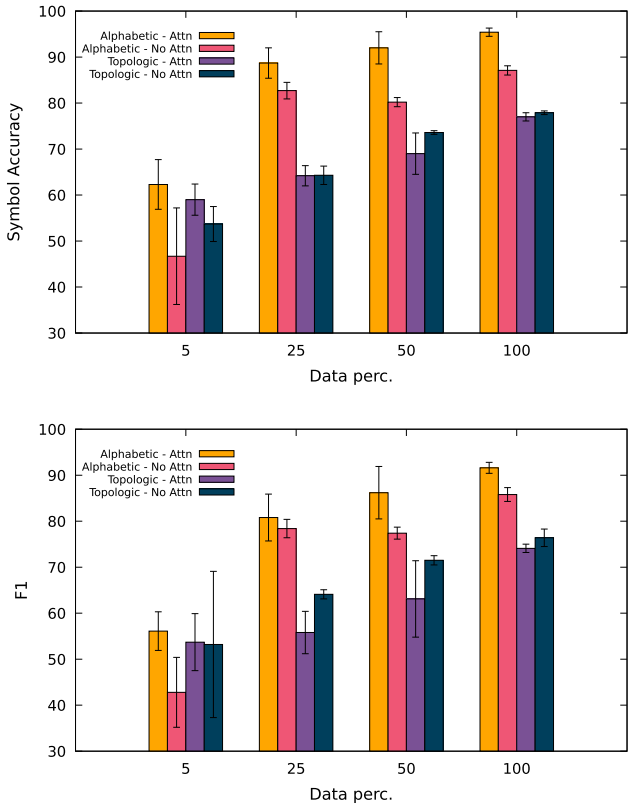


Fig. 5 Barplot visualization (mean and standard deviation) of the results presented in Table 3

is able to correctly classify the symbols, it is also able to correctly reconstruct the edges of the graph with high accuracy. This fact empirically demonstrates that the proposed loss function in Sect. 3.1—which joins node classification and edge reconstruction—enables the network to learn to classify nodes and reconstruct their edges equally, with no bias toward either.

With regard to the choice of the Φ function, there are significant differences in the performance. Choosing the Φ that sorts the symbols alphabetically specifically increases both symbol accuracy and F_1 by approximately 20%. The reason for this difference might be the sensitivity to image variations of the topological order and its impact during training. When presented with two similar images that have displaced elements, it is likely that their graph representations will be topologically different, as the nodes appear in a different order and this alters the adjacency matrix. This is problematic in neural network training, since similar inputs that have completely different outputs—usually referred to as noisy data—lead to convergence issues during the optimization process and decrease the robustness of the model. In our case, there are music primitives with this problem, as shapes tend to be similar with slight variations. When using the topological sorting as the Φ function, it is likely to be introducing this kind of phenomena, which downgrades the network performance. In addition, this case is likely to be aggravated when data augmentation is applied. The model improves significantly when a Φ function that sorts the graph nodes independently to the image layout is used. This is probably because similar samples are very likely to obtain a similar graph representation, which facilitates the convergence of the neural network during training.

With regard to the use of the visual-attention mechanism, there were no significant differences in most scenarios. Nevertheless, in scenarios in which data scarcity becomes more prominent, we find that models that use the visual attention mechanism clearly outperform models that do not. This is because attention mechanisms are able to obtain filtered information from specific feature representations, which enable the model to converge with fewer training samples.

In relation to data scarcity, the performance is also strongly linked to the choice of Φ. With the proper ordering function, the model behaves well even when data scarcity is extreme, as in the case of only 5% of the available data. As shown in Table 3, the models with alphabetical sorting and only 25 % of the available data perform better than the best model with topological sorting using the complete dataset.

In order to illustrate the performance, Fig. 7 depicts two predictions in the test set from the best model from Table 3. As shown in Fig. 7a, the model labels all the nodes correctly and successfully retrieves their corresponding edges. Nevertheless, there are some errors in Fig. 7b: one node is wrongly labeled—it predicts an 8th flag rather than a beam, which can be considered a reasonable mistake from the graphics recognition point of view, and the edge between the bottom-left notehead and the top beam is missed. Note that, since

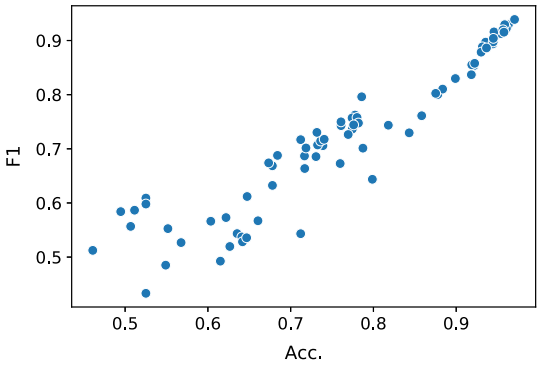
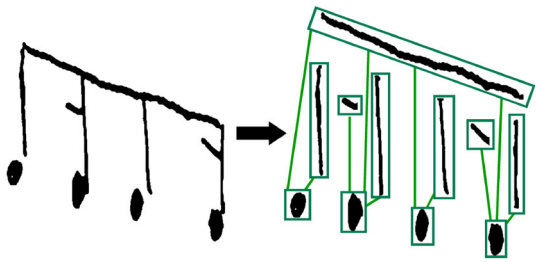
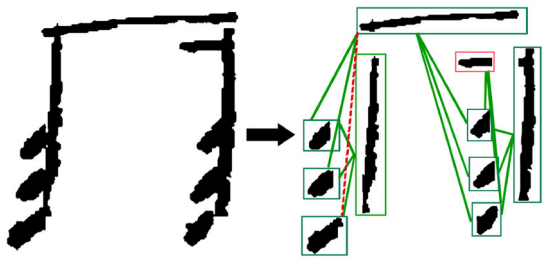


Fig. 6 Scatter plot correlating the results in the test set obtained as regards symbol accuracy (Acc.) and F_1 for each fold of the 5-CV for all models



(a) Example of a correct prediction in which nodes and edges are retrieved successfully.



(b) Example of prediction with errors. The red box represents the fact that the node has been wrongly labeled, while the dashed red line indicates that the edge between the corresponding nodes has not been predicted.

Fig. 7 Examples of test set predictions. Images on the left represent the input image given to the model, while those on the right represent the graphs retrieved by our best model. Note that the bounding boxes depicted are simply a visual aid, as our approach does not retrieve them specifically. Green indicates correct predictions and red indicates errors, which are described in each example

the graphs are inferred without specific indications about the positions of the nodes in the input image, these visualizations are simply a possible interpretation of the relationships that the neural network could have established.

Finally, Fig. 8 shows the obtained attention matrices during the inference of a selected test sample. Although this is just a selected example within the entire dataset, we believe that it is representative of the behavior of the model. Note that while in some timesteps, the token matches with the positions of high activation of the attention matrix (e.g., first stem), in other cases the attention matrix is quite sparse (e.g., beam).

6 Conclusion

In this paper, we propose a new holistic image-to-graph model with which to retrieve a graph representation of the elements present in an input image. We propose a formulation based on sequential node classification and pairwise edge reconstruction. This formulation has been devised for use in the optical music recognition (OMR) field.

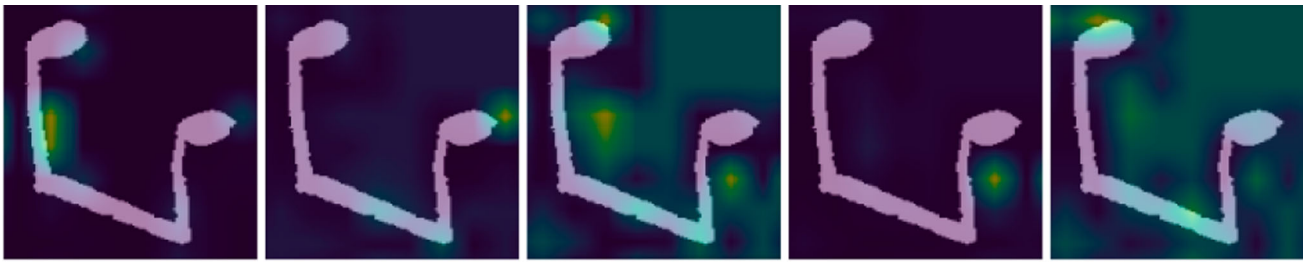
Several experimental scenarios are proposed in our experimentation, such as the suitability of attention mechanisms for the recurrent model, the availability of data, and the means of mapping the set of nodes in the ground truth into an ordered sequence that sticks to a consistent graph representation. These scenarios were tested in the MUSCIMA++ dataset, which contains handwritten music fragments annotated in the form of a graph.

The results showed that the proposed methodology and formulation are successful in solving the image-to-graph task within the specific context of music-notation structures. Our method is able to retrieve the graph from the image without the need for spatial information regarding the elements in the input source.

This paper makes it possible to draw certain interesting conclusions: First, the selection of the graph representation according to the sequential ordering of its nodes would appear to have a strong influence on the performance of the model; second, the model needs to extract edges for every single pair of nodes. This computation can be expensive in large-graph scenarios and require a considerable amount of time to be trained.

The primary objective of our future work is to retrieve graphs from full pages, since the state-of-the-art models are limited as regards the task of retrieving the sequence representation of these document types. This will open up an interesting scenario for the use of graph representation in order to approach the complete OMR problem. However, there is still an additional step to develop in order to compare our method with the state-of-the-art OMR systems, which is to retrieve complete music-notation semantics from the graph. This task is not trivial, as simple rule-based systems are too strict to develop a robust solution to recover these semantics. This will require further efforts in future works.

Furthermore, some drawbacks found in this work must also be addressed, such as creating loss functions or neural network architectures that do not depend on node ordering—which is analogous to that which the DEtection TRansformer [7] or Deep Sets [26] models do when predicting unordered sets—or new methodologies that approach edge retrieval more efficiently. This probably requires estimating an efficient edit distance between graphs representing musical structures, which is an interesting challenge for the future. In addition, we plan to test our approach in other document anal-



(a) stem **(b)** full notehead **(c)** beam **(d)** stem **(e)** full notehead

Fig. 8 Visualization of the attention weights for each timestep during inference in a selected test sample. Subcaptions indicate the node class predicted in each timestep

ysis tasks that might benefit from a holistic image-to-graph formulation.

Author Contributions C.G.-M. and A.R.-V wrote the main manuscript. C.G.-M. performed the experiments. A.R.-V. prepared the figures. J.C.-Z. designed the conceptual methodology. All authors reviewed the manuscript.

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

Declarations

Conflict of interest The authors declare that there is no conflict of interest.

References

1. Baró, A., Riba, P., Calvo-Zaragoza, J., Fornés, A.: From optical music recognition to handwritten music recognition: a baseline. Pattern Recognit. Lett. **123**, 1–8 (2019)
2. Belli, D., Kipf, T.: Image-conditioned graph generation for road network extraction. CoRR (2019)
3. Byrd, D., Simonsen, J.G.: Towards a standard testbed for optical music recognition: definitions, metrics, and page images. J. New Music Res. **44**(3), 169–195 (2015)
4. Calvo-Zaragoza, J., Hajic Jr., J., Pacha, A.: Understanding optical music recognition. ACM Comput. Surv. **53**(4), 77:1–77:35 (2020)
5. 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)
6. Cao, N.D., Kipf, T.: Molgan: an implicit generative model for small molecular graphs. CoRR (2018)
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision - ECCV 2020, pp. 213–229. Springer International Publishing, Cham (2020)
8. Chan, K.F., Yeung, D.Y.: Mathematical expression recognition: a survey. Int. J. Doc. Anal. Recognit. **3**(1), 3–15 (2000)
9. Chang, X., Ren, P., Xu, P., Li, Z., Chen, X., Hauptmann, A.: Scene graphs: a survey of generations and applications. CoRR (2021)
10. Chu, H., Li, D., Acuna, D., Kar, A., Shugrina, M., Wei, X., Liu, M.Y., Torralba, A., Fidler, S.: Neural turtle graphics for modeling city road layouts. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 4522–4530 (2019)
11. 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)
12. Hajič, Jan: 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. Graduate School of Engineering, Osaka Prefecture University, IEEE Computer Society, New York, USA, Dept. of Computer Science and Intelligent Systems (2017)
13. Jonas, E.: Deep imitation learning for molecular inverse problems. In: H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, R. Garnett (Eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc. (2019)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Y. Bengio, Y. LeCun (eds.) 3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
15. Li, Y., Vinyals, O., Dyer, C., Pascanu, R., Battaglia, P.W.: Learning deep generative models of graphs. CoRR (2018)
16. Li, Y., Zhang, L.R., ming Liu, Z.: Multi-objective de novo drug design with conditional graph generative model. J. Cheminform. **10**, 1–24 (2018)
17. 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. Multimed. Inf. Retr. **1**(3), 173–190 (2012)
18. Rossant, F., Bloch, I.: Robust and adaptive OMR system including fuzzy modeling, fusion of musical rules, and possible error detection. EURASIP J. Adv. Sig. Process. **2007**, 1–25 (2007)
19. Simonovsky, M., Komodakis, N.: Graphvae: Towards generation of small graphs using variational autoencoders. CoRR (2018)
20. Torras, P., Baró, A., Kang, L., Fornés, A.: On the integration of language models into sequence to sequence architectures for handwritten music recognition. In: J.H. Lee, A. Lerch, Z. Duan, J. Nam,

P. Rao, P. van Kranenburg, A. Srinivasamurthy (Eds.) Proceedings of the 22nd international society for music information retrieval conference, ISMIR 2021, Online, November 7-12, 2021, pp. 690–696 (2021)
21. Vento, M.: A long trip in the charming world of graphs for pattern recognition. Pattern Recognit. **48**(2), 291–301 (2015)
22. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. CoRR (2019)
23. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: F. Bach, D. Blei (eds.) proceedings of the 32nd international conference on machine learning, proceedings of machine learning research, vol. 37, pp. 2048–2057. PMLR, Lille, France (2015)
24. Yang, C., Zhuang, P., Shi, W., Luu, A., Li, P.: Conditional structure generation through graph variational generative adversarial nets. In: H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, R. Garnett (eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc. (2019)
25. You, J., Ying, R., Ren, X., Hamilton, W.L., Leskovec, J.: Graphrnn: A deep generative model for graphs. CoRR (2018)
26. Zaheer, M., Kottur, S., Ravanbakhsh, S., Póczos, B., Salakhutdinov, R., Smola, A.J.: Deep sets. CoRR (2017)
27. 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 Recognit. **71**, 196–206 (2017)
28. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: a review of methods and applications. AI Open **1**, 57–81 (2020)