MuNG Studio: Annotation Tool for Music Notation Graph

Jiří Mayer
Institute of Formal and Applied Linguistics
Charles University
Prague, Czech Republic
mayer@ufal.mff.cuni.cz

Filip Jebavý
Moravian Library
Brno, Czech Republic
filip.jebavy@mzk.cz

Markéta Herzánová Vlková
Moravian Library
Brno, Czech Republic
marketa.vlkova@mzk.cz

Martina Dvořáková
Moravian Library
Brno, Czech Republic
martina.dvorakova@mzk.cz

Pavel Pecina
Institute of Formal and Applied Linguistics
Charles University
Prague, Czech Republic
pecina@ufal.mff.cuni.cz

Jan Hajič Jr.
Institute of Formal and Applied Linguistics
Charles University
Prague, Czech Republic
hajicj@ufal.mff.cuni.cz

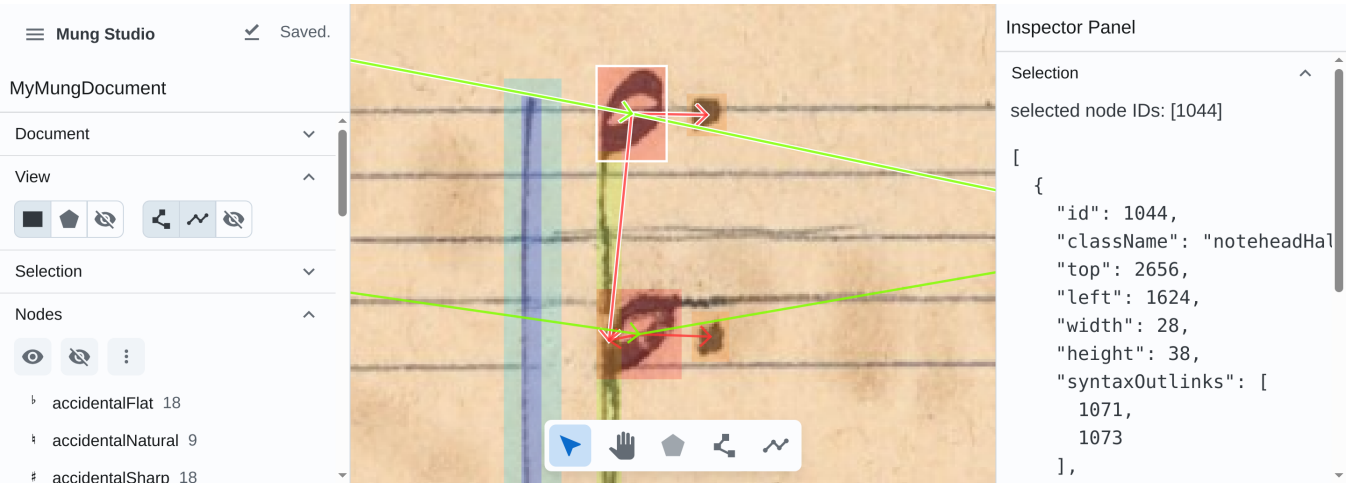


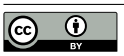
Figure 1: MuNG Studio user interface with a notehead selected (white frame), syntax links in red, precedence links in green.

Abstract

This paper introduces MuNG Studio, a new annotation tool for the Music Notation Graph (MuNG) format. MuNG is a high-detail graphical annotation format designed for Optical Music Recognition tasks, originally proposed for the MUSCIMA++ dataset in 2017. MUSCIMA++ had a significant impact on the OMR community; however, most subsequent datasets made little use of the full MuNG format. This was likely due to the lack of user-friendly tools supporting the format. The new MuNG Studio seeks to provide an easy-to-install web-based viewer and editor for the MuNG format with the goal of expanding and supporting the now growing ecosystem around MuNG.

CCS Concepts

• **Applied computing** → **Graphics recognition and interpretation**; • **Information systems** → **Digital libraries and archives**.



This work is licensed under a Creative Commons Attribution 4.0 International License.
DLfM 2025, Seoul, Republic of Korea
© 2025 Copyright held by the owner/author(s).
ACM ISBN /25/09
https://doi.org/10.1145/3748336.3748379

Keywords

Optical Music Recognition, Annotation Tool, Music Notation Graph

ACM Reference Format:
Jiří Mayer, Filip Jebavý, Markéta Herzánová Vlková, Martina Dvořáková, Pavel Pecina, and Jan Hajič Jr.. 2025. MuNG Studio: Annotation Tool for Music Notation Graph. In *12th International Conference on Digital Libraries for Musicology (DLfM 2025), September 26, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3748336.3748379

1 Introduction

Optical Music Recognition (OMR) is a computer vision task in which the goal is to automatically transcribe a document containing music notation into a machine-readable format [4]. Currently, there are two prevalent approaches to tackling this problem, which are object detection pipelines and end-to-end pipelines.

Object detection pipelines have been used since before the advent of deep learning in computer vision; however, the detection techniques used then consisted of hand-crafted heuristics and visual grammars [1, 6, 8, 27]. With the success of convolutional neural networks, a new wave of object detection pipelines was proposed after 2014 [15, 20, 24–26], setting the environment from which the MUSCIMA++ dataset emerged in 2017 [16]. The authors refined

the previous rule-based approaches into a machine-learnable annotation framework, forming the Music Notation Graph (MuNG) format (Figure 2).

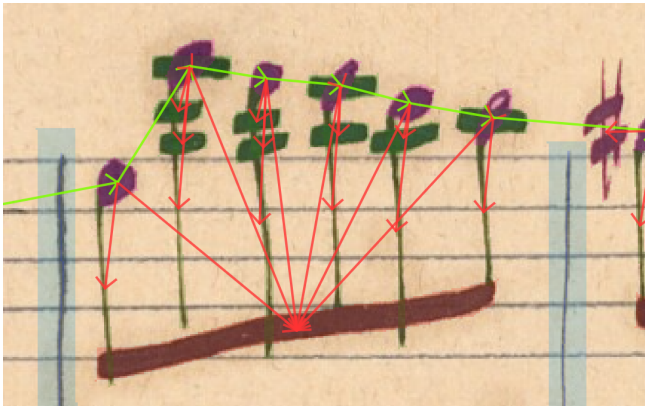


Figure 2: The Music Notation Graph (MuNG) captures musical glyphs at the pixel-level and relationships between them.

Concurrently, end-to-end pipelines started emerging, utilizing neural network architectures that could produce a sequence of tokens from an input image, skipping all preprocessing, binarization, staff removal or object detection steps [2, 5, 22, 28]. Such larger models require far more data than the previously described object detectors, which made it impractical to use the MUSCIMA++ dataset (and similar ones) for training. But the advantage the MuNG format has over other object detection formats is that it allows for deterministic extraction of musical content from the MuNG annotation, which makes the MUSCIMA++ dataset still viable as an evaluation dataset even for end-to-end models. However, the true power of MuNG comes from its low-level, complete, and visual representation of music. This allowed MuNG data to be used as input for data synthesis efforts, such as Mashcima and the recent Smashcima[1] systems [21]. Although MuNG annotation is expensive to produce, the utility of such well-annotated data is inexhaustible.

The MUSCIMA++ dataset that introduced the MuNG format [16] is built on a subset of the CVC-MUSCIMA dataset [8]. The CVC-MUSCIMA dataset provides 1000 binarized images of handwritten music notation, intended for staffline removal and writer identification. It consists of 20 one-page scores, each transcribed by 50 writers for a total of 1000 handwritten pages. The MUSCIMA++ dataset annotates a subset of 140 pages in the MuNG format, capturing segmentation masks of all symbols and the intricacies of their relationships, encoded in the notation graph.

The annotation process for MUSCIMA++ was performed with a custom tool named MUSCIMarker[2] [14], developed as a desktop application in Python 2.7 using the Kivy GUI library which is now obsolete technology. It is therefore very difficult to launch the tool today. Its development also gave rise to the `mung` Python package, which lets the user consume and manipulate the MuNG format from the Python environment. This package is the cornerstone of the MuNG tooling ecosystem.

[1] https://github.com/OMR-Research/Smashcima
[2] https://github.com/OMR-Research/MUSCIMarker

The MuNG authors also designed an ontology that defines a set of class names to be assigned to music notation glyphs, as well as the meaning of syntactic links between them. It is formalized in a grammar, stating which links are allowed and what cardinalities they have. This ontology (the class names) was in 2019 aligned with the SMuFL[3] standard, and the updated MUSCIMA++ dataset was released as version 2.0.[4] The alignment was motivated by a harmonization with the DeepScoresV2 dataset – a collection of born-digital typeset music images, very similar to MUSCIMA++, but unfortunately lacking the graph links [30].

In 2021, the DoReMi dataset was released [29], which, like DeepScores, contains born-digital typeset music images, but also contains proper and complete MuNG annotations. The MuNG format also lends itself to the application of graph neural networks [3, 7, 9]. This avenue of research has led to the creation of the Musigraph dataset in 2022, which is also based on the MuNG format [3].

All of these MuNG datasets, other than MUSCIMA++, contain born-digital typeset images, synthetically created by a computer. This is no doubt because annotating authentic handwritten music images is expensive, but the lack of an easy-to-use annotation tool has not helped either. The currently ongoing OmniOMR project, launched in 2023, aims to build the next handwritten music dataset with MuNG as the ground truth format [17].

In this paper, our aim is to revitalize the MuNG ecosystem by releasing **MuNG Studio**, the successor to MUSCIMarker. It was developed to solve practical problems identified during the annotation effort within the OmniOMR project. MuNG Studio is open-source and available on GitHub[5], with a link to a ready-to-use deployment. Currently, it supports viewing and link editing. We plan on adding object and mask editing in the near future.

2 Music Notation Graph (MuNG)

Several formats are used to represent music and music notation in machine-readable form. MIDI and MusicXML (or MEI) would be the two most widely known outside the OMR research community. The existence of these two very different formats for seemingly the same purpose points to a crucial fact. There are two possible goals for doing music recognition: replayability and reprintability [4].

Replayability aims at reconstructing music in the audio domain, usually focusing on the extraction of played notes. A note is characterized by three properties: pitch, onset, and duration. This is the exact data that the MIDI format encodes. However, if we want to query information about the physical music document, modify it, re-flow, or re-print it, MIDI discards too much information. The MusicXML format reconciles this by encoding music notation – the graphical system for encoding music on a sheet of paper.

In the domain of reprintability, there are several annotation formats as well. They could be characterized by how much of the original image they retain. MusicXML is at the highest level of abstraction, discarding all low-level graphical information included in the image. COCO and other object detection formats encode the position and class labels of symbols, either as bounding boxes, polygons, or full pixel-masks. This is much more fine-grained and useful

[3] https://www.smufl.org/
[4] https://github.com/OMR-Research/muscima-pp/releases/tag/v2.0
[5] https://github.com/OmniOMR/mung-studio

Table 1: Feature comparison of annotation tools applicable to produce MuNG annotations.

	MuNG Studio	MUSCIMarker	Label Studio	Labelbox	CVAT	OMRAT
Runs smoothly at 2K objects	✓	✓	✗	✗	✗	✓
Focused on music notation	✓	✓	✗	✗	✗	✓
MuNG format support	✓	✓	✗	✗	✗	✗
Link annotations	✓	✓	✗	✓	✓*	✓
Bitmap masks	✓	✓	✓	✗	✓	✗
Polygon masks	✓	✗	✓	✓	✓	✗
Backend with ML models	✗	✗	✓	✓	✓	✓
Open source	✓	✓	✓	✗	✓	✗

* Via a hack with geometric line annotations

for low-level OMR tasks, but it lacks the high-level information about the music.

The MuNG format was designed to sit in the middle, encoding objects in much the same way as the COCO format, but also encoding links between them, forming the notation graph. It captures the contextual nature of music notation, allowing deterministic inference of the abstract music semantics (both for reprintability and replayability). This inference is implemented in the `mung` Python package (though it currently only supports export to MIDI).

Before the introduction of MuNG, extracting musical content from object detections was done using procedural methods built on context-free grammars [1, 6, 27]. MuNG took inspiration from dependency trees in natural language processing, which shifts the parsing process away from procedural towards a declarative description. This allows the resulting notation graph to be easily encoded as gold data (which was necessary for the MUSCIMA++ dataset), as well as construing the parsing as a machine-learnable link prediction task called *notation assembly* [23].

MuNG links (graph edges) are of two types: syntax links and precedence links. Syntax links encode the dependencies between notation objects that allow to deterministically infer music semantics (decode notes). They encode that "this stem" belongs to "this notehead", forming a quarter note. The precedence links encode how the notes follow each other to disambiguate complex polyphonic music where the rule "from left to right" does not suffice.

3 Existing Tools and MuNG

In this section, we compare MuNG Studio with existing alternatives for annotation of sheet music: three general-purpose solutions (Label Studio, Labelbox, and CVAT,[6]) and two music-specific tools (MUSCIMarker, OMRAT[7]). The most relevant features of these tools are summarized and compared in Table 1.

Label Studio lacks any support for annotating inter-object links, which is a distinctive feature of the MuNG format. CVAT allows repurposing geometric lines that would then be interpreted as links during postprocessing by using geometric lines and later interpreting them as links during conversion to MuNG, but this would introduce hard-to-revise data quality issues. Labelbox is the only one that supports explicit links between objects.

[6]https://labelstud.io/, https://labelbox.com/, https://www.cvat.ai/
[7]https://music-put.lab.dariah.pl/en/products

MUSCIMarker, despite relying on outdated technologies, could be revitalized and updated. However, its reliance on binarized and pre-segmented stafflines (which were available in CVC-MUSCIMA) vastly limits its applicability for a colored image annotation process.

OMRAT [13] introduced in 2024 is the first music annotation tool after MUSCIMarker that supports the annotation scheme with syntax links between objects. The OMRAT paper [13] references the idea of graph annotation but defines its own ontology, which is closely coupled with the MEI format. This is a divergence from MuNG and its history and alignment with SMuFL. It also lacks mask annotations (only bounding boxes are supported).

For completeness, we would like to mention the MuRET tool [18]. It is also a music-specific annotation tool, but focuses on mensural music notation. As such, it is not relevant for Common Western Music Notation, but it is an important tool in its domain.

3.1 Performance Issues

A very important requirement for the annotation tool when using the MuNG format is its performance and the speed with which it runs with many objects. A typical page of piano music, with 6 systems (lines) and 50 measures of music, contains around 500 noteheads and 1500 notation objects in total plus a similar number of syntax links (see Figure 3). For instance, Labelbox with so many objects becomes unresponsive. OMRAT performs well enough, but lacks the SMuFL-aligned ontology and segmentation masks.

Performance in the web browser is usually lost in the large number of DOM elements (their hit-testing and styling) and in the rendering of the large number of object masks (their rasterization). MuNG Studio currently uses SVG elements to render the scene; however, we are building a WebGL renderer to secure this performance across devices. This approach has already been used, e.g., in the Figma vector editor [31].

4 MuNG Studio

To fill a gap among existing annotation tools, we introduce MuNG Studio: a modern, web-based, lightweight annotation tool specifically for the MuNG format.

4.1 Administrator Experience

The administrator of an annotation process is responsible for setting up the annotation tools and distributing the work to annotators and reviewers. The installation of MuNG Studio is simple because

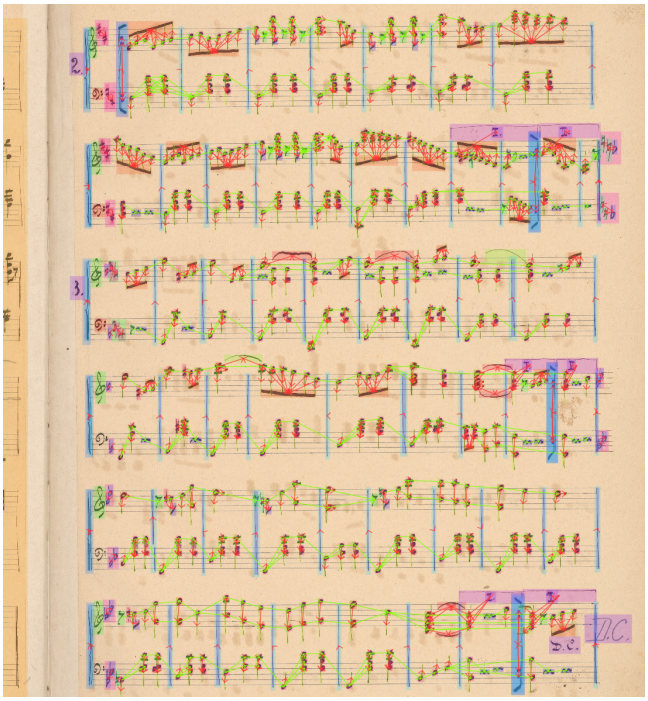



Figure 3: One MuNG document describes a single page of music notation. This one contains 1450 nodes and 1558 links, which corresponds to 538 noteheads.

it is built as a single-page web application and so can be hosted by any common web server. For work distribution, we took inspiration from the Open Score project [10–12] and left the work assignment external to MuNG Studio. The administrator sets up one shared Google Sheet, where annotators assign themselves to documents for annotation and review. However, to avoid the headache of distributing annotated documents via a cloud storage or email, MuNG Studio provides a "`Simple PHP Backend`". It is a single PHP file that can be installed on a web server by copying the file over and configuring it. It exposes one folder to all annotators and this folder contains a list of documents. The backend audits all read and write operations and performs daily backups for each document separately, so accidents can be recovered from.

4.2 User Interface

The design of the editor view is inspired by Figma, Sketch, VS Code, or MuseScore, where in the center there is the scene with the document, the left side is taken up by the overview panel, and the right side contains the inspector panel [19] (Figure 1).

The overview panel displays global document information, such as the document name, main menu, display settings, and the list of all classes and their visibility. The content of this panel does not change based on the selected node or the active editor tool. The inspector panel displays details about the currently selected nodes and instructions on how to use the active editor tool. In the lower part of the screen, in the middle, there is the toolbar which lets the user choose a specialized tool for various annotation actions.

The scene view in the middle shows the music document with annotations overlayed on top. Node annotations can be selected by clicking on them or selecting a group with a selection rectangle by clicking and dragging. The selection logic is then reused in the link annotation tools, both to simplify the MuNG Studio codebase, but mainly to build strong and consistent mental models about its behavior for the user [19].

4.3 Integrated Environment

During the annotation process within the OmniOMR project, we learned that creating node annotations and link annotations in two separate steps is very brittle. If a node annotation mistake is spotted later during link annotation, it is very expensive (if not impossible) to go back and fix it. Especially if the two steps are done in separate tools. Moreover, running our ontology validation scripts from the `mung` Python package after the annotation is just as inconvenient. It is better to give this feedback to the annotator immediately.

For this reason, we plan on supporting node editing in MuNG Studio as well. We also already embedded the Pyodide runtime environment, so that it is capable of running our Python validation scripts directly in the web browser. The user interface for this is not yet present, but we want to take inspiration from modern software development IDEs that successfully combined the text editor with a syntax checker and a compiler into one tool.

4.4 Software Architecture

MuNG Studio is built as a single-page web application, bundled using the Parcel build tool. It uses React to support its UI because of its popularity and our prior experience. It uses the MUI Joy component library to speed up UI development. The state of the application is encapsulated in state stores, which expose the data via plain Javascript for real-time interactions (zooming, pointing) and via the Jotai state management library to React to be displayed in the UI. These state stores solve crucial performance problems related to React re-renders and, without their proper design, the application would not function.

5 Conclusion

We presented MuNG Studio, a new annotation tool for the Music Notation Graph format, which is designed to be easy to install and use, primarily tackling the performance challenge and able to work with 2000 objects on a page. It currently supports viewing and link editing, with node editing planned for the foreseeable future. MuNG Studio was mainly developed to allow annotation of the OmniOMR dataset but we hope that it will be useful for other datasets and become a valuable companion to everyone working with the MuNG format in the future.

Acknowledgments

This work has been supported by the Ministry of Culture of the Czech Republic (project NAKI no. DH23P03OVV008), the project "Human-centred AI for a Sustainable and Adaptive Society" (reg. no. CZ.02.01.01/00/23_025/0008691), co-funded by the European Union, Charles University (project GAUK no. 289623), by SVV project no. 260 821, and by the Ministry of Education, Youth and Sports of the Czech Republic, project no. LM2023062 LINDAT/CLARIAH-CZ.

References

[1] David Bainbridge and Tim Bell. 2003. A music notation construction engine for optical music recognition. *Software: Practise and Experience* 33 (2003), 173–200. doi:10.1002/spe.502

[2] Arnau Baró, Pau Riba, Jorge Calvo-Zaragoza, and Alicia Fornés. 2019. From Optical Music Recognition to Handwritten Music Recognition: A baseline. *Pattern Recognition Letters* 123 (2019), 1–8. doi:10.1016/j.patrec.2019.02.029

[3] Arnau Baró, Pau Riba, and Alicia Fornés. 2022. Musigraph: Optical Music Recognition Through Object Detection and Graph Neural Network. In *18th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. Hyderabad, India, 171–184. doi:10.1007/978-3-031-21648-0_12

[4] Jorge Calvo-Zaragoza, Jan Hajič jr., and Alexander Pacha. 2020. Understanding Optical Music Recognition. *Comput. Surveys* 53, 4 (2020), 77. doi:10.1145/3397499

[5] Jorge Calvo-Zaragoza and David Rizo. 2018. End-to-End Neural Optical Music Recognition of Monophonic Scores. *Applied Sciences* 8, 4 (2018), 606. doi:10.3390/app8040606

[6] Bertrand Coüasnon and Bernard Rétif. 1994. Using A Grammar For A Reliable Full Score Recognition System. In *1st Workshop on Document Analysis Systems (DAS)*. Kaiserslautern, Germany, 15–27.

[7] Gregoire de Lambertye and Alexander Pacha. 2024. Semantic Reconstruction of Sheet Music with Graph-Neural Networks. In *6th International Workshop on Reading Music Systems (WoRMS)*. Online, 12–17. doi:10.48550/arXiv.2411.15741

[8] Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. 2011. CVC-MUSCIMA: A ground truth of handwritten music score images for writer identification and staff removal. *International Journal on Document Analysis and Recognition (IJDAR)* 15 (2011), 243–251. doi:10.1007/s10032-011-0168-2

[9] Carlos Garrido-Munoz, Antonio Rios-Vila, and Jorge Calvo-Zaragoza. 2022. A holistic approach for image-to-graph: application to optical music recognition. *International Journal on Document Analysis and Recognition (IJDAR)* 25 (2022), 293–303. doi:10.1007/s10032-022-00417-4

[10] Mark Gotham, Peter Jonas, Bruno Bower, William Bosworth, Daniel Rootham, and Leigh VanHandel. 2018. Scores of scores: an openscore project to encode and share sheet music. In *5th International Conference on Digital Libraries for Musicology (DLfM)*. Paris, France, 87–95. doi:10.1145/3273024.3273026

[11] Mark Gotham, Maureen Redbond, Bruno Bower, and Peter Jonas. 2023. The "OpenScore String Quartet" Corpus. In *10th International Conference on Digital Libraries for Musicology (DLfM)*. New York, NY, USA, 49–57. doi:10.1145/3625135.3625155

[12] Mark Robert Haigh Gotham and Peter Jonas. 2022. The OpenScore Lieder Corpus. In *Music Encoding Conference*. Alicante, Spain, 131–136. doi:10.17613/1my2-dm23

[13] Stanisław Graczyk, Zuzanna Piniarska, Mateusz Kałamoniak, Tomasz Łukaszewski, and Ewa Łukasik. 2024. An Online Tool for Semi-Automatically Annotating Music Scores for Optical Music Recognition. In *11th International Conference on Digital Libraries for Musicology (DLfM)*. New York, NY, USA, 73–77. doi:10.1145/3660570.3660571

[14] Jan Hajič and Pavel Pecina. 2017. Groundtruthing (Not Only) Music Notation with MUSICMarker: A Practical Overview. In *14th International Conference on Document Analysis and Recognition (ICDAR)*. Kyoto, Japan, 47–48. doi:10.1109/ICDAR.2017.271

[15] Jan Hajič jr. and Pavel Pecina. 2017. Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression. arXiv:1708.01806 [cs.CV] https://arxiv.org/abs/1708.01806

[16] Jan Hajič, jr. and Pavel Pecina. 2017. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. In *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Kyoto, Japan, 39–46. doi:10.1109/ICDAR.2017.16

[17] Jan Hajič jr., Petr Žabička, Jan Rychtář, Jiří Mayer, Martina Dvořáková, Filip Jebavý, Markéta Vlková, and Pavel Pecina. 2023. The OmniOMR Project. In *5th International Workshop on Reading Music Systems (WoRMS)*. Milan, Italy, 12–14. doi:10.48550/arXiv.2311.04091

[18] Jose M. Inesta, David Rizo, and Jorge Calvo Zaragoza. 2019. MuRET as a software for the transcription of historical archives. In *2nd International Workshop on Reading Music Systems (WoRMS)*. Delft, The Netherlands, 12–15. https://sites.google.com/view/worms2019/proceedings

[19] Martin Keary. 2019. Music Software & Interface Design: MuseScore. https://youtu.be/4hZxo96x48A?si=7d4W5XBpsphGdcp_&t=906

[20] Sangkuk Lee, Sung Joon Son, Jiyong Oh, and Nojun Kwak. 2016. Handwritten Music Symbol Classification Using Deep Convolutional Neural Networks. In *International Conference on Information Science and Security (ICISS)*. Pattaya, Thailand, 1–5. doi:10.1109/ICISSEC.2016.7885856

[21] Jiří Mayer and Pavel Pecina. 2021. Synthesizing Training Data for Handwritten Music Recognition. In *16th International Conference on Document Analysis and Recognition (ICDAR)*. Lausanne, Switzerland, 626–641. doi:10.1007/978-3-030-86334-0_41

[22] Jiří Mayer, Milan Straka, Jan Hajič jr., and Pavel Pecina. 2024. Practical End-to-End Optical Music Recognition for Pianoform Music. In *18th International Conference on Document Analysis and Recognition (ICDAR)*. Athens, Greece, 55–73. doi:10.1007/978-3-031-70552-6_4

[23] Alexander Pacha, Jorge Calvo-Zaragoza, and Jan Hajič jr. 2019. Learning Notation Graph Construction for Full-Pipeline Optical Music Recognition. In *20th International Society for Music Information Retrieval (ISMIR)*. Delft, Netherlands, 75–82. doi:10.5281/zenodo.3527744

[24] Alexander Pacha and Horst Eidenberger. 2017. Towards a Universal Music Symbol Classifier. In *14th International Conference on Document Analysis and Recognition (ICDAR)*. Kyoto, Japan, 35–36. doi:10.1109/ICDAR.2017.265

[25] Alexander Pacha and Jan Hajič jr. 2018. A Baseline for General Music Object Detection with Deep Learning. *Applied Sciences* 8, 9 (2018), 1488–1508. doi:10.3390/app8091488

[26] Roberto M. Pinheiro Pereira, Caio E.F. Matos, Geraldo Braz Junior, João D.S. de Almeida, and Anselmo C. de Paiva. 2016. A Deep Approach for Handwritten Musical Symbols Recognition. In *22nd Brazilian Symposium on Multimedia and the Web*. New York, NY, USA, 191–194. doi:10.1145/2976796.2988171

[27] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jamie S. Cardoso. 2012. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval* 1 (2012), 173–190. doi:10.1007/s13735-012-0004-6

[28] Antonio Ríos-Vila, David Rizo, José M. Iñesta, and Jorge Calvo-Zaragoza. 2023. End-to-end optical music recognition for pianoform sheet music. *International Journal on Document Analysis and Recognition (IJDAR)* 26, 3 (2023), 347–362. doi:10.1007/s10032-023-00432-z

[29] Elona Shatri and György Fazekas. 2021. DoReMi: First glance at a universal OMR dataset. In *3rd International Workshop on Reading Music Systems (WoRMS)*. Alicante, Spain, 43–49. doi:10.48550/arXiv.2107.07786

[30] Lukas Tuggener, Yvan Putra Satyawan, Alexander Pacha, Jürgen Schmidhuber, and Thilo Stadelmann. 2021. The DeepScoresV2 Dataset and Benchmark for Music Object Detection. In *25th International Conference on Pattern Recognition (ICPR)*. Milan, Italy, 9188–9195. doi:10.1109/ICPR48806.2021.9412290

[31] Evan Wallace. 2015. Building a professional design tool on the web. https://www.madebyevan.com/figma/building-a-professional-design-tool-on-the-web/