Design of a music recognition, encoding, and transcription online tool

David Rizo[1,2], Jorge Calvo-Zaragoza[1], Juan C. Martínez-Sevilla[1], Adrián Roselló[1], and Eliseo Fuentes-Martínez[1] *

[1] Universidad de Alicante
[2] Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana (ISEA.CV)
drizo@dlsi.ua.es

Abstract. In recent years, Optical Music Recognition (OMR) technologies have experienced a notable boost thanks mainly to the use of new pipelines based on machine learning, specially on deep neural networks. These methods are usually studied just from the point of view of the accuracy of the output of the networks. However, from a practical perspective in a real-world context, this is not enough. In this paper we present a design of a tool devised for allowing the scientific study of the complete OMR workflow in different scenarios and notations, including both the possibility of analyzing the real impact of improvements in automatic recognition models and how they are integrated for practical purposes in the work of the transcriber.

Keywords: Optical music recognition, encoding, transcription, user experience

1 Introduction

Digitizing sheet music and other music-related documents can provide several benefits, including easier access for researchers, music practitioners, musicologists, and the general public, as well as preservation of musical heritage. Digitized sheet music can be searched, played, analyzed, and annotated using specialized software tools, allowing for new discoveries and insights into musical history and culture.

One notable example of an effort to digitize music collections into digital images is the International Music Score Library Project[3] (IMSLP), which aims to create a virtual library of public domain sheet music. The IMSLP has digitized thousands of scores from various composers and genres, making them freely available for download and use. Other organizations and institutions, such as libraries, museums, and universities, are also scanning their music collections into image files to increase access and preserve musical heritage.

* This work has been supported by the Spanish Ministerio de Ciencia e Innovación through project MultiScore (No. PID2020-118447RA-I00), supported by UE FEDER funds.

[3] https://www.imslp.org (accessed April 19th, 2023).

As a matter of fact, having the musical content information, i.e., the audio and the scans, not digitally encoded ends up being a waste of resources given that current music information retrieval pipelines require it. By this means, we cannot consider a score digitization process as finished until the digital score version encoding is produced.

Alfaro-Contreras et al. [3] showed that the most efficient way to obtain these digital scores is to resort to an automated reading of documents by using the so-called Optical Music Recognition (OMR) [5]. This technology has achieved different levels of recognition accuracy depending on the type of documents, the quality of the medium, and the type and complexity of the notation. But still, in most cases, the OMR does not yield perfect results. The need of post-editing depends on the tasks to be performed on the recognized content. Some projects such as F-Tempo[4] directly use—possibly with errors—OMR output to approximate perform searches. When requiring a curated transcription, a manual correction process has to be done. For instance, this pipeline was used to encode a large number of files of the KernScores repository.[5]

The low accuracy of OMR is not the only obstacle in real use-cases. There is no OMR system yet able to process the whole set of sysmbols found in early notations. The processing of orchestral scores of varying layout, the presence of *ossias*, or dealing with works where the different parts are written in separate sheets, make it even more challenging. This is why, in many real projects, the encoding process is eventually performed by transcribers using computerized notation tools such as Finale[6], Dorico[7], MuseScore[8], or Sibelius[9]. In projects such as Didone [26], around 4 000 Eighteenth-Century Italian Opera arias are being manually copied in Finale to be later stored into MusicXML files [13]. The same approach was used to obtain the encodings of the modern version of the renaissance works from the "Josquin Research Project" (JRP).[10]

In this scenario, the main advances in OMR are achieved by modern artificial intelligence techniques based on machine learning, namely deep learning [5]. Improvements are attained through the correct selection of neural network architectures and the availability of training data in sufficient quantity and quality for those architectures. The research-oriented OMR tool "Music Recognition, Encoding, and Transcription" (MuRET) [19] was introduced to push the development of both OMR techniques and the creation and curation of datasets. This tool was developed with JavaFX[11] as a desktop application. It included our first OMR models [8], which allowed for the curation of a number of training sets and the development of new OMR approaches [24, 9].

Once the usefulness of MuRET became clear, we decided to port it to a web application for two main reasons. The first is that the technology is continuously evolving, which made it difficult to deploy the application on a daily basis. The second, and more important, is to naturally maintain a growing repository of both ongoing transcription documents and trained OMR models shared among all users. Having evaluated this re-

[4] `https://f-tempo.org` (accessed April 19th, 2023).
[5] `http://kern.ccarh.org/` (accessed April 19th, 2023).
[6] `https://www.finalemusic.com` (accessed April 19th, 2023).
[7] `https://www.steinberg.net/dorico` (accessed April 19th, 2023).
[8] `https://musescore.org` (accessed April 19th, 2023).
[9] `https://www.avid.com/sibelius` (accessed April 19th, 2023).
[10] `https://josquin.stanford.edu` (accessed April 19th, 2023).
[11] `https://www.oracle.com/es/java/technologies/javase/javafx-overview.html` (accessed April 19th, 2023).

search oriented OMR online tool in real scenarios, in this paper the main decisions to build it are described with which we expect to contribute to the improvement of ongoing and future OMR investigations.

The paper is organized as follows. Section 2 details other alternatives to MuRET that, due to the fact that this tool is eminently research-oriented, may be most suitable to be used in transcription projects. Next, the requiresments taken into account in Section 3 and decisions made during the development of the current online version are discussed in Section 4 that may be useful for other similar projects. Finally, conclusions will be drawn and future works outlined in Section 5.

2 State of the art

There are several Optical Music Recognition (OMR) tools available to transcribe Common Western Modern Notation (CWMN). Only one open-source Audiveris[12] is available, and a number of commercial packages such as SmartScore[13], PhotoScore[14], or PlayScore 2[15]. The effectiveness of each tool can vary depending on the complexity and quality of the sheet music being analyzed. A brief analysis of them for recognizing music theory books can be found in [14], that shows how far they are from retrieving successful results on complex scenarios.

The transcription of notations other than CWMN is very restricted to very few applications. In the context of the SIMSSA project [11], two applications in the past years have been used to automatically extract musical information from images, although they are no longer maintained: Gamut and Aruspix [17]. In addition, within this project, an OMR meta-workflow called Rodan was built in which users can create their own systems using predefined image processing and machine learning blocks [12]. Although not designed for any specific notation, most of the existing blocks are intended for neume recognition. More recently, another approach based on convolutional neural networks was developed specifically for mensural notation, which reported high accuracies [25].

Given this context, to the best of our knowledge, no tool ready to deal both with handwritten and printed sources of several kinds of notation is available other than our proposal MuRET.

3 Requirements

The ultimate goal of MuRET is to facilitate OMR research from a holistic perspective. This means that the tool must support the research of all individual steps of the workflow to obtain a final digital score from the different images, considering both the automatic processes and user manual interactions. Being this a research tool, it must be prepared to be scaled to any possible scenario in terms of notation type, parts arrangement, document layout, calligraphies and fonts, and transcription purposes.

[12] `https://github.com/Audiveris/audiveris` (accessed April 19th, 2023).
[13] `https://www.musitek.com` (accessed April 19th, 2023).
[14] `https://www.neuratron.com/photoscore.htm` (accessed April 19th, 2023).
[15] `https://www.playscore.co` (accessed April 19th, 2023).

From an end-user point of view, the user should be first allowed to manage collections of works made of digital images of any format and resolution. For transcribing a new work, the constituent elements in each image such as pages, staves, and lyrics must be identified. Then, in the case of being a polyphonic work, they must be assigned to the different instruments, voices, or parts. The contents in staves and text regions must be recognized using a variety of approaches that, after being combined, will make up a final digital score that will be exported to standard encoding formats. All possible aids that the machine can compute, such as displaying early notations in modern forms, or hints in the final scoring-up, should be provided. Additionally, the tool must incorporate process-oriented functionalities, as an aid to account for the current status of work on several simultaneous works, or the inclusion of comments both for the whole work or elements inside it.

Ultimately, the system must be able to perform all processes in an assisted manner, so that the output of the various automatic classifiers is corrected when necessary by the user as quickly as possible.

From the OMR process research perspective, the tool must be ready to accommodate different approaches to convert a set of input images into a digital score. For each of those approaches, in the case of being based on machine learning, the extraction of new training sets from the already processed works and the posterior training, upload, and use of new models, must be supported. For analyzing the actual behavior of any paradigm and model besides the usual model performance metrics, all actions of the user must be recorded and categorized for being later analyzed.

Several non-functional requirements arise that can influence the design of the tool. It must allow the simultaneous transcription of the same work by different, possibly remote, users. To allow the accommodation of new repertories with a minimum amount of effort, and to make the user unconcerned about formats and resolutions, the use of IIIF framework[16] for exchanging the processed works to other external tools is recommended. Finally, the system must be usable with a standard computer setup.

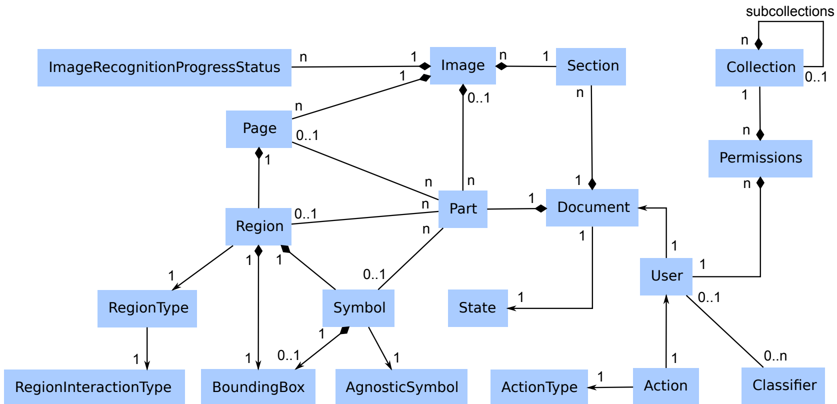


Fig. 1: Domain model. Some classes and relations, and all attributes have been omitted for easier reading.

[16] `https://iiif.io` (accessed April 19th, 2023).

4 Solution design

The current online version of MuRET [17] presents a possible approach to solve all the requirements detailed above. In order to implement it, several decisions have been made that will be described in this section.

The system has been structured using a three-tier web architecture style. The *presentation layer* has been solved using Angular with the Redux pattern[18], the *application layer* has been implemented using Spring Boot[19], and the *data layer* served by a MariaDB[20] relational database.

All the system data is eventually stored in records of the database that is converted to the object-oriented hierarchy shown in Fig. 1 through the Hibernate Object-Relation Mapping (ORM)[21]. The names of the classes can be easily understood from the explanation in the following lines.

4.1 User collections and images

Users must be registered by a system administrator to work on the tool. No self-registering process is offered. All works are organized into collections and sub-collections, whose access is granted by the administrator.

A document is the core entity of a transcription project. After being uploaded as individual image files or inside a PDF document, the images of a document to be transcribed are grouped, at least, into one default section. This is useful for dealing with compound works such as masses and operas. Images and sections can be deleted, edited, and reordered. Images that contain cover sheets, or non-musical content, can be hidden for subsequent automatic recognition processes. Users are allowed to assign to each work metadata such as the notation type, manuscript type (printed or handwritten), composer and printer.

For the purpose of helping the user in daily work tasks, the work in progress and image recognition annotations stages can be marked up to the final transcription (see buttons to mark this progress below in the bottom-right of Fig. 5c).

4.2 Document analysis

After organizing the images, the first step to transcribe a work, known as *document analysis*, is to segment each image into separate components, a series of regions of different types such as staves and lyrics are identified (Fig. 2). Usually, each image contains just one page, but it is also usual to receive scans of books where images contain several pages as in the case of the image in that figure. This process can be performed either manually by drawing bounding boxes on top of the image and assigning a region type to each drawn box, or by using an automatic classifier that identifies the different segments in the image. Across MuRET, when an operation can be performed automatically, the

[17] `https://muret.dlsi.ua.es/muret`
[18] `https://angular.io` and `https://redux.js.org` (accessed April 19th, 2023).
[19] `https://spring.io/projects/spring-boot` (accessed April 19th, 2023).
[20] `https://mariadb.org` (accessed April 19th, 2023).
[21] `https://hibernate.org/orm/` (accessed April 19th, 2023).

user can select and apply a classifier (see the two available models of the drop-down control at the top-right of the Fig. 2), and correct the output if necessary. Classifiers are run currently in the user browser using TensorFlow.js. [22] This decision has the advantage that it allows avoiding collapsing the server machine when several users are using MuRET at the same time running different models that have to be loaded in memory. The main drawback is that the used models have had to be tuned to keep their size at the minimum for being used in standard computers.

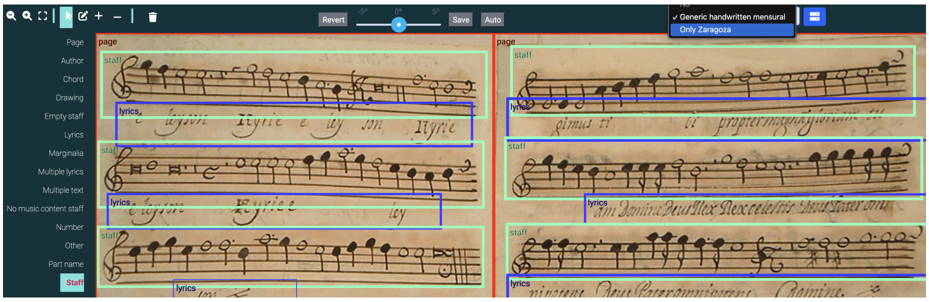


Fig. 2: Document analysis screenshot. In this example, only the staves and lyrics regions are segmented. The snapshot shows two possible classifiers to perform a document analysis (top-right), and controls to rotate, manually or automatically, the image (top-center). The current catalog of region types shown at the left of the image can be easily modified.

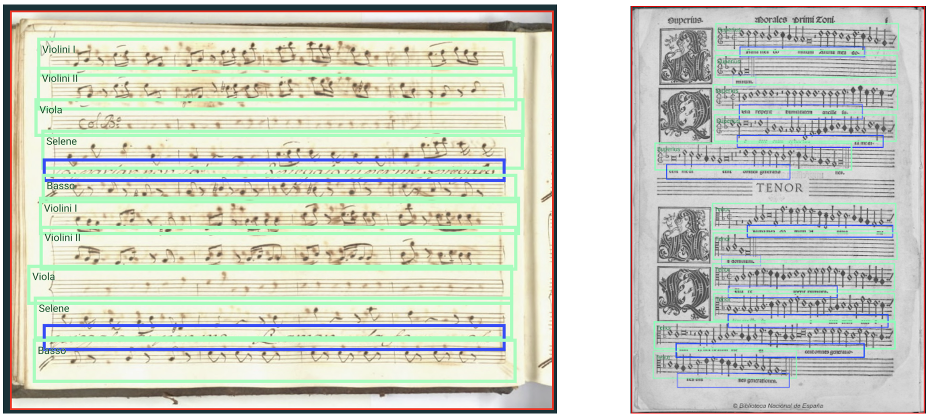
4.3 Part management

Most of the sources to be processed are polyphonic, consisting of several voices, instruments, or parts. There is a variety of arrangements, such as works made of parts distributed across pages (the image in Fig. 2 corresponds just to a part), choir-books where the same page contains two voices (Fig. 3b), ensembles or orchestral scores (Fig. 3a). The kind of book to transcribe could be of a totally different nature. For instance, it can be a compilation of songs not related to any instrument, such as a jazz *Real Book*, or be a catalog containing lists of incipits. In some cases, the volume to be transcribed describes music theory as it is the case of music treatises [14], where most of the content is textual with some illustrative music examples. The process of dealing with parts is performed currently manually. In all cases, the internal implementation of all those situations is reduced to the case where the whole image or page is linked to a part, or when each region must be assigned to a part. The system is also prepared to deal with chorale layouts with two staves for four voices. In that case, each individual symbol inside the region must be linked to each part. For reducing user effort, the system offers aids to manage the set of instruments and to reuse the different layouts between pages.

4.4 Region contents recognition

Once the different staves are identified and assigned to the part they belong to, the musical contents inside the image crop that corresponds to each region must be recognized and

[22] `https://www.tensorflow.org/js` (accessed April 19th, 2023).

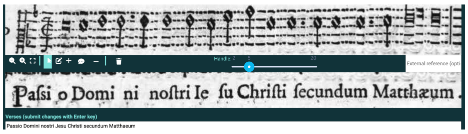


(a) Parts in orchestral score

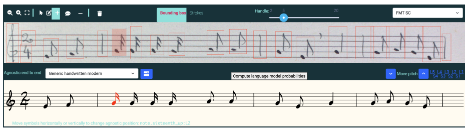
(b) Parts in choir book

Fig. 3: Different parts and arrangements. All regions must be attributed to a part.

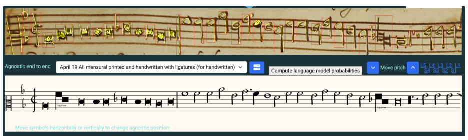
encoded. Currently, lyrics can be encoded as text but they are just stored in the database without any further treatment (Fig. 4a). New region types can be easily incorporated in the future. There are several approaches that can be followed to obtain an encoding from the image, either manually or by applying an automatic classifier. The first consists of manually tracing the graphic symbols so a classifier [8], by using both the stroke (Fig. 4b) and the image obtained from the bounding box that encloses the stroke, identifies each symbol among a set of possible *agnostic symbols* [7] (i.e. graphical symbols without an attributed musical meaning yet), and the vertical position in the staff as an absolute value regardless the clef. Although we use this *symbol-agnostic* concept where we identify complete glyphs, we could easily adapt it to recognize primitives (note head, stem, etc.) as proposed in [16].

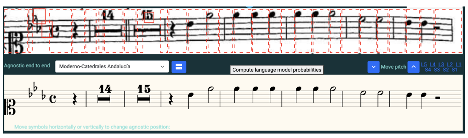


(a) Lyrics. See text transcribed at the bottom of the snapshot.



(c) Bounding boxes of agnostic symbol





(b) Strokes

(d) Staff-level end-to-end

Fig. 4: Transcription of regions.

An alternative is just to draw the bounding box surrounding each symbol to use just the image clipping as input information for the classifier [15] to identify the agnostic symbol type (Fig. 4c). In either of the two options, an agnostic symbol sequence is obtained, i.e., a sequence of symbols ordered from left to right, top to bottom. For instance, the beginning of the agnostic sequence in Fig. 4c is: `clef.G2:L2 digit.2:L4 digit.4:L2 note.8th:L1 note.8th:S1 verticalLine:L1`. The rendering of the bottom staff is performed by using fonts that have a glyph for each agnostic symbol that are just placed in the position of the symbol in the image. For white mensural notation, the Capitan [20] font is used that was developed on-purpose, and for modern notation, Petaluma font[23] has been utilized.

The next possibility is to use a staff-level end-to-end classifier that identifies the agnostic symbols in the image in such a way that the sequence order is respected but the bounding boxes of each symbol are not detected but their approximate horizontal position [7] (dashed lines in Fig. 4d show those approximate positions). The user can move and correct any of the symbols. In that case, to take advantage of the interaction for obtaining a new training sample, the bounding box is drawn (see the second flat in the key signature in Fig. 4d).

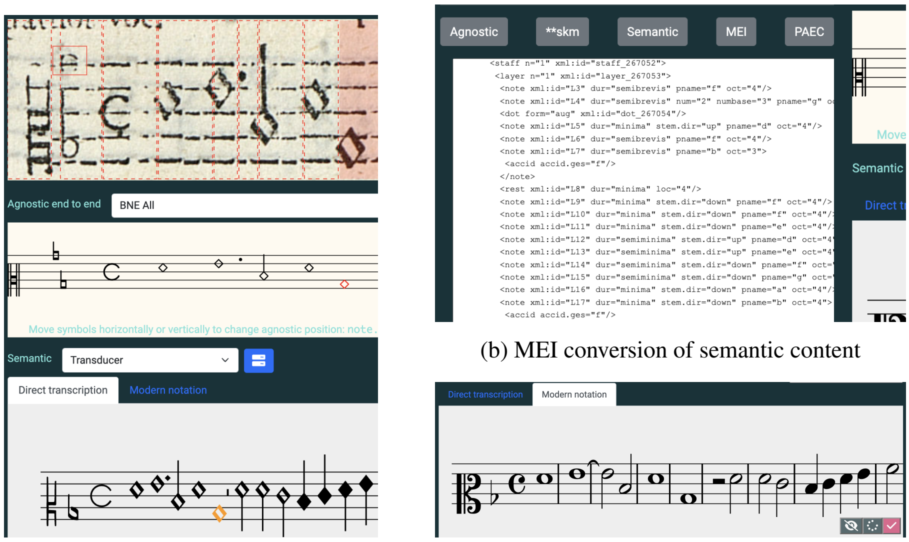
4.5 Music encoding of individual staves

The agnostic sequence must be converted to a meaningful music encoding denoted as *semantic encoding* [7]. For example, the sequence of three flats at the beginning of the agnostic sequence in Fig. 4d must be converted to a E♭ major (or its relative minor) key. When encoding the pitch of the notes, this key signature must be taken into account for correctly assigning if necessary the right accidental. This translation from *agnostic* to *semantic* (Fig 5a) can be performed either using a rules-based automaton transducer [20], or translation technologies based on machine learning approaches [1][24]. For early notations, a valid conversion into modern notation is performed with any consideration of transposition or metric change (Fig. 5c). The rendering of the bottom staff is delegated to Verovio [18], through a previous conversion of our internal format introduced below into MEI (Fig. 5b). If the staff to convert is not the top staff, the contextual information from previous staves, such as the previous time signature, is propagated. The conversion to Plaine and Easie Code [4] for cataloging in RISM[25] is performed by that library as well.

It is important to note that MEI or Verovio do not always support all required features. For instance, bar-lines crossing a note in late mensural notations or the rendering of *signum congruentiae*. In those cases, our principle has been to internally store a specific tag for each unsupported feature and print text marks to visualize them.

A key decision taken to design MuRET was the method to store those semantic sequences. We have not chosen any standard format as the internal encoding, but an ad-hoc representation extended from the Humdrum [23] formats **kern and **mens [21] in what we name **skm. Later, when exporting the final encoding, standard formats are

[23] `https://www.smufl.org/fonts/` (accessed April 19th, 2023).
[24] Note that there is an agnostic representation for dealing with more complex situations such as the presence of chords [2]. In any case, the workflow is the same regardless of the agnostic encoding.
[25] `https://rism.info/` (accessed April 19th, 2023).



(a) Agnostic to semantic

(b) MEI conversion of semantic content

(c) Modern transcription

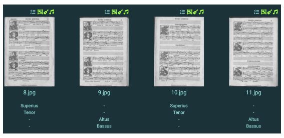
Fig. 5: Semantic contents recognized from the image.

used. This has allowed us to adapt the format to encode specific situations that were not possible with standard **kern/**mens when we required it (e.g. *custos* symbol, the position of rests, canceling accidentals in mensural notation, or dots after a barline). This encoding contains also information about the agnostic symbol each semantic element is related to, which allows for later exporting this kind of graphical information to formats such as the *facsimile* element in MEI. The choice of extending Humdrum formats and not other more comprehensive ones such as SCORE [23], MusicXML [13] or MEI [22], is the ease with which users can fully manually encode or correct the output of the classifiers. In any case, the translation process generates a sequence of objects of a musical object-oriented hierarchy that are just converted into **skm in order to serialized them allowing its presentation in the interface and storage.

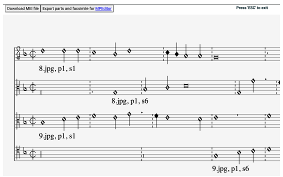
MuRET does not include yet the direct recognition from the image to the **skm encoding because we have experimented to be faster and more accurate to use this intermediate representation as shown in [6]. If a classifier was proven to yield better results, both the transcription and correction processes, it could be easily introduced in the application.

4.6 Scoring up and exporting

Finally, when all previous steps have been finished, the user can select the images (Fig. 6a) that want to be used to generate a final score (Fig. 6b). This operation is accomplished by concatenating all the staves in the selected images grouped by the parts they belong to, exporting them from our internal format to MEI, and letting Verovio engrave the score. In order to share the transcription with external services, the previous MEI can be exported as it is rendered by Verovio. MuRET also is able to convert to a parts-based MEI format including graphical information in the facsimile element (Fig. 6c). This functionality was included for exchanging information with specialized tools such as MP-Editor to perform scoring-up processing in mensural notation [10].



(a) Image selection



(b) Score preview

(c) Parts based MEI including facsimile (see more details in [10]).

Fig. 6: Previsualizing and exporting

4.7 Model training

As mentioned above, the classifiers that support the automatic processes use machine learning models, that need training data for being built. The system allows downloading different training sets from selected collections and works. This training data is just a set of JSON files containing an export of the objects in our internal data model (Fig. 1). After being trained offline, these models can be uploaded again to the system. This incremental workflow, i.e., fixing the output of the different classifiers, downloading datasets with corrected data, building new models, and uploading them to improve the performance of the system, is being shown in our transcription projects to be a proper way to proceed. Using this approach for transcribing a work by Jacob van Eyck, printed by Paulus Matthijsz in 1649, the recognition and post-editing effort were reduced by a factor of 10, allowing the end user to obtain a complete and correct encoding of a standard book page in under one minute per page.

4.8 User action logging and user experience

Some models perform better than others in a theoretical way, but the corrections required to fix their outputs lead to a higher effort by the user. This can be measured by analyzing the actions each individual user performs on the tool that is logged and conveniently categorized into meaningful operations, such as the editing of regions, symbols, semantic content, part management or classifier use. The timestamps of all operations is also registered, as well as the document, region or symbol involved in each operation. Currently, we have stored more than 300 000 actions from different users.

These action logs have helped us to improve the user experience of the system by evincing many operations that are frequent and repetitive, decreasing the final throughput. These issues, such as those related to the feedback of the system in error messages, long actions, or the graphical design of interaction controls, have been gradually corrected or taken into account to include new capabilities to the system.

5 Conclusions and future work

This paper has depicted the main blocks required to integrate an OMR system scalable to work with any kind of notation and scenario that besides being useful for real transcription projects can help in the improvement of OMR research.

This tool is being continuously improved as new features in projects arise. Most of the effort is performed on improving the OMR models by using the increasing quantity of already transcribed works of different kinds. We are working towards addressing the current weaknesses of the system, namely: adding new front-end deep learning frameworks and formats such as ONXX [26], the direct use of the IIIF manifest from servers without the need of uploading any image to the system, the possibility of performing an automatic classification of a whole work to be later corrected to complement the current totally interactive workflow, the possibility of directly training models online, and the endless task of improving the user experience of the tool.

References

1. Ríos-Vila; A., M. Esplà-Gomis, D. Rizo, P.J. Ponce de León, and J.M. Iñesta. Applying automatic translation for optical music recognition's encoding step. *Applied Sciences*, 11(9), april 2021.
2. M. Alfaro-Contreras, J. Calvo-Zaragoza, and J.M. Iñesta. Approaching end-to-end optical music recognition for homophonic scores. In *Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA 2019*, pages 147–158, Madrid, August 2019. Springer.
3. M. Alfaro-Contreras, D. Rizo, J.M. Iñesta, and J. Calvo-Zaragoza. OMR-assisted transcription: a case study with early prints. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 35–41, Online, November 2021. ISMIR.
4. B.S. Brook. The Simplified Plaine and Easie Code System for Notating Music: A Proposal for International Adoption. *Fontes Artis Musicae*, 12(2-3):156–160, jan 1965.
5. J. Calvo-Zaragoza, J. Hajič, and A. Pacha. Understanding optical music recognition. *ACM Computing Surveys (CSUR)*, 53(4):1–35, 2020.
6. J. Calvo-Zaragoza and D. Rizo. 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, September 23-27, 2018*, pages 248–255, 2018.
7. J. Calvo-Zaragoza and D. Rizo. End-to-end neural optical music recognition of monophonic scores. *Applied Sciences*, 8(4), 2018.
8. J. Calvo-Zaragoza, D. Rizo, and J.M. Iñesta. Two (note) heads are better than one: Pen-based multimodal interaction with music scores. In *International Society for Music Information Retrieval Conference*, 2016.
9. J. Calvo-Zaragoza, A.H. Toselli, and E. Vidal. Hybrid hidden markov models and artificial neural networks for handwritten music recognition in mensural notation. *Pattern Anal. Appl.*, 22(4):1573–1584, 2019.
10. K. Desmond, L. Pugin, J. Regimbal, D. Rizo, C. Sapp, and M. E. Thomae. Encoding polyphony from medieval manuscripts notated in mensural notation. In *Music Encoding Conference Proceedings 2021*, page 197–219. Humanities Commons, May 2022.

[26] `https://onnx.ai/` (accessed April 19th, 2023).

11. I. Fujinaga, A. Hankinson, and J.E. Cumming. Introduction to SIMSSA (single interface for music score searching and analysis). In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*, DLfM '14, page 1–3, New York, NY, USA, 2014. Association for Computing Machinery.
12. I. Fujinaga and G. Vigliensoni. The art of teaching computers: The SIMSSA optical music recognition workflow system. In *27th European Signal Processing Conference, EUSIPCO 2019, A Coruña, Spain, September 2-6, 2019*, pages 1–5. IEEE, 2019.
13. M. Good and G. Actor. Using MusicXML for File Interchange. *Web Delivering of Music, International Conference on*, 0:153, 2003.
14. F. Moss, N. Nápoles-López, M. Köster, and D. Rizo. Challenging sources: a new dataset for omr of diverse 19th-century music theory examples. In *Proceedings of the 4th International Workshop on Reading Music Systems (WoRMS 2022)*, November 2022.
15. A. Nuñez-Alcover, P.J. Ponce de León, and J. Calvo-Zaragoza. Glyph and position classification of music symbols in early music manuscripts. In *Proc. of the 9th Iberian Conference on Pattern Recognition and Image Analysis, LNCS vol. 11867*, pages 159–168, Madrid, Spain, July 2019. AERFAI, APRP, Springer.
16. A. Pacha, K. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi, and HorsH. Eidenberger. Handwritten music object detection: Open issues and baseline results. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 163–168. IEEE, 2018.
17. L. Pugin, J. Hockman, J.A. Burgoyne, and I. Fujinaga. Gamera Versus Aruspix: Two optical music recognition approaches. In *9th International Conference on Music Information Retrieval, Drexel University, Philadelphia, USA, September, 2008*, pages 419–424, 2008.
18. L. Pugin, R. Zitellini, and P. Roland. Verovio - A library for Engraving MEI Music Notation into SVG. In *International Society for Music Information Retrieval*, jan 2014.
19. D. Rizo, J. Calvo-Zaragoza, and J.M. Iñesta. MuRET: A music recognition, encoding, and transcription tool. In *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, DLfM '18, page 52–56, New York, NY, USA, 2018. Association for Computing Machinery.
20. D. Rizo, B. Pascual, J.M. Iñesta, A. Ezquerro, and L.A. González. Towards the digital encoding of hispanic white mensural notation. *Anuario Musical*, (72):293–304, 2017.
21. D. Rizo, N. Pascual-León, and C. S. Sapp. White mensural manual encoding: from humdrum to mei. *Cuadernos de Investigación Musical*, (6):373–393, 2018.
22. P. Roland. The music encoding initiative (MEI). In *Proceedings of the First International Conference on Musical Applications Using XML*, pages 55–59, jan 2002.
23. E. Selfridge-Field, editor. *Beyond MIDI: The Handbook of Musical Codes*. MIT Press, Cambridge, MA, USA, 1997.
24. J. Sober-Mira, J. Calvo-Zaragoza, D. Rizo, and J.M. Iñesta. Pen-based music document transcription with convolutional neural networks. In *Graphics Recognition, Current Trends and Evolutions - 12th IAPR International Workshop, GREC 2017, Kyoto, Japan, November 9-10, 2017, Revised Selected Papers*, volume 11009 of *Lecture Notes in Computer Science*, pages 71–80. Springer, 2017.
25. J. Stoessel, D. Collins, and S. Bolland. Using optical music recognition to encode 17th-century music prints: The canonic works of paolo agostini (c.1583–1629) as a test case. In *7th International Conference on Digital Libraries for Musicology*, DLfM 2020, page 1–9, New York, NY, USA, 2020. Association for Computing Machinery.
26. A. Torrente and A. Llorens. The Musicology Lab: Teamwork and the Musicological Toolbox. In *Music Encoding Conference Proceedings 2021*, pages 9–20. Humanities Commons, 2022.