Proceedings of the
6th International Workshop on
Reading Music Systems

22nd November, 2024

Organization

General Chairs

Jorge Calvo-Zaragoza	University of Alicante, Spain
Alexander Pacha	TU Wien, Austria
Elona Shatri	Queen Mary University of London, United Kingdom

Logo made by Freepik from www.flaticon.com. Adapted by Alexander Pacha.

Preface

Dear colleagues,

We are proud to present the proceedings of the 6th International Workshop on Reading Music Systems (WoRMS).

Over the past few years, interest in Music Reading Systems has continued to grow. This year marks a new record, with a total of 22 submissions, 15 of which have been accepted to the workshop. A few papers are omitted from the proceedings by request of the authors. We took great care to provide comprehensive feedback to authors whose works were not accepted, highlighting areas for improvement to meet the quality standards of WoRMS. We hope to see these authors submit their revised works next year.

Due to logistical reasons, we have decided to host this year's edition online again. This format allows participants from all over the world to join easily and learn about the latest developments without the need for extensive travel. However, we acknowledge that an online format cannot fully replace the experience of face-to-face interactions, and we aim to make future editions on-site events once more.

We would like to take this opportunity to promote the GitHub organization `https://github.com/omr-research` once more, which welcomes contributions from everyone and serves as a central hub for publishing and discovering research-related repositories. Additionally, we encourage you to explore our public YouTube channel, `https://www.youtube.com/OpticalMusicRecognition`, which has nearly 250 subscribers and hosts recordings of previous years' sessions. This year's presentations will also be uploaded there. If you have additional content, beyond your WoRMS submission, that you would like to share on this channel, please get in touch with us.

We look forward to engaging presentations and discussions and hope to see many of you again next year.

Jorge Calvo-Zaragoza, Alexander Pacha, and Elona Shatri

Contents

Can multimodal large language models read music score images?

Jorge Calvo-Zaragoza, Eliseo Fuentes-Martínez, Noelia Luna-Barahona, Antonio Ríos-Vila

Pattern Recognition and Artificial Intelligence Group, *University of Alicante*, Spain

Abstract—This paper investigates whether multimodal large language models (MLLMs), which combine visual and textual understanding, can effectively read and interpret music score images. Given their ability to process and integrate information from multiple modalities, MLLMs present a promising approach for Optical Music Recognition (OMR). Through empirical evaluation, we demonstrate that while MLLMs exhibit potential in recognizing musical structures, challenges remain in addressing the complexity of music notation. This work highlights the need for further refinements in MLLM architectures to improve their effectiveness in OMR tasks.

Index Terms—Multimodal Large Language Models, Optical Music Recognition, Music Information Retrieval.

I. Introduction

Optical Music Recognition (OMR) is a challenging area of research that studies how to computationally read music notation in documents [1]. Traditional OMR systems rely on specific computer vision and machine learning techniques to identify musical symbols [2], but modern advances of deep learning, particularly the development of multimodal large language models (MLLM), have opened up new possibilities for interpreting music scores.

MLLMs integrate information from both visual and textual inputs and have shown remarkable success in tasks that require an understanding of multiple modalities, such as image captioning and visual question answering [3], [4]. This paper explores whether MLLMs can be leveraged to interpret music score images by processing both the visual aspects of the score and the symbolic structure of the music.

The question we seek to answer is: Can MLLM be used to the task of reading music score images? We hypothesize that while MLLM have the potential to recognize some elements of music notation, the unique challenges posed by the structure and complexity of music require further adaptation of existing architectures. While this might be no surprise, no previuos work has evaluated this scenario.

II. Methodology

This is a preliminary work to evaluate the capabilities of general MLLM for reading music scores. Each model is tested with the same set of cropped music score images, and their outputs were analyzed to determine the extent to which they can interpret and describe music notation. For such reason, we selected a tiny sample of music score image crops to these

The authors appear in alphabetical order.

general-purpose MLLM. We informally build specific prompts to assess different capabilities regarding sheet music reading. All the components of our study are described below.

A. General models

We aim to assess how general-purpose MLLMs, which have been successful in disparate fields, perform when faced with the task of reading sheet music or retrieving some specific information from music score images. Below, we provide a brief overview of the general-purpose MLLMs tested in our experiments: **ChatGPT (GPT-4V)** [5], **Gemini** [6], **Llama 3.2** [7], **Mistral 7B**, and **Claude 3.5** [8].

B. Sample

The (tiny) set of samples selected to evaluate the MLLMs are depicted in Fig. 1, including different textures such as monophonic (mono), pianoform (piano), and vocal textures.[1] As can be observed, apart from the variability in textures, the images are relatively simple in terms of graphic complexity. In addition, they are rather well-known culturally and socially.

C. Capabilities

We identify four interesting capabilities to assess the MLLMs. These were translated into four questions (prompts) that are outlined below:

- **Q1: Piece recognition**: Evaluates whether the model can identify the composition from a cropped score. This capability tests the model's broader cultural understanding and whether it can associate visual notation with specific compositions or composers.
- **Q2: Transcription**: Assesses the model's ability to convert music notation to a symbolic format. This is a core OMR task, requiring the model to interpret the visual layout of the music notation.
- **Q3: Tonality identification**: Tests if the model can infer the score's tonality. It requires both graphical recognition and some understanding of basic music notation.
- **Q4: Texture classification**: Examines if the model can recognize the type of musical texture. This is a simple graphical task, but requires understanding of the layout of sheet music.

[1]The images were taken from **IMSLP Petrucci Music Library**. Accessed September 30, 2024. International Music Score Library Project. https://imslp.org/.

(a) **Mono**: Excerpt of *Symphony No. 9 in D minor, Op. 125* by L.V. Beethoven"

(b) **Piano**: Excerpt of *Piano Sonata No. 11 in A major, K. 331* by W.A. Mozart.

(c) **Vocal**: excerpt of *My Way*, lyrics by Paul Anka, music by Claude François and Jacques Revaux.

Fig. 1: Sample of images used for evaluating the capabilities of the MLLM, involving different music textures (monophonic, pianoform, and vocal).

Each task was designed to capture a specific facet of music reading, from broad cultural knowledge (Q1) to technical transcription skills (Q2), as well as simpler graphic recognitions (Q3 and Q4).

The specific prompts for answering these questions were carefully formulated with the help of the MLLM itself to ensure that they were worded in the best possible way.

III. RESULTS

The evaluation of the models across the four questions reveals significant differences in their capabilities. A summary of our evaluation is given in Table I.

For **Q1**, the models generally failed to identify the musical piece from the score, as they could not interpret enough musical information. The exception was a vocal example, where some models successfully identified the song "My Way" due to their ability to recognize and process the lyrics, highlighting their reliance on textual rather than musical data for recognition.

In **Q2**, all models performed poorly, unable to transcribe the music notation into any symbolic format.[2] A minor exception was observed in vocal music, where the models managed to

---

[2]As mentioned above, the prompts were built using the model itself. In this sense, each model was asked for the output format they claimed to know (MusicXML or ABC, mainly).

MLLM Model	Input	Questions			
		Q1	Q2	Q3	Q4
GPT-4V	Vocal	✓	∼	✓	✓
	Mono	×	×	✓	✓
	Piano	×	×	×	✓
Gemini	Vocal	∼	×	✓	∼
	Mono	×	×	✓	✓
	Piano	×	×	×	∼
Llama 3.2	Vocal	×	∼	✓	×
	Mono	∼	×	✓	∼
	Piano	×	×	×	✓
Mistral 7B	Vocal	✓	×	✓	✓
	Mono	×	×	×	∼
	Piano	×	×	∼	∼
Claude 3.5	Vocal	✓	∼	✓	✓
	Mono	×	×	✓	✓
	Piano	×	×	✓	✓

TABLE I: Observed performance of the MLLM. ✓denotes the cases where the model is able to provide a reasonable or accurate answer; ∼means that the model does not provide a correct answer but exhibits some knowledge about the task; ×indicates the cases where the model clearly fails.

transcribe the lyrics accurately, but their transcription of music notation remained inaccurate.

In contrast, the models performed better in **Q3** and **Q4**. For Q3, most models could infer the tonality with reasonable accuracy, suggesting that they could identify key signatures based on visual cues, even without detailed transcription of the music. For Q4, the models were generally accurate, particularly GPT-V4 and Claude 3.5, demonstrating that they can detect visual patterns related to musical structure, even struggling with specific notational details.

Overall, the results indicate that while the models are not equipped to read music, they are capable of extracting some visual information. This highlights the potential of MLLMs for founding music reading systems, although significant improvements are required for their use in detailed OMR tasks.

IV. CONCLUSIONS

In this paper, we explored the potential of multimodal large language models (MLLMs) for understanding and interpreting music score images, a task traditionally handled by Optical Music Recognition (OMR) systems. Our preliminary experiments demonstrated that while MLLMs exhibit certain capabilities, such as recognizing lyrics in vocal music and identifying musical features like tonality and texture, they still struggle significantly with tasks that require detailed interpretation of musical notation.

Future work could focus on fine-tuning these MLLMs specifically for music score reading tasks, using techniques such as low-rank adaptation (LoRA) to adjust their weights for OMR tasks, or retrieval-augmented generation (RAG) approaches to enhance their ability to reference symbolic music knowledge.

References

[1] Jorge Calvo-Zaragoza, Jan Hajič Jr, and Alexander Pacha. Understanding optical music recognition. *ACM Computing Surveys (CSUR)*, 53(4):1–35, 2020.

[2] Elona Shatri and György Fazekas. Optical music recognition: State of the art and major challenges. *arXiv preprint arXiv:2006.07885*, 2020.

[3] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

[4] Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*, 2024.

[5] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[6] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[7] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[8] Anthropic. Claude 3.5. https://www.anthropic.com, 2024. Language model developed by Anthropic.

Towards Sheet Music Information Retrieval:
A Unified Approach Using Multitask Transformers

Antonio Ríos-Vila[1], Eliseo Fuentes-Martinez[1], Jorge Calvo-Zaragoza[1]
[1] Pattern Recognition and Artificial Intelligence Group, University of Alicante, Spain

Abstract—Sheet Music Information Retrieval (SMIR) is a novel and rapidly evolving field within Music Information Retrieval that aims to extract, analyze, and retrieve information from sheet music documents. This discipline encompasses a wide range of tasks, including Optical Music Recognition (OMR), Optical Character Recognition (OCR), layout analysis, and content-based retrieval. SMIR has significant applications in musicology, digital libraries, and music education, enabling researchers and musicians to interact with and analyze large collections of sheet music more efficiently. Recent advancements in SMIR have been largely driven by Deep Learning-based approaches dedicated to specific tasks, which currently show remarkable improvements in accuracy and robustness compared to traditional methods. However, these approaches are isolated for their specific tasks, leading to a fragmented landscape of solutions and increased complexity in developing comprehensive SMIR applications. In this paper, we research in briefly defining SMIR and addressing its challenges through an end-to-end approach using multitask learning and language modeling techniques. We present the Sheet Music Information Retrieval Transformer (SMIReT), an Transformer-based deep learning model that unifies multiple SMIR tasks within a single framework. Built upon the Sheet Music Transformer architecture, SMIReT leverages task-specific prompting and a unified vocabulary to handle diverse SMIR tasks seamlessly. We evaluate our model on the Mottecta corpus, a collection of early notation documents from the 17th century. Results demonstrate the ability of to perform multiple SMIR tasks within a single framework, showing promising results and challenges for the future of SMIR.

Index Terms—SMIR, Transformer, Mensural notation, Multitask learning

I. Introduction

The field of Optical Music Recognition (OMR) has evolved significantly from its conception [1], evolving from multi-stage statistical learning pipelines [2], [3] to *end-to-end* deep learning-based approaches, where notation primitive detection and assembly detection [4]–[8] and sequence generation-based transcription [9]–[12] mainly domain the state of the art in the field. This progress has led to advanced systems capable of extracting more than just the content of music scores, having for example Layout Analysis for detecting regions of interest [13]–[15] or search systems based on transcriptions [16]. This progress has led to multiple practical applications in the musicology field, where users are able to work hands-on with these technologies to process music scores [17], [18]. Despite these significant advances, practical OMR applications often still require task-specific models to extract all the information form a music score. This is inconvenient in terms of computing resources and maintainability. The main reason this has happened is because there has never

been the perspective of recognizing music as a whole when developing these systems. The same way analogous fields, such as Optical Character Recognition (OCR) and Handwritten Text Recognition (HTR) are shifting towards a task unification of seemingly isolated tasks through end-to-end models under the umbrella of Document Understanding [19], [20], music should shift towards this. This end-to-end paradigm represents a promising direction for automatic information extraction from documents, as not only all tasks are resolved through the same method, but their information helps to produce more accurate results in other tasks. In this paper, we briefly define how this evolution can be formulated through the Sheet Music Information Retrieval (SMIR) challenge. Then, we explore if end-to-end state-of-the-art OMR can also go make this step further. To do so, we introduce a first solution based on autoregressive Transformers, curriculum learning and task-specific prompting [21]. We test this approach with an early notation corpus, which is one of the application targets of musicological tools [17]. Results indicate that the approach is viable with promising performance, although several improvements are still required.

II. Sheet Music Information Retrieval

This paper refers to the challenge of SMIR. Whereas there is not a formal definition, SMIR is a specialized field within Music Information Retrieval (MIR) that focuses on extracting, analyzing, and retrieving information from sheet music documents. That is, SMIR serves as an umbrella term for several tasks, such as OMR, Layout Analysis or content-based search. Since this is the first time the term is defined and proposed, there are no specific tasks that settle the challenge. In this paper, we propose which tasks—based on state of the art— should be considered to compose the SMIR challenge. These tasks are grouped in three families: parsing, layout recognition and queries.

A. Parsing tasks

The first group is composed by the tasks that involve end-to-end content extraction from the music scores. This mainly involves OMR, as it is the field that primarily studies the extraction of music content from score documents. Note, however, that text extraction-related tasks should also be considered, as some score documents may contain text paragraphs or lyrics. Given this, we propose parsing tasks to be Full Parsing, where all the content of the document is extracted,

OMR, where only music is recognized and OCR, where the model should output only the text sections of the document.

B. Layout recognition tasks

The second group is related to the detection and extraction of the graphical elements of the music score, this is mainly addressed by the Layout Analysis field. We, therefore, formulate this group as an object detection task, where we benchmark both region of interest extraction and classification.

C. Query-based tasks

The queries group refers to all the tasks where the system must give an answer based on the input document and specific instructions. This group serves as a proxy for user interaction with the system. In this case, we propose two main tasks. The first one, named *selective* OMR, refers to the partial transcription of the music score given a specific set of bounding boxes. This way, we assess the awareness of the model to the structure of the music score, as well as its capability to guide the reading in a non-hierarchical reading order[1]. Then, we also propose pattern search queries, where the user inserts a specific music sequence—of a varying number of notes— and the model outputs the bounding boxes of the regions in the score that contain the pattern, or none if not found.

III. SHEET MUSIC INFORMATION RETRIEVAL TRANSFORMER

In this paper, we present the Sheet Music Information Retrieval Transformer (SMIReT) model, which is a next-step of the Sheet Music Transformer (SMT) transcription architecture to address SMIR.

A. Sheet Music Transformer

The SMT is an autoregressive neural network designed for music transcription [22], [23]. It is composed of two key components: an encoder and a decoder. The encoder functions as a feature extractor, taking an input image x and producing a feature map x'_e. The decoder, built upon an autoregressive conditioned language model, predicts the probability of each symbol in the vocabulary at a given timestep. This prediction is based on both the output of the encoder and the sequence of previously generated symbols, formalized as:

$$\hat{y} = \underset{\hat{y} \in \Sigma}{} P(\hat{y}_t \mid x'_e, (\hat{y}_0, \hat{y}_1, \hat{y}2, ..., \hat{y}_{t-1})) \tag{1}$$

Here, Σ represents the comprehensive symbol vocabulary encoding musical content, x'_e is the encoded feature map, and t denotes the current timestep.

1) Encoder: The encoder processes an input image $x \in \mathbb{R}^{c \times h \times w}$, where h, w, and c represent height, width, and number of channels, respectively. Leveraging Convolutional Neural Networks, the encoder transforms this input into a set of c_e two-dimensional feature maps, denoted as $x_e \in \mathbb{R}^{h_e \times w_e \times c_e}$. The dimensions h_e and w_e are related to the original image dimensions by factors r_h and r_w, representing the downscaling effect of the network.

---

[1]Bear in mind that full parsing always follows a specific reading order given by the layout of the page.

2) Decoder: The decoder is built upon the Transformer architecture, currently the state-of-the-art approach for conditional sequence generation tasks. At each timestep t, the decoder generates a probability distribution $p_t \in \mathbb{R}^{|\Sigma|}$ over the symbol vocabulary Σ. This distribution is conditioned on both the output of the encoder x'_e and the previously predicted tokens $(\hat{y}0, \ldots, \hat{y}t-1)$. The prediction process begins with a special start-of-transcription symbol and concludes upon generating an end-of-transcription token. To bridge the dimensional gap between the 2D output of the encoder and the sequential nature of the decoder, the feature map is flattened. To preserve the spatial intricacies of full-page music scores, a two-dimensional positional encoding is integrated into the feature maps before flattening [24], [25].

B. SMIReT: a multitask SMT for SMIR

To achieve multitask processing capabilities, the SMIReT model adapts the SMT through task prompting, as in other Document Understanding approaches [19]. These prompts act as task-specific cues, allowing the model to adapt its behavior based on the desired output. Referring to Equation 1, we modify the input of the decoder as:

$$d = \mathbf{p} \cup (\hat{y}_0, \hat{y}_1, \hat{y}2, ..., \hat{y}_{t-1}) \tag{2}$$

where d is the decoder input and $\mathbf{p}$ is the prompt sequence, tokenized through the prompt vocabulary Σ_p. Therefore, equation 1 is expressed as:

$$\hat{y} = \underset{\hat{y} \in \Sigma}{} P(\hat{y}_t \mid x'_e, d) \tag{3}$$

By incorporating different tokens to the prompts and unifying the input and output vocabulary, the SMT can perform all the tasks that are described in Section II end-to-end. An example of this mechanism is shown in Figure 1.

One of the challenges of following this specific formulation is the unification of the SMIR vocabulary, which is multimodal by the diversity of tasks, through a single language model. To approach this, the SMIReT output vocabulary is composed of music tokens in agnostic encoding, where notes are depicted by its shape and position, characters for text, absolute positions for bounding boxes—following the Pix2Seq methodology [26]—and special tokens for music region categories.

C. Training procedure

Since we are dealing with an autoregressive Transformer, we perform a curriculum-based learning with synthetic generation, which is composed of two main processes:

- **Full parsing training**: The model is trained in full-page parsing with synthetic samples. This training is done incrementally, feeding the model with pages with an increasing amount of music staves with text regions to transcribe.
- **Target fine-tuning**: The pretrained model is fine-tuned in all the SMIR tasks at the same time. In this case, we follow also an incremental curriculum learning, where synthetic samples are interleaved with target ones.
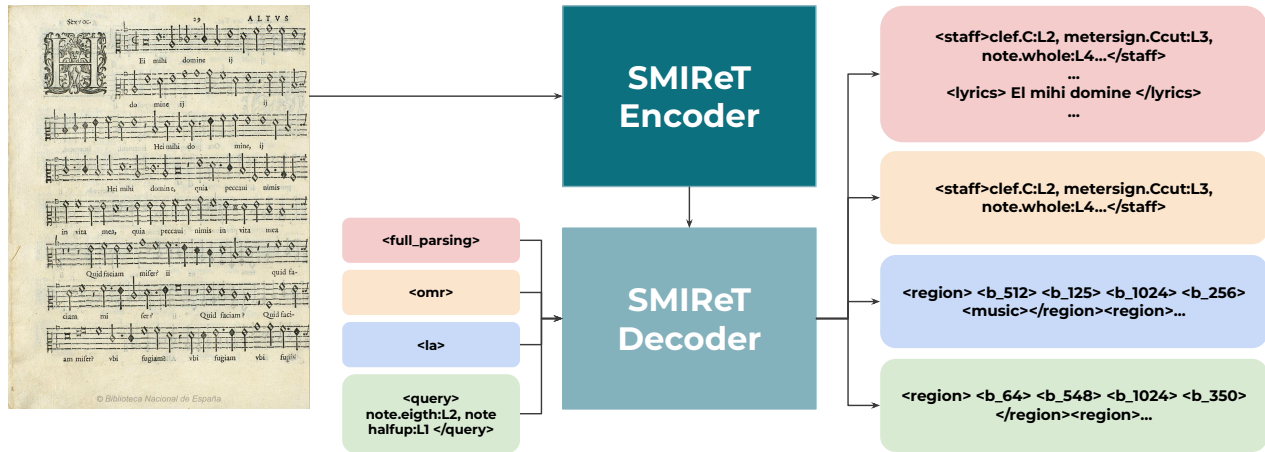
Fig. 1: Architecture of the SMIReT model with some examples of task prompting.

IV. EXPERIMENT

A. Data

We evaluated the SMIReT model in the application of information extraction from early notation documents. These datasets are, up to the moment, the ones that contain the majority of information required to target SMIR, due to the effort hat has been put for their digital preservation [17]. We experimented with the MOTTECTA corpus [27], which is a set of 297 printed pages from a collection of Mensural books of the "Biblioteca Digital Hispánica" dated from the 17th century completely labeled, both regions and text, covering both parsing and layout analysis tasks. In the case of query-based approaches, we randomly generate them through the information given in the dataset, by selecting a specific set of regions in the case of selective OMR and by picking up chunks of the ground truth from pages in the case of pattern-matching queries. The datasets has been split into fixed partitions, where 60% of the samples have been used for training, 20% have been used for validation, and 20% for testing.

For synthetic generation, we construct samples through the PRIMENS dataset, which is a large collection of synthetically rendered mensural music incipits [27].

B. Results

Table I discloses the performance reported of the SMIReT model with the test set of the MOTTECTA dataset.

TABLE I: Results of the performance obtained by the SMIReT model on the different tasks proposed for SMIR for the MOTTECTA dataset.

Task	Metric	SMIReT
Parsing tasks		
Full parsing	Music SER	6.05
	Text CER	15.30
OMR	SER	5.92
OCR	CER	10.08
Layout recognition tasks		
Region detection	IoU	70.23
Classification	F1	97.00
Query-based tasks		
Selective OMR	SER	41.55
Pattern match	Accuracy	73.80
	IoU	75.03

First of all, we observe that the SMIReT model is capable of learning all the SMIR tasks successfully and show acceptable performance.

Results on parsing tasks reveals intriguing dynamics in the SMIReT multitask learning capabilities. In isolated tasks, the model demonstrates superior performance in, 5.92% SER in OMR and 10.08% CER in OCR. However, when faced with the full parsing task that combines both music and text recognition, we observe a slight degradation—2.9%—in music recognition and a more substantial decline in text recognition, a 51.78%. This disparity suggests a potential bias in the

attention mechanism of the model towards musical elements when processing mixed content. The relative stability of music recognition performance in the presence of text, contrasted with the more significant deterioration of text recognition in the presence of musical notation, indicates that the features learned for music recognition are more robust and less susceptible to interference. This phenomenon may be attributed to the more structured and standardized nature of musical notation compared to the variability inherent in textual elements found in sheet music.

When analyzing the layout recognition tasks, a paradox emerges, where the model demonstrates high classification accuracy, 97.00% F1 score, but moderate performance in region detection, with 70.23% of IoU, notably below to state-of-the-art Layout Analysis techniques [15]. This discrepancy suggests that while the model excels at recognizing the nature of different regions within a sheet music image, it struggles with precisely localizing these regions or defining their boundaries. This suggests that, perhaps, the model is learning the structure of the music document through the language model, but it may not necessarily correspond to precise spatial information, where some regions, as Figure 2 shows, may be avoided by the network.
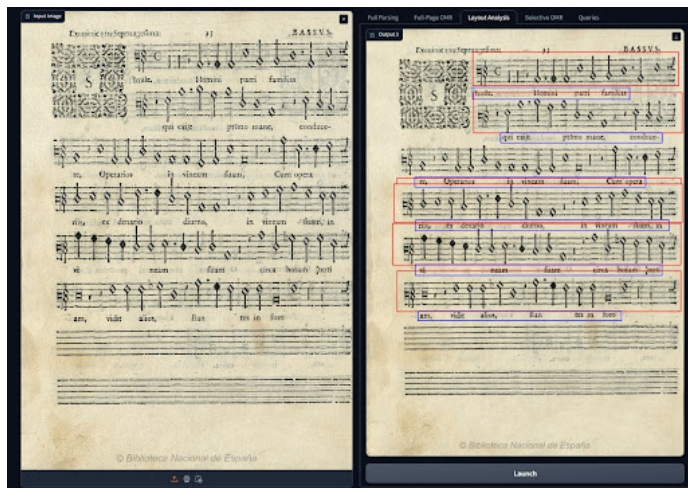

Fig. 2: Visualization of the SMIReT performance on a test sample of the MOTTECTA dataset in the tasks of layout recognition.

The analysis on the query-based tasks reveals that the model is able to detect patterns and correlate them to the image general features, shown by the 73% accuracy on the pattern matching task. However, when giving pixel-wise contextual information both in the input, through the selective OMR, and in the output, reported 75.03% IoU, the model struggles in the same way as in the layout recognition tasks. This points to a potential shortcoming in the integration between the model visual processing capabilities and its natural language understanding or instruction-following modules. Addressing this limitation could involve developing more sophisticated attention mechanisms that can dynamically focus on relevant

parts of the input based on query requirements, as well as improving the ability of the model to ground natural language queries in the visual domain of sheet music.

V. CONCLUSION

In this paper, we present the Sheet Music Information Retrieval (SMIR) challenge, a novel research field in Music Information Retrieval that seeks to extract the information from music score documents. We research on the capabilities of deep learning models to be able to address the challenge in an end-to-end fashion. To do so, we propose the Sheet Music Information Retrieval Transformer (SMIReT) model, a Transformer-based model that adapts state-of-the-art OMR to adress multitask learning.

Our model demonstrates the feasibility of addressing SMIR tasks—including full parsing, OMR, OCR, layout recognition, and query-based retrieval—within a single, unified framework. The evaluation on the MOTTECTA corpus reveals promising results. However, our study also uncovers several challenges that warrant further investigation. The performance disparity between music and text recognition in mixed content scenarios suggests a need for more balanced feature learning. The discrepancy between high classification accuracy and moderate region detection performance in layout analysis tasks indicates room for improvement in spatial understanding. Additionally, the model struggles with pixel-wise contextual information in query-based tasks highlight the need for enhanced integration between visual processing and language understanding components.

These findings open up several avenues for future research. Developing more sophisticated attention mechanisms could improve the ability of the model to focus on relevant parts of the input based on task requirements. Furthermore, exploring ways to balance the learning of features for different modalities (music notation, text, spatial information) could lead to more robust performance across all SMIR tasks.

REFERENCES

[1] Jorge Calvo-Zaragoza, Jan Hajič Jr., and Alexander Pacha. Understanding optical music recognition. *ACM Comput. Surv.*, 53(4), 2020.

[2] David Bainbridge and Tim Bell. The challenge of optical music recognition. *Computers and the Humanities*, 35:95–121, 2001.

[3] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre RS Marcal, Carlos Guedes, and Jaime S Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, 2012.

[4] Alexander Pacha and Horst Eidenberger. Towards a universal music symbol classifier. In *14th International Conference on Document Analysis and Recognition*, pages 35–36, Kyoto, Japan, 2017. IAPR TC10 (Technical Committee on Graphics Recognition), IEEE Computer Society.

[5] Yaqi Song, Yun Shen, Peng Ding, Xuezhi Zhang, Xiaohou Shi, and Yuying Xue. Optical music recognition based deep neural networks. In *Signal and Information Processing, Networking and Computers*, pages 1051–1059, Singapore, 2022. Springer Nature Singapore.

[6] Francisco Fernández De Vega, Jorge Alvarado, and Juan Villegas Cortez. Optical Music recognition and Deep Learning: An application to 4-part harmony. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 01–07, 2022.

[7] Alexander Hartelt and Frank Puppe. Optical medieval music recognition using background knowledge. *Algorithms*, 15(7), 2022.

[8] Ali Yesilkanat, Yann Soullard, Bertrand Coüasnon, and Nathalie Girard. Full-page music symbols recognition: State-of-the-art deep model comparison for handwritten and printed music scores. In *Document Analysis Systems*, pages 327–343, Cham, 2024. Springer Nature Switzerland.

[9] Jorge Calvo-Zaragoza and David 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*, pages 248–255. ISMIR, November 2018.

[10] Jorge Calvo-Zaragoza, Alejandro H Toselli, and Enrique Vidal. Handwritten Music Recognition for Mensural notation with convolutional recurrent neural networks. *Pattern Recognition Letters*, 128:115–121, 2019.

[11] María Alfaro-Contreras, Antonio Ríos-Vila, Jose J. Valero-Mas, José M. Iñesta, and Jorge Calvo-Zaragoza. Decoupling music notation to improve end-to-end optical music recognition. *Pattern Recognition Letters*, 158:157–163, 2022.

[12] Pau Torras, Arnau Baró, Lei Kang, and Alicia Fornés. On the Integration of Language Models into Sequence to Sequence Architectures for Handwritten Music Recognition. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 690–696. ISMIR, 2021.

[13] Vicente Bosch Campos, Jorge Calvo-Zaragoza, Alejandro H Toselli, and Enrique Vidal Ruiz. Sheet music statistical layout analysis. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 313–318. IEEE, 2016.

[14] Francisco J Castellanos, Jorge Calvo-Zaragoza, and Jose M Iñesta. A neural approach for full-page optical music recognition of mensural documents. In *Proc. of the 21th Int. Society for Music Information Retrieval Conference*, pages 12–16, 2020.

[15] Francisco J. Castellanos, Carlos Garrido-Munoz, Antonio Ríos-Vila, and Jorge Calvo-Zaragoza. Region-based layout analysis of music score images. *Expert Systems with Applications*, 209:118211, 2022.

[16] Ichiro Fujinaga, Andrew Hankinson, and Julie 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*, pages 1–3, 2014.

[17] David Rizo, Jorge Calvo-Zaragoza, and José M Iñesta. Muret: A music recognition, encoding, and transcription tool. In *Proceedings of the 5th international conference on digital libraries for musicology*, pages 52–56, 2018.

[18] Andrew Noah Hankinson. *Optical music recognition infrastructure for large-scale music document analysis*. McGill University (Canada), 2014.

[19] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, 2022.

[20] Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents, 2023.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[22] Antonio Ríos-Vila, Jorge Calvo-Zaragoza, and Thierry Paquet. Sheet music transformer: End-to-end optical music recognition beyond monophonic transcription. In *Document Analysis and Recognition - ICDAR 2024*, pages 20–37, Cham, 2024. Springer Nature Switzerland.

[23] Antonio Ríos-Vila, Jorge Calvo-Zaragoza, David Rizo, and Thierry Paquet. End-to-end full-page optical music recognition for pianoform sheet music, 2024.

[24] Denis Coquenet, Clément Chatelain, and Thierry Paquet. Dan: a segmentation-free document attention network for handwritten document recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8227–8243, 2023.

[25] Sumeet S. Singh and Sergey Karayev. Full page handwriting recognition via image to sequence extraction. In Josep Lladós, Daniel Lopresti, and Seiichi Uchida, editors, *16th International Conference on Document Analysis and Recognition, ICDAR 2021, Lausanne, Switzerland, September 5-10, 2021, Proceedings, Part III*, volume 12823 of *Lecture Notes in Computer Science*, pages 55–69. Springer, 2021.

[26] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021.

[27] Juan C Martinez-Sevilla, Adrian Rosello, David Rizo, and Jorge Calvo-Zaragoza. On the performance of optical music recognition in the absence of specific training data. In *ISMIR*, pages 319–326, 2023.

Semantic Reconstruction of Sheet Music with Graph-Neural Networks

Grégoire de Lambertye
TU Wien, Austria
e12202211@student.tuwien.ac.at

Alexander Pacha
*Institute of Visual Computing
and Human-Centered Technology
TU Wien, Austria*
alexander.pacha@tuwien.ac.at

Abstract—**Optical Music Reconstruction (OMR) is a field of research that investigates how to computationally read music notation. Many OMR systems operate by first detecting all objects in an image, and then using heuristics to recover the relationships between the musical primitives to reconstruct their semantics. These heuristics are inherently limited, and there is a significant lack of research on performing the semantic reconstruction more adequately.**

This paper investigates how Graph Neural Networks (GNNs) can be used to perform the semantics reconstruction of notated music. We developed a versatile pipeline and demonstrated the capacity of GNNs to effectively recover the relations between the musical primitives. However, challenges related to the instability and sensibility of the GNNs indicate that, despite their potential, these models may not be the optimal solution for this task either.

Index Terms—**Optical Music Recognition, Graph Neural Network, Link Prediction, Semantic Reconstruction**

I. Introduction

Many OMR systems divide the task of reading music into a 4-stage pipeline. This pipeline starts with *image pre-processing*, followed by the *music object detection* stage, which retrieves the locations of all musical primitives, and assigns each element a class label. The third stage is the *semantic reconstruction*, which attempts to recover the relationships between the primitives. The last stage is called *encoding* and converts the internal representation into a standardized format such as MusicXML.

A useful representation for recovering the semantics of music notation is the Music Notation Graph (MuNG), illustrated in Figure 1. The notion of a MuNG has been used before [1], [2], [3], but there is no commonly accepted definition; the shared understanding is that musical primitives (e.g., noteheads, accidentals, or clefs) are the nodes of the graph, and an edge represents a relationship between two primitives. Definitions of MuNGs vary primarily in how these edges are constructed: for instance, some MuNGs include edges between accidentals and noteheads, while others exclude them.

This paper investigates how to construct MuNGs from the output of a music object detector with GNNs, more specifically how to predict the existence of syntactic edges between primitives that form notes. GNNs are a class of machine learning models introduced by Gori and Scarselli [4]. Unlike traditional neural networks that operate on regular grid-like structures, GNNs directly process graph-structured data. They can learn node embeddings over aggregated information from a neighborhood, and demonstrate state-of-the-art capabilities in link prediction.
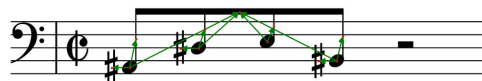

Fig. 1: Music Notation Graph

II. Related Work

The term Music Notation Graph (MuNG) has first been used by Hajič et al. [5] to build the MUSCIMA++ dataset. This format has then been adopted by other datasets such as MusiGraph [1], and DoReMi [6]. In Pacha et al. [3], the authors formulate the link prediction task as a binary classification problem and apply a Convolutional Neural Network to construct a MuNG. In [1] Baró et al. construct MuNGs by leveraging GNNs. While most of their architecture is kept private, they claim very good results and a Music Error Rate of 5%.

III. Semantic Reconstruction With Graph Neural Networks

GNNs require graph-structured data as input. To make use of the output of a Music object detector, we can transform the list of detected objects into a feature matrix. Instead of directly constructing the adjacency matrix, we propose to build an over-complete graph called *Candidate Graph* (CG), which is then pruned using the GNN to obtain the final graph. Figure 2 illustrates the proposed pipeline.

A. Building the Feature Matrix

The feature matrix contains 1 row for each detected musical primitive and stores its size and position on the page, as well as a one-hot encoding of its class label.

The first challenge is the *set of classes*—the vocabulary that is used to encode the musical primitives—which can be different for each encoding and each dataset. In the simplest case, the same concept just has a different name, e.g., *noteheadFull* vs. *noteheadBlack*. However, in some cases it gets more complicated when datasets have a different granularity, e.g., a *flag* being split into the classes *flagUp* and *flagDown*.
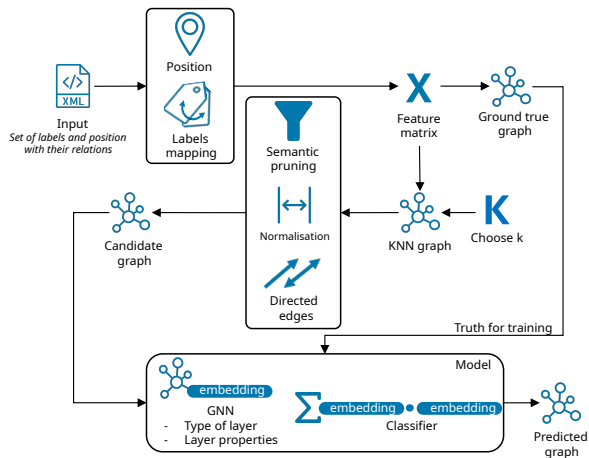
Fig. 2: Training Pipeline

Generally, it's beneficial to have the most detailed granularity, as it can be simplified to a reduced set of classes while maintaining potentially relevant information (e.g., the direction of a stem might help to determine the voice in a polyphonic staff).

We propose to map the set of classes of a given dataset to a reduced set of classes that is optimized for reconstructing the relationships between primitives. Input classes that never form relations can be filtered and removed. The mapping also doesn't need to be bijective, because we can use the IDs of each object that is coming from the object detector to retrieve the original class for each object.

In our experiments, the models learned more efficiently with smaller, coarser sets of classes, e.g., when using a single class for all *flags* instead of multiple classes (*8thFlagUp, 8thFlagDown, 16thFlagUp, ...*). The 2 final sets of classes in our experiments have only 6 and 10 classes respectively. The details are given in the Appendix.

B. Building the Adjacency Matrix

To directly construct the adjacency matrix using a GNN would be ideal; however, GNNs require an input graph to operate. After processing, undesired edges can be removed to perform link prediction. The simplest approach to obtain an initial graph would be to construct a fully-connected graph. However, this is computationally prohibitive for large graphs. If we used GNNs without any connections, they would not work efficiently either, as they use the edges for information to flow from one node to another.

Given that related music primitives are spatially close to one another, a K-Nearest Neighbors (kNN) approach seems suitable. An exploration of the MusiGraph dataset shows, that after removing primitives that never form relations, k=13 is sufficient to include every relation from the ground truth. It is important to choose k sufficiently high, as a missing edge from the CG would also be missing from the final adjacency matrix, given we only cut edges.

To further optimize our pipeline, grammatical rules are applied to semantically prune the initial kNN graph, removing edges that would not exist in an errorless MuNG; for example, links between 2 noteheads are pruned. We also normalize the scores: the top-most, bottom-most, left-most, and right-most primitive bounding box edges are used to perform a min-max normalization. This normalization ensures that the pipeline can work with different fonts or handwritten notation as well as images of any size. While normalization is usually beneficial, we observe a negative impact for GNNs: the distance between certain objects (e.g., notehead and stem) is usually stable due to the typesetting process. With normalization, these scores are distorted and the model cannot learn from these distances.

C. Model architecture

The model consists of a GNN, which learns a node embedding, and a classifier, which decides whether an edge should be kept or pruned. The CG edges are considered undirected to enable the information to flow both ways along an edge in the GNN. The GNN is composed of 3 GraphSAGE layers, separated by ReLU activation functions. The intermediate representation has a size of 2048 and the final output embedding has a size of 1024. The classifier takes those embeddings and calculates the cross-product between 2 edge endpoints. If the value is above 0.5, the edge is kept, otherwise it is pruned.

We use the Binary Cross-Entropy loss function to train our models and a learning rate scheduler which reduces the learning rate if the validation loss stagnates for 15 epochs. While we started with a standard early-stopping mechanism, we observed that the training and validation losses initially decreased and then started to increase again. To force the model to explore promising areas we implement a novel mechanism that we call *jump back on learning rate change*, illustrated in Figure 3. It resets the weights to the best-saved configuration when decreasing the learning rate. This approach automates a training restart from a checkpoint that previously had the lowest validation loss.
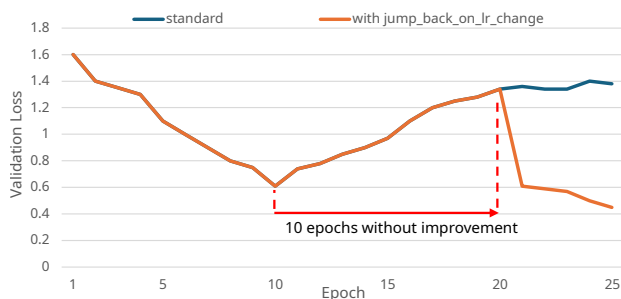

Fig. 3: Illustration of the *jump back on learning rate change* mechanism: Epoch 10 yields the best validation loss with an initial learning rate. After 10epochs without improvement, the learning rate is reduced and the snapshot from epoch 10 gets loaded.

IV. METRICS

OMR lacks a universally accepted, intrinsic metric that is easily interpretable and corresponds directly to the proportion of recovered musical information. Such a metric would enable reliable comparisons of OMR system performance without the need for expensive user studies. However, some standard metrics remain useful for evaluating OMR systems. In the specific case of MuNGs, we can define 2 types of metrics: those based on the binary classification aspect of the task and those based on its graph structure.

The task of music semantic reconstruction can be formulated as a binary classification problem, where each edge in the CG is classified as either existing or not. This allows the evaluation of a model using standard binary classification metrics such as accuracy. However, the informative value of these metrics is limited: Adding trivial or easily removable edges to a CG can increase accuracy without improving the meaningfulness of the predicted graph. Thus, while binary classification metrics reflect how well a model has learned, they do not necessarily capture how valuable its output is.

Some limitations of these metrics disappear if we evaluate the output as a graph. A more robust approach is to use the Graph Edit Distance (GED) as defined in equation 1. It measures the sequence of least-cost edit operations required to transform one graph into another. This metric is effective when comparing different models on the same score; however, it becomes problematic when comparing performance across different scores, such as monophonic vs. polyphonic music. Differences in the size of ground truth graphs make GED values incomparable. To address this, the Music Error Rate (MER), defined in equation 2 can be more meaningful. It normalizes the number of edit operations by the size of the graph (see [7] for more details).

$$\text{GED}(G_1, G_2) = \min_{(o_1, o_2, \ldots, o_k)} \sum_{i=1}^{k} c(o_i) \qquad (1)$$

With $o_1, \ldots, o_k$ a set of operations that transforms G_1 into $G2$ and $c(o_i)$ the cost of the operation i.

$$\text{MER} = \frac{I + R + S}{T} = \frac{GED}{T} \qquad (2)$$

Where I, R, and S are the number of insertions, deletions, and substitutions to obtain the ground truth sequence. T is the number of edges in the ground graph.

While the MER provides a more balanced comparison, it is still not flawless, as the length of the ground truth graph does not always correlate with the complexity of the musical notation. Moreover, there is no single standard for constructing edges in a MuNG, meaning that multiple valid MuNGs could represent the same musical score. Consequently, graph-based metrics are only comparable across MuNGs that have been built according to the same rules.

V. INSTABILITY

In the course of our experiments, we encountered significant challenges related to the instability and reproducibility of results. The first barrier to reproducible and stable results is inherent to the library we used: PyTorch Geometric. Although a seed is set to control many sources of randomness, some operations retain non-deterministic behavior during GPU execution [8]. The second barrier is inherent to GNNs, which are known to be unstable [9]. While we aimed for reproducible experiments, we noted that different seeds led to vastly different outcomes.

VI. RESULTS

It is important to acknowledge that achieving a 0% Graph Edit Distance or Music Error Rate is not a realistic expectation in this study. The datasets employed, such as MusiGraph, inherently contain an unknown number of errors. The scores in the dataset MUSCIMA++ and DoReMi have been divided by measure to align with MusiGraph characteristics. This division process certainly introduced some errors as well [7]. Table I shows the performance of different models for each dataset using the *10-labels* class set, described in Appendix. To get a better impression of the performance of these models, Figures 4, 5, and 6 visualize the predicted graph for 3 different models.

The algorithm that is used to divide the scores has a couple of drawbacks including that some one-page scores (notably for MUSCIMA++) are considered as a single measure. In addition, the increased complexity of the scores makes the 13 nearest neighbors insufficient to obtain inclusive CGs. To account for the different types of scores, we set k to 20 for the datasets MUSCIMA++ and DoReMi cut by measures. Setting k to 20 does not guarantee the CG to be inclusive either. In fact for *MUSCIMA_measure_cut*, 80% of the ground true edges are included in the CG, and for *DoReMi_measure_cut* the share of edges included in the CG is 91%. An improved algorithm for dividing scores by measure should be used to select a more meaningful value for k. One improvement that we didn't implement, would be to integrate the grammar of music notation directly into the construction of the graph and only connect a node with the k nearest neighbors that it theoretically could connect to instead of connecting each node to its k nearest neighbors and then pruning the graph. We hypothesize that with this improvement, k could be even smaller, leading to smaller CGs.

Not all models perform equally across different link types, a link type is defined by the class of its 2 endpoints. Based on this observation, we can imagine an ensemble approach, where multiple models are employed together. During the prediction phase, various models generate predictions. For each link, depending on its specific type, we select the prediction from the model that has demonstrated the best performance for that particular link type. Table II shows the performance of 4 models across the link types. Leveraging model ensemble strategy and combining these 4 models, we obtained the results presented in Table III. One of these models is based on a Geo-GCN layer [10] instead of a graphSAGE layer. This pipeline

TABLE I: Best models obtained for the different datasets with the 10-labels class set

Models	Dataset	k	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	MER (%)	GED
model 1	MusiGraph	13	94.45	96.97	95.15	97.00	13.48	2.41
model 2	DoReMi_measure_cut	20	89.71	71.20	84.75	91.00	37.02	8.39
model 3	MUSCIMA_measure_cut	20	84.37	64.71	72.00	88.00	38.08	13.87

TABLE II: Accuracy obtained for each link type by a selection of models trained and evaluated on MusiGraph for the 6-labels class set

Models	Layer	noteheadBlack - stem	noteheadBlack - accidental	noteheadBlack - Flag	noteheadBlack - Beam	Notehead WholeOrHalf - stem	noteheadWholeOrHalf - accidental
model 4	graphSAGE	98.84	**99.00**	99.38	70.41	**98.80**	**90.87**
model 5	graphSAGE	**98.89**	98.96	99.37	69.02	**98.80**	90.83
model 6	geoGCN	91.52	90.53	90.34	**89.43**	96.63	89.99
model 7	graphSAGE	98.85	98.97	**99.41**	68.45	98.61	90.83

TABLE III: Metrics obtained with the model ensemble and the 6-labels class set

Dataset	k	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	MER (%)	GED
MusiGraph	13	97.09	97.76	92.70	99.06	6.09	0.70

also uses a different set of classes, *6-labels*, more suited to MusiGraph. It encodes less primitives but has a greater granularity (see Appendix for details).

VII. Discussion and Conclusion

Our developed pipeline incorporates certain decisions that may be subject to discussion. The first criticism we can address is the limiting aspect of the solution regarding the set of classes. Our model relies on specific class sets, and the primitives' labels must be one-hot encoded to form the feature vectors. Such a framework makes it impossible to adapt a pre-trained model for accepting new classes. Another critical area to address is the normalization step; While intended to standardize musical scores for versatility, the approach was excessive. A more moderate strategy, using staff size as a reference for normalization, would have aligned better with the inherent properties of the musical data and potentially improved the model's performance. An important majority of scores leverage typeset staff and considering them different from one to another is probably excessive.

Despite these criticisms, the testing framework itself is robust and provides a solid foundation for evaluating model performance. It allows for an objective comprehensible measurement of how good the models are.

We have demonstrated that GNNs can be applied to the semantic reconstruction stage of the OMR pipeline with acceptable performances. The performances could probably be improved with a better combination of parameters. However, the sensitivity and instability of these models may limit their suitability as the optimal solution.

References

[1] A. Baró, P. Riba, and A. Fornés, "Musigraph: Optical music recognition through object detection and graph neural network," in *Frontiers in Handwriting Recognition - 18th International Conference, ICFHR 2022, Hyderabad, India, December 4-7, 2022, Proceedings*, ser. Lecture Notes in Computer Science, U. Porwal, A. Fornés, and F. Shafait, Eds., vol. 13639. Springer, 2022, pp. 171–184. [Online]. Available: https://doi.org/10.1007/978-3-031-21648-0_12

[2] J. Hajič jr., M. Dorfer, G. Widmer, and P. Pecina, "Towards full-pipeline handwritten omr with musical symbol detection by u-nets," in *International Society for Music Information Retrieval Conference*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53048053

[3] A. Pacha, J. Calvo-Zaragoza, and j. Jan Hajič, "Learning notation graph construction for full-pipeline optical music recognition," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR 2019)*, 2019, pp. 75–82. [Online]. Available: https://doi.org/10.5281/zenodo.3527744

[4] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, 2005, pp. 729–734 vol. 2.

[5] J. Hajič and P. Pecina, "The MUSCIMA++ dataset for handwritten optical music recognition," in *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017*. IEEE, 2017, pp. 39–46.

[6] E. Shatri and G. Fazekas, "Doremi: First glance at a universal OMR dataset," *CoRR*, vol. abs/2107.07786, 2021. [Online]. Available: https://arxiv.org/abs/2107.07786

[7] G. de Lambertye, "Music semantic reconstruction with deep learning," Master's thesis, Technical University of Vienna, Wien, Austria, Oct. 2024.

[8] PyTorch Contributors, *Reproducibility — PyTorch 2.0 documentation*, pytorch.org. [Online]. Available: https://pytorch.org/docs/stable/notes/randomness.html

[9] P. Veličković, "Everything is connected: Graph neural networks," *CoRR*, vol. abs/2301.08210, 2023.

[10] P. Spurek, T. Danel, J. Tabor, M. Smieja, L. Struski, A. Slowik, and L. Maziarka, "Geometric graph convolutional neural networks," *CoRR*, vol. abs/1909.05310, 2019. [Online]. Available: http://arxiv.org/abs/1909.05310
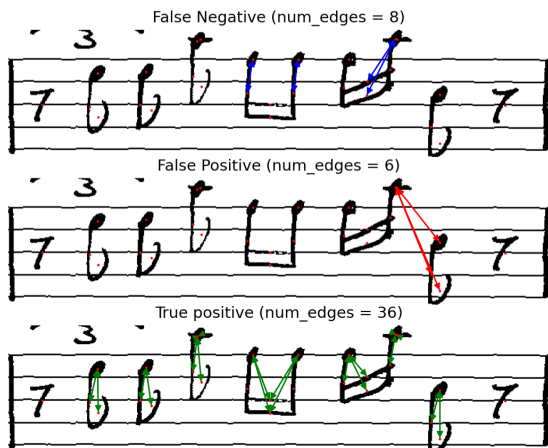
APPENDIX

Fig. 4: Example of model 2's prediction on the MUSCIMA++ dataset (cut by measure).
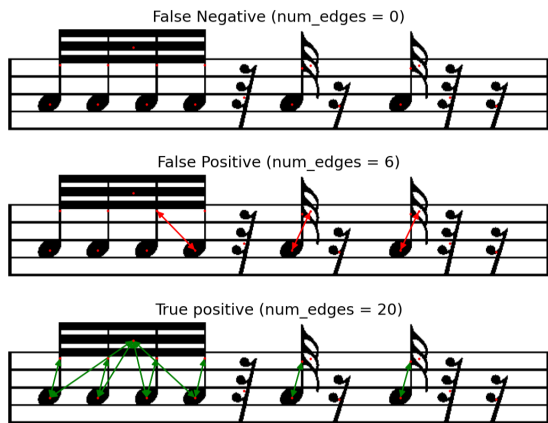

Fig. 5: Example of model 3's prediction on the DoReMi dataset (cut by measure). In this example, we see an error in the dataset: the links between the triple flags and their noteheads have been correctly predicted but are classified as false positives.
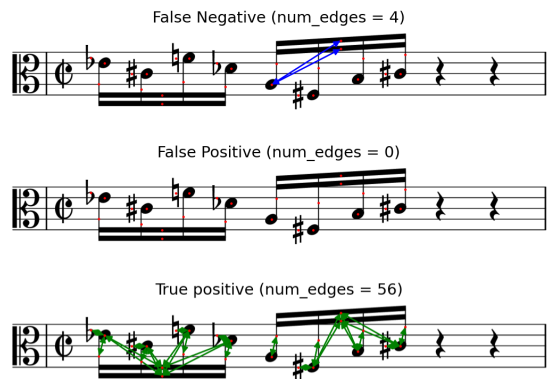

Fig. 6: Example of the model ensemble prediction on the MusiGraph dataset.

TABLE IV: 6-labels class set

MusiGraph	MUSCIMA++	DoReMi	6-labels
stem	stem	stem	stem
notehead-full	noteheadFull	noteheadBlack	noteheadBlack
notehead-empty	noteheadHalf, noteheadFullSmall, noteheadWhole	noteheadHalf, noteheadWhole	noteheadWholeOrHalf
beam	beam	beam	beam
sharp, flat, natural	augmentationDot, accidentalSharp, accidentalFlat, accidentalNatural, accidentalDouble Sharp	augmentationDot, accidentalSharp accidentalFlat, accidentalNatural accidentalDoubleFlat accidentalQuarterToneSharpStein accidentalQuarterToneFlatStein accidentalDoubleSharp accidentalThreeQuarterTonesSharpStein	accidental
16th_flag, 8th_flag	flag16thUp, flag16thDown,flag8thDown, flag8thUp, flag16thUp, flag16thDown, flag8thDown, flag8thUp	flag16thUp, flag16thDown, flag8thDown, flag8thUp, flag32ndUp, flag32ndDown	flag

TABLE V: 10-labels class set

MusiGraph	MUSCIMA++	DoReMi	10-labels
stem	stem	stem	stem
notehead-full, notehead-empty	noteheadFull, noteheadHalf, noteheadFullSmall, noteheadWhole	noteheadBlack, noteheadHalf, noteheadWhole	notehead
beam	beam	beam	beam
(missing)	augmentationDot	augmentationDot	augmentationDot
sharp, flat, natural	accidentalSharp, accidentalFlat, accidentalNatural, accidentalDoubleSharp	accidentalSharp accidentalFlat, accidentalNatural accidentalDoubleFlat accidentalQuarterToneSharpStein accidentalQuarterToneFlatStein accidentalDoubleSharp accidentalThreeQuarterTonesSharpStein	accidental
16th_flag, 8th_flag	flag16thUp, flag16thDown, flag8thDown, flag8thUp	flag16thUp, flag16thDown, flag8thDown, flag8thUp, flag32ndUp, flag32ndDown	Flag
(missing)	tie	tie	tie
(missing)	legerLine	(missing)	legerLine
(missing)	dynamicCrescendoHairpin, dynamicDiminuendoHairpin slur	dynamicForte, dynamicPiano, dynamicFFF, dynamicPPP, dynamicFF, dynamicText, dynamicMP, dynamicFortePiano, dynamicPP, dynamicSforzato, dynamicMF, dynamicForzando, gradualDynamic, slur	others_slur_dynamics_etc
8th_rest, 16th_rest, quarter_rest, half_rest	rest8th, rest16th, restQuarter, restHalf, restHBar, restWhole	rest8th, rest16th, rest32nd, restQuarter, restHalf, restWhole	rest

Staff Layout Analysis Using the YOLO Platform

Vojtěch Dvořák, Jan Hajič jr., Jiří Mayer (✉)

Institute of Formal and Applied Linquistics
Charles University, Prague, Czech Republic
Email: v.dvorak@matfyz.cz, hajicj@ufal.mff.cuni.cz, mayer@ufal.mff.cuni.cz
ORCID: 0009-0007-8423-5139, 0000-0002-9207-567X, 0000-0001-6503-3442

Abstract—**Detecting staffs, systems, and measures, collectively known as layout analysis, matters for Optical Music Recognition (OMR), both because most systems today expect staff-level inputs, and because even if these are replaced by systems that can process the whole page, the staffs and systems are useful elements of OMR user interfaces and applications. It receives comparatively little attention, which is justified, as it avoids many class imbalance, small object, and object assembly phenomena, which is what makes OMR difficult and interesting. However, the main publicly available tool for layout analysis, the MeasureDetector, has not been updated for several years, and off-the-shelf object detection has progressed: not just in accuracy, but also in speed. Therefore, in this paper, we bring an update on the performance of OMR layout analysis with the state-of-the-art YOLO platform. Compared to the MeausreDetector, it achieves a similar or better accuracy across both in-domain and out-of-domain tests over three different datasets that we harmonized, it is more than 20x faster, and requires more than 4 times less memory.**

Index Terms—**Optical Music Recognition, Layout Analysis, Deep Learning**

I. INTRODUCTION

One of the first steps in many Optical Music Recognition systems is detecting which regions of the music score image correspond to high-level elements of music notation: system, staff, and measure, usually as a staff detection step of the traditional OMR pipeline [4], [5], [17]. Assigning written music to these objects, especially staffs and systems, determines the reading order of the written page. This step can be done before or after individual image pixels are assigned to layers such as background, staff, and foreground [3], [6], [7].

Is staff layout analysis still a relevant task for OMR in the presence of end-to-end methods? Most end-to-end recognition methods to date have also operated on single staffs or systems [1], [15], [18]–[20]. Even though there are attempts to perform full-page recognition that learns to read the whole page without splitting it into these basic elements of music score layout [19], these are still initial experiments (though promising). Therefore, while system, staff and measure detection may not represent a key element of every OMR system, it is currently still a broadly applicable initial step that, while perhaps not as exciting as end-to-end recognition itself, has its place in the ecosystem.

Furthermore, even in the presence of well-performing end-to-end methods for processing the entire page, we believe having an explicit staff layout detected before processing entire pages may be highly useful in practice. Computing resources are not unlimited and transformer-based models and other recurrent models that represent the current state of the art [15], [19] are computationally expensive. Errors in staff layout (such as not assigning staffs to systems correctly) are extremely expensive to fix manually and lead to many compounding errors downstream. So, in user-facing applications, it may be highly desirable to get staff layout information verified interactively, before running the relatively expensive recognition model itself.

Finally, also in the spirit of lowering computational (and therefore energy) costs, when one is trying to detect music notation in large collections of documents (in the millions of pages or more), the staff is the visually most distinct and clear sign of music notation's presence – Common Western Music Notation (CWMN) as well as mensural or medieval.[1] Sending an image into full OMR processing only when a staff is detected with a very high probability is thus a reasonable component in a practical library-scale system.

At the same time, detecting systems, staffs and measures is a sub-task of OMR that is not particularly affected by the music notation phenomena that make OMR as a whole so difficult [2], [4]. These notation objects occupy large convex regions of the image, and there aren't as many on any single page. Hence, existing object detection methods are expected to be entirely applicable. This justifies why the task of staff layout detection has not received much scholarly attention in the past few years. But, significant progress in object detection has been made since [8], [12], [14], [24], most importantly on the YOLO platform [22]. And the most popular publicly available measure detector that the field has produced, the veritable MeasureDetector[2], has last been updated in 2020, it is based on TensorFlow version 1.13.1, which is outdated and requires Python 3.7, a version that reached end-of-life in 2023. Therefore, we believe it is time to update the OMR field's collective intuition on how well (and how fast) this auxilliary task of sheet music layout detection can in fact be performed today.

II. CONTRIBUTIONS

The central contribution of this paper is not surprising: we find that the current state-of-the-art YOLOv8m model [13] reaches similarly good or slightly better performance as the older R-CNN [16], but is significantly faster and smaller,

---

[1]Aside from early adiastematic chant manuscripts, of which there aren't millions of pages extant.

[2]https://github.com/OMR-Research/MeasureDetector

and therefore it is preferable. Pre-trained models are made available both for the previous Faster R-CNN architecture and YOLOv8m.[3]

Furthermore, this work:

- harmonizes, combines and extends already existing datasets (extended with systems and grand staffs), and provides scripts to convert and merge them into COCO and YOLO format;
- adds MZKBlank[4], a new dataset that contains background images representative of archival collections;
- trained object detection models available both as Faster R-CNN and YOLOv8m;
- provides an estimate of out-of-domain generalization for several basic classes of scores;

All of the scripts and models are available on GitHub[5]. Secondary outputs include hotfixes to A. Pacha's 2019 MeasureDetector[6], done to prepare datasets and train R-CNN models.

Taken together, we believe these contributions are a substantial update – especially in terms of quality-of-life – for putting into practice those OMR systems that rely on layout detection as a preprocessing step.

III. LAYOUT OBJECTS

We use five classes of layout objects: staff, grand staff, system, staff measure, and system measure.

Staff. Contains (typically) five parallel lines, all of the same length. One staff is one "line of sheet music" for an instrument. Many end-to-end methods assume as their input the image of a single staff and its associated symbols. Staffline spacing is also the basic element of music notation scaling.

Grand staff. A pair of staffs meant for a single instrument with a large range (typically keyboard instruments, or the harp). It is practical to treat the grand staff as a separate class because it implies the presence of more complex classes of notation that might be better handled by a more complex but more demanding model (polyphonic and pianoform [4]).

System. A set of staffs (some of which may be grand staffs) that are to be read in parallel. Barlines may be drawn across the whole system, to provide the readers (usually the conductor, or singers) clear information on synchronization. Technically, e.g. in a violin part, each staff is also a system, but systems are most useful to detect when one needs to decide which staffs should be concatenated and which should not (for instance, to correctly assemble the staffs for individual instruments in a string quartet score).

Staff measure. One measure (a region of notation that corresponds, typically, to one metrical cycle of a downbeat and other beats, as denoted by the time signature) on a staff. Measures are useful for instance as units of indexing for

fast sheet music retrieval, and they can also be used for "sanity checks" when assembling scores from individual staff components to catch de-synchronization between parts early (and correct for it).

System measure. All the measures belonging to the same system that should be played in parallel.

These classes are sufficiently generic that they apply across many different CWMN datasets,[7] as evidenced by the unproblematic harmonization of multiple datasets for this work, and at the same time are useful objects that someone might want to extract from a score, for instance to establish an unambiguous reading order.

IV. DATASETS

The resulting dataset is a combination of three already existing datasets and a new one, for numbers of images and annotations (see Tab. I). All datasets mentioned have annotations available in COCO format, concrete implementations differ.

A. AudioLabs v2

AudioLabs v2 is an extension of the AudioLabs v1 dataset. Its annotations were generated with the help of a neural network and the original dataset [23], the images are generated from CSV files. Grand staffs and system bounding boxes were added manually to the dataset.

B. MUSCIMA++

MUSCIMA++ [11] is a dataset of handwritten music notation for musical symbol detection that is based on the CVC-MUSCIMA dataset [9]. Grand staffs and system bounding boxes were added manually to the dataset.

C. Open Score Lieder – OSLiC

OpenScore Lieder is a collection of digital editions of accompanied songs by 19th century composers transcribed using the MuseScore editor [10]. The annotations were parsed from SVGs that were generated along with PNGs using MuseScore from the dataset's MSCX scores. Because of many inconsistencies, some scores were ruled out of the final dataset; however, these are all still pixel-accurate annotations.

D. MZKBlank

For the best training results, 1–10 % of the images in the dataset should be background images (negative samples) [13], but the datasets mentioned above do not contain enough of these examples, only 56 images of 6 007 do not contain any annotations. The Moravská Zemská Knihovna (MZK) offers access to more than two thousand public domain sheet music documents with more than nine thousand labeled pages that do not contain any music[8] – our negative samples. Reducing

---

[3]https://github.com/v-dvorak/omr-layout-analysis/releases/tag/evaluation-release

[4]https://github.com/v-dvorak/omr-layout-analysis/blob/main/app/MZKBlank

[5]https://github.com/v-dvorak/omr-layout-analysis

[6]https://github.com/v-dvorak/MeasureDetector

[7]For menusral music, measures are (in the vast majority of cases) not applicable, and different configurations than a *partitura* such as choirbooks or partbooks where the concept of systems is much less trivial (even though at least choirbooks take care so that all parts need turning the page at the same time, in case of longer compositions).

[8]Blank, front cover, front end sheet, title page, table of contents and more.

TABLE I
NUMBERS OF ANNOTATIONS AND IMAGES IN EACH DATASET

	images	system measures	staff measures	staffs	systems	grand staffs
AudioLabs v2	940	24 186	50 064	11 143	5 376	5 375
MUSCIMA++	140	2 888	4 616	883	484	94
OSLiC	4 927	72 028	220 868	55 038	17 991	17 959
MZKBlank	1 006	0	0	0	0	0
total	7 013	99 102	275 548	67 064	23 851	23 428

the number of samples while maintaining the relative ratios, we get the new MZKBlank dataset that contains 1 006 semi-randomly chosen images that are related to sheet music but do not contain any sheet music.

Fig. 1. Overview of front covers from MZKBlank resized to squares.

V. EXPERIMENTS

In our evaluation, we compare the YOLOv8m model with the Faster R-CNN model implemented using TensorFlow, previously utilized for a measure detector by A. Pacha. (We train both architectures on the same datasets, we do not re-use the trained MeasureDetector.) All trained models and results are available online.[9]

One in-domain test was performed using a 90/10 train/test split across the combination of all datasets. Then, three out-of-domain tests (with one non-blank dataset left out as the test set, see Tab. II) were performed. The results were measured with mAP50 and mAP50-95[10]. We think that specifically for layout analysis, the higher IoU thresholds are more relevant, because the accuracy of the bounding box matters, especially when layout analysis is used as a preprocessing step (as opposed to localization of e.g. clefs or stems in a hypothetical downstream object detection step). We used the `pycocotools` Python library to calculate these metrics.[11]

[9]https://github.com/v-dvorak/omr-layout-analysis/releases/tag/evaluation-release

[10]Mean average precision at an IoU at threshold of 0.50, and the average of the mAP calculated at varying IoU thresholds ranging from 0.50 to 0.95.

[11]The YOLO platform provides its own evaluation script that is not suitable to evaluate the R-CNN models. In fact, they use less strict parameters, so a custom script is used to evaluate both types of models.

TABLE II
OUT-OF-DOMAIN TEST DATASETS CONTENTS

id	training datasets	validation dataset
IV	AudioLabs v2, MUSCIMA++, MZKBlank	OSLiC
V	MUSCIMA++, OSLiC, MZKBlank	AudioLabs v2
VI	AudioLabs v2, OSLiC, MZKBlank	MUSCIMA++

A. Test results

In case of the in-domain test (see Tab. III), the YOLO model outperforms R-CNN: slightly, in the mAP50 setting, and significantly in mAP50-95, where it on average gets halfway closer to a perfect score.

In the case of out-of-domain tests on printed music (see Tab. IV and V, both models perform comparably, with the R-CNN slightly beating YOLOv8m in mAP50 scores, but YOLO performing better when better localization is required. When MUSCIMA++ is the out-of-domain dataset, however, YOLO is better in both metrics, and while nowhere near usable overall, it reaches 0.75 mAP50-95 for grand staffs and 0.72 mAP50 for staffs, which compared to YOLO's 0.164 and 0.061. YOLO can apparently to some extent abstract away the unexpectedly handwritten context in which the staffs exist, while R-CNN has practically no chance.

B. Speed comparison

Using the same hardware and running on a CPU[12], the inference times for both models were measured. Pacha's R-CNN averaged an inference time of 21.33 seconds per image, whereas YOLOv8m averages at just 0.83 seconds, nearly 26 times faster. YOLOv8m's speed can be further improved when ran on GPU[13], with an average of 0.42 seconds per image, where the inference itself (with pre- and post-processing) takes an average of 0.16 seconds.

C. R-CNN's overlap problem

One of the specifics of staff layout analysis are overlapping bounding boxes – for every grand staff there has to be a system (that may or may not contain other staff). In our dataset 98%[14] instances of grand staffs are a system. This overlap has an unwanted effect on the R-CNN model. It has no problem identifying systems and grand staff with confidence > 0.9 when grand staff $\subsetneq$ system. But when grand staffs are also entire systems, the confidence drops for both predictions

[12]Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz, 32GB RAM

[13]NVIDIA GeForce GTX 1080 Ti, 32GB RAM

[14]23 428 : 23 851 ≈ 98%, see Tab. I.

(sometimes even bellow 0.5) and the bounding boxes are less accurate. We conjecture that this may be caused by the structure of Faster R-CNNs, specifically the region proposals. YOLOv8 is a one-stage model [21] in contrast to the two-stage R-CNN, and we did not notice any significant performance drops in the same situation (see Fig. 2).

One can rightly argue that this is an artifact of ground truth design. However, we contend that at least for library-scale OMR systems, which are one of the major use-cases, the ability to determine that a page is e.g. purely for the piano (or one instrument), which can be inferred precisely from (grand) staffs and systems occupying the same area, should still be automated, and hence this property of R-CNN is not an *irrelevant* artifact of our ground truth design.
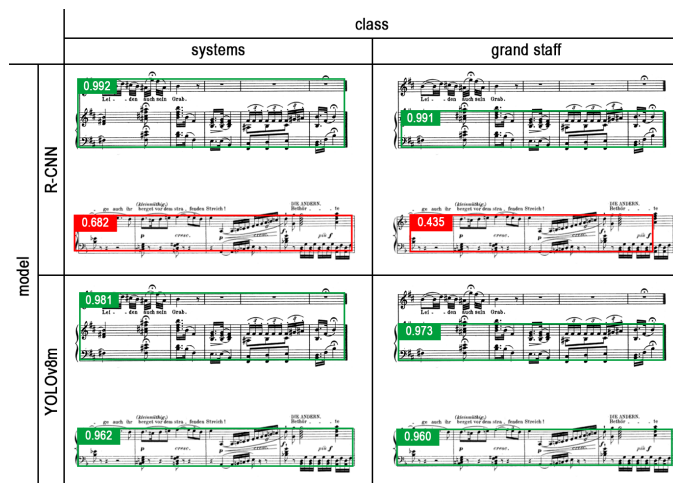

Fig. 2. Visualization of R-CNN's overlap problem. Each cell contains predictions for a 2- and 3-staff system. R-CNN 's confidence and precision significantly drops (red rectangles) when system and grand staff overlap (2-staff system), YOLOv8 does not exhibit this behavior.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

Given that the MeasureDetector tool[15] has not been updated for several years, we supply a new, up-to-date measure detector for anyone to use in this still-important OMR preprocessing step. This is no revolution of OMR, of course, but we observed clear performance benefits, both in correctness (esp. on the in-domain task, where the switch to YOLO and more extensive training eliminated between a third and a half of the remaining errors across diverse datasets), in speed (by an order of magnitude, a 26x increase, for an average of 0.83 s per page), and in memory requirements (50 vs 220 MB, which matters for instance for running the model in a browser, if one feels so inclined). However, for some out-of-domain settings, YOLO still does significantly worse when one does not need to use higher IoU thresholds, so we do not advocate phasing out the MeasureDetector completely.

We plan to extend the dataset further by adding more handwritten music, possibly synthetic, and to further explore

[15] https://github.com/OMR-Research/MeasureDetector

possibilities of the YOLO platform by experimenting with both smaller and larger models available (N, S, L, X), and to provide more pre-trained models that can be easily embedded into complete OMR workflows.

Looking at the progress of object detection [22], [24], layout analysis for OMR should be on its way to become a solved problem and a step that practitioners can easily plug into their systems. While in-domain detection results are coming close to this goal, out-of-domain layout analysis still has a long way to go. Overall, we believe that these models are a useful step in that direction.

TABLE III
IN DOMAIN EVALUATION, 90/10 TRAIN/TEST SPLIT.

class	instances	Pacha's R-CNN		YOLOv8m	
		mAP50	mAP50-95	mAP50	mAP50-95
system measures	9 151	0.989	0.943	0.987	0.975
staff measures	27 294	0.979	0.831	0.989	0.930
staffs	6 816	0.980	0.854	0.989	0.888
systems	2 326	0.990	0.947	0.990	0.986
grand staff	2 285	0.996	0.931	1.000	0.993
all	47 872	0.987	0.901	0.991	0.954

TABLE IV
OUT OF DOMAIN: EVALUATED ON OSLiC.

class	instances	Pacha's R-CNN		YOLOv8m	
		mAP50	mAP50-95	mAP50	mAP50-95
system measures	72 028	0.727	0.507	0.554	0.571
staff measures	220 868	0.678	0.204	0.580	0.249
staffs	55 038	0.921	0.295	0.829	0.334
systems	17 991	0.945	0.697	0.978	0.949
grand staff	17 959	0.982	0.701	0.901	0.792
all	383 884	0.851	0.481	0.790	0.579

TABLE V
OUT OF DOMAIN: EVALUATED ON ALv2.

class	instances	Pacha's R-CNN		YOLOv8m	
		mAP50	mAP50-95	mAP50	mAP50-95
system measures	24 186	0.989	0.827	0.934	0.770
staff measures	50 064	0.976	0.494	0.921	0.535
staffs	11 143	0.939	0.511	0.939	0.584
systems	5 376	0.989	0.832	0.960	0.860
grand staff	5 375	0.973	0.699	0.960	0.859
all	96 144	0.973	0.673	0.943	0.722

TABLE VI
OUT OF DOMAIN: EVALUATED ON MUSCIMA++.

class	instances	Pacha's R-CNN		YOLOv8m	
		mAP50	mAP50-95	mAP50	mAP50-95
system measures	2 888	0.256	0.140	0.153	0.123
staff measures	4 616	0.196	0.026	0.420	0.174
staffs	883	0.061	0.008	0.723	0.329
systems	484	0.237	0.111	0.192	0.140
grand staff	94	0.393	0.164	0.758	0.747
all	8 965	0.229	0.090	0.449	0.303

[16] https://github.com/Kristyna-Harvanova/Bachelor-Thesis

REFERENCES

[1] María Alfaro-Contreras, José M. Iñesta, and Jorge Calvo-Zaragoza. Optical music recognition for homophonic scores with neural networks and synthetic music generation. *12th International Journal of Multimedia Information Retrieval*, 12(1), May 2023. `doi:10.1007/s13735-023-00278-5`.

[2] Donald Byrd and Jakob Grue Simonsen. Towards a standard testbed for optical music recognition: Definitions, metrics, and page images. *Journal of New Music Research*, 44(3):169–195, 2015. `doi:10.1080/09298215.2015.1045424`.

[3] Jorge Calvo-Zaragoza and Antonio-Javier Gallego. A selectional auto-encoder approach for document image binarization. *Pattern Recognition*, 86:37–47, 2019. `doi:10.1016/j.patcog.2018.08.011`.

[4] Jorge Calvo-Zaragoza, Jan Hajič Jr., and Alexander Pacha. Understanding optical music recognition. *ACM Comput. Surv.*, 53(4), 2020. `doi:10.1145/3397499`.

[5] Jorge Calvo-Zaragoza, Juan C. Martinez-Sevilla, Carlos Penarrubia, and Antonio Rios-Vila. Optical music recognition: Recent advances, current challenges, and future directions. In Mickael Coustaty and Alicia Fornés, editors, *Document Analysis and Recognition Workshops*, pages 94–104, Cham, 2023. Springer Nature Switzerland. `doi:10.1007/978-3-031-41498-5_7`.

[6] Jorge Calvo-Zaragoza, Luisa Mico, and Jose Oncina. Music staff removal with supervised pixel classification. *International Journal on Document Analysis and Recognition*, 19:211–219, sept 2016. `doi:10.1007/s10032-016-0266-2`.

[7] Francisco J. Castellanos, Antonio Javier Gallego, and Ichiro Fujinaga. A few-shot neural approach for layout analysis of music score images. In *24th International Society for Music Information Retrieval Conference*, pages 106–113, Milan, Italy, 2023. URL: https://archives.ismir.net/ismir2023/paper/000011.pdf.

[8] Wei Chen, Jinjin Luo, Fan Zhang, and Zijian Tian. A review of object detection: Datasets, performance evaluation, architecture, applications and current trends. *Multimedia Tools and Applications*, 83:1–59, 01 2024. `doi:10.1007/s11042-023-17949-4`.

[9] Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. CVC-MUSCIMA: A ground-truth of handwritten music score images for writer identification and staff removal. *International Journal on Document Analysis and Recognition*, 15(3):243–251, 2012. `doi:10.1007/s10032-011-0168-2`.

[10] Mark Robert Haigh Gotham and Peter Jonas. The OpenScore Lieder Corpus. In Stefan Münnich and David Rizo, editors, *Music Encoding Conference Proceedings 2021*, pages 131–136. Humanities Commons, 2022. `doi:10.17613/1my2-dm23`.

[11] Jan Hajič jr. and Pavel Pecina. In search of a dataset for handwritten optical music recognition: Introducing MUSCIMA++. *Computing Research Repository*, abs/1703.04824:1–16, 2017. URL: http://arxiv.org/abs/1703.04824.

[12] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3296–3297, 2017. `doi:10.1109/CVPR.2017.351`.

[13] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, January 2023. URL: https://github.com/ultralytics/ultralytics.

[14] Jaskirat Kaur and Williamjeet Singh. A systematic review of object detection from images using deep learning. *Multimedia Tools and Applications*, 83:1–86, 06 2023. `doi:10.1007/s11042-023-15981-y`.

[15] Jiří Mayer, Milan Straka, Jan Hajič Jr., and Pavel Pecina. Practical end-to-end optical music recognition for pianoform music. In Elisa H. Barney Smith, Marcus Liwicki, and Liangrui Peng, editors, *Document Analysis and Recognition*, pages 55–73, Cham, 2024. Springer Nature Switzerland. `doi:10.1007/978-3-031-70536-6`.

[16] Alexandr Pacha. MeasureDetector, April 2019. URL: https://github.com/OMR-Research/MeasureDetector/releases/tag/v1.0.

[17] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R.S. Marcal, Carlos Guedes, and Jamie dos Santos Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, 2012. `doi:10.1007/s13735-012-0004-6`.

[18] Antonio Ríos-Vila. Rotations are all you need: A generic method for end-to-end optical music recognition. In Jorge Calvo-Zaragoza, Alexander Pacha, and Elona Shatri, editors, *Proceedings of the 5th International Workshop on Reading Music Systems*, pages 34–38, Milan, Italy, 2023. `doi:10.48550/arXiv.2311.04091`.

[19] Antonio Ríos-Vila, Jorge Calvo-Zaragoza, and Thierry Paquet. Sheet music transformer: End-to-end optical music recognition beyond monophonic transcription. In *International Conference on Document Analysis and Recognition*, pages 20–37, Athens, Greece, 2024. Springer. `doi:10.48550/arXiv.2402.07596`.

[20] Antonio Ríos-Vila, Jose M. Iñesta, and Jorge Calvo-Zaragoza. End-to-end full-page optical music recognition of monophonic documents via score unfolding. In Jorge Calvo-Zaragoza, Alexander Pacha, and Elona Shatri, editors, *Proceedings of the 4th International Workshop on Reading Music Systems*, pages 20–24, Online, 2022. `doi:10.48550/arXiv.2211.13285`.

[21] Jane Torres. Yolov8 architecture explained, March 2024. URL: https://yolov8.org/yolov8-architecture-explained/.

[22] Ajantha Vijayakumar and Subramaniyaswamy Vairavasundaram. Yolo-based object detection models: A review and its applications. *Multimedia Tools and Applications*, pages 1–40, 2024. `doi:10.1007/s11042-024-18872-y`.

[23] Frank Zalkow, Angel Villar Corrales, TJ Tsai, Vlora Arifi-Müller, and Meinard Müller. Tools for semi-automatic bounding box annotation of musical measures in sheet music. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019. URL: https://www.audiolabs-erlangen.de/resources/MIR/2019-ISMIR-LBD-Measures.

[24] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276, 2023. `doi:10.1109/JPROC.2023.3238524`.

On Designing a Representation for the Evaluation of Optical Music Recognition Systems

Pau Torras
Computer Vision Center
Computer Science Department
Universitat Autònoma de Barcelona
Bellaterra, Spain
ptorras@cvc.uab.cat

Sanket Biswas
Computer Vision Center
Computer Science Department
Universitat Autònoma de Barcelona
Bellaterra, Spain
sbiswas@cvc.uab.cat

Alicia Fornés
Computer Vision Center
Computer Science Department
Universitat Autònoma de Barcelona
Bellaterra, Spain
afornes@cvc.uab.es

Abstract—**Optical Music Recognition (OMR) is currently fragmented, with incompatible datasets and methodologies making it difficult to combine or compare systems. This paper proposes the Music Tree Notation (MTN) format as a unified framework to promote collaboration, technology reuse, and fair evaluation in OMR research. MTN represents music using an abstract tree built upon the concept of visual primitives, a trade-off between fully graph-based and sequential-based formats. The authors also introduce a set of metrics and a typeset score dataset.**

Index Terms—**Optical Music Recognition, Representation, Evaluation, Datasets, Computer Vision**

I. INTRODUCTION

Written music has been a key part of human cultural heritage for centuries, from ancient neumes to modern Western notation, with countless pages preserved over time. Given the vast number of notable, often forgotten works, scholars now turn to computers for help in preserving and analysing them. Optical Music Recognition (OMR) plays a crucial role in this process, converting scanned or imaged scores into a computer-readable format for further analysis [1].

However, the field of OMR is quite fragmented nowadays. Only a small community is fully devoted to it, and each of its members have developed their unique point of view and methodology [1], [2]. This is particularly evident when analysing the available datasets [3], as most of them are restricted to specific steps or approaches and almost none of them are compatible with each other (the most notable exception being the DoReMi dataset introduced in 2021 [4], which incorporates its ground truth in multiple formats).

Another point of disagreement among OMR researchers is the matter of the evaluation of models and systems [1], [5]–[7]. Evaluation of OMR models is currently performed on a per-methodology basis [1], [8], disregarding the full reconstruction of music scores. Most metrics revolve around measuring the fidelity of intermediate representations such as object bounding boxes or agnostic text representations without creating a final notation file. Moreover, the benchmarks these

evaluation metrics are computed upon are rarely widespread, with each methodology using their own.

To advance toward unifying the efforts of the OMR community, we propose establishing a shared framework. The first step is to define a target final representation that supports a wide range of use cases within Common Western Music Notation (CWMN), the system used in Europe from the early 18th century to the present. Our focus on CWMN, rather than the broader range of related western notations, is intentional. While CWMN evolved from Mensural notation and shares some graphical elements, its distinct semantic concepts set it apart. Given the unique nature of each notation system and the specific needs of OMR for CWMN, we believe it is best to focus on this system exclusively. Once a shared representation is chosen, a set of evaluation metrics can be fairly defined.

Thus, the contribution of this work can be summarised by the following claims:

- We try to bridge the gap between the different benchmark suites in OMR literature with a universal tree-based notation format designed to represent musical scores at the graphical level[1].
- We also present an evaluation toolkit which aims towards unify existing benchmark OMR tasks for fairer comparison.
- We have produced a typeset dataset using public domain works with permissive licenses [2].

II. THE MUSIC TREE NOTATION FORMAT

The cornerstone of the MTN format is understanding the task of structured OMR as the reconstruction of the score at the visual domain. The core idea of this format is therefore to build a notation that exclusively models relationships between graphical symbols and defers inference of music semantics until a later stage. Only those high-level music concepts that are strictly required to reconstruct the score unambiguously are kept iff there is a direct graphical cue that allows straight-forward inference. MTN is designed in order to

- normalise the set of music primitives to be recognised,

[1]Repository of the project: https://github.com/CVC-DAG/comref-converter
[2]Link to the dataset https://datasets.cvc.uab.cat/comref/comref.zip
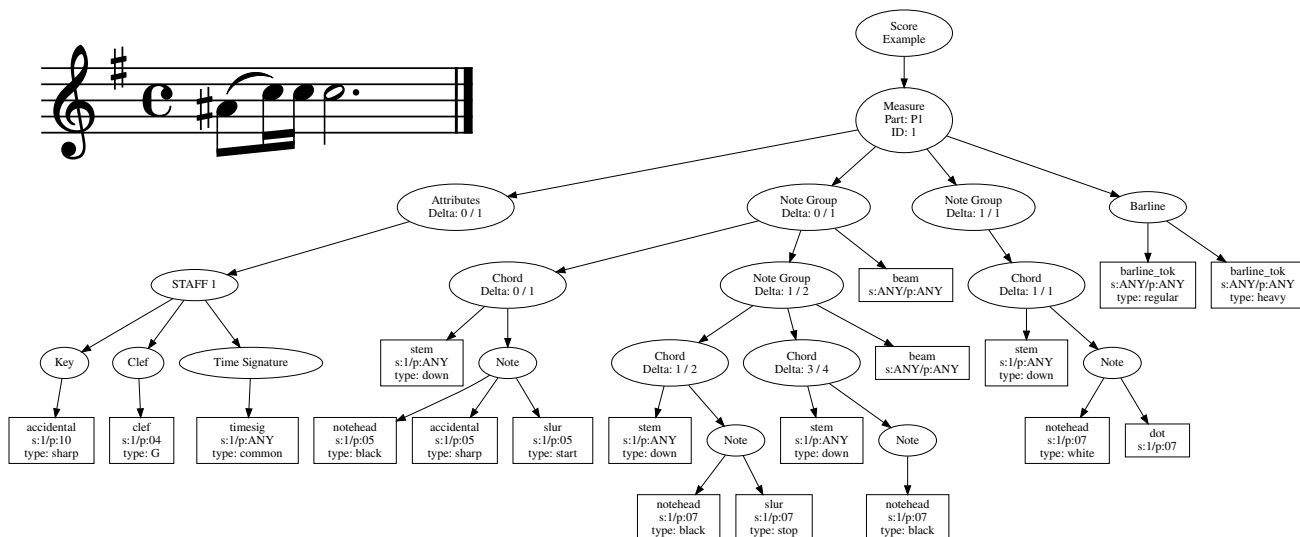
Fig. 1. Example showing a fragment of a measure in which the annotation format for attributes and staff-modifying elements is shown. Rectangular nodes represent primitives as tokens and rounded nodes are abstract elements.

- simplify conversion to a final structured format,
- enable comparison of diverse OMR methods on equal grounds and
- facilitate the usage of non-OMR-specific data.

A graphical representation of a simple measure engraved in MTN can be seen in Figure 1. The core element of this format is the Musical Primitive, a concept that is quite widespread in the OMR literature [9]–[12] and can be defined as any of the independent structural elements that may or may not be combined together to form a semantic unit in the music score. The set of musical primitives includes all graphical elements in a score that are self-contained and require no other symbols to convey meaning (rests, clefs or time signature symbols), the set of graphical elements that compose notes (noteheads, stems, flags, dots, accidentals, etc.) and other miscellaneous elements such as numbers for compound time signatures. Every primitive is given a unique work-level identifier.

These primitives associate together to form more abstract constructs. This is modelled in MTN using a tree-like structure of higher-order elements, which defines the set of dependencies among objects in the score. This idea, present in works such as [13], emulates parsing the contents of the score using a grammar, enabling the bulk of tools and research on parsers, parser generators and AST analysis and processing to be used in the context of music. Furthermore, it is a structure that can be modelled very easily using an exchange format such as XML.

There are some elements in music that break the tree-like structure assumption. These are elements that connect multiple notes together outside their local note group structure: slurs, ties, parentheses and tuplets, among others. Both MEI and MusicXML acknowledge this limitation and circumvent it through the use of identifiers. MTN is no different: it provides a unique starting and ending token for each side of the object

and gives both ends the same identifier.

In order to describe the position of MTN elements, two magnitudes are used. Firstly, for every token a tuple of two integers denotes the staff the element belongs to and its position within the staff. The position is denoted counting the number of steps from the first ledger line below a staff. For those elements without a specific position (such as rests or stems), a null value is used. Secondly, for any object immediately below the class measure, an exact timing value is provided. It is measured in fractions of a quarter note from the start of the measure itself. This information is also provided for every chord in a note group even if this information can be inferred for the sake of simplifying evaluation procedures.

Finally, to produce unambiguous scores, a reading order of sorts must be established. We propose the following ordering criterion:

- By starting time counting from the beginning of the measure.
- By top level class (in this order): Attributes, Directions, Rests, Note Groups and Barlines.
- By staff position: first objects on upper staves and lower positions within them.
- In case of Note Groups, by direction of the first stem: first stems looking upwards.
- For anything else, token alphabetical order. This also guarantees stability of the notation if new token types are added.

For other elements such as text or bounding boxes, we propose the use of extensions to complement the format.

III. EVALUATION METRICS

We propose a set of evaluation metrics that both acknowledge the existence of multiple paradigms for OMR while also setting ways to compare any structured output equally. These metrics draw inspiration from the currently used Symbol Error

Rate and some ideas from Hajič jr. [14]. We have divided our proposed metrics as tiers depending on the abstraction level they address and the problems they can help diagnose.

A. Tier 0: Methodology-Specific Metrics

In Tier 0 any methodology-specific metrics should be logged. This includes bounding box-level mean average precision, symbol error rate or other metrics that are standard for an OMR approach but are not regulated by MTN.

B. Tier 1: Primitive detection

The first set of metrics addresses the presence or absence of terminals within the MTN string. These metrics do not take into account structural matters, making them quick to compute. Given a set of Predicted Terminals P and a set of Ground truth Terminals G, we define

- **Primitive-level precision**

$$precision = \frac{\|P \cap G\|}{\|P\|} \qquad (1)$$

- **Primitive-level recall**

$$recall = \frac{\|P \cap G\|}{\|G\|} \qquad (2)$$

These metrics are computed per-class for the entire dataset. In order to produce a single precision and recall measure, results are aggregated per-class using a weighted average, where the weights are the relative frequency of each token in the ground truth.

C. Tier 2: Structure Reconstruction

This tier takes into account the structure of the produced MTN and compares it directly with that of the ground truth. A matching from ground truth elements to those present in the prediction is performed using a tree edit distance algorithm. In particular, since there is a restriction on the ordering of sibling labels, the $O\left(n^3\right)$ solution from Zhang and Sasha can be employed [15]. In practice, we use a Python implementation [16] of Pawlik *et al.*'s APTED algorithm [17].

Given the following operations:

- **Substitution:** Changing the label of a single node within the tree.
- **Deletion:** Removing a single node of the tree and setting its children as siblings.
- **Insertion:** Adding a new node under a parent one and setting a consecutive subsequence of its siblings as children.

Given a predicted tree and a ground truth tree whose set of vertices is G and assuming an equal edit cost of 1 for all operations, the Tree Error Rate (TER) is defined as

$$TER = \frac{S + D + I}{\|G\|} \qquad (3)$$

where S, D and I are the number of substitution, deletion and insertion operations required to produce the ground truth tree from the predicted tree. This metric is designed mostly for benchmarking and is defined by analogy to the ubiquitous Symbol Error Rate (SER).

D. Tier 3: Semantic Reconstruction

This tier considers whether the subset of music semantics required by MTN has been extracted correctly. It depends on the matching extracted from the structural level in order. Thus, the **False Positive Rate** and **Missing Note Rate**(MNR) metrics are defined as the ratio of ground truth notes that do or do not have a corresponding prediction.

In MTN, a note n is defined semantically from its graphical properties: position and time. From this idea and the matching extracted from the previous tier, we define a few metrics. **Pitch** and **Time Precision** are defined as the number of correctly predicted graphical pitches and times w.r.t. the ground truth. **Average Pitch Shift** (APS) and **Time Average Shift** (TAS) are defined as the average offset in pitch and time from the predicted note w.r.t. its corresponding ground truth note. Signedness is kept in order to identify the direction in which the underlying OMR system tends to move the notes.

In order for all of these metrics to be independent of the sequence length, they should be computed and accumulated for the entire dataset and not averaged on a by-prediction basis.

IV. A PROOF OF CONCEPT

We have developed a dataset built on transcriptions of public domain works as a proof of concept of the notation format. In particular, we have used the OpenScore project's transcriptions of widely known works such as The Art of the Fugue by J.S. Bach or the Planets by Gustav Holst, among others. We have also incorporated the Lieder Corpus [18] and the String Quartet corpus [19]. All these scores are engraved from MusicXML files.

In summary, the dataset is produced by processing of 894 individual works into images at the measure level (including all staves that belong to it), to produce a total of 435.162 images after cleanup. Page-level images are also provided.

The process through which the dataset was produced is summarised in Figure 2. Scores are engraved through Verovio [20] into page-level SVG files. Using the hierarchical structure within the SVG and exploiting the optional identifier information Verovio can be instructed to attach, measures are engraved individually. It also marks those measures at the beginning of a line to insert attribute elements.

Once the images are produced, the converter uses the MusicXML file and produces the MTN notation. In order to ensure all images have their corresponding ground truth, we use a cleaning script that finds matching identifiers for images in the MTN files. It also checks for outliers in case there are blatant mistakes in the notation. Although we have taken precautions to minimise the number of errors, there are a few images with objects far from the staff, either temporally or graphically. We remove these outliers heuristically to ensure the quality of the data.

We conducted a simple proof-of-concept experiment on this dataset to assess the feasibility of the methodology proposed in this paper. For this purpose, we used an off-the-shelf OMR system to produce a transcription of the test partition and we analysed its results.

TABLE I

EVALUATION RESULTS. FROM LEFT TO RIGHT, PRIMITIVE PRECISION AND RECALL, TREE ERROR RATE, AVERAGE TIME SHIFT, AVERAGE PITCH SHIFT, TIME PRECISION, PITCH PRECISION, STAFF PRECISION, FALSE POSITIVE RATE AND MISSING NOTE RATE

Prec.	Rec.	TER	Time Shift	Pitch Shift	Staff Shift	Time Prec.	Pitch Prec.	Staff Prec.	FPR	MNR
0.894	0.733	0.372	-0.096	-0.091	0.022	0.802	0.749	0.963	0.097	0.216
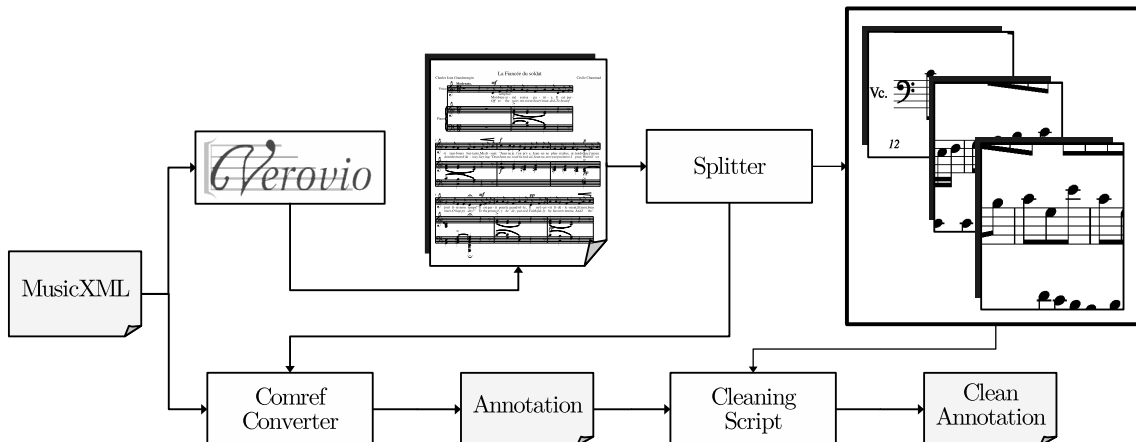
Fig. 2. The pipeline through which the COMREF dataset has been generated.

The OMR system used for this experiment is Audiveris [21], an Open Source page-level system capable of generating a MusicXML output from a single input image. The page-level images of the dataset are used as input, since Audiveris requires the information of the clef, key and beat. The output MusicXML is then converted to MTN and a simple matching between predicted and ground truth samples is generated by imposing a top-down reading order given the samples known to be present on each page. If a prediction has more measures per page than the ones in the ground truth, the extra ones are just discarded.

With the setup outlined above, Audiveris predicted 45822 measures from the 52884 present on the ground truth. Out of these, 40622 measures from both sets could be matched together, corresponding to a coverage of 76.9%. The missed predictions are as a result of the engine failing to give an output on certain pages. Results for all tiers are shown in I.

In general, the model tends to identify objects quite reliably but misses objects, as the precision is higher than the recall. Inspecting the per-class precision and recall values we see that attributes and the smaller objects of the score are the ones that tend to be recognised worse. The 80% of recall on black noteheads is alarming because this can cause a very significant drop of performance in note detection. Consequentially, note groups are missed and a temporal shift forward appears.

Overall, even if the results for this specific tool on the dataset still leave room for improvement, we consider that our proposed format and metric fulfil their design purposes: unique representation of scores and evaluation. Therefore, we consider this simple trial successful.

V. CONCLUSIONS

In this paper we have argued for the implementation of an Optical Music Recognition Framework through the development of a notation format in which score reconstruction is independent from the recognition methodology. Moreover, the resulting scores can be evaluated fairly an unambiguously. Our proposed reification of this idea is the MTN format. Since this method builds upon some of the most widely used abstractions of the community (e.g. symbols as combinations of primitives, time from ordering, etc) it stands as a good candidate for a common endpoint for OMR as a whole. Of course, CWMN is a tremendously complex notation system which has been optimised and streamlined for hundreds of years. Nevertheless, we believe the subset of music that can be expressed in this format is large enough to be useful for the community.

In this work, we have also presented a concrete implementation of a set of metrics for OMR practitioners with the hopes of bringing together the community to speak the same language; a *lingua franca* thanks to which research can be shared and compared fairly and easily. We provide a simple baseline from which to demonstrate how the evaluation framework works.

The work that lies ahead now is building a corpus of music that can be employed with this format into a benchmark for CWMN recognition, both in typeset and handwritten domains, which shall be the focus of our next efforts.

ACKNOWLEDGMENT

REFERENCES

[1] J. Calvo-Zaragoza, J. Hajič Jr., and A. Pacha, "Understanding Optical Music Recognition," *ACM Comput. Surv.*, vol. 53, pp. 1–35, July 2021.

[2] A. Pacha, "Advancing OMR as a Community: Best Practices for Reproducible Research," in *1st International Workshop on Reading Music Systems* (J. Calvo-Zaragoza, J. Hajič jr., and A. Pacha, eds.), (Paris, France), pp. 19–20, 2018.

[3] A. Pacha, "The OMR Datasets Project," 2017.

[4] E. Shatri and G. Fazekas, "DoReMi: First glance at a universal OMR dataset," in *Proceedings of the 3rd International Workshop on Reading Music Systems* (J. Calvo-Zaragoza and A. Pacha, eds.), (Alicante, Spain), pp. 43–49, 2021.

[5] J. Hajič and P. Pecina, "The MUSCIMA++ Dataset for Handwritten Optical Music Recognition," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, pp. 39–46, Nov. 2017. ISSN: 2379-2140.

[6] J. Hajič jr., "A Case for Intrinsic Evaluation of Optical Music Recognition," in *1st International Workshop on Reading Music Systems* (J. Calvo-Zaragoza, J. Hajič jr., and A. Pacha, eds.), (Paris, France), pp. 15–16, 2018.

[7] L. Mengarelli, B. Kostiuk, J. G. Vitório, M. A. Tibola, W. Wolff, and C. N. Silla, "OMR metrics and evaluation: a systematic review," *Multimedia Tools and Applications*, Dec. 2019.

[8] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes, and J. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *Int J Multimed Info Retr*, vol. 1, pp. 173–190, Oct. 2012.

[9] A. Baró, P. Riba, and A. Fornés, "Towards the Recognition of Compound Music Notes in Handwritten Music Scores," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 465–470, Oct. 2016. ISSN: 2167-6445.

[10] A. Baró, C. Badal, and A. Fornés, "Handwritten Historical Music Recognition by Sequence-to-Sequence with Attention Mechanism," in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 205–210, Sept. 2020.

[11] J. Calvo-Zaragoza and D. Rizo, "Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores," in *19th International Society for Music Information Retrieval Conference*, (Paris, France), pp. 248–255, 2018.

[12] L. Tuggener, I. Elezi, J. Schmidhuber, M. Pelillo, and T. Stadelmann, "DeepScores - A Dataset for Segmentation, Detection and Classification of Tiny Objects," in *24th International Conference on Pattern Recognition*, (Beijing, China), ZHAW, 2018.

[13] F. Foscarin, F. Jacquemard, and R. Fournier-S'niehotta, "A diff procedure for music score files," in *Proceedings of the 6th International Conference on Digital Libraries for Musicology*, DLfM '19, (New York, NY, USA), pp. 58–64, Association for Computing Machinery, Nov. 2019.

[14] J. Hajič jr., J. Novotný, P. Pecina, and J. Pokorný, "Further Steps towards a Standard Testbed for Optical Music Recognition," in *17th International Society for Music Information Retrieval Conference* (M. Mandel, J. Devaney, D. Turnbull, and G. Tzanetakis, eds.), (New York, USA), pp. 157–163, New York University, 2016. Backup Publisher: New York University.

[15] K. Zhang and D. Shasha, "Simple Fast Algorithms for the Editing Distance between Trees and Related Problems," *SIAM J. Comput.*, vol. 18, pp. 1245–1262, Dec. 1989. Publisher: Society for Industrial and Applied Mathematics.

[16] "JoaoFelipe/apted: Python APTED algorithm for the Tree Edit Distance." https://github.com/JoaoFelipe/apted/tree/master, 2017. Accessed: 2024-03-10.

[17] M. Pawlik and N. Augsten, "Tree edit distance: Robust and memory-efficient," *Information Systems*, vol. 56, pp. 157–173, Mar. 2016.

[18] M. R. H. Gotham and P. Jonas, "The OpenScore Lieder Corpus," in *Music Encoding Conference Proceedings 2021* (S. Münnich and D. Rizo, eds.), pp. 131–136, Humanities Commons, 2022.

[19] "String quartet corpus," 2023. Accessed: 2023-10-10.

[20] L. Pugin, "Verovio, a music notation engraving library." https://www.verovio.org/, 20?? Accessed: 2024-03-14.

[21] A. Project, "Audiveris - open-source optical music recognition." https://github.com/Audiveris/audiveris/, 20?? Accessed: 2024-03-14.

Enhanced User-Machine Interaction for Historical Sheet Music Retrieval: a Musical Notation Approach

Aitana Menárguez-Box
PRHLT Research Center
Universitat Politècnica de València
Valencia, Spain
amenbox@prhlt.upv.es

Alejandro H. Tosselli
PRHLT Research Center
Universitat Politècnica de València
Valencia, Spain
ahector@prhlt.upv.es

Enrique Vidal
PRHLT Research Center
Universitat Politècnica de València
Valencia, Spain
evidal@prhlt.upv.es

Abstract—Searching for musical information in historical handwritten scores poses a significant challenge, particularly for musicians and history researchers. Until now, an existing tool had enabled to query the Cod. 253 of the Vorau Abbey library through staff-relative-position symbols referred to as "geometrical notation".

We present advancements in user-machine interaction by enabling queries expressed with pitch-relative symbols (musical notation) in this system, offering more intuitive and precise means of interaction. Leveraging a web piano interface, users can now input queries using real musical notes, enhancing both usability and accuracy.

Even though previous works have already explored this kind of implementation, it has only been tested in already transcribed and digitalized sheet music. Our approach, based on fully automatic Probabilistic Indexing (PrIx) of a manuscript, addresses the intricacies inherent in historical scores, including variations in clef types and positions, to transform musical queries into complex Boolean geometric expressions. By integrating these enhancements into an existing search engine, we provide researchers with a more accessible and efficient means of exploring vast collections of historical sheet music.

This paper underscores the significance of user-machine interaction improvements in facilitating meaningful discoveries and insights in musicology and historical research.

Index Terms—Musical Probabilistic Indexing, Musical Information Retrieval, Historical Handwritten Music Recognition.

I. Introduction

Historical sheet music collections are invaluable resources for musicologists, historians, and musicians. These collections contain a wealth of information about the evolution of music notation, composition styles, and cultural practices. However, searching for specific musical information within these collections can be challenging due to the complexity of historical scores and the limitations of existing search tools.

In particular, searching for musical information in handwritten scores can be difficult because of variations in notation styles, clef types, and other notational conventions. To address these challenges, some work such as [1], [2] or [3] has been made to develop technologies based on Optical Music Recognition (OMR) to automatically transcribe and index historical scores. These enable the creation of tools that allow users to search for musical information in ancient sheet music.

In this paper, we took as a starting point the demonstrator[1] developed thanks to the work done in [4], which allowed to query the Cod. 253 of the Vorau Abbey library (Vorau-253). We have tried to overcome the limitations in terms of usability and accuracy this tool had. These were mainly due to the fact that they required users to input queries using abstract symbols (positions within the staves) which had little to do with musical notation. This made the search process less intuitive and precise, especially for users who were familiar with music theory.

Here we present an enhanced user-machine interaction approach for historical sheet music retrieval that enables users to input queries using pitch-relative symbols (musical notation) within the same web-based search engine. Our approach leverages a web piano interface that allows users to input queries using real notes within the staff, making the search process more intuitive and precise. We also address the complexities in historical scores, such as variations in clef types and positions, by transforming musical queries into complex Boolean geometric expressions that can be used to search for matches inside the mentioned manuscript collection.

II. Limitations of Current Music Search Systems

When comparing our approach to existing music search tools, several critical distinctions emerge. Many current systems do not operate on automatically recognized sheet music but instead rely on transcriptions prepared by musicologists. A prime example of this is the search tool available within the Cantus database [5]. This kind of tools, while robust for exploring already by-hand-transcribed chant manuscripts, do not address the intricacies involved in working with emphuntranscribed manuscripts, which constitute the vast majority of the millions of historical sheet music books in archives and libraries.

---

[1]Available at https://prhlt-carabela.prhlt.upv.es/musica.

A. Recognition Level Approaches

Methodologies employed for music recognition have evolved over time. Early systems like Aruspix or Gamera utilized traditional approaches, such as k-nearest neighbors (kNN) or hidden Markov models (HMMs) [6], to segment and recognize musical symbols. These and other similar systems [7] are built on workflows that treat musical elements in isolation as described in [2]. Although effective in specific cases, these segmentation-based approaches are inherently limited in accuracy and scalability.

More recent advancements, such as the OMMR4ALL[2] and the Cantus Ultimus[3] projects employ modern deep learning techniques to enhance Handwritten Music Recognition (HMR)[4] capabilities. While these neural network-based systems represent significant progress, they still share a common limitation with their predecessors: they focus on the most probable hypothesis during recognition for performing the information retrieval. In both old and new implementations, the recognition pipeline reduces the output to a single, definitive interpretation, which simplifies but also constrains the search process. In contrast, our approach introduces a novel dimension by utilizing Probabilistic Indexing [4]. Rather than selecting a single hypothesis for each recognized musical symbol, all recognition hypotheses are preserved alongside their probabilities. This comprehensive approach, as will bee seen further in this work, allows for more flexible and accurate search queries, as it accommodates uncertainty and variability of historical handwritten scores. Searching within a probabilistic framework enables users to retrieve not just the most likely match but also alternative possibilities, thereby improving the overall accuracy and reliability of the system.

B. Web Interface Design

Another significant difference can be noticed in the design of the user interface. Several existing systems offer web-based search functionalities, but these often suffer from poor usability. For example, the Liber Usualis Search[5], while pioneering in its intent, lacks the intuitive input mechanisms necessary for efficient query formulation. Similarly, the F-tempo[6] system, which employs the Aruspix engine, inherits many of the above mentioned limitations associated with segmentation-based recognition. There are also systems that offer interfaces which are closer to our proposal, such as Musiconn.scoresearch[7], that allows users to input queries using musical notation through a simulated piano keyboard. However, like many others, it still relies on a deterministic search model that limits results to the highest-probability recognition, potentially overlooking valuable alternative interpretations. On the other hand, our system addresses these

shortcomings by introducing a more intuitive, pitch-relative input method using a web piano interface. This allows users to query the system with real musical notes, making the search process far more accessible, particularly for those familiar with music theory. Coupled with the recognition approach used [8] and the Probabilistic Indexing [4], this interface enables a more nuanced and accurate search experience, offering multiple layers of recognition possibilities that are absent in other systems. By preserving the ambiguity and flexibility inherent in historical music notation, our approach enhances both the usability and the accuracy of music information retrieval.

III. QUERYING THE SYSTEM

Before the proposal in this paper, querying the Vorau-253 music collection was conducted using *geometrical notation* which relied on the positional information of musical elements within the staff. This notation system, as described in previous works [4], was convenient for basic training and testing experiments with handwritten music images.

A. Geometrical Notation Drawbacks

In the geometrical notation, basic lowercase symbols (l for notes on lines, s for notes in spaces) were utilized, with appended numbers indicating the vertical position in the staff. Additionally, other symbols were used to represent clefs (c or f followed by a number depending on the line they were located) and accidentals (i.e., the word flat). While this geometrical notation facilitated optical modeling and decoding of staff images, it fell short in representing melodic patterns adequately from a musical point of view. An example of this notation for an extract of the manuscript can be seen in Fig. 1.
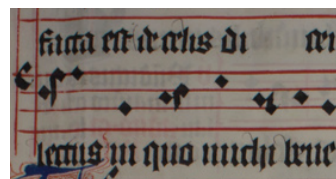

Fig. 1. A small staff fragment of a real sheet music image from the dataset Vorau-253 used in this work. The sequence of notes (and clef) on this image becomes represented as ⟨c4,l3,l4,l4,l2,s2,l3,s3,s2,l2,s2,l2⟩.

One significant limitation of geometrical notation is its inability to capture the contextual nuances in traditional musical notation systems. Unlike conventional musical notation, where a note's interpretation relies heavily on its relationship with other musical symbols (thanks to its conversion into a specific pitch), geometrical notation treats each note as an isolated entity, solely determined by its position on the staff. This lack of contextual information poses challenges in accurately representing and querying musical patterns, hindering the system's usability.

To overcome these limitations and enhance user-machine interaction, our approach introduces querying capabilities using

---

pitch-relative symbols (musical notation) within the existing web-based search engine. By leveraging a web piano interface, users can now input queries using real notes within the staff, offering a more intuitive and precise search experience. Furthermore, our system addresses the complexities of historical scores, including variations in clef types and positions, by transforming musical queries into complex Boolean geometrical expressions. This integration of musical notation querying enhances accessibility and efficiency, empowering researchers to explore vast collections of historical sheet music with greater ease and accuracy.

B. From Geometrical to Musical Notation

A single geometrical symbol, such as `l2`, does not inherently denote a specific pitch. Its interpretation depends on the preceding clef symbol. For instance, if a `c3` clef precedes it, the symbol represents an `A3`, whereas a `c4` clef would render it as a `F3`. Moreover, a single note may have multiple geometric equivalents based on different clef positions. For example, the geometrical representation of `D4` could be `s3` with a preceding `c3` clef or `s4` with a `c4` clef. These variations illustrate that each note can have a series of alternatives depending on the context provided by the clef.

In a search system based on geometrical notation, queries are typically constrained to a single clef at a time, leading to potential bias in the expressed information and the retrieved results. To address this limitation, a new querying approach is proposed, which involves converting queries into multiple translations corresponding to each possible clef position. This enables the representation of a note's context (i.e., its clef) within the query.

Consider the sequence of notes "C4 D4 F4" in musical notation to be converted into geometrical notation. Each note in the sequence is translated for every possible clef, and the resulting translations are combined into a single query. Additionally, certain constraints, such as excluding notes outside the potential pitch range (`E2` to `D5`), are applied to refine the query and improve its relevance.

The translation of the proposed note sequence into geometrical notation yields a complex query structure, as shown below:

```
(c1 &̄ [l1 s1 s2]) || (c2 &̄ [l2 s2 s3]) ||
(c3 &̄ [l3 s3 s4]) || (c4 &̄ [l4 s4 s5]) ||
(f1 &̄ [l3 s3 s4]) || (f2 &̄ [l4 s4 s5])
```

In this query, the `||` symbol denotes a boolean OR operation, the square brackets `[` and `]` indicate a sequence of notes and the `&̄` symbol represents an AND operator, ensuring that the clef must precede the sequence of notes on the same staff.

C. Expanding the Search: Ignoring the Key

After considering the use of this new querying approach, we must also take into account that sometimes, the user may want to look for melodies that are not exactly the same as the one they input. This could be due to a mistake in the transcription of the pitch or because they want to find all melodies that share the same intervallic relation between notes.

For example, if the user wants to search for an ascending melody of three notes separated by whole tones, a sequence query proposal would be "C4 D4 E4". Although the search is explicitly for these three notes, the option to search for all melodies that maintain the intervallic relationship should be given to satisfy the initial query. Thus, the notes may not be looked for as they are and the sequences "G4 A4 B4" or "C3 D3 E3" could also be found, both corresponding to the initial query of the three ascending notes separated by whole tones. We have also implemented this type of queries inside the demonstrator. Now the queries will be referred to as queries *with key* (if they take into account the original pitch) or *without key* (if they do not).

Both search forms necessarily induce a significant increase in the complexity compared to pure geometric queries, leading to potential performance implications for the search engine. Further research is needed to evaluate their impact on the system's performance.

IV. THE WEB PIANO INTERFACE

The musical input to the web platform is facilitated through the piano tab, which has been implemented using an HTML dialog containing various elements for melody insertion and visualization. This dialog is accessible via a button on the right side of the interface. Below we describe the main components of the tab:

- **Piano Keyboard**: HTML buttons simulating the keys of a real piano, each button is linked to the corresponding note in musical notation. Pressing a button (by clicking it or through the computer keyboard) transmits the note's name directly. Additionally, each button is associated with a mp3 audio file, enabling users to hear the sound of the note. Furthermore, notes which will for sure not appear in the manuscript are lighter in color. If one is pressed, the corresponding sound is played, but the note is not transmitted. The appearance of the piano keyboard is shown in Fig. 2.
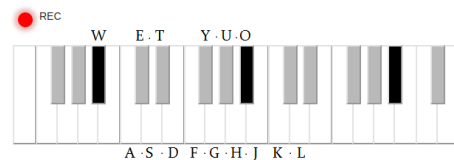

Fig. 2. Screenshot of the piano keyboard, simulating a real one. The equivalence with the computer keyboard is included next to the corresponding keys.

- **Record Button**: Positioned at the top left corner of the piano tab, enables users to play keys without transmitting the notes. This feature is beneficial for sound testing or practicing melodies without affecting search queries. When illuminated in bright red, recording is active, indicating that played notes will be transmitted. Activation or deactivation of recording is achieved with a simple click.
- **Search Bar**: Featuring an HTML input element, the search bar allows users to input queries in musical

notation. As users play notes on the piano keyboard, the search bar dynamically populates with the played melody, contingent upon the recording's activation.

- **Convert and Erase Buttons**: The translation button initiates the conversion process of the musical query into complex Boolean geometrical expressions for search purposes. The resulting query is then transferred to the external search bar of the website. The erase button clears the content of the search bar from the piano dialog. An example of usage for the translation of a melody can be seen in Fig. 3.
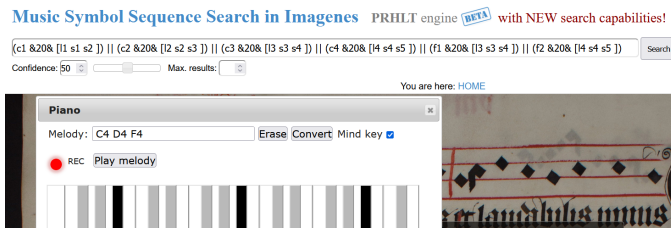

Fig. 3. Screenshot of the piano tab after a melody has been played and translated into a query. The resulting query is shown in the search bar.

- **Play Button**: After melody input, users can utilize the play button to audibly review the entered melody. This functionality enables users to verify the correctness of the melody before executing the search. Or even listening to the melody after the search has been performed.
- **Mind Key Checkbox**: Allows users to determine whether the search should consider the original pitch of the entered melody. When activated ensures that the search respects the original pitch.

The piano interface supports MIDI input, permitting users to play keys using a connected MIDI instrument. These inputs are managed through the MIDI API[8], included in most browsers. Further testing is essential to ascertain the full functionality and performance of the web piano interface. Refinement of its appearance and usability may be necessary to optimize user experience. As such, the interface remains in a testing phase, subject to iterative improvements based on user feedback and evaluation.

V. Evaluation and Results

In this section we aim to determine whether allowing queries based on sequences in musical notation, along with the possible translation without considering the original pitch, result in a good performance of the search engine.

To conduct the evaluation, it is necessary to send a substantial number of melody queries to the demonstrator to obtain a reasonably representative reference (185 musical queries in this case). Then, we can measure the quality of the results depending on the technology used to retrieve the information, i.e. the search method used (three different approaches), together with the type of queries performed (with and without tonality) to assess the search engine's performance.

[8]More information can be found at https://webaudio.github.io/web-midi-api/.

TABLE I
Average Precision (AP) results for the different detection methods and the type of queries.

Method	AP	
	With key	Without key
OP	0.72	0.82
ROP	0.73	0.82
GP	**0.87**	**0.91**

To simulate a realistic testing environment while maintaining stylistic criteria, melodies used for training the developed tool in [9] were employed to create the queries. The effectiveness of information retrieval systems is generally measured using recall and interpolated precision standards [10]. We report results in terms of Average Precision (AP), defined as the area under the recall-precision (R-P) curve. The higher its value the better the system's performance. The set of staves in which the search has been performed is the same as in [4].

Within this evaluation, different approaches to detect queries of musical sequences in handwritten scores were employed in the experimentation. Three detection alternatives were tested: based on the logical positions of the indexed detections (OP), based on the logical positions whose order is consistent with their geometric location (ROP), and based solely on the geometric positions of the indexed detections (GP).

Tab. I presents the Average Precision (AP) results for all these combinations, where the use of GP together with tonality-free queries achieves the best result.

VI. Conclusion

In this paper, we introduced advancements in user-machine interaction for searching historical handwritten scores. By enabling queries using musical notation symbols within the web-based search engine in https://prhlt-carabela.prhlt.upv.es/musica, we enhanced its usability. Thanks to the implementation of the piano interface, users can now input queries intuitively with real notes on the staff.

Our approach addresses complexities in historical scores, such as variations in clef types and positions, by transforming musical queries into Boolean geometrical expressions. Evaluation results demonstrate the effectiveness of our method, especially when combining geometric positions with tonality-free queries.

While our results provide a promising starting point, further refinement and testing (specially of the web piano interface) are necessary. It is important to note that direct comparisons with previous studies [4] may not be feasible due to differences in methodologies and evaluation criteria.

These enhancements represent a meaningful advance in enabling discoveries in musicology and historical research, especially given the widespread use of the treated notation style across the vast corpus of musical documents. Future research could extend this methodology to other notational systems and later musical sources, incorporating additional elements (rhythm, polyphony, etc.). Furthermore, continued testing and refinement will be essential for optimizing user experience and maximizing the impact of our approach in terms of computational efficiency.

REFERENCES

[1] J. Calvo-Zaragoza, J. H. Jr., and A. Pacha, "Understanding optical music recognition," *ACM Comput. Surv.*, vol. 53, no. 4, jul 2020. [Online]. Available: https://doi.org/10.1145/3397499

[2] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, vol. 1, pp. 173–190, 2012.

[3] M. Villarreal and J. A. Sánchez, "Handwritten music recognition improvement through language model re-interpretation for mensural notation," in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, pp. 199–204.

[4] J. Calvo-Zaragoza, A. H. Toselli, E. Vidal, and J. A. Sánchez, "Music symbol sequence indexing in medieval plainchant manuscripts," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 882–887.

[5] D. Lacoste, "The cantus database: Mining for medieval chant traditions," *Digital Medievalist*, vol. 7, 2012.

[6] L. P. J. H. J. Ashley and B. I. Fujinaga, "Gamera versus aruspix two optical music recognition approaches," *ISMIR 2008*, p. 139, 2008.

[7] Y.-H. Huang, X. Chen, S. Beck, D. Burn, and L. Van Gool, "Automatic handwritten mensural notation interpreter: From manuscript to midi performance." in *ISMIR*, 2015, pp. 79–85.

[8] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, "Handwritten music recognition for mensural notation with convolutional recurrent neural networks," *Pattern Recognition Letters*, vol. 128, pp. 115–121, 2019.

[9] P. P. Cruz-Alcazar and E. Vidal-Ruiz, "Modeling musical style using grammatical inference techniques: a tool for classifying and generating melodies," in *Proceedings Third International Conference on WEB Delivering of Music*. IEEE Comput. Soc, 2004.

[10] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 39.

The CollabScore project – From Optical Recognition to Multimodal Music Sources

Bertrand Coüasnon *Univ. Rennes, CNRS, IRISA, INSA Rennes* bertrand.couasnon@irisa.fr
Mathieu Giraud *CNRS/Univ. Lille* mathieu.giraud@univ-lille.fr
Christophe Guillotel Nothmann CNRS/Sorbonne Univ. christophe.guillotel-nothmann@cnrs.fr
Aurélie Lemaitre *Université Rennes 2, CNRS, IRISA* aurelie.lemaitre@irisa.fr
Philippe Rigaux Cnam, philippe.rigaux@lecnam.net

Abstract—We introduce CollabScore, a project funded by the French National Research Agency, devoted to the design and production of tools and methods to improve accesses to large collections of sheet music scans. The new optical music recognition (OMR) approach developed in CollabScore is part of a larger goal, namely that of interlinking multimodal documents related to music works. In this perspective, the music notation obtained from the OMR process is seen as a *pivot* that associates related fragments of images, audio, video, XML, or text sources. As an application of this principle, CollabScore supports the *synchronization* of sources, leveraging the raw content of digital libraries with listening and visualization experiences. The present paper introduces the project and exposes some of its current achievements.

I. Overview

The core concept of the project is that of *multimodal music sources* and the main project's efforts aim at creating tools and methods to *interlink* these sources. We begin with an overview of this perspective before surveying some more technical aspects.

A. Multimodal music sources

Given a music work (say, the Goldberg variations) seen as an abstract entity, we can find many concrete documents that provide a specific representation. These documents can be recordings, in audio or video format, images (scans) of score sheets, editable scores in MusicXML or MEI, and even textual sources that comment/annotate/enrich the music. It turns out that each representation is difficult to use beyond its specific purpose. For non specialists, we know it is hard to "hear" the music from a score and, conversely, it is hard to "replay" or analyse the music from a performance, live or recording. Moreover, sources are usually self-contained, independent documents, encoded in some specific format. This keeps from easily mapping music components (a voice, an harmonic sequence, a phrase) from one source to another, at a finer level of granularity than the whole document itself.

In CollabScore, we address these issues with *multimodal music scores* (MMS). A MMS combines an encoding of the music notation (a MEI file) with links that associate the notation elements to the corresponding fragments of multimedia sources, e.g., a region on an image, a time frame in an audio/video source, as section of a textbook. Music notation is thus used as a description language for music content, which

serves as a reference, or *pivot* to link heterogeneous sources that encode the same content.

CollabScore implements this model in a data store[1] which provides (i) a management of such *pivot scores*, (ii) a storage of each pivot with external or internal multimedia sources, and (iii) an annotation mechanism that maps the pivot fragments to the corresponding part of each source [1], [2]. Figure 1 shows an example of a MMS: the *pivot score* (here, *La coccinelle*, a melody from Saint-Saëns) stored as a MEI document in Neuma is the central piece that glues together several sources: an image (taken from the Gallica digital library), a video accessible on YouTube, a MIDI file (internal source).
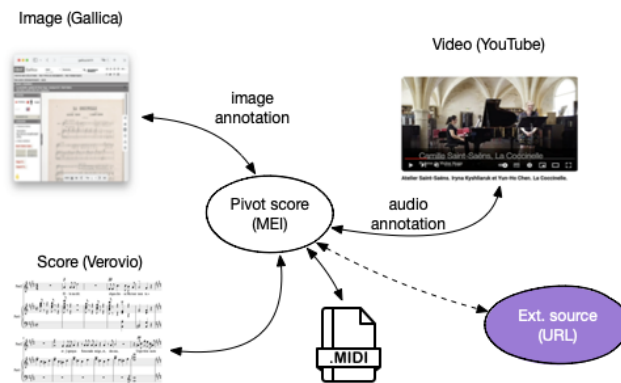

Fig. 1. A multimodal score and its sources

The project's work consists in designing tools to produce and manage MMS, including a powerful OMR system which the privileged mean to obtain a pivot. They are briefly summarized below.

B. Producing the pivot via optical recognition and crowdsourcing

Although pivot scores could be obtained by edition or transcription, CollabScore integrates Optical Music Recognition (OMR) as the primary mean to produce a notation from image sources. In this context, our definition of "OMR" corresponds to the class of "structured encoding" OMR in [3]:

---

we ambition to produce an editable score featuring all the notation elements visible on the sheet scan, along with their proper interpretation. In others words, we implement a process that attempts to invert the production of a printed score from specifications entered in a music notation engraver. Moreover, we combine this process with crowdsourcing phases to achieve a high-quality output, as discussed in [4]. The process is validated on a corpus mostly taken from the BnF Gallica Digital Library. These aspects are covered in Section II.

C. Alignment of sources

Multimedia sources are *aligned* with the pivot as shown on Fig. 2. The XML encoding of the notation (in MEI) identifies each component (here, a chord) with a unique id which is the target of annotations that refer to the corresponding fragments of sources. In the case of image, the annotation specifies a region on the image; in the case of audio/video, a time frame gives the start/end of the fragment.

Fig. 2. Aligning sources: A multimodal score with three documents

The alignment methods depends on the sources. In the case of images, annotations are supplied by the OMR system as a side effect of the recognition process. For other sources, dedicated interfaces have been implemented (Section III).

D. Applications

Finally, through the music description available in the pivot score, the content of two sources can be associated at a fine granularity level. The OMR output for instance can be controlled by a side by side display of both the source image and the pivot score rendering. Textual annotation (e.g., analytic comments) can be added on a score image at precise positions. An interface developed in COLLABSCORE allows to listen an audio/video source while highlighting the music being played on the original image source. Among many other advantages, this is likely to greatly leverage the content of digital libraries with attractive features (details in Section III).

II. THE OMR PROCESS

Among the various works on OMR [3], [5], two main types of approach can be observed in recent work. One is based on the detection of musical symbols [6], [7], inspired by architectures developed for object detection in natural scenes, with problems specific to OMR related to the large size of the images to be processed and the very small size of some musical symbols. The other is based on end-to-end recognition methods that directly produce a representation of the recognized score, which initially tackled monophonic scores and only very recently have been able to start to handle polyphonic systems [8], [9]. For the moment, these methods do not produce the localization of the recognized information required, for example, for image-sound synchronization.

The OMR process we propose in COLLABSCORE to deal with polyphonic orchestra scores is founded on DMOS method, completed with a collaborative process that aims at clarifying the interpretation of symbols that have been identified as ambiguous. We experiment this combination of a large corpus for which a reference encoding has been produced.

A. Automatic syntactic OMR with DMOS

DMOS [10] relies a grammatical method that enables the combination of visual clues with syntactic rules, in order to describe both the physical and the logical content of the document. The process follows two steps, as shown in Fig. 3.
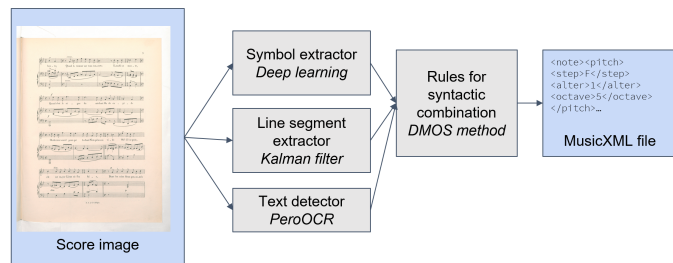

Fig. 3. Overview of DMOS: combination of low level detectors and high level syntactic rules

In a first step, three low level extractors are applied on the image:

- a symbol extractor based on deep learning (Cascade R-CNN - FocalNet architecture), dedicated to the extraction of small musical symbols [11] from high-resolution full-page images;
- an existing line segment extractor, based on Kalman filtering [12], used to extract linear elements, such as staff lines and stems;
- the existing PeroOCR [13], for the extraction of textual elements, such as titles, lyrics, instrument names.

In a second step, those elements are given as input to a syntactic system, based on DMOS method [10]. It produces a description of the graphical and syntactic content of the musical content of a score image: a score is made of staff systems, containing measures, and each measure contains musical objects (notes, rests, ...) that respect time constraints.

Recognizing a measure involves three steps of analysis. First, the staves and barlines are identified. Then, inside of a score, the graphical content is detected based on the position and assembly constraints of both the symbols detected by the deep object detector and the linear elements extractor: key, notes, rests, dots, accidentals, ties, slurs, dynamics, articulations marks, lyrics... Each detected content is localized in the image, and produced with is associated bounding box (Fig. 4).

Finally, the system organises the content into voices. After the distribution of notes into voices, the system checks the global consistency of the recognition, and produces warning if the detected elements do not follow some given rules. For example, if a eight note is miss-detected, the system will trigger a warning because the time signature is not respected.

Fig. 4. The OMR process: Detection of graphical content

Fig. 5. The collaborative process, phase 1: checking parts and their staves

Moreover, based on the vertical alignment of notes in a system, it is possible to locate or even correct the note with the wrong duration. Applying these rules makes detection more reliable in a context of ancient noisy documents.

All the elements identified by DMOS are organized in a document compliant with the music notation grammar. This document is the main source of information to initiate a multimodal music source in COLLABSCORE. Indeed, from the music notation symbols, a symbolic score in MEI is reconstructed – the pivot, and the source image is aligned with this pivot thanks to the regions identified by DMOS. Moreover, the alerts raised by DMOS are recorded and subsequently submitted to the collaborative process.

B. The collaborative process

The "raw" music score obtained from the OMR process enters in a phase of corrections via a sequence of dedicated interfaces. A design choice of COLLABSCORE is to limit user actions to the list of alerts raised by the DMOS component. While this may seem restricted, we believe that going beyond would ultimately lead to implement a full online score editor[2].

The advantage of considering only the DMOS alerts is that we remain within the scope of an automatic recognition process, augmented with a one-time human assistance to solve difficult cases. This limits the competency expected from users, as well as the complexity of the required actions since they essentially consist in answering a question. This choice also provides a sound basis to evaluate the performance of DMOS: Given a ground truth, we can compare it first to the raw output, and second, to the corrected one, identifying the impact of human interaction on the final quality.

The list of alerts raised by DMOS are classified in three categories, based on how globally the potential error may impact the resulting score. These categories result in three correction phases:

- The first one, called *Instrumentation*, refers to the identification of music parts, and to the correct assignment of staves to the parts. Any error on these structural aspect has a dramatic impact on the whole score. This is the case for instance of a double-staff piano part not recognized as such, or when some parts are introduced/removed from one system to the other (e.g., a solo/melody arriving after an instrumental introduction, resulting in the introduction of a new staff in systems). Special cases difficult to
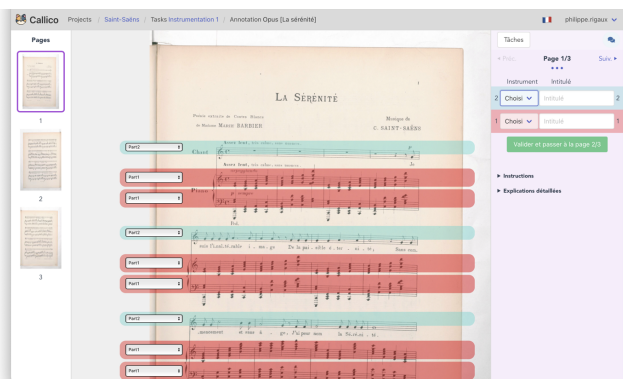
identify automatically (e.g., transposing instrument) can also be solved during this step.
- The second one, *Transcription context*, refers to all the notation element that dictates the transcription of music events: clefs, key signatures and time signatures. Here again, any misinterpretation severely hinders the music notation accuracy.
- Finally, the last phase, *Music objects*, addresses the notation of musical events: notes, chords, rests, ties. At this point, the user cann *locally* correct a property of a faulty music object: duration, height, etc.

For each phase, a list of microtasks is produced, and submitted to a group of users. At the end of each phase, the list of validated corrections is applied to the score, and this corrected version is proposed to the following phase.

Fig 5 shows an example of the user interface dedicated to the first phase (Instrumentation). It heavily relies on information obtained from the DMOS analysis which comes as the default interpretation. Here, the list of parts (chant and piano) has been identified, and each staff (or pair of staves) assigned to a part. The user can correct this information if needed.

The subsequent phases imply a display of both the initial image and the score for comparison purpose (see Fig. 6 for phase 2). Elements to be controlled (here, clefs and signatures) can be highlighted on both the image and the target score, thanks to the regions provided by the OMR and to the links between both sources. We implemented an interface that lets the user directly correct an object (a clef, Fig. 6), each action being immediately reported on the score.

At the time of writing, we are finalizing the implementation of the collaborative system. It is based on the Open-source Callico system [14] and available at https://collabscore.cnam.fr. An experiment will be conducted in early 2025 with a group of users on a large corpus to be described next.

C. The reference corpus

The reference corpus comprises all the works by Camille Saint-Saëns (1835-1921) with the exception of dramatic works (operas, oratorios, incidental music). Aside from considerations relating to the BnF's promotion policy – COLLABSCORE coincided with a project to promote the composer's work on

---

[2]Note that it always remain possible to import the MEI or MusicXML output in a standard score engraver
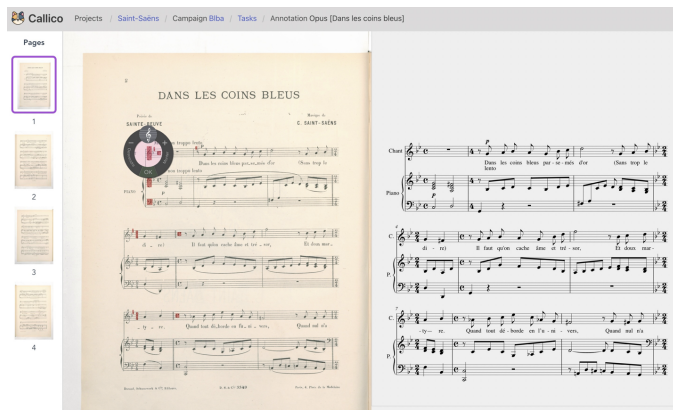
Fig. 6. The collaborative process, phase 2: checking the transcription context

the occasion of the hundredth anniversary of his death in 2021 – two criteria prevailed in the selection of this corpus, which totals more than 500 compositions.

1) **Variety of genres and instrumentation.** The compositions include sacred and secular works for a *capella* choir, chamber music, melodies for voice and piano, compositions for brass or military bands, keyboard repertoire and symphonic works with or without solo instruments. This diversity allows the software solution to be tested in different situations that present particular challenges, such as cross-staff notation in piano works, transposing instruments in orchestral works, or syllable positioning in melodies, etc.

2) **Particularities of French printed music from the period 1850-1920.** These scores, which have been made available by the BnF on Gallica, differ from modern, standardised notation, with regard to their implicit features (e.g. triplet notation), special signs (crochet rests), complexities relating to the placement of the text and the presence of artifacts in the preserved scores. Thus, we see this case study as an appropriate starting point for follow-up projects dedicated to printed music from earlier periods and handwritten notation.

For all the items, MEI files were created containing mei-headers with metadata extracted from Gallica including title, date of creation, genre, authorial attribution(s), historical print identifier, location and physical description. A sample of 18 scores was then transcribed in full, either manually or using commercial software (PhotoScore) with post-correction.

The reference corpus will serve as a ground truth to evaluate the performance of DMOS (for raw output) and of the collaborative phases (for users-corrected output). OMR evaluation is a notably difficult task [15]–[17] and we hope to contribute to progresses in this field. We started using the MusicDiff tool, designed by one of the project's partners [18] and now available as a Python package [3], but additional work is required with the OMR community to achieve a commonly accepted yardstick.

III. Sources alignment and synchronisation

Once obtained, the pivot score can be aligned with multimedia sources. We tailored the Dezrann platform [19] of our partner Algomus to propose tools for synchronization and synchronized score playback. Regarding images, as shown on Fig. 4, we can rely on the bounding box supplied by DMOS for each detected symbol, but also for all the measures, staves and systems. We link this region to the corresponding element ID in the pivot document.

Aligning with recordings (audio or video) involves identifying the time frame at the finest possible temporal granularity (we target the beat level). The fields of audio-score alignment and score following are actively researched [20]–[22]. Common methods involve dynamic time-warping algorithms or, more recently, deep learning approaches. In particular when sections are repeated. user interaction is often necessary to achieve a satisfying correspondence. We designed a simple interface to let users add and update alignment timestamps [19].

Finally, as a demonstration of the potential of our work to promote the content of digital libraries to a wide audience, COLLABSCORE proposes an interface where the sources of a multimodal score can be displayed simultaneously for an improved user experience. Fig 7 shows how the original Gallica image, the pivot score and a YouTube recording can be associated, exhibiting at any moment a close correspondence between the performance, the notation, and the original image.
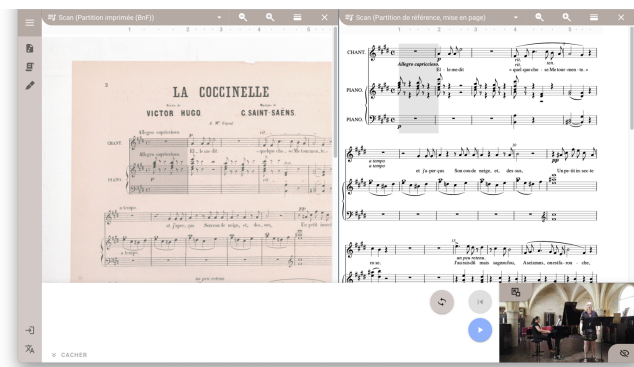

Fig. 7. COLLABSCORE interface showing three synchronized sources on *La Coccinelle* with the Dezrann libraries: the original image, the pivot score, and a YouTube performance.

IV. Conclusion

The COLLABSCORE project addresses many challenges in modeling and interlinking multimodal documents related to music, and has already required a lot of efforts to achieve its current state in OMR, collaborative process, score synchronization and playback. Each aspect would obviously deserve a much more detailed presentation and require further research and development, but we believe the the results obtained so far seem very promising. We are keen to showcase the COLLABSCORE project with the community, and obtain in return an informed feedback.

---

[3]https://github.com/gregchapman-dev/musicdiff

[3]https://gallica.bnf.fr/ark:/12148/bpt6k1162049x

REFERENCES

[1] S. Cherfi, C. Guillotel, F. Hamdi, P. Rigaux, and N. Travers, "Ontology-Based Annotation of Music Scores," in *Intl. Conf. on Knowledge Capture (K-CAP'17)*, 2017, austin, Texas, Dec. 4-6 2017.

[2] R. Sanderson, P. Ciccarese, and B. Young, "Web annotation data model," Technical report, W3C Recommendation, 23 February, Tech. Rep., 2017.

[3] J. Calvo-Zaragoza, J. Hajic, and A. Pacha, "Understanding optical music recognition," *ACM Computing Surveys (CSUR)*, vol. 53, pp. 1 – 35, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID: 199543265

[4] C. Saitis, A. Hankinson, and I. Fujinaga, "Correcting large-scale OMR data with crowdsourcing," in *1st International Workshop on Digital Libraries for Musicology*. ACM, 2014, pp. 1–3.

[5] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marçal, C. Guedes, and J. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, vol. 1, pp. 173–190, 2012.

[6] L. Tuggener, Y. P. Satyawan, A. Pacha, J. Schmidhuber, and T. Stadelmann, "The DeepScoresV2 dataset and benchmark for music object detection," in *Proc. ICPR*, 2021, pp. 9188–9195.

[7] Y. Zhang, Z. Huang, Y. Zhang, and K. Ren, "A detector for page-level handwritten music object recognition based on deep learning," *Neural Comput. Appl.*, 2023.

[8] J. Mayer, M. Straka, J. Hajič, and P. Pecina, "Practical end-to-end optical music recognition for pianoform music," in *Document Analysis and Recognition - ICDAR 2024*, E. H. Barney Smith, M. Liwicki, and L. Peng, Eds. Cham: Springer Nature Switzerland, 2024, pp. 55–73.

[9] A. Ríos-Vila, J. Calvo-Zaragoza, and T. Paquet, "Sheet music transformer: End-to-end optical music recognition beyond monophonic transcription," in *Document Analysis and Recognition - ICDAR 2024*, E. H. Barney Smith, M. Liwicki, and L. Peng, Eds. Cham: Springer Nature Switzerland, 2024, pp. 20–37.

[10] B. Coüasnon, "DMOS, a generic document recognition method: Application to table structure analysis in a general and in a specific way," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 8(2), pp. 111–122, 2006.

[11] A. Yesilkanat, Y. Soullard, B. Coüasnon, and N. Girard, "Full-page music symbols recognition: state-of-the-art deep models comparison for handwritten and printed music scores," in *DAS 2024 Workshop on Document Analysis System*, Sep. 2024.

[12] C. Queguiner, J. Camillerapp, and I. Leplumey, "Kalman Filter Contributions Towards Document Segmentation," in *ICDAR 1995 Third International Conference on Document Analysis and Recognition*, Montreal, Canada, Aug. 1995, pp. 765–769.

[13] O. Kodym and M. Hradis, "Page layout analysis system for unconstrained historic documents," *CoRR*, vol. abs/2102.11838, 2021.

[14] C. Kermorvant, E. Bardou, M. Blanco, and B. Abadie, "Callico: A versatile open-source document image annotation platform," in *Document Analysis and Recognition - ICDAR 2024: 18th International Conference, Athens, Greece, August 30 – September 4, 2024, Proceedings, Part III*. Berlin, Heidelberg: Springer-Verlag, 2024, p. 338–353. [Online]. Available: https://doi.org/10.1007/978-3-031-70543-4_20

[15] D. Byrd and J. G. Simonsen, "Towards a standard testbed for optical music recognition: Definitions, metrics, and page images," *Journal of New Music Research*, vol. 44, no. 3, pp. 169–195, 2015.

[16] J. j. Hajič, "A case for intrinsic evaluation of optical music recognition," in *1st International Workshop on Reading Music Systems*, J. Calvo-Zaragoza, J. H. jr., and A. Pacha, Eds., Paris, France, 2018, pp. 15–16. [Online]. Available: https://sites.google.com/view/worms2018/proceedings

[17] P. Torras, S. Biswas, and A. Fornés, "A unified representation framework for the evaluation of Optical Music Recognition systems," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 27, no. 3, pp. 379–393, 2024.

[18] F. Foscarin, F. Jacquemard, and R. Fournier-S'niehotta, "A diff procedure for music score files," in *6th International Conference on Digital Libraries for Musicology*, 2019, pp. 58–64.

[19] L. Garczynski, M. Giraud, E. Leguy, and P. Rigaux, "Modeling and editing cross-modal synchronization on a label web canvas," 2022.

[20] M. Dorfer, F. Henkel, and G. Widmer, "Learning to listen, read, and follow: Score following as a reinforcement learning game," in *Proceeding of International Conference on Music Information Retrieval (ISMIR)*, 2018.

[21] J. Thickstun, J. Brennan, and H. Verma, "Rethinking evaluation methodology for audio-to-score alignment," *arXiv preprint arXiv:2009.14374*, 2020.

[22] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync Toolbox: A Python package for efficient, robust, and accurate music synchronization," *Journal of Open Source Software (JOSS)*, vol. 6, no. 64, pp. 3434:1–4, 2021.

Semi-Automatic Annotation of Chinese Suzipu Notation Using a Component-Based Prediction and Similarity Approach

Tristan Repolusk*† and Eduardo Veas*†

*Graz University of Technology, †Know Center Research GmbH

Email: †trepolusk@know-center.at, *eveas@tugraz.at

ORCID: 0009-0009-8435-1185, 0000-0002-0356-4034

Abstract—In recent years, the development of Optical Music Recognition (OMR) has progressed significantly. However, historical or smaller music cultures have only recently been considered in this process. This also includes Chinese music notations such as *suzipu*. In this paper, a component-based clustering and similarity approach for this kind of notation is introduced. Its goal is to facilitate and automate the manual annotation of such notation instances through artificial intelligence, resulting in a more efficient digitization of machine-readable digital archives which can also be used for OMR applications. The *suzipu* notation studied in this work is taken from the KuiSCIMA dataset, containing Jiang Kui's influential collection *Baishidaoren Gequ* 白石道人歌曲 from 1202. This contribution serves as the basis for the further development of OMR algorithms acting on *suzipu* and similar kinds of notations, thus fostering the dissemination, preservation and computational analysis of historical Chinese music.

Index Terms—Chinese music, Jiang Kui, optical music recognition, suzipu, banzipu, Baishidaoren Gequ

I. Introduction

The field of OMR is similar to optical character recognition (OCR) regarding the extraction of information that is available as optical data. However, recovering the musical semantics (such as pitch, onset, duration, velocity) is a crucial step and often involving implicit rules. This makes OMR tasks extremely challenging [3]. A well established process pipeline of OMR comprises phases of preprocessing, symbol recognition, notation assembly, and encoding [12, 3].

OMR systems are at best incomplete when it comes to transcribing music scores as regarding the fulfillment of all transcription phases, therefore inviting the use of at least partial manual workflows [16]. This is perhaps even more accentuated when dealing with handwritten scores [2]. The majority of the works concern common practice period music scores, and most approaches do not support notations of other kinds of musical traditions.

Baishidaoren Gequ is an important work in the history of Chinese music. It is a compilation of the works of Jiang Kui 姜夔, a renowned poet, calligrapher, and music theorist of the Song dynasty (1127-1279 CE), also known by his courtesy name Baishi 白石. This collection is one of the earliest surviving examples of melodized lyrics in Chinese history [9], reflecting the sophisticated musical culture of the Southern Song period. It provides researchers with valuable information about the musical practices and aesthetics of the Song Dynasty, making it an important resource for studying the history of Chinese music. The collection primarily consists of *ci* poetry set to music, covering various themes including nature, emotions, and historical events.

17 out of the 109 pieces featured in *Baishidaoren Gequ* are endowed with the *suzipu* 俗字谱 (literal meaning: common character notation) notation, also known as *banzipu* 半字谱 (literal meaning: half character notation). This kind of notation was especially common in China in Song dynasty (960−1279), with the 17 pieces in *Baishidaoren Gequ* being the largest historical source of this notation. Five handwritten editions of *suzipu* notation with optical annotations are contained in the publicly available KuiSCIMA[1] dataset [13]. Also some contemporary musical practices such as *Xi'an Guyue* 西安鼓乐 use related notations [7].

In the context of *suzipu*, many challenges arise for OMR:

- The notations consist of a pitch and secondary component that can be realized in notation by different kinds of compositions (such as top-bottom or left-right).
- The number of existing labeled samples is scarce and some symbols appear rarely.
- The music notations may have implicit relationships to the poetry that is accompanying them.
- Only a handful of experts worldwide have deep insights into the musical semantics of the score.

Therefore, in this work the manual digitization is facilitated through AI techniques that are embedded into the graphical user interface of the *Chinese Musical Annotation Tool* [14], thus giving rise to a semi-automated annotation approach that is designed to guide and support the human annotator.

---

[1]https://github.com/SuziAI/KuiSCIMA

II. Related Works

The simultaneous recognition and encoding of music scores has been shown feasible, albeit for notation types for which large annotated datasets exist [15, 10]. A major problem arises with the lack of large volume data for training and testing the benchmark model in OMR systems, which can be overcome with a semi-automated human-in-the-loop approach [6, 8].

Handwritten scores pose additional challenges due to nuances inherent in individual handwriting. MuRET (Music Recognition, Encoding, and Transcription), designed for music transcription and OMR, focuses on repertoires of handwritten monodic melody scores of traditional Spanish music and white mensural notation from 16th to 18th-century manuscripts [16].

Regarding Chinese notations, an OMR architecture was designed and evaluated with 100 songs of a Chinese songbook with a regular structure of monophonic score involving Chinese number notation *jianpu* 简谱 [17]. Two other works focused on *gongche* notation in *Kunqu* opera: In [5] a comparison of different algorithms for recognition of *gongche* pitches is presented, while [4] focuses on extracting semantic information taking into account the spatial structures of *Kunqu* opera pieces.

III. Methods

The *suzipu* notation is characterized by having two properties: a pitch component indicating the syllable's pitch, and an optional secondary component providing rhythmical and ornamentation information [13]. The individual components with their machine-readable representations are found in Table I.

In this section, the three intelligent user interface components facilitating the manual annotation of *suzipu* notation instances are introduced. The user interface can be seen in Figure 1.

Button (1) in the GUI starts the automatic OMR prediction, and button (2) can be used to overwrite the annotations in the tool with the model predictions. (3) and (4) provide additional context, where (3) contains the OMR model predictions with confidence scores, and in (4), the most similar notations with respect to their optical features are shown. (5) is a display of all instances in KuiSCIMA that have the same annotation as the one assigned to the symbol by the user.

A. Suzipu Classification

For the prediction, the currently available state-of-the-art OMR model introduced in [13] is used. The classifier consists of two small convolutional neural networks leveraging the special structure of *suzipu* notation. The first classifier is for the pitch component, while the second classifier deals with the notation's secondary component. Similar methods have already successfully been used in settings where images share features that can be described by product spaces, e.g. the Ethiopic script in [1].

Suzipu (Pitch)	Name	ASCII Representation
ㄙ	合	"HE"
ㄱ	四	"SI"
一	一	"YI"
�厶	上	"SHANG"
ㄴ	勾	"GOU"
人	尺	"CHE"
フ	工	"GONG"
ㄱ	凡	"FAN"
ㄨ	六	"LIU"
め	五	"WU"
る	高五	"GAO_WU"

Suzipu (Secondary)	Name	ASCII Representation
め	大頓	"DA_DUN"
り	小住	"XIAO_ZHU"
フ	丁住	"DING_ZHU"
カ	大住	"DA_ZHU"
ら	折	"ZHE"
J	拽	"YE"

TABLE I: The machine-readable representations of each of the 11 pitch and 6 secondary components of *suzipu* notation. The representation is the capitalized pinyin realization of the symbol's name without tone marks.
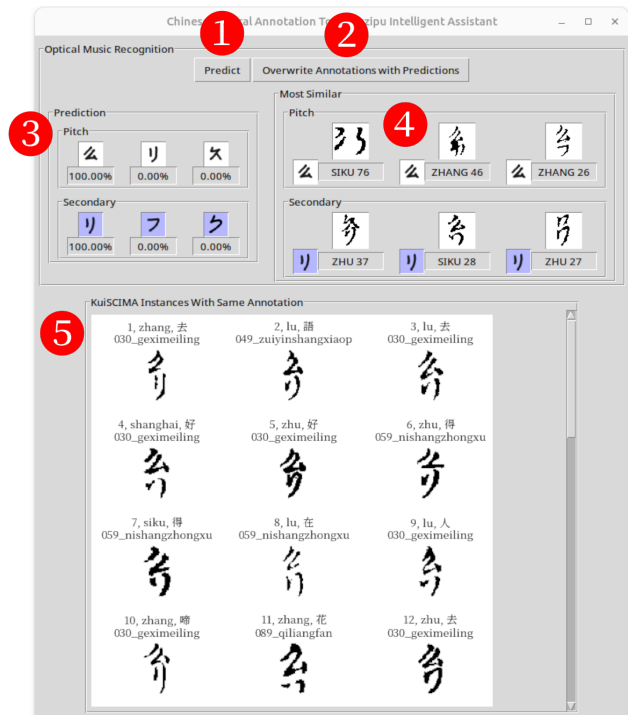

Fig. 1: The `Suzipu Intelligent Assistant` window allows for visualization and automatic labeling of notation instances.

This decomposition is meaningful in settings of limited or imbalanced data, as is the case with the KuiSCIMA dataset. It is quite small with 7212 non-empty notation annotations, highly imbalanced, and out of the 77 classes only 66 occur. This makes the OMR on this kind of notation especially challenging.

The details about the training and validation processes regarding the classifier is found in [13]. On unseen test data (consisting of *Baishidaoren Gequ*'s Shanghai MS edition), the best classifier achieves a single character error rate (CER) of 10.4% on the best model combination. Therefore, a user must only correct around every 10th model prediction instead of manually annotating every single instance.

To make the OMR algorithm more interpretable to users of the GUI, the classifiers are calibrated using a temperature scaling approach on their respective validation sets. For the pitch classifier, the temperature is trained with 10 epochs and an SGD learning rate of 0.01, resulting in a temperature of 1.4574. The secondary classifier's temperature is trained with a learning rate of 0.005, yielding a temperature of 1.2430. The reliability plots in Figure 2 show that the calibrated models of the *suzipu* pitch classifier are well-calibrated for confidences greater than 50%. Since the test dataset is with 1439 instances quite small, a good calibration on the whole confidence scale is not feasible.

In (3), the model predictions with the corresponding confidence scores are displayed.

B. Similarity Visualization

As an additional guidance for the user, the similarity visualization in (4) displays the three most similar notation instances found in in the KuiSCIMA dataset with respect to their optical features.

For each of the two component classifiers models, i.e., the layer `fc2` (which is a 120-dimensional vector) is extracted as a feature encoding of the respective property. The features are collected for each notation in the dataset, and an unsupervised UMAP [11] dimensionality reduction to 2D is applied with a `random_state` of 42. Visualizations of the UMAP spaces with dataset samples are found in Figure 3.

The UMAP representation of the currently investigated notation instance is compared against the precomputed UMAP representation of all instances in the KuiSCIMA dataset, and the three nearest neighbors are retrieved using K-means. The displayed similarity score is calculated as the inverse euclidean distance between the current instance and the neighbor.

C. Display of Instances with Same Annotation

In (5), KuiSCIMA dataset instances with the same annotation as the currently investigated notation instance are displayed, which is useful to validate the currently annotated sample against already annotated instances
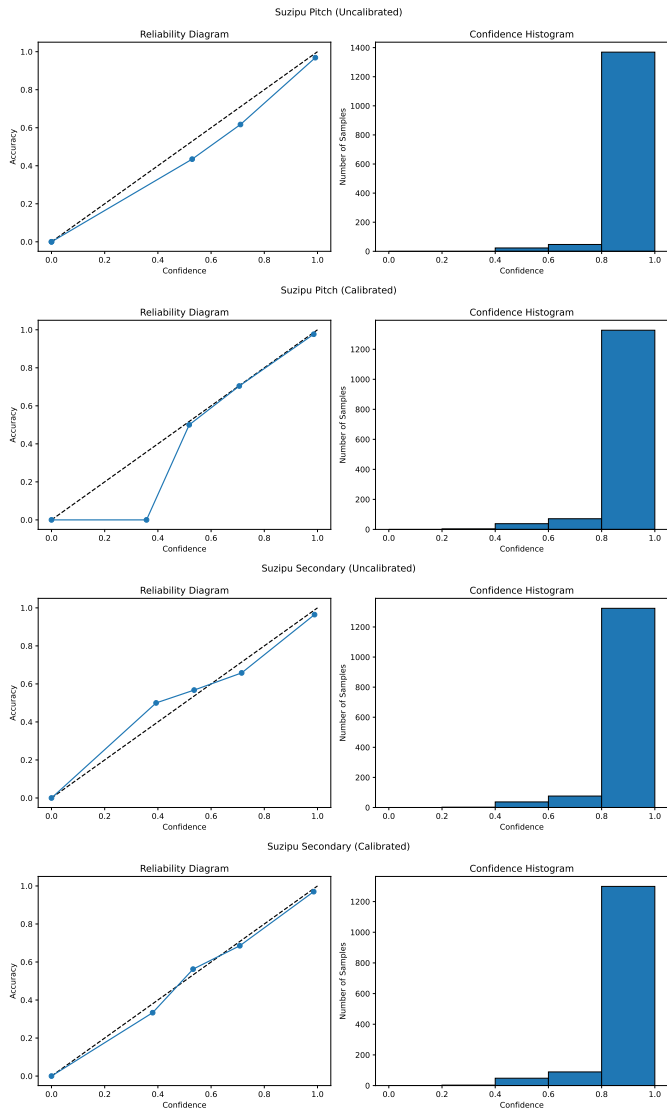

Fig. 2: Reliability scores for the individual OMR classifiers indicate a good calibration for confidences greater than 50%.

featuring the same label, and no optical features are used for this. However, since some classes are very frequent and occur more than 800 times, not all instances can be displayed, and an intelligent selection is made.

In the case that an annotation occurs up to 39 times in the dataset, this selection is just the instances themselves. However, if there are more than 39 instances with this annotation in the dataset, the raw 28x28 pixel images are clustered into 39 classes using K-means. From each of those clusters, the first element is chosen as a representative. With this method, the possibly large amount of total samples is reduced to a small selection of most diverse samples in pixel space.

The images for each annotation class are pre-generated and loaded when starting the tool to reduce loading times.

Suzipu Pitch Embeddings (UMAP)

Suzipu Pitch Embeddings (UMAP)

Suzipu Secondary Embeddings (UMAP)

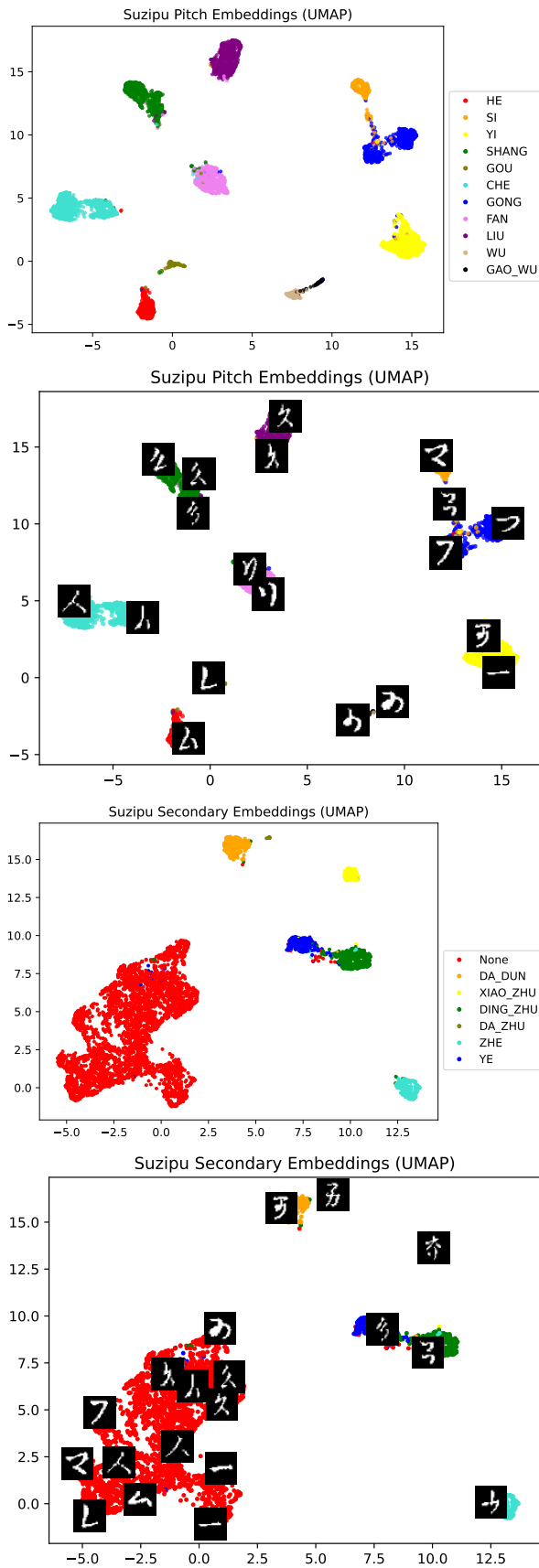Suzipu Secondary Embeddings (UMAP)

Fig. 3: UMAP embedding space visualization for both the pitch and secondary component classifiers. The second and fourth images show the space with some dataset examples.

IV. Conclusion and Future Work

In this paper, an extension of the *Chinese Musical Annotation Tool suzipu* musical notation was introduced for semi-automatic collection of OMR datasets and ethnomusicological archives. This is fostering the dissemination and preservation of cultural heritage and laying the foundation for computational studies of *suzipu*. Since the involved datasets and the annotation tool are open source and publicly available[2], a great impact on these areas of research is expected.

In order to provide a good user experience and make the annotation of *suzipu* as easy as possible, the tool is endowed with intelligent systems: OMR algorithms for the automated annotation of notation symbols, similarity visualization to discover dataset samples that exhibit similar optical features, and a clustering of all annotation instances that occur in the KuiSCIMA dataset.

In the case of *suzipu* notation OMR, a user must only correct around every 10th model prediction instead of manually annotating every single instance. A model calibration allows the user to interpret the confidence of the model's predictions. With this, a significant reduction of human effort is achieved.

For future work, we propose the following directions:

1) Conducting user studies for a practical evaluation of the annotation tool involving multiple human annotators.
2) Developing suitable OMR algorithm for the other two kinds of musical notations that appear in *Baishidaoren Gequ* (*lülüpu* 律吕谱 and *jianzipu* 减字谱) and its integration into the GUI.
3) Incorporating other kinds of Chinese or related musical notations, such as *gongchepu* 工尺谱 as it is used for Chinese music theaters *Kunqu* 昆曲 or *Jyutkek* 粤劇, or even the extension to Japanese or Korean musical notations.
4) Creating an educational intelligent user interface for interactive teaching and learning of ancient Chinese music notations.

References

[1] Birhanu Belay et al. "Factored Convolutional Neural Network for Amharic Character Image Recognition". In: *2019 IEEE International Conference on Image Processing (ICIP)*. 2019, pp. 2906–2910. DOI: 10.1109/ICIP.2019.8804407.

[2] Manuel Burghardt and Sebastian Spanner. "Allegro: User-centered Design of a Tool for the Crowdsourced Transcription of Handwritten Music Scores". In: *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage*. DATeCH2017. Göttingen, Germany: Association for Computing Machinery, 2017, pp. 15–20. ISBN: 9781450352659. DOI: 10.1145/3078081.

[2]https://github.com/SuziAI

3078101. URL: https://doi.org/10.1145/3078081.3078101.

[3] Jorge Calvo-Zaragoza, Jan Haji Jr., and Alexander Pacha. "Understanding Optical Music Recognition". In: *ACM Comput. Surv.* 53.4 (July 2020). ISSN: 0360-0300. DOI: 10.1145/3397499. URL: https://doi.org/10.1145/3397499.

[4] Gen-Fang Chen. "Music sheet score recognition of Chinese Gong-che notation based on Deep Learning". In: *2021 International Conference on Big Data Analysis and Computer Science (BDACS)*. 2021, pp. 183–190. DOI: 10.1109/BDACS53596.2021.00048.

[5] Gen-Fang Chen and Jia-Shing Sheu. "An optical music recognition system for traditional Chinese Kunqu Opera scores written in Gong-Che Notation". In: *EURASIP Journal on Audio, Speech, and Music Processing* (2014), pp. 7–17. DOI: 10.1186/1687-4722-2014-7.

[6] Liang Chen, Rong Jin, and Christopher Raphael. "Human-Guided Recognition of Music Score Images". In: *Proceedings of the 4th International Workshop on Digital Libraries for Musicology*. DLfM '17. Shanghai, China: Association for Computing Machinery, 2017, pp. 9–12. ISBN: 9781450353472. DOI: 10.1145/3144749.3144752. URL: https://doi.org/10.1145/3144749.3144752.

[7] Yu Cheng. "Xi'an Guyue −Xi'an Old Music in New China. 'Living fossil' or 'flowing river'?" Dissertation. School of Oriental and African Studies, University of London, 2005. URL: https://eprints.soas.ac.uk/29336/1/10731431.pdf (visited on 08/03/2023).

[8] Stanisaw Graczyk et al. "An Online Tool for Semi-Automatically Annotating Music Scores for Optical Music Recognition". In: *Proceedings of the 11th International Conference on Digital Libraries for Musicology*. DLfM '24. Stellenbosch, South Africa: Association for Computing Machinery, 2024, pp. 73–77. ISBN: 9798400717208. DOI: 10.1145/3660570.3660571. URL: https://doi.org/10.1145/3660570.3660571.

[9] Joseph S. C. Lam. "Ci Songs From the Song Dynasty: A Ménage à Trois of Lyrics, Music, and Performance". In: *New Literary History* 46.4 (2015), pp. 623–646. ISSN: 00286087, 1080661X. URL: http://www.jstor.org/stable/24772762 (visited on 08/02/2023).

[10] Aozhi Liu et al. "Residual Recurrent CRNN for End-to-End Optical Music Recognition on Monophonic Scores". In: *Proceedings of the 2021 Workshop on Multi-Modal Pre-Training for Multimedia Understanding*. MMPT '21. Taipei, Taiwan: Association for Computing Machinery, 2021, pp. 23–27. ISBN: 9781450385305. DOI: 10.1145/3463945.3469056. URL: https://doi.org/10.1145/3463945.3469056.

[11] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". In: *Journal of Open Source Software* 3.29 (2018), p. 861. DOI: 10.21105/joss.00861. URL: https://doi.org/10.21105/joss.00861.

[12] Ana Rebelo et al. "Optical music recognition: state-of-the-art and open issues". In: *International Journal of Multimedia Information Retrieval* 1.3 (2012), pp. 173–190. DOI: 10.1007/s13735-012-0004-6. URL: https://doi.org/10.1007/s13735-012-0004-6.

[13] Tristan Repolusk and Eduardo Veas. "The KuiSCIMA Dataset for Optical Music Recognition of Ancient Chinese Suzipu Notation". In: *Document Analysis and Recognition - ICDAR 2024*. Ed. by Elisa H. Barney Smith, Marcus Liwicki, and Liangrui Peng. Cham: Springer Nature Switzerland, 2024, pp. 38–54. DOI: 10.1007/978-3-031-70552-6_3.

[14] Tristan Repolusk and Eduardo Veas. "The Suzipu Musical Annotation Tool for the Creation of Machine-Readable Datasets of Ancient Chinese Music". In: *Proceedings of the 5th International Workshop on Reading Music Systems (WoRMS)*. Ed. by Jorge Calvo-Zaragoza, Alexander Pacha, and Elona Shatri. Milan, Italy, 2023, pp. 7–11. DOI: 10.48550/arXiv.2311.04091. URL: https://sites.google.com/view/worms2023/proceedings.

[15] Antonio Ríos-Vila, Jorge Calvo-Zaragoza, and David Rizo. "Evaluating Simultaneous Recognition and Encoding for Optical Music Recognition". In: *Proceedings of the 7th International Conference on Digital Libraries for Musicology*. DLfM '20. Montréal, QC, Canada: Association for Computing Machinery, 2020, pp. 10–17. ISBN: 9781450387606. DOI: 10.1145/3424911.3425512. URL: https://doi.org/10.1145/3424911.3425512.

[16] David Rizo, Jorge Calvo-Zaragoza, and José M. Iñesta. "MuRET: a music recognition, encoding, and transcription tool". In: *Proceedings of the 5th International Conference on Digital Libraries for Musicology*. DLfM '18. Paris, France: Association for Computing Machinery, 2018, pp. 52–56. ISBN: 9781450365222. DOI: 10.1145/3273024.3273029. URL: https://doi.org/10.1145/3273024.3273029.

[17] Fu-Hai Frank Wu and Jyh-Shing Roger Jang. "An Architecture for Optical Music Recognition of Numbered Music Notation". In: *Proceedings of International Conference on Internet Multimedia Computing and Service*. ICIMCS '14. Xiamen, China: Association for Computing Machinery, 2014, pp. 241–245. ISBN: 9781450328104. DOI: 10.1145/2632856.2632930. URL: https://doi.org/10.1145/2632856.2632930.

OMR on Early Music Sources at the Bavarian State Library with MuRET – Prototyping, Automating, Scaling

Janosch Umbreit, M.A.
Bavarian State Library, Music Department
Munich, Germany
janosch.umbreit@bsb-muenchen.de

Silvana Schumann, M.A.
Bavarian State Library, Music Department
Munich, Germany
silvana.schumann@bsb-muenchen.de

Abstract—**We describe the application of the OMR software MuRET to a corpus of mensural printed and handwritten music for the purpose of making a collection of music sources held at the Bavarian State Library searchable via the application *musiconn.scoresearch*. We focus on our workflow with MuRET, describe the improvements and roadblocks we have noticed, and discuss coming challenges concerning automation and batch processing of such a heterogeneous dataset.**

Index Terms—**Optical Music Recognition, Mensural Notation, MuRET**

I. INTRODUCTION

The digital availability of ever more musical sources from libraries and archives creates a growing desire for efficient and granular search options, not only of metadata, but also of the musical texts themselves. This desire for the searchability of music texts presents institutions with a twofold challenge: on the one hand, search entry points must be created that allow search queries to be formulated both expressively and intuitively, and on the other hand, the music sources must not only be digitized, but their content must be recognized as accurately as possible and made machine-readable. The development of an automated OMR workflow based on MuRET – an OMR application developed by Prof. David Rizo (University of Alicante) [1] – for the search portal musiconn.scoresearch showcases the particular difficulties of this task when working with a corpus of early modern mensural music. The aim of the collaboration – to create a robust OMR model that can reliably process sources of varying quality from the 16th and 17th centuries and to automate the recognition of musical characters over well over 1,000 sources – illustrates both the strides OMR is making and the challenges heterogeneous and idiosyncratic sources such as mensural choir- and partbooks pose for the OMR itself, as well as its automation.

II. AIM AND SCOPE OF THE PROJECT

musiconn.scoresearch[1] is an interface that facilitates the search for digitized music notation based on optical music recognition (OMR) and a search mask that allows users to search for melodies via a digital keyboard. It is currently being developed by the Specialized Information Service Musicology (FID Musikwissenschaft – musiconn) at the Bavarian State Library [2]. At present, the corpus of searchable music consists of roughly 159,000 scanned pages of works by selected composers of the 18th and 19th centuries, such as Beethoven, Händel, and Schubert, as well as the first two series of the "Denkmäler Deutscher Tonkunst" and older music prints from the Schott archive. For the next project phase (2024–2026), one of the goals for musiconn.scoresearch is to expand the searchable repertoire by including older music sources from the 16th and 17th centuries, including over 1,800 printed part books and 75 choir books in manuscript form held at the Bavarian State Library.

In order to achieve high quality OMR results for this corpus of mostly white mensural notation, which differs substantially from the more recent sources that are already searchable, Prof. David Rizo and musiconn are cooperating to further extend the OMR software MuRET (Music Recognition, Encoding, and Transcription). The goal of this collaboration is to train a robust model for OMR on a representative selection of sources from the corpus, to develop needed additions for MuRET, such as new mensural meters and extended ligatures, and finally to implement a full offline pipeline that includes OMR, layout recognition, and indexing of the resulting MEI files. This approach presents a change from the workflow for scoresearch thus far – which relied on the SmartScore[2] application and MusicXML output – due to the special requirements of the early modern repertoire.

MuRET itself was initially developed for the HISPAMUS project, which has aims that are very similar to those of musiconn.scoresearch. However, unlike HISPAMUS, our focus lies less with providing material for editors and performers and more with devising a workflow for indexation that can be scaled and automated sufficiently to be applied to the collection held at the Bavarian State Library [3]. The F-TEMPO project, too, which is concerned with OMR, indexing, and

[1] https://scoresearch.musiconn.de/ScoreSearch/about?lang=en

[2] https://www.musitek.de/produkte/smartscore.php

query matching as well, acts as a point of reference. The issues with OMR and indexing the project's authors describe are very applicable to the ones we are encountering [4]. However, while F-TEMPO aims at exposing a broad selection of musical sources through its API, scoresearch is mainly concerned with making the holdings of the Bavarian State Library accessible within the framework of the existing research options offered by musiconn.

III. Workflow with MuRET

The training of the OMR model with MuRET follows an iterative approach and comprises three discrete phases: At the beginning, the first page of each source in the training set is processed and then corrected manually. In the second step, the first 20 pages of each source are processed and corrected. Finally, 10 % of each remaining source are processed and corrected.

OMR with MuRET is divided into three individual tasks that are performed for every scan: Document analysis, transcription, and semantic interpretation. Document analysis starts with identifying the individual staves on a given page. Once the staves are identified, all symbols on each staff are transcribed diplomatically. Finally, this diplomatic transcription is interpreted semantically. During both the document analysis and the transcription step, the supervisor can intervene and correct MuRET's classification by redrawing bounding boxes, identifying symbols the model omitted, or changing the values of transcribed symbols.

Over the past months, the project has progressed to the third phase, focusing first on document analysis and now on transcription. In the coming months, more focus will be placed on the semantic interpretation and, based on this work, the connections between individual parts that make up pieces and the assembly of a more automated pipeline from scan to MEI file.

IV. Observations and Improvements

Concerning the accuracy of MuRET, we can report that the training up until now has yielded significantly improved results: The staff detection is very robust and can correctly process even crooked scans. The accuracy of the transcription has steadily improved as well, with features such as dots, rests of varying length, and flats being recognized much better than at the outset, while pitch recognition has been comparatively solid since the beginning. Duration, on the other hand, can sometimes still pose an issue, for example, when it comes to differentiating between fusae (eighths) and semifusae (sixteenths). In weighing the performance of MuRET, especially with regards to the ultimate goals of musiconn.scoresearch, we should keep in mind that in OMR, errors in classifying symbols such as the clef or the meter are disproportionately graver than errors of e. g. pitch, as they propagate throughout whole sections of music. Thankfully, MuRET has yielded consistently good results in recognizing both clefs and meters.

Some features MuRET has to address when transcribing the musiconn.scoresearch corpus are unique to the music publishing industry of the early modern period. For example, sharp accidentals are frequently used instead of naturals. Furthermore, accidentals often do not appear exactly on the same line or space as the note to which they refer. The architecture of MuRET facilitates dealing with these issues, as they force us to differentiate between the correct recognition and diplomatic transcription of the symbol on the one hand, and its semantic interpretation, which may yield a more standardized representation, on the other.

V. Further Development

The application of a tool like MuRET to a task such as the expansion of musiconn.scoresearch highlights the special requirements and adjustments that are necessary for OMR on mensural notation: On a fundamental level, mensural notation requires a somewhat different set of glyphs, such as different meters and ligatures. The latter in particular can pose problems, as they can be synthesized in a great number of combinations and require specialized rules for their semantic interpretation. The enhancement of the glyph repertoire and the further development of ligature processing are some of the most imminent tasks in our continuing work with MuRET.

Further considerations include the next steps to be taken in recognizing parts and whole pieces. Early modern polyphonic music differs from common modern western layouts in that voices are typically either split up in partbooks, or arranged in choir book format. Of the two, choir books are simpler to assemble from their parts: As voices are arranged in the corners of a given spread of two pages, they cover the same musical duration and the difficulty lies mainly in distinguishing between the individual parts on a given page. In the case of partbooks, the different voices belonging to a polyphonic composition might be far apart in terms of scanned pages and their alignment requires a sophisticated understanding of the given musical structures, as well as additional information derived from non-musical texts or other visual cues, such as initials. At present, boundaries between individual pieces can only be designated manually. Part names, too, can only be assigned by users, although this task can be sped up significantly wherever parts repeat in a predictable pattern, which is often the case for this repertoire. However, in order to automate the OMR and the subsequent export to MEI, more work needs to be done here. Finding a balance between automatic segmentation & labeling and human intervention to ensure valid and meaningful encoding will be a major task for the further development of the project.

The collaboration between the Bavarian State Library and Prof. Rizo and his team is still in the early stages. However, we can already report very promising results that showcase the great reliability of MuRET in analyzing page structure and recognizing musical characters. We are confident that our ongoing work on the improvement of the model's performance, as well as the layout recognition, will further advance the automatic processing of early modern music sources, a task that has proven just as intriguing as it is complex.

REFERENCES

[1] J. M. Iñesta, D. Rizo, and J. Calvo-Zaragoza. "MuRET as a Software for the Transcription of Historical Archives." In Proceedings of the 2ⁿᵈ International Workshop on Reading Music Systems (WoRMS). Delft: 2019, pp. 12–15.

[2] S. P. Achankunju. "Music search engine from noisy OMR data." in 1ˢᵗ International Workshop on Reading Music Systems. Paris: 2018, pp. 23–24.

[3] J. M. Iñesta, P. J. Ponce de León, D. Rizo, J. Oncina, L. Micó, J. R. Rico-Juan, C. Pérez-Sancho, and A. Pertusa. "HISPAMUS: Handwritten Spanish Music Heritage Preservation by Automatic Transcription." In Proceedings of the 1ˢᵗ International Workshop on Reading Music Systems. Paris: 2018, pp. 17–18.

[4] T. Crawford, D. Lewis, and A. Porter. 2023. "Exploring Early Vocal Music and Its Lute Arrangements: Using F-TEMPO as a Musicological Tool. In Proceedings of the 10ᵗʰ International Conference on Digital Libraries for Musicology (DLfM '23). Association for Computing Machinery, New York: 2023, pp. 77–81.

OMMR4all revisited – a Semiautomatic Online Editor for Medieval Music Notations

1ˢᵗ Alexander Hartelt
Dep. for Artificial Intelligence and Knowledge Systems)
University of Wuerzburg
D-97074 Germany
alexander.hartelt@uni-wuerzburg.de

2ⁿᵈ Frank Puppe
Dep. for Artificial Intelligence and Knowledge Systems
University of Wuerzburg
D-97074 Germany
frank.puppe@uni-wuerzburg.de

Abstract—**Five years ago, OMMR4all has been presented at the WoRMS 2019 workshop. In this paper we report on advances and new evaluations of the editor for medieval music notations. The main contribution is the handling of lyrics (text) within the chants to be transcribed. The 2019 version of OMMR4all recognized staff lines, layout and symbols but required the syllables of the text to be entered in advance. In addition, the current version recognizes the start and end of a chant within a page, transcribes the text with an automatic alignment to the most similar chant from a chant database and assigns the symbols to the syllables of the text. Evaluations show a speed-up of the transcription process of medieval square notations by a factor of 6 to 9 compared to a factor of 1.2 to 1.3 in the 2019 version with a comfortable manual editor used as baseline.**

Index Terms—**optical music recognition, web app, medieval manuscripts, square notation, user interface, pipeline**

I. INTRODUCTION

A major challenge for optical music recognition is the alignment of music notation and lyrics which is essential for cultural heritage, where vocal music is prevalent [1]. This is particularly challenging for medieval chants, where the lyrics is written in a style which is very demanding for HTR-systems even with finetuning (s. Fig. 1). While there are generally few systems dealing with medieval music notations (see chapter 2), the text recognition and alignment issues are largely circumvented by assuming, that the text with syllables, which are the units for alignment, is given as input like e.g. in OMMR4all in a WoRMS paper from 2019 [2].

In this paper we report major advances in the OMMR4all system during the last five years and in particular, how we solve the lyrics transcription and alignment problem. The main idea is to use background knowledge from a large corpus of transcribed chants from e.g. the Cantus database[1] or our Corpus Monodicum project[2] extracting a chant text repository and a word dictionary. The chances are high, that a new chant to be transcribed uses the same or very similar lyrics than other chants with different music. Together with general advances in the deep learning technology for OMR we achieve a significant speed-up in the transcription process of medieval chants from a factor of 1.3 reported in OMMR4all-2019 compared to manual transcription used as baseline to a factor of 8.9 in the best

constellation in the new OMMR4all-2024 system[3].

II. RELATED WORK

The literature presents a lack of workflows that display an entire OMR pipeline for historical data. While recent years have witnessed a surge in research on end-to-end OMR workflows [3], [4], these studies predominantly focus on clean and modern datasets. Such works typically rely on sequence-to-sequence architectures trained with the CTC loss for recognition tasks. Moreover, they often limit their scope to symbol recognition, frequently using a musical region as input, thereby necessitating a preceding segmentation step. The combined assignment of symbols, text and syllables is rarely addressed in these studies.

In [5] Fujinaga et. al presented an OMR workflow for processing and encoding Medieval music manuscripts used in the SIMSSA [6] project. It uses a combination of convolutional neural networks, interactive classifiers, and web-based tools to process and encode the music in the MEI format. The workflow includes steps for layout analysis (Pixel.js), symbol classification by using an interactive classifier, neume processing, text recognition (using Calamari OCR engine) and MEI generation. The layout analysis divides the image into distinct components such as staff lines, text, and musical symbols. The symbols are then classified using a web-based k-nearest neighbor classifier. Afterwards the pitch is determined using the output of a CNN. In addition to the text layer, the transcription of the text layers serves as input for the text recognition. Calamari is then used to calculate text alignment information (position of the text/chars/syllables on the page). The workflow is managed by management software called Rodan[4], which allows for configuration of the entire process. The resulting encoded music is afterwards merged with metadata from the Cantus Manuscript Database and displayed on a website. The project has encoded already over 8000 chants from Medieval manuscripts.

In summary, both OMMR4all and SIMSSA share a common objective: to expedite the transcription of historical documents

---

[1] https://cantus.uwaterloo.ca/
[2] https://corpus-monodicum.de/

[3] A detailed description of the pipeline used in the OMMR4all-2024 system is given in [7]
[4] http://ddmal.music.mcgill.ca/e2e-omr-documentation/overview/rodan.html
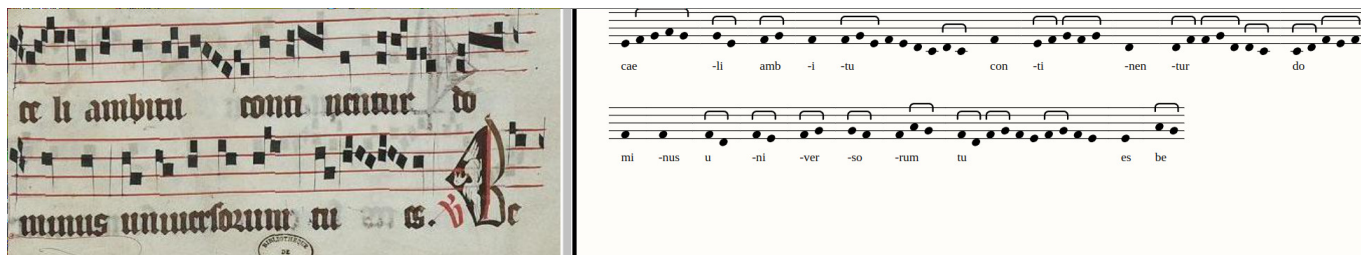
Fig. 1. Snippet of the split-view component of OMMR4all-2024. Displayed is the transcription result next to the original image. Different overlays (e.g. display of Symbols, Text, Syllabels, Layout, etc.) can be applied to the original image. Many letters of the lyrics in the original image look alike, posing a great challenge for automatic transcription and syllable separation.

by providing tools and interfaces that support editors. While OMMR4all offers a comprehensive interface encompassing all functionalities (pipeline, training, correction), the SIMSSA project leverages numerous separate projects. Moreover, the projects employ various ground truth (GT) types. SIMSSA necessitates pixel-precise labeling of the original documents, whereas OMMR4all utilizes polygon-based GT for training staff, layout, and symbol recognition, significantly accelerating the creation of GT. Beyond additional differences in algorithm functionalities and post-processing, OMMR4all-2024 uniquely supports chant segmentation, both manually and automatically.

III. Advances in OMMR4all

In OMMR4all-2019 the problem of recognition of the handwritten chant texts and their assignment to the note symbols was circumvented by entering the text manually including syllables and assigning each syllable consecutively to a neume component. This approach requires a perfect aggregation of symbols to neume components, which needs a manual correction step before the aggregation. In OMMR4all-2024 we experimented with different HTR engines (Handwritten Text Recognition) but did not get satisfactory results due to the chant notation style, where among other problems letters like e.g. u, v, n, m, I, r, t, e and c look very similar (compare Fig. 1). However, we had a large corpus of transcribed chants from the Corpus Monodicum project and found that there are many duplicated texts within this chant corpus. Therefore, we used the faulty transcription results of the HTR engine for a new chant and correct some words with a dictionary generated from the corpus and then select the most similar chant from the library, which succeeded in more than 80% of the new chants. The assignment of symbol sequences to syllables uses the corrected transcription results, because usually, the position of a syllable and the position of the corresponding note symbols agree with each other. However, there are some exceptions requiring additional knowledge. This new approach speeds up the efficiency of the transcription drastically. Further changes from OMMR4all-2019 to OMMR4all-2024 include an improved algorithm for recognizing staff lines achieving nearly 100% accuracy, enabling a complete automatic layout recognition for separating symbols and text regions. The recognition of symbols is also improved in OMMR4all-2024: both approaches still use an U-NET [8] architecture with an

encoder and a decoder, but the custom encoder in OMMR4all-2019 has been replaced in OMMRall-2024 by an EfficientNet-b3 [9] architecture. The decoder was custom fine-tuned for this encoder. The clefs at the beginning (and sometimes in the middle) of a line and the duplicated symbols at the end of line (they are identical to the first symbol of the next line, compare Fig. 2) are now recognized very well. Since OMMR4all is trained with a large corpus, a pretraining on the same source as in OMMR4all-2019 is not always needed; the evaluation results of OMMR4all-2024 (see next chapter) use a generic model for square notations. Further on, OMMR4all-2024 tackles the task of document separation: In the sources, a new document (chant) usually starts in the middle of a page and even in the middle of a line and is marked by a prominent drop capital, i.e. a decorated first letter of the new chant. Recognizing these drop capitols with a separate component based initially on a Mask-RCNN and later on a YOLOv8 architecture allows the segmentation of the transcribed symbols and texts in chants, which is a prerequisite for the corpus-based text recognition approach (s. above). Since chants can span about more than one page, the new approach includes a management component for sources containing many pages instead of transcribing individual pages only as in OMMR4all-2019. Finally, the editor support for validating and correcting the transcription results has been improved. Usually, a one-step correction step is sufficient in OMMR4all-2024, because the error rates are low. Since the exact position of note symbols relative to the lines is sometimes difficult to assess, the editor highlights the notes lying on a line resp. between two lines with different colors. An example of the editor is shown in Figure 2 containing two chants with two drop capitals in the main view and a list of consecutive pages from a source on the left. The editor provides different views and split views, where the transcription result can be compared directly to the original scan.

IV. Evaluation results

OMMR4all-2019 published the following evaluation results for transcription of chants in square notation in minutes per page for postcorrection (average of five pages with 267 symbols per page): 0.6 minutes for correcting staff lines, 3.3 minutes for correcting symbols and 2.9 minutes for correcting for correcting the assignment of symbols to syllables, in total
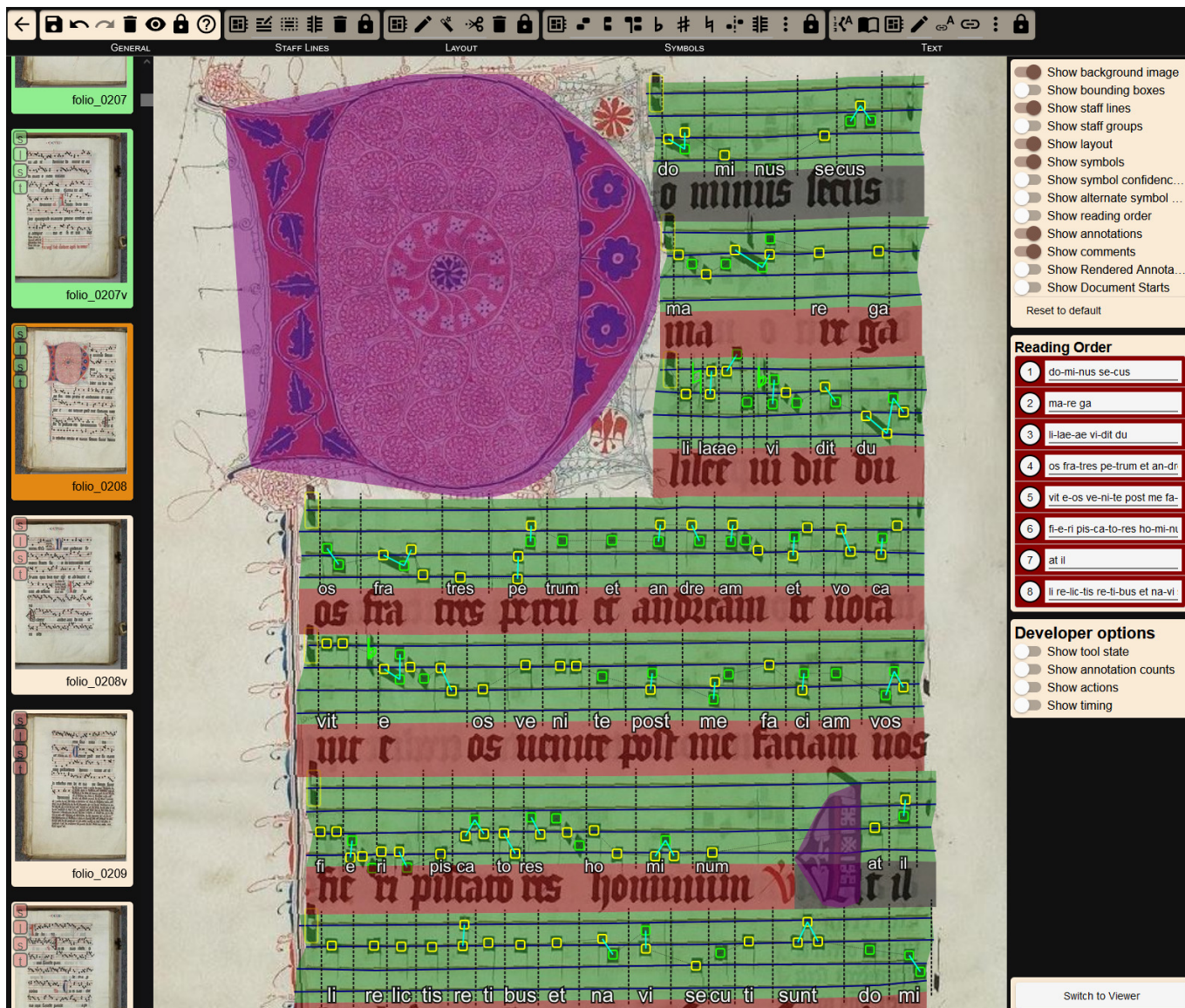
Fig. 2. The images were cropped from the OMMR4all-2024 overlay editor. The purple areas mark drop capitals. Green areas are music regions. Red areas are lyric regions. Gray regions are also lyric regions, but they additionally mark the start of a new chant. Yellow and green squares within the music region mark symbols. The different colors of the symbol indicate whether they lie on a staff line or between two staff lines. The reading order of the symbols is represented by a thin line that connects the symbols to each other. Turquoise lines between symbols mark graphic connections. Vertical lines in the music regions mark note sequences to which a syllable is assigned

6,9 minutes per page. Compared to completely manual entry with the Monodi editor [10], which took 8.5 minutes, a speed-up of a factor of 1.3 was observed. However, transcribing the text of the chant and its separation in syllables was not part of the evaluation, since the text was given in advance for both editors. In OMMR4all-2024 these steps are covered too, and the full postcorrection time including transcription of the text in syllables and determining the begin and end of the chants is evaluated. The speed-up factor compared to manual transcription with the Monodi editor jumped from 1.3 in OMMR4all-2019 to 5.7 with a two-step process and even 8.9 with a one-step process (see table I):

V. DISCUSSION

OMMR4all-2024 excells at staffline recognition, which is nearly errorfree, and symbol recognition including differenti-ating between normal notes, clefs, accidentals including dupli-cated notes at the end of a line and pitch determination relative to the stafflines with combined error rates of 2,2% without finetuning and 1,5% with finetuning. The main error sources are recognizing graphical connections and assignment of these note complexes to syllables of the text. Using a large chant corpus and a corresponding word dictionary for automatic correction of the erroneous transcription of the handwritten texts was a major breakthrough, but does not deliver perfect results, since there are often minor variations in the chant texts,

TABLE I

Time in minutes for postcorrection per page with ca. 200 symbols and chant text (computed as average from 20 pages) from three different persons, where person3 corrected the output of OMMR4all-2024 in one step, whereas person1 and person2 used two steps for correction of symbols first and then for the chant text including assignment of syllables to symbols. For comparison, the manual transcription time with the Monodi editor is stated for person2 (computed as average from 5 pages).

	OMMR4all-2024			Monodi
Person	Person1	Person2	Person3	Person2
Symbol Level	1.1	1.0	-	11.1
Text Level	2.0	2.1	-	6.7
Total	3.1	3.1	2.0	17.8

which currently must be corrected manually. In addition, the position of the syllables cannot be determined precisely in cases of bad handwritten recognition results and corresponds usually, but not always with the start of its matching note complex, which requires manual correction steps as well.

VI. Conclusion and future work

OMMR4all was developed in the context of the Corpus Monodicum project, which consists of three parts: "editions", "transcriptions", and "graduale synopticum". While editions are transcribed mainly manually with the Monodi editor (to a large degree before OMMR4all-2024 was available), the transcriptions and the graduale synopticum (a retro-digitalisation project based on analog transcriptions from http://gregorianik.uni-regensburg.de/gr/) were made available with OMMR4all and manual postcorrection. Recently, we completed the transcription of the Graduale Synopticum[5], involving the transcription of over 5,500 chants using the pipeline. Currently, we are engaged in the transcription of additional manuscripts employing square notation, such as the "Köln, Dombibl. 1001b"[6] manuscript. Furthermore, we are actively exploring the applicability of the pipeline to other similar notations, particularly "Hufnagel" notation (e.g., Geesebook[7])

References

[1] Calvo-Zaragoza, J., Martinez-Sevilla, J.C., Penarrubia, C., Rios-Vila, A. (2023). Optical Music Recognition: Recent Advances, Current Challenges, and Future Directions. In: Coustaty, M., Fornés, A. (eds.) Document Analysis and Recognition – ICDAR 2023 Workshops. ICDAR 2023. Lecture Notes in Computer Science, vol 14193. Springer, Cham. https://doi.org/10.1007/978-3-031-41498-5_7

[2] Wick, Christoph and Puppe, Frank, OMMR4all — a Semiautomatic Online Editor for Medieval Music Notations, In: 2nd International Workshop on Reading Music Systems, 2019, pp 31–34

[3] Ríos-Vila, A., Rizo, D., Iñesta, J.M. et al. End-to-end optical music recognition for pianoform sheet music. IJDAR 26, pp. 347–362 (2023). https://doi.org/10.1007/s10032-023-00432-z

[4] Ríos-Vila, A., Rizo, D., Calvo-Zaragoza, J. (2021). Complete Optical Music Recognition via Agnostic Transcription and Machine Translation. In: Lladós, J., Lopresti, D., Uchida, S. (eds) Document Analysis and Recognition – ICDAR 2021. ICDAR 2021. https://doi.org/10.1007/978-3-030-86334-0_43

[5] Fujinaga, Ichiro and Vigliensoni, Gabriel, "Optical Music Recognition Workflow for Medieval Music Manuscripts,"5th International Workshop on Music Reading Systems, 2023.

[6] Fujinaga, Ichiro and Hankinson, Andrew and Cumming, Julie E., Introduction to SIMSSA (Single Interface for Music Score Searching and Analysis, In: Proceedings of the 1st International Workshop on Digital Libraries for Musicology 2014, pp. 1–3

[7] Hartelt, Alexander and Eipert, Tim and Puppe, Frank, Optical Medieval Music Recognition—A Complete Pipeline for Historic Chants, In: Applied Sciences, 2024

[8] Olaf Ronneberger and Philipp Fischer and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241

[9] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, pp. 6105–6114

[10] Eipert, T., Haug, A., Herrmann, F., Puppe, F., and Wick, C., Editor Support for Digital Editions of Medieval Monophonic Music, In: 2nd International Workshop on Reading Music Systems (WoRMS), 2019

[5] http://gregorianik.uni-regensburg.de/gr/

[6] https://digital.dombibliothek-koeln.de/hs/handschriften/content/zoom/312082

[7] http://geesebook.ab-c.nl/

Crafting Handwritten Notations: Towards Sheet Music Generation

1ˢᵗ Nivesara Tirupati, 2ⁿᵈ Elona Shatri, 3ʳᵈ György Fazekas

Centre for Digital Music, Queen Mary University of London

London, UK

Emails: n.tirupati@se23.qmul.ac.uk, e.shatri@qmul.ac.uk, george.fazekas@qmul.ac.uk

Abstract—**Handwritten musical notation represents a significant part of the world's cultural heritage, yet its complex and unstructured nature presents challenges for digitisation through Optical Music Recognition (OMR). While existing OMR systems perform well with printed scores, they struggle with handwritten music due to inconsistencies in writing styles and the quality of scanned images. This paper addresses these challenges by applying Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) to generate high-quality, synthetic handwritten music sheets. The generated sheets can then be used to improve OMR handwritten datasets with more style variability. Experimental results demonstrate that ESRGAN outperforms conventional models, producing detailed and high-fidelity synthetic music sheets. This research offers a practical approach to improving the preservation and digitisation of handwritten music, benefiting musicologists, educators, and archivists.**

Index Terms—**Handwritten Musical Notation, OMR, Enhanced Super-Resolution Generative Adversarial Networks (ESRGANs), Synthetic Handwriting**

I. INTRODUCTION

Handwritten music manuscripts form a critical part of the world's musical heritage, preserving unique details about composition, performance styles, and historical notations. Digitising these manuscripts is essential for future generations and for advancing musicological research, education, and archival work. However, transforming handwritten music into digital formats through Optical Music Recognition (OMR) is a highly complex task due to variations in handwriting styles, inconsistencies in notation, and degradation of physical manuscripts. While OMR systems have achieved considerable success with printed scores, where symbols and layouts are standardised, they falter when applied to handwritten music. The natural variability in handwriting, along with issues such as blurred ink, faded writing, and physical wear, leads to significant recognition errors. These challenges create a pressing need for more sophisticated methods to accurately process and digitise handwritten music.

Recent advances in machine learning, particularly with Generative Adversarial Networks (GANs), have opened new possibilities for generating realistic synthetic data. Enhanced Super-Resolution GANs (ESRGANs) [21], known for their ability to produce detailed high-resolution images, offer a promising solution for overcoming the limitations of traditional OMR systems. By generating synthetic handwritten music sheets that replicate the complexity and nuances of real manuscripts, ESRGAN can be leveraged to train OMR systems more effectively.

This paper proposes a novel approach to improving OMR for handwritten music by using ESRGAN-generated synthetic data. Our method enhances the quality of synthetic music sheets, providing OMR systems with training data that closely resembles real-world manuscripts. The results demonstrate improved accuracy in recognizing handwritten music, offering a new pathway for preserving and digitizing the world's musical heritage.

II. RELATED WORK

The preservation of cultural heritage through handwritten music is crucial, as it represents a legacy passed down through generations. The digitisation of such notations is essential for ensuring their longevity and accessibility to future generations. As noted in [9], the introduction of the MUSCIMA++ dataset marked a significant advancement in the field of OMR. This dataset offers a comprehensive collection of notated manuscript music, encompassing a broad array of symbols, from basic notes and rests to complex articulations and dynamic markings. MUSCIMA++ provides researchers with essential tools for developing and evaluating OMR systems, serving as the ground truth for addressing key challenges in recognising music symbols, such as determining their occurrence and location.

Despite the availability of this dataset, OMR systems continue to face substantial challenges due to the variability in handwriting styles and the complexity of music notation. Even with the comprehensive MUSCIMA++ dataset, the accuracy of current OMR technologies remains a significant issue, as outlined in [6]. This highlights the need for more advanced techniques to improve the digitisation process.

GANs have had a transformative impact on image processing and offer potential solutions to some of the persistent challenges in OMR. Initially proposed in [10], GANs are capable of generating images with highly realistic features, which can enhance the optical recognition of digitised handwritten music. The authors in [2] introduced StyleGAN, a style-based generator architecture that offers enhanced control over generated attributes, improving the quality of image generation. Previous research has demonstrated the effectiveness of deep learning techniques [7], [8], particularly GANs, in enhancing image resolution and improving recognition accuracy. To address

the specific challenges of handwritten music digitisation, [21] developed the ESRGAN, which focuses on improving image resolution while preserving fine details. In this study, ESRGAN is applied to enhance the clarity of handwritten music sheets, building upon previous work to address the accuracy challenges in OMR.

III. METHODOLOGY

Our methodology is divided into several key stages: dataset acquisition and preparation, image preprocessing, model architecture design, and training. Each stage is designed to ensure that the ESRGAN model generates high-fidelity handwritten music sheets that replicate the complexity and variability found in real-world music manuscripts.

The following sections will outline each stage of the methodology in detail, starting with the preparation of the MUSCIMA++ dataset, followed by the image preprocessing steps, the architecture of the ESRGAN model, and the training procedure designed to ensure optimal performance.

A. Dataset Acquisition and Image Preprocessing

For training the ESRGAN model, we utilised the MUSCIMA++ dataset [9], a collection of handwritten music. This dataset includes a variety of music notation styles, from simple notes to complex symbols from different writers, which allows the model to generalise effectively across different handwriting styles. Its diversity is crucial for training ESRGAN to accurately replicate real-world handwritten music.
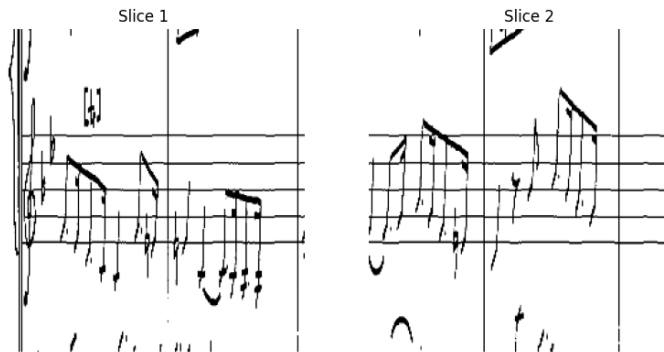

Fig. 1: Segmentation of handwritten music sheets into smaller patches (256x256 pixels) for training.

1) Image Preprocessing and Cropping: To standardise the input data for ESRGAN, all music sheets were resized to 256x256 pixels and converted to greyscale. This ensures a uniform input format while preserving sufficient detail for effective model training [21].

a) Contrast Enhancement: Handwritten music often contains fine details [3], [4] that can be lost during digitisation, such as thin lines or varying ink densities. To preserve these details, we applied contrast enhancement to ensure the ESRGAN model receives clear, well-defined input images [6]. This step is especially important for distinguishing staff lines from background noise in the scanned images [1].

b) Gaussian Noise: To simulate real-world imperfections, such as smudged ink and paper texture variations, we introduced Gaussian noise into the images. This noise makes the dataset more robust, ensuring that ESRGAN can handle the kinds of distortions often encountered in practical applications [10], [11]. By managing pixel values to remain within a valid range [0,1], we prevent the model from over-darkening or brightening the images, thus maintaining overall quality.

c) Image Cropping and Patching: To ensure the model focuses on capturing fine details in handwritten music notation, each music sheet image was divided into 256x256 pixel patches. This approach enables the ESRGAN model to process smaller, more manageable sections, improving its ability to learn subtle and complex features, such as variations in musical symbols [6]. Additionally, patching reduces computational overhead and standardises the input format across the dataset, facilitating efficient and consistent model training, as depicted in Fig. 1.

B. ESRGAN Architecture

The ESRGAN architecture is composed of two primary components: the generator and the discriminator [21]. These components work together in an adversarial framework to enhance the quality and resolution of the synthetic handwritten music sheets. The generator is responsible for transforming input image patches into high-resolution outputs. It achieves this through several residual blocks, which allow for the preservation of important features while avoiding issues like vanishing gradients [15]. Each residual block contains convolutional layers, batch normalisation, and activation functions to progressively refine image details. By focusing on the fine details of music symbols, the generator ensures that high-resolution synthetic images closely resemble real handwritten sheets. The final output is activated with a Tanh function, ensuring pixel values remain within a usable range for image synthesis [11].

The discriminator acts as a binary classifier, distinguishing between real and generated images [10]. It consists of a series of convolutional layers followed by Leaky ReLU activation functions and batch normalisation [24]. These layers downsample the input, progressively focusing on distinguishing fine details. The final layer uses a Sigmoid activation to output a probability score, guiding the generator in producing more realistic images over successive iterations [23].

By training the generator and discriminator together in an adversarial loop, ESRGAN produces high-resolution synthetic handwritten music sheets that effectively capture the nuances of real-world notation. This architecture can become pivotal to improving the performance of OMR systems in handling the variability of handwritten music scores.

The ESRGAN model was trained using an adversarial training loop, where the generator and discriminator were updated alternately. The generator aimed to produce realistic high-resolution images, while the discriminator worked to distinguish these from real images [10]. We employed the Adam optimiser [23] to adjust learning rates and stabilise

convergence. The loss function consisted of adversarial loss, which encourages realism in generated images, and L1 loss, ensuring pixel-wise accuracy between generated and target images [11]. This process was iteratively refined over 80 epochs, allowing the ESRGAN to effectively model the variability in handwritten music notation.

In addition to the standard adversarial and L1 losses, the ESRGAN model incorporates a perceptual loss to enhance image quality. This loss is calculated by comparing high-level features between the generated and real images, as extracted by a pre-trained network. Unlike pixel-based losses, perceptual loss focuses on the overall structure and content of the image, ensuring that the generated music sheets are not only visually similar but also maintain the intricate relationships between musical symbols and notation features. This helps in preserving the subtle variations in handwriting styles.

C. Post-processing and Image Reconstruction

We initially generated these images at a resolution of 256x512 pixels. The images were then upscaled to match the dimensions of the original sheets for comparison purposes. We acknowledge that this resizing could introduce slight downscaling artefacts, which may have affected the visual quality in Figure 6. However, given resource constraints, generating directly at higher resolutions was challenging. To minimise visible seams between patches, a 4x4 pixel overlap was used, and pixel values were averaged across overlapping regions [21]. This ensures smooth transitions between patches, maintaining the fidelity and integrity of the overall image. Once all patches are recombined, the image is resized to match the original handwritten music sheet dimensions.

IV. Experimental Setup

The ESRGAN model was trained iteratively, with the generator producing high-resolution handwritten music sheet images and the discriminator distinguishing between real and synthetic images. This adversarial process continuously refines the output quality, ensuring more realistic and accurate handwritten music representations.

The generator and discriminator networks were initialised and optimised using the Adam optimiser [23], with a learning rate of 0.00001. The adaptive nature of the Adam optimiser allowed for stable convergence and fine-tuning of model parameters, effectively managing the complex loss landscapes typical in GAN training [24].

The training process followed a standard adversarial loop [10], where the generator produced batches of synthetic images, which the discriminator then classified alongside real images [21]. The generator's loss function combined three key components:

- Adversarial Loss: Encouraged realistic image generation [24].
- L1 Loss: Ensured pixel-level accuracy between generated and real images [25].

- Perceptual Loss: Maintained high-level feature alignment between generated and real images, ensuring the structural integrity of the music notation [19].

The discriminator was trained with a binary cross-entropy loss to improve its ability to distinguish real from synthetic images, thereby refining the generator's outputs over successive iterations.

Fig. 2: Handwritten music sheet generated by the ESRGAN model after 80 training epochs

The model underwent 80 training epochs, during which the generator incrementally refined its outputs to better resemble original handwritten music sheets [21]. Intermediate outputs were inspected visually after each epoch to evaluate how well the model captured musical notation details, guiding further parameter adjustments as needed [18]. An example of these outputs is shown in Figures 2 and 7, illustrating the model's ability to synthesise high-fidelity handwritten music sheets.

A. Evaluation Metrics

To assess the performance of the ESRGAN model and compare it with Pix2Pix, we employed two key sets of evaluation metrics: one for the overall quality of the generated images and another for edge detection. These metrics were selected to provide a comprehensive evaluation of the models' ability to replicate handwritten music sheets with both structural and perceptual accuracy.

a) Fréchet Inception Distance (FID) and Inception Score (IS): FID is a widely-used metric that measures the distance between the distribution of real and generated images. Lower FID scores indicate that the generated images are more similar to real images, making it an ideal metric for assessing the realism of generated handwritten music sheets. The IS, on the other hand, evaluates the quality and diversity of the generated images. Higher IS values reflect better image quality and variety, ensuring that the model generates not only realistic images but also diverse representations of handwritten music.

b) Edge Detection (MSE): Edge detection is crucial for preserving the structural details of handwritten music, such as staff lines and note stems. To evaluate this, we used the MSE between the detected edges in the generated images and the ground truth images. Lower MSE values indicate that the model is better at replicating the fine details of the handwritten music sheets. This metric was chosen to assess how well each model retains the critical edge structures necessary for accurate digitisation.

These evaluation metrics were chosen to provide a balanced and thorough assessment of the models' capabilities in terms of both visual realism and structural accuracy..

V. Results and Discussion

After 80 training epochs, the ESRGAN model demonstrated stable performance improvements, with both generator loss

(GLOSS) and discriminator loss (DLOSS) stabilising over time, as shown in Fig. 3. Initially, GLOSS dropped significantly as the generator refined its ability to produce realistic images. Over time, fluctuations in both losses diminished, indicating that the adversarial training loop reached equilibrium, a common characteristic in GAN models [11].
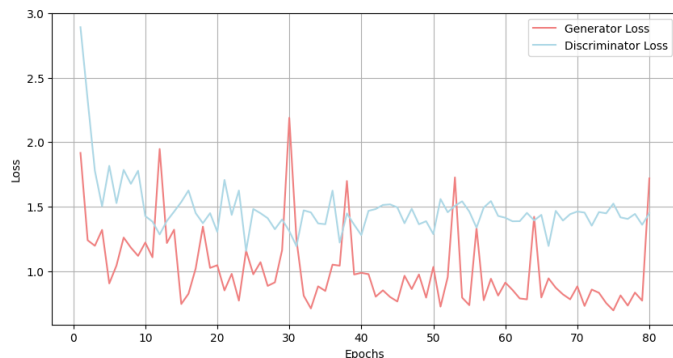

Fig. 3: Graph showing the generator loss (GLoss) and discriminator loss (DLoss) over 80 training epochs for the ESRGAN model.

The t-SNE clustering plot (Fig. 4) highlights how closely the ESRGAN-generated images resemble the original handwritten music sheets, with the generated images clustering tightly with the real ones. This close overlap suggests that ESRGAN successfully captures musical symbol features. [9].

TABLE I: FID and IS Scores Comparison

Model	FID Score	IS Score
ESRGAN	29.47	2.08
Pix2Pix	50.14	2.08

In comparison to Pix2Pix [14], ESRGAN consistently outperformed it in edge detection and image quality, as demonstrated by the lower MSE values (Table II) and Fig. 5. The
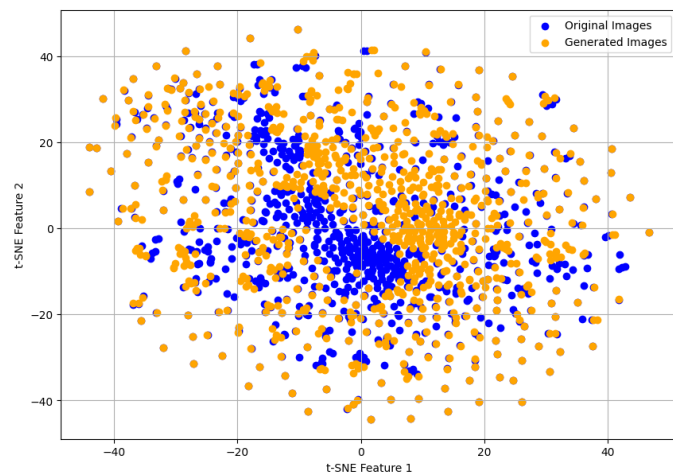
TABLE II: MSE Comparison for Edge Detection between ESRGAN and Pix2Pix

Model	MSE (Edge Detection)
ESRGAN	0.1081
Pix2Pix	0.3781

clearer and sharper edges produced by ESRGAN underscore its ability to retain fine structural details crucial for OMR tasks. Furthermore, ESRGAN achieved a significantly lower FID score (29.47) than Pix2Pix (50.14), indicating its superior ability to generate images that closely resemble real handwritten music sheets (Table I).
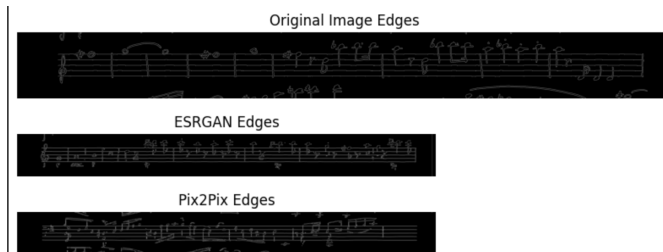

Fig. 5: Edge detection results comparing ESRGAN-generated handwritten music sheets with the ground truth.

VI. CONCLUSIONS

This study demonstrates that ESRGAN is highly effective in synthesising high-fidelity handwritten music sheets, outperforming Pix2Pix in key areas such as edge detection, FID score, and overall image quality. By preserving finer details like musical notations and staff lines, ESRGAN is better suited for tasks requiring precise digitisation of handwritten music sheets.

The lower FID score and more accurate edge detection metrics confirm ESRGAN's ability to generate images that are not only visually similar to real music sheets but also capture critical features necessary for OMR systems. The results suggest that ESRGAN is an ideal candidate for advancing music sheet digitisation and preservation.

Moving forward, integrating ESRGAN with multi-modal datasets, including audio data, could further enhance its applications in music recognition and cultural preservation. Additionally, expanding the diversity of training datasets to include a wider range of notations and historical periods would further improve the model's generalisation, making it applicable to a broader array of musical traditions. The success of ESRGAN in this domain opens the door for future innovations in AI-driven cultural heritage preservation.

ACKNOWLEDGMENTS

Fig. 4: t-SNE clustering plot showing the distribution of real and ESRGAN-generated handwritten music sheets in a reduced feature space.

REFERENCES

[1] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2223–2232.

[2] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4401–4410.

[3] E. Shatri and G. Fazekas, "DoReMi: First glance at a universal OMR dataset," *arXiv preprint*, arXiv:2107.07786, Jul. 2021.

[4] E. Shatri and G. Fazekas, "Knowledge Discovery in Optical Music Recognition: Enhancing Information Retrieval with Instance Segmentation," in *Proc. Int. Conf. Knowledge Discovery and Information Retrieval (KDIR)*, 2024.

[5] A. Brock, J. Donahue, and K. Simonyan, "Large-scale GAN training for high-fidelity natural image synthesis," *arXiv preprint*, arXiv:1809.11096, 2018.

[6] E. Shatri and G. Fazekas, "Optical music recognition: State of the art and major challenges," *arXiv preprint*, arXiv:2006.07885, 2020.

[7] E. Shatri, K. Palavala, and G. Fazekas, "Synthesising Handwritten Music with GANs: A Comprehensive Evaluation of CycleWGAN, ProGAN, and DCGAN," to appear in *2nd Workshop on AI Music Generation (AIMG 2024), IEEE Big Data*, Washington D.C., 2024.

[8] P. Hande, E. Shatri, B. Timms, and G. Fazekas, "Towards Artificially Generated Handwritten Sheet Music Datasets," in *Proc. 5th Int. Workshop on Reading Music Systems*, 2023, p. 25.

[9] J. Hajič and P. Pecina, "The MUSCIMA++ Dataset for Handwritten Optical Music Recognition," in *Proc. 14th IAPR Int. Conf. Document Analysis and Recognition (ICDAR)*, 2017, pp. 39–46, doi: 10.1109/ICDAR.2017.16.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2014, doi: 10.1145/3422622.

[11] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. Bharath, "Generative Adversarial Networks: An Overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, 2017, doi: 10.1109/MSP.2017.2765202.

[12] N. Li, "Generative Adversarial Network for Musical Notation Recognition during Music Teaching," *Computational Intelligence and Neuroscience*, 2022, doi: 10.1155/2022/8724688.

[13] S. Lee, U. Hwang, S. Min, and S. Yoon, "Polyphonic Music Generation with Sequence Generative Adversarial Networks," *arXiv preprint*, arXiv:1710.11418, 2017.

[14] Raghavendra, M., & Sarappadi, P., 2022. Transfer Learning with Pix2Pix GAN for Generating Realistic Photographs from Viewed Sketch Arts. *Journal of Southwest Jiaotong University*. https://doi.org/10.35741/issn.0258-2724.57.4.17.

[15] H. Dong, W. Hsiao, L. Yang, and Y. Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," in *Proc. AAAI Conf. Artificial Intelligence*, 2017, pp. 34–41, doi: 10.1609/aaai.v32i1.11312.

[16] H. Chen, Q. Xiao, and X. Yin, "Generating Music Algorithm with Deep Convolutional Generative Adversarial Networks," in *Proc. IEEE Int. Conf. Electronics Technology (ICET)*, 2019, pp. 576–580, doi: 10.1109/ELTECH.2019.8839521.

[17] M. Liu, X. Huang, J. Yu, T. Wang, and A. Mallya, "Generative Adversarial Networks for Image and Video Synthesis: Algorithms and Applications," *Proc. IEEE*, vol. 109, no. 5, pp. 839–862, 2020, doi: 10.1109/JPROC.2021.3049196.

[18] É. Clabaut, M. Lemelin, M. Germain, Y. Bouroubi, and T. St-Pierre, "Model Specialization for the Use of ESRGAN on Satellite and Airborne Imagery," *Remote Sens.*, vol. 13, no. 20, p. 4044, 2021, doi: 10.3390/rs13204044.

[19] Z. Zhu, Y. Lei, Y. Qin, C. Zhu, and Y. Zhu, "IRE: Improved Image Super-Resolution Based on Real-ESRGAN," *IEEE Access*, vol. 11, pp. 45334–45348, 2023, doi: 10.1109/ACCESS.2023.3256086.

[20] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-object detection in Remote Sensing Images with End-to-End Edge-Enhanced GAN and Object Detector Network," *Remote Sens.*, vol. 12, no. 9, p. 1432, 2020, doi: 10.20944/preprints202003.0313.v1.

[21] X. Wang et al., "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks," in *Proc. Eur. Conf. Computer Vision (ECCV)*, 2018, pp. 63–79, doi: 10.1007/978-3-030-11021-5_5.

[22] T. Le-Tien, T. Nguyen-Thanh, H. Xuan, G. Nguyen-Truong, and V. Ta-Quoc, "Deep Learning-Based Approach Implemented to Image Super-Resolution," *J. Adv. Inf. Technol.*, vol. 11, no. 4, pp. 209–216, 2020, doi: 10.12720/jait.11.4.209-216.

[23] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data," in *Proc. IEEE/CVF Int. Conf. Computer Vision Workshops (ICCVW)*, 2021, pp. 1905–1914, doi: 10.1109/ICCVW54120.2021.00217.

[24] N. Singh, A. F, M. Rastogi, and R. Prasad, "Performance Analysis of Conditional GANs-based Image-to-Image Translation Models for Low-Light Image Enhancement," in *Proc. Int. Conf. Signal Process. and Communication (ICSC)*, 2022, pp. 468–474, doi: 10.1109/ICSC56524.2022.10009340.

[25] Fujioka, T., Satoh, Y., Imokawa, T., Mori, M., Yamaga, E., Takahashi, K., Kubota, K., Onishi, H., & Tateishi, U., 2022. Proposal to Improve the Image Quality of Short-Acquisition Time-Dedicated Breast Positron Emission Tomography Using the Pix2pix Generative Adversarial Network. *Diagnostics*, 12. https://doi.org/10.3390/diagnostics12123114.

[26] J. Calvo-Zaragoza, A. Gallego, and A. Pertusa, "Recognition of Handwritten Music Symbols with Convolutional Neural Codes," in *Proc. 14th IAPR Int. Conf. Document Analysis and Recognition (ICDAR)*, 2017, pp. 691–696, doi: 10.1109/ICDAR.2017.118.

APPENDIX
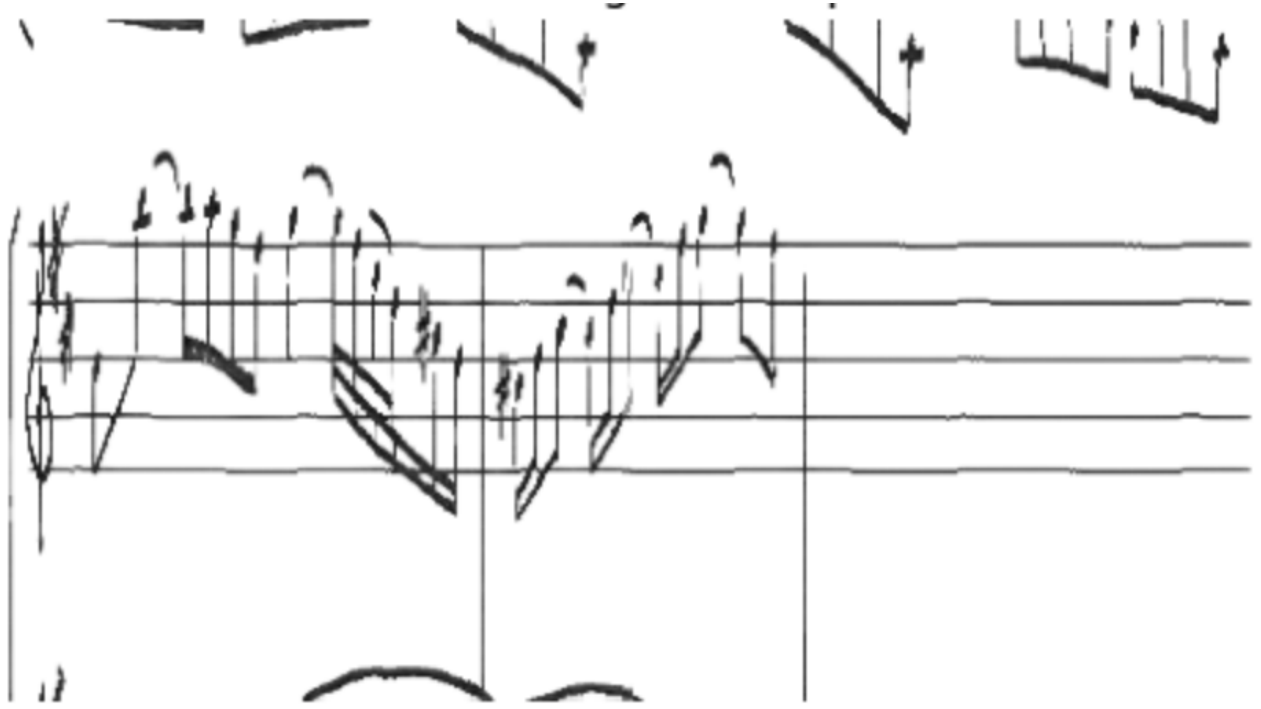
(a)

(b)

Fig. 6: Examples of generated ESRGAN scores with a resolution of 256x512 before post-processing.

Fig. 7: Examples of ESRGAN generated scores after resizing and post-processing.