Proceedings of the 3rd International Workshop on Reading Music Systems

23rd July, 2021
Alicante, Spain

Organization

General Chairs

Jorge Calvo-Zaragoza	University of Alicante, Spain
Alexander Pacha	TU Wien, Austria

Program Committee

Ichiro Fujinaga	McGill University, Canada
Jose M. Iñesta	University of Alicante, Spain
Heinz Roggenkemper	Canamus, United States of America
George Fazekas	Queen Mary University of London, United Kingdom
Gabriel Vigliensoni	Goldsmiths University of London, United Kingdom
Elona Shatri	Queen Mary University of London, United Kingdom
Alicia Fornés	Computer Vision Center, Spain

Proceedings of the 3rd International Workshop on Reading Music Systems, Alicante, 2021

Edited by Jorge Calvo-Zaragoza and Alexander Pacha

Preface

Dear colleagues!

We are more than pleased to present to you the proceedings of the 3rd International Workshop on Reading Music Systems (WoRMS). With the pandemic hopefully slowly fading, we can finally resume our tradition of having an annual workshop that brings together researchers and practitioners that work on music reading systems.

For us, it was always important to create an interactive workshop that brings together people that share a common interest in music reading systems, allowing them to exchange ideas and form relationships with one another. 2020 was a difficult year for many of us - personal situations changed rapidly, research projects were canceled, and not knowing whether an online-only edition of WoRMS would be desirable under these circumstances, we decided, heavy-hearted, to skip WoRMS 2020.

Nevertheless, we are looking forward to this year's edition, which will take place in a hybrid mode with some participants being on-site, while others joining remotely via Zoom. We believe that this way, we can find the balance between enabling interaction and keeping everyone safe. Nothing can replace in-person communication, so we hope that future editions will be fully in-person again. However, we also want to highlight the benefits of this format: offering an online option allows people to join the workshop that couldn't participate otherwise.

This year's edition features 11 contributions, reaching from exciting new datasets to multi-modal methods that might change the way how we think about processing written music. We noted that machine-learning remains a common theme throughout most papers—a trend that we expect to resume in the future.

Finally, we want to thank Escuela Politecnica Superior of University of Alicante for providing the room and the TU Wien for providing Zoom conferencing facilities.

Jorge Calvo-Zaragoza and Alexander Pacha

Contents

The Challenge of Reconstructing Digits in Music Scores

Alexander Pacha
Enote GmbH
alexander.pacha@enote.com

***Abstract*—Digits play an essential role in music scores and need to be reconstructed correctly by any Optical Music Recognition system. This challenge is frequently underestimated and can be a potential source of error if done incorrectly. Typical challenges come from the contextual nature of music notation, ambiguities and general computer-vision issues stemming from image acquisition. In this paper, we discuss the nature of this problem, how image-based methods can be used to tackle them and why they are not enough to fully solve this challenge.**

***Index Terms*—Optical Music Recognition, Digits, Image Classification**

I. Introduction

Music notation tries to capture and encode the most important aspects of music, including the pitch and rhythm, as well as other performance instructions. By convention, many of these aspects are encoded by using numerical information, like the rhythmical division of a measure, the harmony to be played or the finger that should be used to hit a certain note in case of instrumental music.

When developing an Optical Music Recognition (OMR) system [1], these digits have to be reconstructed and assigned to a specific class, to correctly interpret their semantics. For example, a three on top of a note can be an instruction to the musician to use the third finger, or can alter the rhythm of the three closest notes.

However, correctly reconstructing digits can be extremely challenging for multiple reasons, including the assignment problem, intra-class variations, ambiguities, or general recognition problems. In this paper, we want to discuss these challenges in detail, list some approaches that may be used to tackle this problem, and share some open problems that we were not able to solve yet.

II. Digits in Music Scores

Text and digits are integral parts of music notation, as they encode essential information for the musician. Recognizing text and digits can be challenging, but especially digits present some unique challenges that developers of OMR systems should be aware of. When reconstructing the semantics of text, the content is usually sufficient, e.g., the letters *mf* can be classified as dynamics, giving the musician information on how loud he should play. While there are some special cases, where the text changes the music, OMR systems can generally reconstruct the text and remaining music as it is, leaving the interpretation of the text to the performer. In stark contrast, the presence (or deliberate absence) of digits can directly affect the music by changing the duration of nearby notes - information that must be reconstructed and interpreted correctly. To understand if a digit needs special treatment depends on the semantic class it belongs to, like tuplet or fingering. We have identified and grouped digits that appear in music scores into one of the ten classes, shown in Table I.

A. Challenges

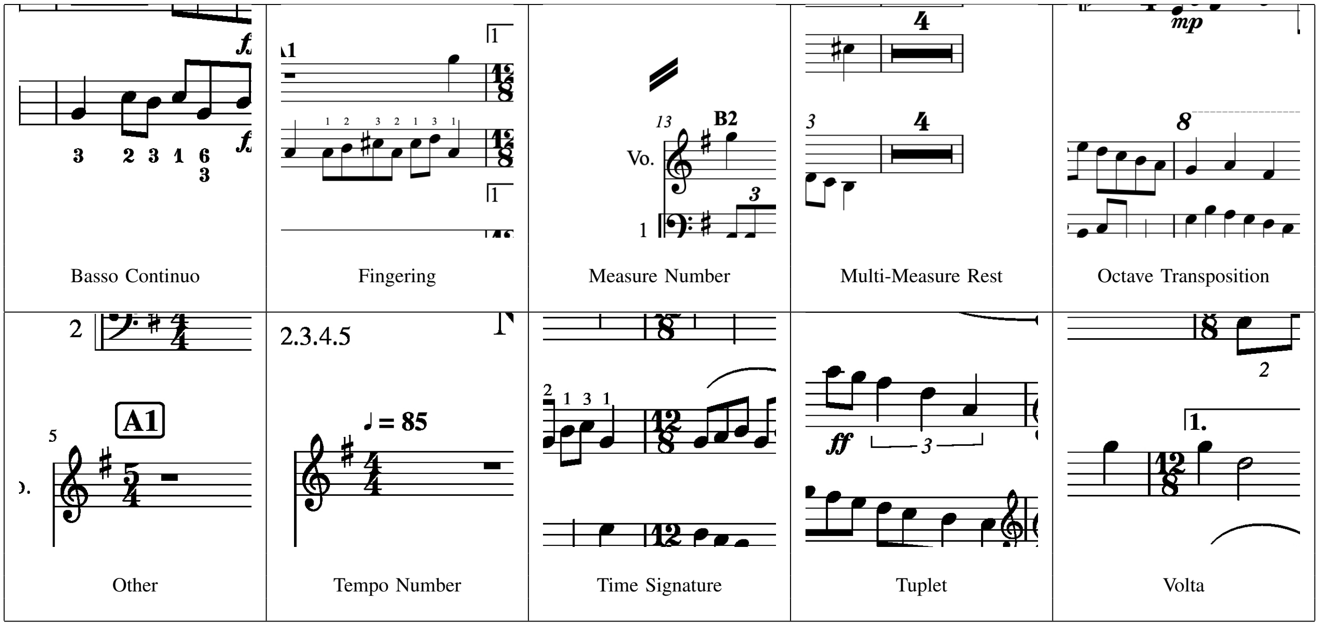
The first challenge one has to tackle is to actually find all digits in an image of music scores. Computer vision methods can be used to solve this, with deep-learning-based approaches showing the most promising results in the last couple of years [2]–[5]. Typical problems that one encounters include bad-quality scans, noise, image distortions, and intra-class variations (e.g., digits being printed in different fonts, italic or bold, or sometimes enclosed in a box or circle). Assuming that this problem can be solved with a powerful neural network, trained on a large number of training samples, we still solved only half of the puzzle. However, knowing which digit appears where, and what number is being depicted is a great start that enables further processing.

The second challenge is the assignment problem of the digits to one of the ten classes, mentioned in Table I. Keep in mind, that these ten classes are categories that we found useful in our work, but are by no means authoritative. This problem is much harder than the first challenge, as it requires a deep understanding not only of the digit, but also of the surrounding context. The reason is that the context often determines the interpretation, thus the class. For example, a three could be a measure number, fingering instruction or the signal that three notes are grouped into a triplet (see Fig. 1).

Some information can potentially be exploited to ease this classification process: While several classes of digits can appear anywhere in the score, other classes only make sense in a very specific context. E.g., a measure number is almost always written at the beginning of a measure (above or below), but usually not in the middle of it. Fingering instructions and multi-measure rests must always have a nearby note and rest, respectively. While this visual context information can be very helpful for a wide range of cases, some problems cannot be resolved by just looking at the image—a deeper semantical analysis must be applied.

The first case, where the visual information is not sufficient are ambiguities between fingering instructions and tuplets. 1(b)

TABLE I
THE TEN CLASSES OF DIGITS THAT WE IDENTIFIED AND USED IN AUTOMATIC IMAGE CLASSIFICATION. THE DIGIT IN QUESTION IS ALWAYS CENTERED.



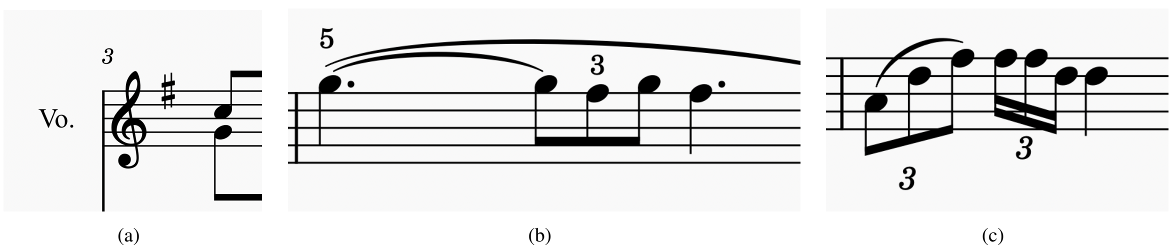


Fig. 1. The number 3, serving as measure number (a), fingering instruction (b) and triplet (c).

and 2(b) contain examples of a digit appearing in a position that is ambiguous when looked at in isolation. Studying a large corpus reveals that these ambiguities aren't rare events, but encountered frequently.

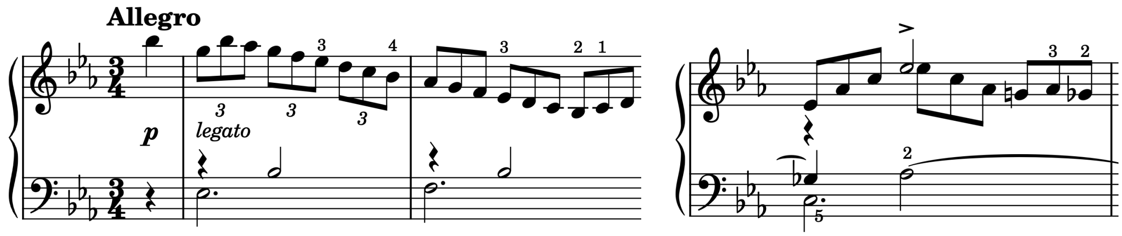
The second case, where the visual information can not be relied upon is when the respective digit is missing. 2(a) contains an example of a common pattern in music notation: If multiple notes are repeatedly grouped into tuplets, the tuplet numbers are omitted after a few measures to not distract from the depicted music. Any OMR system encountering this situation has to "dream up" these invisible numbers when analysing the rhythmical structure, trying to come up with a plausible hypothesis on how to interpret that section of music. 2(b) is particularly tricky, because the groups in the upper stave are indeed triplets, so technically, the number three could correctly indicate a triplet, however, in this case is meant to be a fingering indication. If an OMR system would (correctly) interpret that three as a triplet indicator and hide it, due to the implicit notation, we would lose important information.

To make this challenge even harder, many researchers that tackle this problem and only review a hand-full of scores, discover common patterns that they falsely interpret as universal rules that can be hard-coded into their system. An example for such a pattern is that tuplets frequently appear next to a beamed group of notes. If one builds a system that relies on those beams to detect tuplets, their system will have difficulties with music pieces that fall out of this rule (see Figure 3).

III. CLASSIFYING DIGITS IN MUSIC SCORES

So how can we solve this problem? Admittedly, we underestimated this challenge too and want to give some insights into possible approaches and how they are insufficient to fully solve this problem.

In the era of Deep Learning, it is not uncommon to try if the problem can be solved with machine learning. Given the visual nature of the problem and several previous works that showed promising results for similar challenges [6], convolutional neural networks were evaluated first. Image classification



(a) (b)

Fig. 2. Excerpt of Schubert's Impromptus, Opus 90 (D 899): Triplets are omitted starting in the second measure (left). Musicians are expected to understand the rule of good continuation and fill in the missing thirds afterwards. Later threes that appear in the middle of three grouped notes are to be understood as fingering indications (right).

(a) (b)

Fig. 3. Excerpt of Schubert's Lindenbaum (D 911, Edition Peters): Triplets appear horizontally centered for each beam (a). While triplets are often visually grouped by a beam, it is not always the case (b). In this particular case, a fingering instruction can be ruled out, because it would be incompatible to the used instrument (voice).

problems are nowadays well understood and can typically be solved very well, so a neural network that classifies a digit in an image into one of the ten classes given above presents a sensible choice. We followed a similar approach as proposed for position classification of notes in mensural notation in [7]. Given an image of music that contains a digit in the center, the neural network is challenged to estimate the likelihood of the digit belonging to each class. We assumed, that if enough (visual) context is available, that the problem should be solvable. Additional information can be added as inputs to the network, like the value of the depicted digit (0-9), a-priori distributions of combinations between values and classes (e.g., a tuplet with the value 3 is more likely than the value 2), or compatibility information between the classes and digit values (e.g., fingering instructions are limited to the values 0-5, thus making the values 6, 7, 8, and 9 incompatible to that class). In theory, that should be enough information for a neural network to produce very good results, if enough data is available.

Finally, the semantical reconstruction stage in the OMR system has to evaluate different hypotheses to come up with a final interpretation of a given piece of music, resolving conflicts, such as over-full or under-full measures.

A. Dataset

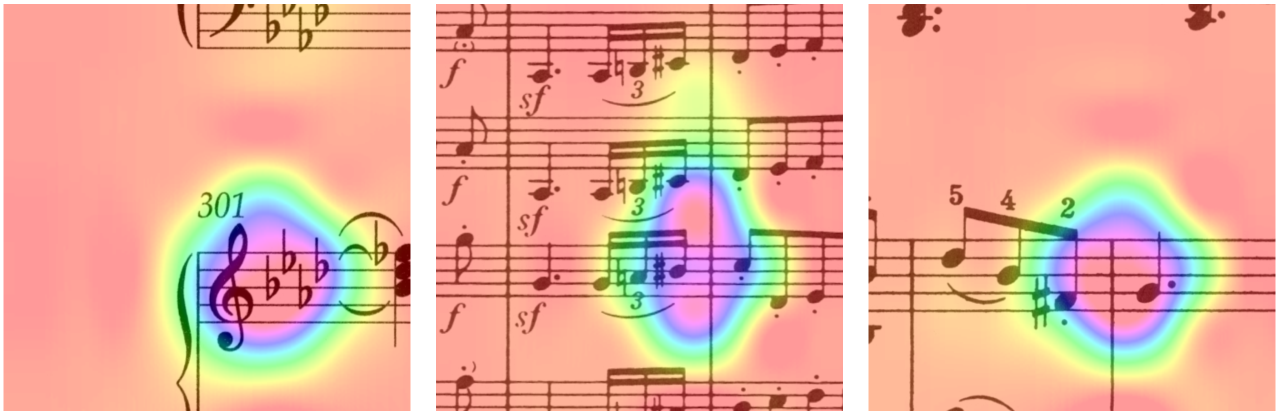
To train such a neural network, we created a dataset with over 10.000 samples, containing both synthetic and real score images. The digit in question was always centered in that image. To ensure the quality, about 7000 samples were manually annotated, based on the detection results of real scores.

B. Results

After training the neural network for several epochs, it typically converged to a validation accuracy of over 95%. However, in practise, it did not live up to that number. Initially, the network was only trained on synthetic examples, and performed quite poorly on real scores. To combat this issue, we added fine-tuning on the 7000 manually annotated, real scores. Unfortunately, the performance in real-world applications was still underwhelming with less than 60% accuracy.

We hypothesise the following reasons could have contributed to that poor performance:

- The network classified the entire image, instead of just the digit in the center of the image. Given that the neural network had no attention mechanisms, it had difficulties figuring out which part of the image is relevant (see Fig. 4).
- The synthetic samples do not cover all real-world cases, and since the network classifies the entire image, it is confronted with contextual information, it has never seen before.
- The problem of classifying digits is inherently more difficult than determining the pitch of a note, as it depends on the context and information that might have appeared some time before in the music.
- Ambiguities force the network to approximate for the most likely case, causing it to struggle with exceptional cases (see Fig. 2(b)).



(a) (b) (c)

Fig. 4. CAM-Attention visualization of a trained network on three samples. The network seems to be focusing on the area that is slightly below and to the right of the middle. This makes sense, as it can help the network to classify measure numbers (a), but might be counter-productive for other classes (b, c).

IV. Conclusion

Classifying digits and assigning it to its musical category is a vital step for correctly reconstructing the musical semantic of music. It allows an Optical Music Recognition system to infer correct durations, and to understand which information just has to be replicated as-is for the musician performing the music. While a purely image-based approach can solve many issues, we believe that the problem is inherently more difficult and cannot be solved entirely by it, because digits are highly context-sensitive, some information is implicit and has to be deduced from earlier bars, there is no standardized way how digits of certain classes are visually represented (e.g., italic vs. non-italic), and some ambiguities can only be solved by carefully examining the example and fusing it with information that might not be present in that image.

Finally, when applying a machine-learning-based approach, one has to ensure that the network actually learns the right thing and not just overfits the training set. Gradient Visualization mechanisms can help to understand what the network actually learnt. Experiments with just an image-based approach did not yield the results that we were hoping for, so we hope that this work encourages future researchers to develop better approaches.

References

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

[2] A.-J. Gallego and J. Calvo-Zaragoza, "Staff-line removal with selectional auto-encoders," *Expert Systems with Applications*, vol. 89, pp. 138–148, 2017.

[3] S. Waloschek, A. Hadjakos, and A. Pacha, "Identification and cross-document alignment of measures in music score images," in *20th International Society for Music Information Retrieval Conference*, 2019.

[4] F. Henkel, R. Kelz, and G. Widmer, "Learning to read and follow music in complete score sheet images," in *Proceedings of the 21st Int. Society for Music Information Retrieval Conf.*, 2020.

[5] A. Baró, P. Riba, J. Calvo-Zaragoza, and A. Fornés, "From optical music recognition to handwritten music recognition: A baseline," *Pattern Recognition Letters*, vol. 123, pp. 1–8, 2019.

[6] A. Pacha, K.-Y. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi, and H. Eidenberger, "Handwritten music object detection: Open issues and baseline results," in *13th International Workshop on Document Analysis Systems*, 2018, pp. 163–168.

[7] A. Pacha and J. Calvo-Zaragoza, "Optical music recognition in mensural notation with region-based convolutional neural networks," in *19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 240–247.

Detecting Staves and Measures in Music Scores with Deep Learning

Marc Kletz
Salzburg University of Applied Sciences
FH Salzburg, Austria
marc.kletz@hotmail.com

Alexander Pacha
Institute of Information Systems Engineering
TU Wien, Austria
alexander.pacha@tuwien.ac.at

***Abstract*—Music scores contain structural elements that provide a reference system and visual guidance, most notably the stave. Finding these structural elements is a fundamental step of every OMR system, often performed together with other preprocessing steps such as binarization or rotation of the page. In this work, we propose a machine learning approach for reliably detecting the basic structuring elements in music scores: staves, measures, and system measures. We evaluate several strategies and show that it is possible to train a neural network to detect these structural elements with high precision. Experiments on a big corpus of handwritten (MUSCIMA++) and typeset (AudioLabs v2) music scores yield a mAP of over 75% when detecting all three classes at the same time and over 90% when detecting them individually. Finally, we describe how we bootstrapped our annotation process, by training a neural network on the MUSCIMA++ dataset and then used the trained network for generating accurate annotations for 940 images of the AudioLabs dataset. The resulting annotations were manually verified, corrected where necessary, and published as AudioLabs v2 dataset.**

***Index Terms*—Optical Music Recognition, Stave Detection, Measure Detection, Object Detection, Deep Learning**

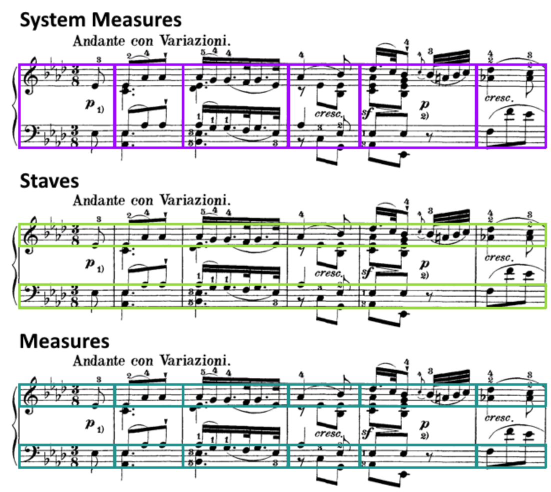


Fig. 1. The basic structural elements of music notation: staves, and measures. For clarity, we distinguish between (stave) measures (bottom) and system measures (top). System measures are measures from multiple staves that should be played simultaneously.

I. Introduction

Music scores can be used to capture very complex musical ideas. To understand these ideas, they are written down in the formal language of music notation. Music notation captures the most relevant events in time, such as the beginning and end of a note, and structures them in a two-dimensional, visual representation. The basic building block is called a stave, typically consisting of one to five parallel lines. The position of objects on these staves corresponds to the desired point in time where they should appear. To give additional visual and musical guidance, staves can be split into measures that are separated by barlines. When more than one instrument is present, they can be notated on multiple staves that are grouped into a system of connected staves, indicating that they should be played at the same time. Understanding this basic structure is crucial for any musician performing a piece, as well as for any Optical Music Recognition (OMR) system trying to read and understand music scores [1].

In this work, we present a machine learning approach that is capable of reliably detecting these basic structuring elements in music scores, more specifically staves, (stave) measures, and system measures (see Fig. 1).

Finding these structural elements is a fundamental step of every OMR system, often performed together with other preprocessing steps such as binarisation or rotation of the page. Because of that, it is rarely singled out as a dedicated task. Although many approaches exist, there is no such thing as a universal structure detector for music scores, even though it would be extremely helpful in many situations. For example, a universal stave detector in combination with a good measure detector can enable easy navigation through the scores by highlighting individual measures. This can be helpful both for humans, browsing through scores, as well as for automatic score-following systems [2], [3] that track a live performance. Furthermore, in a collaborative scenario, detecting staves and measures can be used to provide an empty template with the correct amount of measures. Such a collaborative framework could be used to further enhance and speed up projects such as OpenScore [4] or TROMPA [5]. Finally, knowledge about each measure of a piece of music can already be enough to enable use-cases like searching musical phrases [6], or aligning multiple editions of the same piece of music [7].

Our proposed approach uses state-of-the-art object detection methods (Faster R-CNN) to directly detect rectangular bounding-boxes of staves, measures, and system measures from the raw input image. This approach allows us to find all these structural elements with high precision while also generalizing well to previously unseen scores.

II. Related Work

Finding staves in images of music scores is a topic that has seen a lot of research in the past [8], [9], ranging from simple projection-based approaches [10] to more sophisticated approaches like stable paths [11]. The work that is closest to our approach is described in [12], where an iterative method is used to teach a machine how to detect staves by providing human feedback on a previously trained model. The results, however, are far from ideal, due to the rotated and skewed nature of images.

In contrast to the problem of detecting staves, detecting measures was rarely singled out and more often included as a (pre-)processing step in a bigger OMR pipeline. The work described in [13] is one of the few that did single it out. They propose a traditional computer-vision approach, that relies on staff detection and the discovery of barlines for computing a set of measure candidates, that are filtered to produce the final set of measures. However, it relies on humans providing additional input about the score. In [14], the authors presented a simple pipeline, which detects system measures by clustering horizontal and vertical black pixels of an image and segmenting the image based on these clusters. The only work that proposes a similar approach to ours is the research by Waloschek et al. [7] where the authors also use convolutional neural networks to detect the structure of music scores. However, their model is only trained to detect system measures.

III. Measure Detection with Neural Networks

In this work, we investigate how neural networks can be used to detect staves, measures, and system measures, ideally at the same time.

Several papers have shown that neural networks can reliably detect music symbols [15]–[17]. Considering the lower structural complexity of measures and staves, it seems reasonable that these should be detectable with a similar approach too, especially considering their large size and fewer instances per image. Unfortunately, even though staves are defined as horizontal lines, real scans of music scores can be rotated, bent, and skewed, as the result of camera-based image-acquisition methods, which makes the detection problem significantly harder.

We propose a machine learning approach using convolutional neural networks, that is capable of detecting staves, measures, and system measures within music scores with state-of-the-art accuracy. We evaluate our system on two open-source datasets and show that it performs very well on handwritten and typeset music.

A. Framework

Detectron2 [18] has been chosen as the neural network framework for this work because it is flexible, fast, and easy to adopt. The Detectron2 implementation provides state-of-the-art object detection algorithms, including Panoptic feature pyramid networks, Mask R-CNN, and Faster R-CNN [19].

We use the Faster R-CNN capabilities of Detectron2 to predict bounding boxes in images. Faster R-CNN is a two-stage object detection framework, consisting of a region proposal network and a classification and regression network, all of which can be trained simultaneously end-to-end. This type of network has been found to achieve excellent results for many object detection problems. For the shared backbone network, we evaluated three networks: ResNet-50, ResNet-101 and ResNeXt-101-32x8d [18].

B. Datasets

For our experiments, we used two open-source datasets: The handwritten MUSCIMA++ dataset with additional measure annotations, and the AudioLabs dataset[1] from [14]. Both are available from the OMR-Datasets repository [20]. However, the AudioLabs dataset only contains system measure annotations.

To the best of our knowledge, MUSCIMA++ is the only dataset that contains annotations for staves, measures, and system measures. This means that there is only one handwritten dataset[2] with these annotations and no typeset dataset that contains all three annotation types. Training a neural network on only handwritten data seems insufficient, considering a vast amount of music being typeset. This led to the idea of adding the missing annotations to the AudioLabs dataset, to create the first typeset dataset with complete measure annotations.

C. Bootstrapping Annotations

We trained a neural network to detect the stave measures from the MUSCIMA++ dataset and used that model to predict the stave measures in the AudioLabs dataset (see Fig. 2, top). The predicted stave measures were used to calculate the staves by grouping them based on their y-coordinates on the music sheet. If two measures have overlapping y-coordinates, it indicates that they belong to the same stave. Averaging over all y-coordinates for grouped measures results in an approximation for the staves. The obtained staves were then sliced with the existing system measures to calculate precise stave measures (see Fig. 2, bottom). In both steps, we manually corrected any errors that we found. We call the newly annotated dataset AudioLabs v2 and it can be downloaded from https://github.com/apacha/OMR-Datasets/releases/download/datasets/AudioLabs_v2.zip and used under the same license as the original AudioLabs dataset.

[1]The original name is “typeset bounding box annotations of musical measures”, but we took the liberty to shorten that into the “AudioLabs dataset”

[2]Even though the notes and symbols in the MUSCIMA++ dataset are handwritten, it should be noted that the staves are synthesized, binarized and of very high quality with homogenous thickness and spacing between them.

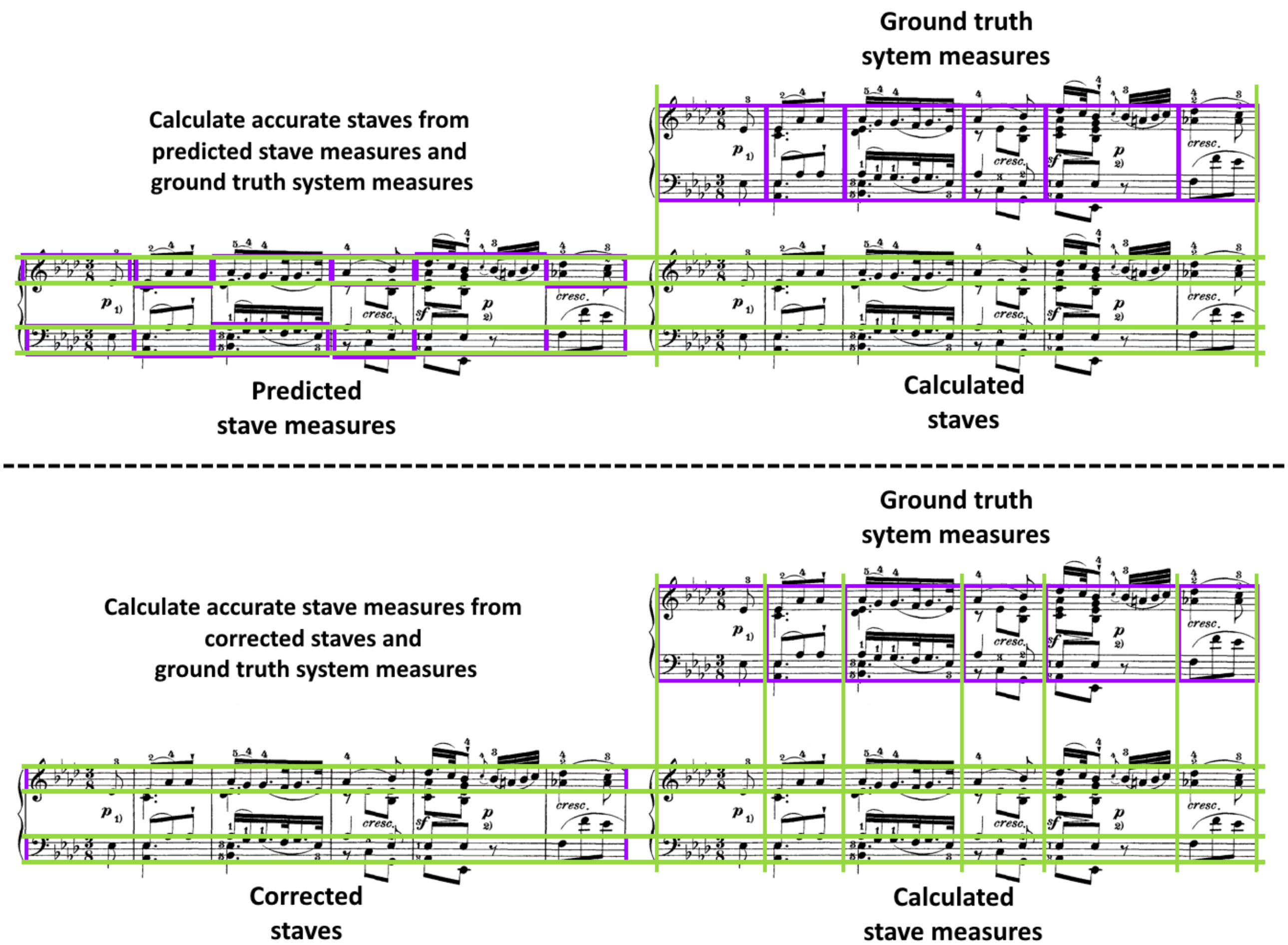


Fig. 2. Beethoven Sonate Op. 26, Depicting our bootstrapping process in a nutshell: First, we predict stave measure, group them by their y-coordinates and take the average height to calculate the stave height. Then we intersect the resulting staves with the far right and far left ground truth system measures to delimit staves horizontally for the final approximation of staves. Afterwards we manually correct any potential errors. Finally, we use the corrected staves and intersect them with the ground truth system measures to get accurate stave measures.

For our experiments, we used both the MUSCIMA++ handwritten dataset, consisting of 140 annotated music scores in combination with the 9 additional augmentations for each of these images from the CVC-MUSCIMA dataset [21], summing up to a total of 1400 images and AudioLabs v2 dataset consisting of 940 images. The number of bounding boxes for each category is given in Table I. The distribution of annotations is quite even between both of the datasets.

D. Training Regime

Both datasets were split into 60% training data, 20% test data, and 20% validation data. This split happens before merging the datasets to give both datasets the same relative distribution. This means that the training dataset contains 60% of the MUSCIMA++ data and 60% from the AudioLabs v2 dataset. We also made sure that every image from the MUSCIMA++ dataset, along with its 9 augmentations would only appear in either the test, training, or validation dataset to prevent data leaks.

We trained several models that were either challenged to produce predictions for a single class or a combination of multiple classes at the same time. In total, we evaluated five model combinations: Three single-class models, one for the combination of system measures and staves, and one model for detecting all three classes at the same time.

In our experiments, the used models performed best when training between 10000 and 20000 steps, upon which the validation performance plateaued. So we are training for a maximum of 20000 steps with a learning rate of 0.005 and a batch size of three (the maximum number of images that fit into the memory of the graphics card). We calculate the average precision (AP) and mean average precision (mAP) metrics on the validation set every 300 steps. The model that performs best on the validation dataset is then selected for the final evaluation on the test-set (see Table II).

TABLE I
THE NUMBER OF PAGES AND BOUNDING BOXES IN THE MUSCIMA++ AND AUDIOLABS V2 DATASETS.

	Nr. of pages	System Measures	Staves	Stave Measures	Total boxes
MUSCIMA++	1400	28.880	8.830	46.160	83.870
AudioLabs v2	940	24.186	11.143	50.651	85.970
Total	2330	53.066	19.973	96.811	169.850

IV. RESULTS AND DISCUSSION

We evaluated several combinations of models and backbones, with Table II listing the obtained results. All backbones perform similarly well with the greatest impact coming from whether we trained one model for all three classes or three different models. Models that were trained on all three classes at the same time, performed much worse than models which were trained on just a single class or the two classes staves and system measures. We hypothesize that the three-class model is performing worse due to pages containing only one instrument, in which case the stave measures and system measures are the same. When applying non-maximum suppression, which is used in Faster R-CNNs models to eliminate nearby prediction duplicates, one of the two overlapping boxes will be suppressed and is then missing in the output.

Although the accuracy of the networks seems extremely high, it should be noted that these models might not work for every kind of sheet music, as our dataset lacks diversity. They work very well for typeset music, but might not work as well for music with handwritten staves or images that have perspective distortions. More training data that is realistic and diverse could help to improve robustness in the future.

If training and running three separate models is too expensive, a two-class model with staves and system measures can be a suitable alternative, because the remaining stave measures can automatically be derived from these detections (see Fig. 2).

We conclude that training neural networks to predict stave measures, system measures, and staves in music notation can work well under the following conditions: (a) The available training set must be large enough to cover a wide range of possible situations, ideally with a mixture of handwritten and typeset music scores, (b) Rotation of pages is handled appropriately, either by training on rotated images (e.g., with image augmentation) or by correcting the page rotation before detecting the structure (e.g., with an image dewarping system as a preprocessing step), and (c) Annotations must not be completely overlapping when a non-maximum suppression mechanism is used in the detection process.

While the recognized structure of a music score is only a starting point for any OMR system, we hope that our work still proves to be useful for future researchers, working on OMR. We showed how a modern neural network approach can yield excellent results, given a large enough, annotated dataset. For future research in this area, we make our source code and datasets publicly available (https://github.com/MarcKletz/OMR-MeasureRecognition). We believe that the biggest advances in this area can be achieved by expanding the dataset to be more versatile, and by employing a robust dewarping system that can correct page rotations and other deformations before detecting the structure of music scores. If page rotations are not corrected, the neural network tries to fit non-rotated rectangles around rotated targets, which inevitably leads to poor results.

TABLE II
TEST-SET PERFORMANCE FOR ALL MODELS, GROUPED BY THEIR CATEGORIES AND BACKBONES.

Single Class Models		mAP	AP75	AP50
System Measure	ResNet-50	95.828	98.785	98.982
	ResNet-101	95.996	98.823	98.988
	ResNeXt-101	95.907	98.931	99.008
Stave measures	ResNet-50	87.639	97.582	98.933
	ResNet-101	88.882	97.515	98.938
	ResNeXt-101	89.625	97.785	99.001
Staves	ResNet-50	92.578	99.003	99.010
	ResNet-101	93.650	100.00	100.00
	ResNeXt-101	93.457	99.009	100.00

Two Class Models		mAP	AP75	AP50
System measures & Staves	ResNet-50	88.190	95.423	95.519
	ResNet-101	88.886	96.962	97.018
	ResNeXt-101	88.941	95.319	95.693

Three Class Models		mAP	AP75	AP50
System measures & Stave measures & Staves	ResNet-50	75.970	85.549	86.422
	ResNet-101	75.041	85.297	86.713
	ResNeXt-101	75.922	86.017	87.059

REFERENCES

[1] J. Calvo-Zaragoza, J. Hajič Jr., and A. Pacha, "Understanding optical music recognition," *ACM Comput. Surv.*, vol. 53, no. 4, 2020.

[2] A. Cont, "A coupled duration-focused architecture for real-time music-to-score alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 974–987, 2010.

[3] F. Henkel, R. Kelz, and G. Widmer, "Learning to read and follow music in complete score sheet images," in *Proceedings of the 21st Int. Society for Music Information Retrieval Conf.*, 2020.

[4] M. Gotham, P. Jonas, B. Bower, W. Bosworth, D. Rootham, and L. VanHandel, "Scores of scores: An openscore project to encode and share sheet music," in *5th International Conference on Digital Libraries for Musicology*. Paris, France: ACM, 2018, pp. 87–95.

[5] D. M. Weigl, W. Goebl, A. Hofmann, T. Crawford, F. Zubani, C. C. S. Liem, and A. Porter, "Read/write digital libraries for musicology," in *7th International Conference on Digital Libraries for Musicology*, ser. DLfM 2020. New York, NY, USA: Association for Computing Machinery, 2020, pp. 48–52.

[6] J. Hajič jr., M. Kolárová, A. Pacha, and J. Calvo-Zaragoza, "How current optical music recognition systems are becoming useful for digital libraries," in *5th International Conference on Digital Libraries for Musicology*. Paris, France: ACM, 2018, pp. 57–61.

[7] S. Waloschek, A. Hadjakos, and A. Pacha, "Identification and cross-document alignment of measures in music score images," in *20th International Society for Music Information Retrieval Conference*, 2019.

[8] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. d. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 3, pp. 173–190, 2012.

[9] A. Fornés, V. C. Kieu, M. Visani, N. Journet, and A. Dutta, "The ICDAR/GREC 2013 music scores competition: Staff removal," in *Graphics Recognition. Current Trends and Challenges*, B. Lamiroy and J.-M. Ogier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 207–220.

[10] D. Bainbridge and T. Bell, "The challenge of optical music recognition," *Computers and the Humanities*, vol. 35, no. 2, pp. 95–121, 2001.

[11] J. d. S. Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. Pinto da Costa, "Staff detection with stable paths," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1134–1139, 2009.

[12] A. Pacha, "Incremental supervised staff detection," in *2nd International Workshop on Reading Music Systems*, J. Calvo-Zaragoza and A. Pacha, Eds., Delft, The Netherlands, 2019, pp. 16–20.

[13] G. Vigliensoni, G. Burlet, and I. Fujinaga, "Optical measure recognition in common music notation," in *14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013.

[14] F. Zalkow, A. V. Corrales, T. Tsai, V. Arifi-Müller, and M. Müller, "Tools for semi-automatic bounding box annotation of musical measures in sheet music," in *Late Breaking/Demo at 20th International Society for Music Information Retrieval*, Delft, The Netherlands, 2019.

[15] A. Pacha, J. Hajič jr., and J. Calvo-Zaragoza, "A baseline for general music object detection with deep learning," *Applied Sciences*, vol. 8, no. 9, pp. 1488–1508, 2018.

[16] J. Calvo-Zaragoza and D. Rizo, "End-to-end neural optical music recognition of monophonic scores," *Applied Sciences*, vol. 8, no. 4, 2018.

[17] L. Tuggener, I. Elezi, J. Schmidhuber, and T. Stadelmann, "Deep watershed detector for music object recognition," in *19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 271–278.

[18] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99.

[20] A. Pacha, "The OMR datasets project," https://apacha.github.io/OMR-Datasets, 2017.

[21] A. Fornés, A. Dutta, A. Gordo, and J. Lladós, "CVC-MUSCIMA: A ground-truth of handwritten music score images for writer identification and staff removal," *International Journal on Document Analysis and Recognition*, vol. 15, no. 3, pp. 243–251, 2012.

Sequential Next-Symbol Prediction for Optical Music Recognition

Enrique Mas-Candela
University of Alicante, Spain
emc89@alu.ua.es

María Alfaro-Contreras
University of Alicante, Spain
m.alfaro@ua.es

***Abstract*—State-of-the-art technologies for Optical Music Recognition typically follow an end-to-end approach that retrieves the series of symbols that appear in the image of a single staff in a sole stage. This type of model demands a training set of sufficient size; however, the existence of many music manuscripts of reduced size questions the usefulness of this framework. In order to address such a drawback, we propose a sequential classification-based approach for music documents that processes sequentially the staff image. This is achieved by predicting, in the proper reading order, the symbol locations and their corresponding music-notation labels. Our experimental results report a noticeable improvement over previous attempts in scenarios of limited ground truth (for instance, decreasing the Symbol Error Rate from 70 % to 37 % with just 80 training staves), while still attaining a competitive performance as the training set size increases.**

***Index Terms*—Optical Music Recognition, Handwritten Music Recognition, Deep Learning, Reading Order**

I. Introduction

As in many other fields, modern Machine Learning techniques, namely Deep Neural Networks (DNN), have brought new learning-based approaches to Optical Music Recogniton (OMR) [1]. End-to-end systems, based on Convolutional Recurrent Neural Networks (CRNN), that retrieve the series of symbols that appear in a single-section staff image, can be considered the current state of the art [2]–[5]. To develop these approaches, only training pairs (problem images together with their corresponding transcript solutions) are needed. As long as there is sufficient training data, the results achieved can be considered effective for transcribing music notation.

However, the size of the training set might become an issue especially, when transcribing small music manuscripts. In these cases, which are quite common in historical music heritage, the amount of data needed to train an accurate system might be close to the total amount of data to be transcribed. This could lead to a scenario in which the use of automatic technology is not useful at all.

In this paper, we propose an alternative classification-based system. More concretely, we introduce a Next-Symbol Prediction (NSP) model, which aims at predicting the sequence of locations of each symbol of a staff in their reading order. For that, a Convolutional Neural Network (CNN) is trained to predict the location of the next symbol in the staff, conditioned to a current symbol location. Once a symbol is located, we retrieve both its shape and its vertical position in the staff, which are features related to the rhythm and the tone, respectively. This is achieved by simple and common classification CNNs, widely used in symbol classification [6].

Our results show that the proposed approach yields good results with a small amount of training data, as opposed to the state of the art, while still providing a competitive performance as the ground-truth size increases. In addition, the present work opens up new avenues for research that will be discussed below.

II. Framework

We define the OMR problem here as the task of retrieving the music-notation symbols that appear in a given staff-section image. As in the state-of-the-art works referenced above, we also assume that a previous process isolated each staff of the pages, much in the same way as most Handwritten Text Recognition systems assume a previous line-level segmentation.

We propose an OMR system that moves over a single-staff section image, focusing each time on the *next* symbol and predicting its shape and position. By iteratively repeating the process, we are able to decode the full staff.

Formally, let us assume that a staff section $\mathbf{x} = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ is a collection of symbols with sequential reading order. Each symbol is, in turn, modeled by a pair $\sigma_i = (c_i, l_i)$ where $c_i \in \mathbb{R}^2$ represents the center of the symbol in the image and l_i represents its label. In addition, given that music symbols are defined by both its shape and its vertical position within the staff (height), the label l_i consists of a pair (s_i, h_i), where $s_i \in \Sigma_s$ and $h_i \in \Sigma_h$ represent the shape and height components, respectively, from fixed alphabets. This is illustrated in Fig. 1.

Broadly speaking, our approach seeks to estimate two functions, both conditioned to a symbol center c_i. The first function must predict the label l_i of the symbol whose center is given. A second function must predict the center c_{i+1} of the next symbol to be read. In this work, we resort to simple classification schemes for the former, while the latter is performed by means of the aforementioned NSP module.

A graphical overview of our proposal is illustrated in Fig. 2. We below delve into the architectures performing each function.

Second author's research is funded by the Spanish Ministerio de Universidades through grant FPU19/04957.

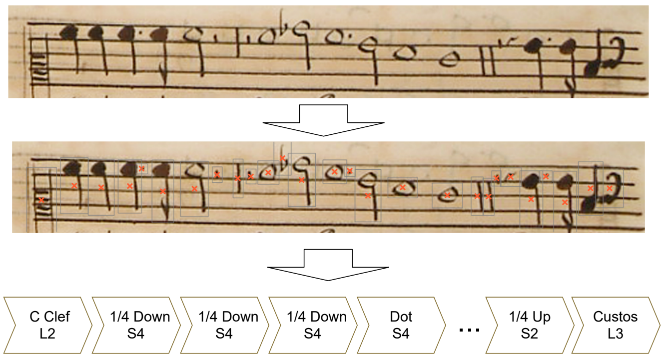


Fig. 1. Illustration of our reading music process. From top to bottom: the single staff image, the location of the symbols as the center (red crosses) of hypothetical bounding boxes, and the decoded sequence (comprising two features: shape and vertical position of the symbol in the staff).

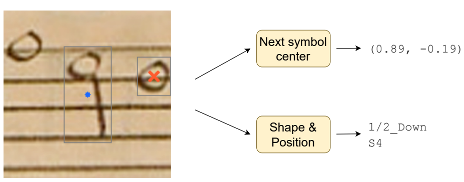


Fig. 2. Graphical description of the proposed approach. Given the blue ground truth symbol center, the shape and position models properly recognize the symbol in terms of its graphical meaning while the NSP module predicts the location of the next symbol in the sequence, illustrated in red. In light gray are represented the labeled bounding boxes which are used to compute the center of each symbol. The predicted coordinates are in the range $[-1, 1]$, as the NSP module works in this range. The labels *1/2 down* and *S4* correspond to the shape and height of the symbol respectively—the first one means the symbol is a down-oriented note that lasts half of the duration of a whole note, and the second one means that the note is in the fourth space between lines.

A. Shape and height prediction

We consider that every music symbol can be completely recognized by two graphics components: shape and height, which typically condition its duration and pitch, respectively. Note that even those that do not represent any duration (clefs or alterations) or any pitch (rests), still have a specific shape and are placed in a specific location within the staff lines.

We resort to two different classifiers, one CNN for each component, for the label prediction module. Given a symbol center $c_i = (x_i, y_i)$, we take a neighboring region from the top-left corner $\left(x_i - C_h^L, y_i - C_w^L\right)$ to the bottom-right corner $\left(x_i + C_h^L, y_i + C_w^L\right)$, where C_h^L and C_w^L represent the *context* height and width, respectively, of the corresponding label $L \in \{\text{shape}, \text{height}\}$ classifier. The specific values for these neighboring regions are determined empirically. We then input the corresponding region to each CNN so that they predict the label, s_i or h_i, as appropriate.

B. Next-Symbol Prediction module

The NSP module is intended to predict, in the proper reading order, a sequence of locations corresponding with each of the symbols on the given staff image.

It works as follows: given a current symbol location $c_i = (x_i, y_i)$, we follow the same procedure as in previous section and take a neighboring region from the top-left corner $\left(x_i - C_h^{\text{NSP}}, y_i - C_w^{\text{NSP}}\right)$ to the bottom-right corner $\left(x_i + C_h^{\text{NSP}}, y_i + C_w^{\text{NSP}}\right)$, where C_h^{NSP} and C_w^{NSP} represent the *context* height and width of the NSP module, respectively—whose values are empirically studied. We then feed a model with such region to estimate the center of the next symbol $c_{i+1} = (x_{i+1}, y_{i+1})$.

In our framework, the coordinates of the next symbol c_{i+1} are always within the range $[-1, 1]$, in normalized relation to the current center c_i. Therefore, we map the neighboring region, also referred to as *context image*, onto the $[-1, 1]^2$ space.

When starting the process, the first context image is centered horizontally in the first column of the staff image, with zero padding on the negative side, and vertically centered in the staff region. Similarly, zero padding is included at the end of the staff, so that when the location of the next symbol predicted by NSP falls in that area (that is, outside the actual staff), the reading of the section is finished.

To implement the NSP, we consider a CNN, which is formed by an initial convolutional stage, that acts as a *backbone*, followed by a regression layer, that acts as an *output block*. Given an RGB image of the context of the current symbol, the backbone performs a feature extraction of that image that the output block later synthesizes into a numerical coordinate representation, (x, y), indicating the estimated location of the next symbol center.

III. Experiments

In this section, we describe the data set, the evaluation protocol, and the architectures of the involved models.

A. Corpus

We consider the *Capitan* corpus [7], which contains a manuscript from the 17th century of a *missa* (sacred music) in the so-called Mensural notation.[1] An example of a staff section from this corpus is depicted in Fig. 3.

Fig. 3. Handwritten music staff-section from *Capitan* dataset.

The ground-truth data already provides the segmentation of the pages into staves. This leads to a total of 99 pages, 704 single-staff sections, that amount to 17,112 running symbols, belonging to 53 and 16 different classes for shape and height, respectively. The annotation of this corpus includes the bounding boxes of the symbols and, therefore, we assume the center of those boxes as their locations.

[1]Music notation system used for most part of the XVI and XVII centuries in Europe.

B. Evaluation protocol

Taking into account the different modules of the proposed framework, we consider several metrics to measure the performance of each of them individually, and also as a whole, namely:

- The shape and height classifiers are evaluated individually with simple categorical accuracy (cACC).
- The performance of the NSP module is evaluated with the Continuous Euclidean Error Rate (CEER) computed as the Euclidean Distance between the ground-truth position of a symbol and the location predicted considering the previous predicted location as the initial location.
- The NSP module together with the shape and height classifiers constitute a complete OMR model. The performance of such model is evaluated with the Symbol Error Rate (SER) computed as the average number of elementary editing operations (insertions, deletions, or substitutions) necessary to match the predicted sequence with the ground truth one, normalized by the length of the latter. It must be noted that this figure of merit is applied when both labels, s_i and h_i, are treated as a unique category, i.e., the label space considered is $\Sigma_s \times \Sigma_h$. This is the common metric of evaluation in state-of-the-art OMR.

For the experiments, we follow a 5-fold cross-validation, each fold containing its corresponding training (61 pages), validation (19 pages), and test (19 pages) partitions. The average result over the 5 folds will be reported. This figure constitutes that of the test data partition for the case in which the validation data achieve its best performance.

C. NSP configuration

The backbone of the NSP module follows the well-known convolutional layers of the VGG16 [8]—pretrained with *ImageNet* [9]—a common reference in computer vision.

On the other hand, we consider a *differentiable spatial to numerical transform* (DSNT) layer [10] for the output block as it was specifically designed for predicting image coordinates. This layer is fed with a single-channel normalized heatmap, $\hat{\boldsymbol{Z}}$. The term "normalized" indicates that all values of $\hat{\boldsymbol{Z}}$ are non-negative and sum to one. Such normalized heatmap is obtained after applying a two-dimensional softmax activation to the feature map predicted by the backbone. Then, through matrix conversion, it is output as numerical coordinates of range $[-1, 1]$, as mentioned in Section II-B.

As a last remark, locations are predicted from a context image of size $\left(2\,C_h^{\text{NSP}} \times 2\,C_w^{\text{NSP}}\right)$ as described in Section II-B. By means of informal experimentation, we found that (192×192) was an appropriate size as it was large enough to encompass sufficient information without causing difficulties in the learning process.

D. Label classifiers configuration

The two CNN models for retrieving the shape and height features of music notation symbols, respectively, are set as an equally-configured CNN.

The CNN consists of five convolutional blocks. Each block consists, in turn, of a 3×3 convolution layer with $32 \times 2^{n-1}$ filters, followed by 2×2 max-pooling for downsampling and a dropout of 30%, where n represents the layer (from 1 to 5). After the 5th convolutional block, we set a fully-connected layer with 512 units followed by a batch normalization layer and a dropout of 50%. Then, the last layer of the model consists of a fully-connected layer with the number of units equals the number of classes for each classifier, along with a softmax function that converts the activations into probabilities.

As mentioned in Section II-A, given a context image of $(2\,C_h^L \times 2\,C_w^L)$, the corresponding label, s_i or h_i, is predicted. We empirically found that (200×200) and (250×125) were appropriate for L = shape and L = height, respectively. These sizes provide sufficient relevant information for the classification process. However, to reduce the complexity of the learning process, we resized these context images to (96×96) and (100×50), respectively, before feeding the information to the CNN.

IV. Results

In this section, we present the results obtained in our experiments in the following order: first, those obtained in a performance analysis of our method, and second, those concerning a comparison between our method and the baseline.

A. Performance analysis

To properly assess the performance of our proposal, we evaluate the contribution of each module. Table I provides the average results of this analysis over the test set. As it may be checked, the classifiers perform reasonably well, considering a general neighboring region and non-perfect localization. The height classifier reports a lower accuracy than the shape one because height is usually harder to learn as there is greater variability between samples of the same class, despite having a smaller number of classes. Furthermore, we can observe that the NSP generalizes well. For a qualitative analysis, Fig. 4 illustrates the NSP performance over a selected part of one test image. Note that most of the centers are not perfectly located but close enough for assuming a correct detection. As reported in the table, all these modules yield a 14.2 % SER when combined to perform a complete pipeline.

TABLE I

Average over a 5-fold cross-validation over the test set attained by each module of the proposal and the method as a whole (Complete pipeline).

Modules		
	Label classifier (shape)	98.3 % cACC
	Label classifier (height)	89.8 % cACC
	NSP	15.9 % CEER
Complete pipeline		14.2 % SER

In order to gain more insights about which module of the proposed method has a greater impact on the complete pipeline, we compute a histogram of the edit operations

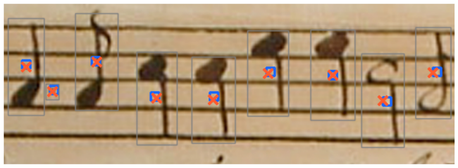


Fig. 4. Illustration of a qualitative evaluation for a selected part of a single test sample. The blue squared marks represent the ground truth symbol locations, computed from the light gray bounding boxes (ground-truth annotations of the corpus); and the red crosses denote the symbol locations predicted by NSP.

(insertions, deletions, and substitutions) when measuring the edit distance between the ground-truth symbol sequences and the predicted ones. We consider that each insertion or deletion is caused by an error on the NSP module, which means predicting a sequence shorter or longer than the ground truth one—and therefore an incorrect symbol location—and each substitution is attributed to the shape or the height classifiers, which are assigning an incorrect symbol feature. The computation report that out of the total editing operations performed to match the predicted sequences to the ground truth ones, 21% correspond to the NSP module (6.5% and 14.5% of insertions and deletions, respectively) and the remaining 79.0% to substitutions (that is, classifiers). Hence, the shape and height prediction models are most likely to be causing most of the errors of the complete pipeline. This fact suggests two possible reasons: (i) the neighboring regions and/or the CNNs considered might not be very appropriate for these classifications, or (ii) the NSP subtle errors might cause difficulties in the classification processes.

B. Comparison with the state of the art

We want to observe how the amount of training data impacts the learning process. For that, we carried out an incremental training experiment. As a starting point, we evaluated the model using only 11 pages out of the total 61 that make up each training partition. We then repeat the process by increasing the number of training pages by 10 and so on, until we reach the scenario where all training pages are used. In addition, the performance of the considered approach is compared with the baseline algorithm. Specifically, a CTC-trained CRNN has been implemented following the details provided in the work by Calvo-Zaragoza et al. [2].

The results of this experiment are shown in Fig. 5. An inspection of the reported figures reveals two relevant conclusions. First, a sequential classification-based approach, based on the combined use of our modules (NSP and label classifiers), allows the retrieval of the series of symbols that appear in the image of a single staff successfully. Second, the aforementioned approach drastically improves all the results obtained with CTC-trained CRNN in the cases when using a limited amount of training data. For sufficient-data cases, our approach shows a competitive performance, since the differences in the figure errors with respect to those of the baseline are relatively low.

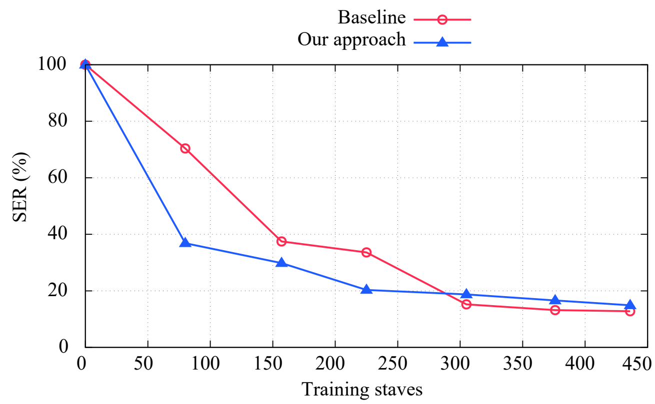


Fig. 5. SER (%) attained by the compared methods with respect to the number of training staves.

V. Conclusions

Further improvements of state-of-the-art OMR systems are limited by the size of the available ground truth. To address such shortcoming, we propose a sequential classification-based OMR system, that retrieves the series of symbols that appear in the image of a single staff by performing symbol detection and classification as two consecutive but dependent tasks.

For that, we present the Next-Symbol Prediction module, implemented as a CNN that predicts the location of the next symbol in the staff as a pair of cartesian coordinates. This prediction is conditioned to a current symbol location, characterized in terms of a neighboring context image. We use it along with two CNNs that classify the located symbol by its two-dimensional nature: shape and height (vertical position within the staff lines), respectively.

In our experiments over a handwritten 17th-century manuscript, the recognition results show that when given a small amount of training data, the presented system improves considerably the state-of-the art results (for instance, it decreases the symbol error rate from 70 % to 37 % with 80 training images), while still reporting a competitive approach as the training set size increases. In spite of this benefit, the main disadvantage of our proposal is that it involves fine labeling symbol positions, which requires more effort to create labeled corpora than in the CRNN-CTC approach, that only needs sequence transcripts.

Given the variability of music notation and the relative scarcity of existing labeled data, we aim at exploring *transfer learning* techniques to study different strategies to properly exploit the knowledge gathered from a given corpus on a different one. We also consider that *data augmentation* could be exploited to make the neural models much more robust. The use of a classification-based approach, rather than a holistic approach, suggests that these future research avenues may yield new insights on how to improve the overall performance of our model.

REFERENCES

[1] J. Calvo-Zaragoza, J. H. Jr, and A. Pacha, “Understanding optical music recognition,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–35, 2020.

[2] 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.

[3] A. Baró, C. Badal, and A. Fornês, “Handwritten historical music recognition by sequence-to-sequence with attention mechanism,” in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, pp. 205–210.

[4] 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.

[5] C. Wick and F. Puppe, “Experiments and detailed error-analysis of automatic square notation transcription of medieval music manuscripts using cnn/lstm-networks and a neume dictionary,” *Journal of New Music Research*, pp. 1–19, 2021.

[6] A. Pacha and H. Eidenberger, “Towards a universal music symbol classifier,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 2. IEEE, 2017, pp. 35–36.

[7] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, “Handwritten music recognition for mensural notation: Formulation, data and baseline results,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1081–1086.

[8] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[10] A. Nibali, Z. He, S. Morgan, and L. Prendergast, “Numerical coordinate regression with convolutional neural networks,” *Computer Research Repository*, vol. abs/1801.07372, 2018. [Online]. Available: http://arxiv.org/abs/1801.07372

Multimodal Audio and Image Music Transcription

Carlos de la Fuente, Jose J. Valero-Mas, Francisco J. Castellanos, Jorge Calvo-Zaragoza
Department of Software and Computing Systems
University of Alicante, Alicante, Spain
cdlf4@alu.ua.es, {jjvalero, fcastellanos, jcalvo}@dlsi.ua.es

***Abstract*—Optical Music Recognition (OMR) and Automatic Music Transcription (AMT) stand for the research fields which aim at obtaining a structured digital representation of the music content present in either a sheet music image or an acoustic recording, respectively. While these fields have historically evolved separately, the fact that both tasks share the same output representation poses the question of whether they could be combined in a multimodal framework that exploits the individual transcription advantages depicted by each modality in a synergistic manner. To assess this hypothesis, this paper presents a proof-of-concept work that combines the predictions given by end-to-end AMT and OMR systems over a corpus of monophonic music pieces considering a local alignment approach. The results obtained, while showing a narrow improvement with respect to the best individual modality, validate our initial premise, thus opening avenues for further research in combined OMR-AMT transcription.**

***Index Terms*—Optical Music Recognition, Automatic Music Transcription, Multimodal Recognition**

I. Introduction

The attainment of structured digital representations of music sources, typically known as *transcription*, remains as one of the key, yet challenging, tasks in the Music Information Retrieval (MIR) field [1]. Such digitization not only improves the issue of music heritage preservation and dissemination [2], but it also enables the use of computer-based tools which allow indexing, analysis, and retrieval, among many other tasks [3].

Under this transcription framework, two particular research lines stand out within the MIR community: on the one hand, when tackling music scores, Optical Music Recognition (OMR) is the field that investigates how to computationally read music notation from these documents and to store them in a digital structured format [4]; on the other hand, when considering acoustic music signals, Automatic Music Transcription (AMT) represents the field that researches on the design of computational algorithms to transcribe them into some form of structured digital music notation [5].

Nevertheless, despite pursuing the same goal, these two fields have historically worked in a disjoint manner due to the different nature of the source data, either scores or acoustic pieces. This is not unusual since, in non-musical scenarios, the equivalent fields of Text Recognition (TR) and Automatic Speech Recognition (ASR) have traditionally worked in a separate manner as well. However, recent research works have initially explored the possibility of combining both TR and ASR [6], with successful results. Hence, if applicable to text and speech, could music transcription benefit from such multimodal approach? That is, assuming that we have both a score and a recording of a performance of a given music composition, is it possible to combine the individual OMR and AMT systems to obtain a digital transcription of the piece which benefits from the particular advantages of each method?

Under this premise, one might argue whether it would be possible, or even practical, to have both the acoustic and score representations of the piece to be transcribed. This same point is posed for TR and ASR and is justified in that, considering the availability of a written text, producing a uttering of that text requires less effort than manually transcribing the text or correcting the errors produced by the TR system [7]. In a similar manner, we assume that for a certain musician it would be, at least, more appealing to play a composition reading a music sheet rather than manually transcribing it.

This work, therefore, aims at exploring, as a proof of concept, whether the transcription results of a multimodal combination of sheet scores and acoustic performances of music pieces improves those of the stand-alone modalities. For that, we consider a basic fusion policy based on the combination of the most probable hypotheses depicted by each source of data (prediction-level fusion) for monophonic compositions considering end-to-end OMR and AMT systems. The results obtained prove that such combination improves the transcription capabilities of the stand-alone systems, thus validating our initial premise and opening opportunities for further explorations aligned to fusion strategies in the context at issue.

The rest of the paper is structured as follows: Section II introduces the proof-of-concept methodology proposed; Section III presents the experimental set-up considered as well as the results obtained and a discussion about them; finally, Section IV concludes the work and poses future work to tackle.

II. Methodology

As commented, we are considering two end-to-end transcription systems as the base methods for validating our multimodal prediction-level combination proposal. To properly describe these methods and design principles, we shall introduce some notation.

Formally, let $\mathcal{T} = \{(x_i, \mathbf{z}_i) : x_i \in \mathcal{X},\ \mathbf{z}_i \in \mathcal{Z}\}_{i=1}^{|\mathcal{T}|}$ represent a set of data where signal x_i drawn from space $\mathcal{X}$ is related to a sequence of symbols $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iN_i})$ from space $\mathcal{Z}$ considering the underlying function $g : \mathcal{X} \rightarrow \mathcal{Z}$. Note that this latter space is defined as $\mathcal{Z} = \Sigma^*$ where Σ represents the symbol vocabulary.

Since we are dealing with two sources of information, we have different representation spaces $\mathcal{X}^s$ and $\mathcal{X}^a$ with vocabularies Σ^s and Σ^a related to the staff images and audio signals, respectively. While not strictly necessary, for simplicity we are constraining both systems to consider the same vocabulary, i.e., $\Sigma^a = \Sigma^s$. Also note that, for a particular i-th element, while staff $x_i^s \in \mathcal{X}^s$ and audio $x_i^a \in \mathcal{X}^a$ signals depict a different origin, the target sequence $\mathbf{z}_i \in \mathcal{Z}$ is deemed to be the same.

A. End-to-end base recognition systems

Concerning the end-to-end neural architectures, we have considered a Convolutional Recurrent Neural Network (CRNN) scheme [8] together with the Connectionist Temporal Classification (CTC) training algorithm [9] to approximate the $g(\cdot)$ underlying function as $\hat{g}(\cdot)$. This network is formed by an initial block of *convolutional* layers devised to learn the adequate features for the particular recognition task followed by another group of *recurrent* stages which model the temporal/spatial dependencies of those features.

As commented, the network is trained using the CTC training function as it allows training the CRNN scheme using unsegmented sequential data. In a practical sense, this method only requires the different input signals to the scheme and their associated sequences of characters as its expected output, without any specific input-output alignment. It must be mentioned that CTC requires the inclusion of an additional "*blank*" symbol within the Σ vocabulary, i.e., $\Sigma' = \Sigma \cup \{blank\}$ due to its particular training procedure. This symbol is used for enabling the detection of consecutive repeated elements.

Since CTC assumes that the architecture contains a fully-connected network of $|\Sigma'|$ outputs with a *softmax* activation, the actual output is a posteriogram with a number of frames given by the recurrent stage with $|\Sigma'|$ tokens each. Most commonly the final prediction is obtained out of this posteriogram using a *greedy* approach which retrieves the most probable symbol per step and a posterior squash function which merges consecutive repeated symbols and removes the *blank* label. In our case, we slightly modify this decoding approach for allowing the multimodal fusion of both sources of information.

B. Fusion policy

As aforementioned, among the different combination possibilities, in this case, we resort to a predication-level policy that allows the individual recognition systems to be trained in an isolated manner.

The proposed policy takes as starting point the posteriograms of the two recognition modalities, OMR and AMT. For each posteriogram, a greedy decoding policy is applied to each of them for obtaining their most probable symbols per frame together with their per-symbol probabilities.

After that, the CTC squash function merges consecutive symbols for each modality with the particularity of deriving the per-symbol probability by averaging the individual probability values of the merged symbols. For example, when any of the models obtains a sequence in which it predicts the same symbol for 4 consecutive frames, the algorithm combines them and computes the average probabilities of these involved frames. Note that the *blank* symbols estimated by CTC are also removed.

Given that the resulting sequences for each modality may not match in terms of length, it is necessary to align both estimations for properly merging them. In this regard, we make use of the Smith-Waterman (SW) local alignment algorithm [10] which performs a search for the most similar regions between pairs of sequences.

Eventually, the final estimation is obtained from these two aligned sequences following these premises: (i) if both sequences match on a token, it is included in the resulting estimation; (ii) if the sequences disagree on a token, the one with the highest probability is included in the estimation; (iii) if one of the sequences poses a *blank* symbol, that of the other sequence is included in the estimation.

III. Experimentation

This section presents the experimental part of the work. For that, we introduce the particular CRNN schemes considered for the OMR and AMT recognition systems, we describe the corpus and metrics considered, and finally we present and discuss the results obtained.

A. CRNN models

The different CRNN topologies considered for both the OMR and the AMT systems are described in Table I. Note that, as aforementioned, the last recurrent layer of the schemes is connected to a dense unit with $|\Sigma^a| + 1 = |\Sigma^s| + 1$ output neurons and a softmax activation.

B. Materials

For the evaluation of our approach, we considered the Camera-based Printed Images of Music Staves (Camera-PrIMuS) database [11]. This corpus contains 87,678 real music staves of monophonic incipits[1] extracted from the *Répertoire International des Sources Musicales* (RISM). For each incipit, different representations are provided: an image with the rendered score (both plain and with artificial distortions), several encoding formats for the symbol information, and a MIDI file of the content.

Regarding the particular type of data used by each recognition model, the OMR system takes as input the artificially distorted staff image of the incipit scaled to a height of 64 pixels, maintaining the aspect ratio. Regarding the AMT model, an audio file is synthesized from the MIDI file for each incipit with the FluidSynth software[2] and a piano timbre considering a sampling rate of 22,050 Hz; then a time-frequency representation is obtained by means of the Constant-Q Transform with a hop length of 512 samples, 120 bins, and 24 bins per octave. This result is embedded as an image whose height is scaled to 256 pixels, maintaining the aspect ratio.

[1] Short sequence of notes, typically the first measures of the piece, used for indexing and identifying a melody or musical work.

[2] https://www.fluidsynth.org/

TABLE I
CRNN CONFIGURATIONS CONSIDERED. NOTATION: CONV$(f, w \times h)$ STANDS FOR A CONVOLUTION LAYER OF f FILTERS OF SIZE $w \times h$ PIXELS, BATCHNORM PERFORMS THE NORMALIZATION OF THE BATCH, LEAKYRELU(α) REPRESENTS A LEAKY RECTIFIED LINEAR UNIT ACTIVATION WITH NEGATIVE SLOPE VALUE OF α, MAXPOOL2D$(w_p \times h_p)$ STANDS FOR THE MAX-POOLING OPERATOR OF DIMENSIONS $w_p \times h_p$ PIXELS, BLSTM(n) DENOTES A BIDIRECTIONAL LONG SHORT-TERM MEMORY UNIT WITH n NEURONS, AND DROPOUT(d) PERFORMS THE DROPOUT OPERATION WITH d PROBABILITY.

Model	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6
OMR	Conv(64, 5 × 5) BatchNorm LeakyReLU(0.20) MaxPool(2 × 2)	Conv(64, 5 × 5) BatchNorm LeakyReLU(0.20) MaxPool(1 × 2)	Conv(128, 3 × 3) BatchNorm LeakyReLU(0.20) MaxPool(1 × 2)	Conv(128, 3 × 3) BatchNorm LeakyReLU(0.20) MaxPool(1 × 2)	BLSTM(256) Dropout(0.50)	BLSTM(256) Dropout(0.50)
AMT	Conv(8, 2 × 10) BatchNorm LeakyReLU(0.20) MaxPool(2 × 2)	Conv(8, 5 × 8) BatchNorm LeakyReLU(0.20) MaxPool(1 × 2)	BLSTM(256) Dropout(0.50)	BLSTM(256) Dropout(0.50)		

An initial data curation process was applied to the corpus for discarding samples which may cause a conflict in the combination, resulting in 67,000 incipits. Since this reduced set still contains a considerably large amount of elements, we constrained our experiments to approximately a third of this curated set, which results randomly selected set of 22,285 incipits with a label space of $|\Sigma^a| = |\Sigma^s| = 1,180$ tokens.

It must be pointed out that, in our experiments, this latter set was exclusively used for training the AMT system while the OMR considered a subset of it. The reason for such set-up is that, in preliminary experimentation, the OMR system remarkably outperformed the AMT one, thus hindering the possible improvement of the multimodal proposal as the latter recognition model rarely corrected any flaw of the former one. In this regard, since we want to determine whether OMR and AMT may collaborate in the transcription process in a synergistic manner, we reduced the training set of the OMR system not to eclipse the possible contribution of AMT to the combined result. On this subject, Table II summarizes the details of the data considered for each modality and partition.

TABLE II
NUMBER OF INCIPITS CONSIDERED FOR EACH MODALITY AND PARTITION.

Modality	Train	Validation	Test
OMR (Image)	802	4,457	4,457
AMT (Audio)	13,371	4,457	4,457

Finally, regarding the performance evaluation, we considered the Symbol Error Rate (Sym-ER) as in other neural-based transcription systems. This measure is defined as:

$$\text{SER}\ (\%) = \frac{\sum_{i=1}^{|\mathcal{S}|} \text{ED}\left(\mathbf{z}_i,\ \mathbf{z'}_i\right)}{\sum_{i=1}^{|\mathcal{S}|} |\mathbf{z}_i|} \tag{1}$$

where $\text{ED}\left(\cdot,\cdot\right)$ stands for the string Edit distance [12], $\mathcal{S}$ a set of test data, and $\mathbf{z}_i$ and $\mathbf{z'}_i$ the target and estimated sequences, respectively.

Using the same definitions, for a more in-depth analysis we also considered the Sequence Error Rate (Seq-ER) which estimates the error as the ratio of erroneous sequence and the total number of predictions. Mathematically it is defined as:

$$\text{Seq-ER}\ (\%) = \frac{|\{x \in \mathcal{S} : g\left(x\right) \neq \hat{g}\left(x\right)\}|}{|\mathcal{S}|} \tag{2}$$

C. Results

The results obtained with the experimental set-up considered for the AMT and OMR systems as well as the presented fusion policy are depicted in Table III. It must be pointed out that these results constitute the ones achieved after optimizing the alignment parameters of the SW algorithm on the validation partition considering the Sym-ER metric as reference.

TABLE III
SYMBOL AND SEQUENCE ERROR RATE RESULTS FOR THE OMR, AMT, AND FUSION POLICY FOR THE TEST PARTITION CONSIDERED.

Metric	OMR	AMT	Fusion
Sym-ER (%)	14.29	27.53	12.95
Seq-ER (%)	76.67	99.15	84.92

As it can be observed, the stand-alone OMR method consistently outperforms the AMT one, achieving the former system a Sym-ER figure approximately 13% lower than that of the latter. In this context, one could argue that combining the outputs of these two systems may report an improvement due to being the OMR system considerably more robust than the AMT one. Nevertheless, the fusion method is able to decrease the error rate obtained by the best transcription model when the alignment method is properly adjusted. More precisely, the fusion method achieves a Sym-ER 1.4% lower than that of the OMR model, i.e. the error is reduced over 9.4%. Regarding the Seq-ER results, despite the fact that these figures do not show the same improvement as in the other metric, it should be noted that the experimentation was optimized considering the Sym-ER one.

While the previous Sym-ER figures prove the validity of the fusion proposal, we may further analyze the results to analyze the error typology by each method as well as the incorrect hypotheses the fusion policy is able to correct. For that Table IV shows an example of the results obtained for a given incipit with the stand-alone OMR and AMT systems

TABLE IV
EXAMPLE OF THE MULTIMODAL FUSION ON A MUSIC INCIPIT. THE OMR AND AMT COLUMNS DEPICT THE ESTIMATED SEQUENCES BY THE STAND-ALONE SYSTEMS WHILE THE FUSION ONE SHOWS THE COMBINED ESTIMATION. THE GROUND-TRUTH TRANSCRIPTION IS ALSO PROVIDED. DISAGREEMENTS BETWEEN MODALITIES ARE HIGHLIGHTED IN BLUE.

OMR	**AMT**	**Fusion**	**Ground Truth**
clef-G2	clef-C1	clef-G2	clef-G2
keySignature-FM	-	keySignature-FM	keySignature-FM
timeSignature-C	timeSignature-C	timeSignature-C	timeSignature-C
rest-half	rest-half	rest-half	rest-half
note-A4_eighth	note-A4_eighth	note-A4_eighth	note-A4_eighth
note-D5_eighth	note-D5_eighth	note-D5_eighth	note-D5_eighth
note-D5_sixteenth	note-D5_sixteenth	note-D5_sixteenth	note-D5_sixteenth
note-C5_sixteenth	note-C#5_sixteenth	note-C#5_sixteenth	note-C#5_sixteenth
note-D5_sixteenth	note-D5_sixteenth	note-D5_sixteenth	note-D5_sixteenth
note-E5_sixteenth	note-E5_sixteenth	note-E5_sixteenth	note-E5_sixteenth
barline	barline	barline	barline
note-F5_eighth	note-F5_eighth	note-F5_eighth	note-F5_eighth
note-D5_eighth	note-D5_eighth	note-D5_eighth	note-D5_eighth
rest-eighth	rest-eighth	rest-eighth	rest-eighth
note-C5_eighth	note-C#5_eighth	note-C#5_eighth	note-C#5_eighth
note-D5_eighth	note-D5_eighth	note-D5_eighth	note-D5_eighth

as well with the multimodal fusion proposed. The reference transcription is also provided for a proper analysis.

It is observed that, for this particular case, there is a strong agreement between the OMR and AMT modalities, being only four cases in which the two sequences estimate different labels: one related to the clef, another one for the key signature, and the remaining related to actual music notes. We shall now examine how these conflicts are solved by the merging policy.

Focusing on the clef and key errors, note that the devised fusion policy estimates the correct labels to be the ones by the OMR recognition system. Given that this disagreement is solved, on a broad sense, by taking the token with a superior probability among the different modalities, it is possible to affirm that the OMR performs better on this particular information than the AMT system. This conclusion is somehow expected since these two pieces of information are explicitly drawn in the score while, for the case of audio data, this information must be inferred.

On the other hand, the errors present in the notes of the piece are better estimated by the AMT system rather than the OMR one. Again, this behaviour is totally coherent since, while the note information is explicitly present in the audio data, in scores some information is elided due to the graphical representation rules. As an example, if the music piece depicts pitch alterations (sharp and/or flat notes), this information is explicitly engraved in the key signature of the piece and not represented in the notes to be recognized; oppositely, audio data directly contains the note with its possible alteration.

Finally, it must be highlighted that the improvement over a 9.4% of the Sym-ER metric supports the initial hypothesis that the multimodal combination of OMR and AMT technologies may enhance that of stand-alone systems, being, hence, worthwhile studying this new paradigm for transcription tasks. Note that this work constitutes a proof-of-concept research piece meant to validate the commented hypothesis.

IV. CONCLUSIONS

Music transcription, understood as obtaining a structured digital representation of the content of a given music source, is deemed as key challenge in the Music Information Retrieval field for its applicability in tasks such as music heritage preservation, dissemination, and analysis, among others.

This work presents a proposal that combines the predictions depicted by a couple of end-to-end Automatic Music Transcription and Optical Music Recognition systems over a set of monophonic music data considering a local alignment approach. This proof-of-concept experience validates the initial hypothesis that the multimodal combination of these two sources of information is capable of retrieving an improved transcription result.

In light of these results, different research paths may be now explored to further improve the results obtained. The first one is the actual combination of the hypotheses by the individual systems on a probabilistic framework, such as that of *word graphs* or *confusion networks*. Besides, while these proposals work on a prediction-level, it may be also explored the case in which this combination is previous stages in the general pipeline as, for instance, the feature extraction one. Finally, in a more practical sense, experimentation may be also extended to more challenging data as, for instance, handwritten scores, audio recordings of different instrumentation, or even polyphonic music.

ACKNOWLEDGMENT

This work is supported by the Spanish “Ministerio de Educación y Formación Profesional” through grant 20CO1/000966 and the “Programa I+D+i de la Generalitat Valenciana” through grants ACIF/2019/042 and APOSTD/2020/256.

REFERENCES

[1] X. Serra, M. Magas, E. Benetos, M. Chudy, S. Dixon, A. Flexer, E. Gómez Gutiérrez, F. Gouyon, P. Herrera, S. Jordà, et al. *Roadmap for music information research*. The MIReS Consortium, 2013.

[2] 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 *1st International Workshop on Reading Music Systems*, pages 17–18, 2018.

[3] M. Schedl, E. Gómez, and J. Urbano. Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, 8:127–261, 01 2014.

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

[5] E. Benetos, S. Dixon, Z. Duan, and S. Ewert. Automatic music transcription: An overview. *IEEE Signal Processing Magazine*, 36(1):20–30, 2018.

[6] A. Singh, A. Sangwan, and J. H. L. Hansen. Improved parcel sorting by combining automatic speech and character recognition. In *2012 IEEE International Conference on Emerging Signal Processing Applications*, pages 52–55, 2012.

[7] E. Granell and C.-D. Martínez-Hinarejos. Multimodal output combination for transcribing historical handwritten documents. In *International Conference on Computer Analysis of Images and Patterns*, pages 246–260. Springer, 2015.

[8] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2017.

[9] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 369–376, New York, NY, USA, 2006. ACM.

[10] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[11] J. Calvo-Zaragoza and D. Rizo. Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 248–255, Paris, France, September 2018.

[12] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710, 1966.

Hybrid Annotation Systems for Music Transcription

Ioannis Petros Samiotis*, Christoph Lofi†, and Alessandro Bozzon‡
Delft University of Technology
Email: *i.p.samiotis@tudelft.nl, †c.lofi@tudelft.nl, ‡a.bozzon@tudelft.nl

***Abstract*—Automated methods and human annotation are being extensively utilized to scale up knowledge-intensive information systems. However, tasks such as music transcription are still challenging due to the complexity of the domain and the expertise needed to read and process music scores. In this work, we examine how music transcription could benefit from hybrid annotation workflows combining automated AI methods with crowdsourcing. We show that through careful task and interaction design utilizing microtask crowdsourcing principles, a non-specialist crowd can meaningfully contribute to such hybrid transcription systems despite the complexity of the domain.**

***Index Terms*—crowd computing, crowdsourcing, music transcription, hybrid annotation systems**

I. Introduction

Traditionally, digital music transcription involves highly trained experts who understand music structures and notations. They also know how to use specialised software tools and have the ability to identify and fix errors of previous editions. Through Optical Music Recognition (OMR), researchers try to automate this process (or parts of it) with several processing steps such as score segmentation, symbol recognition and semantic reconstruction of a scanned music score.

State-of-the-art methods show acceptable performance in the case of clean music scores, but their quality quickly degrades in case of hand-written notes [1]. In general, they still require substantial human intervention to provide results with consistent quality [1], [2], while interactive systems that could utilize human evaluation in an efficient and scalable way are still an open issue [3].

Microtask crowdsourcing is a popular approach for scaling up digital content annotation tasks. On online microtask crowdsourcing platforms, such as Amazon Mechanical Turk, large groups of individuals (workers) perform *microtasks* like image categorization, and audio or text transcription. By splitting a complex and cognitively intensive task into simpler steps, *microtasks* crowdsourcing allows people with little to no expertise to contribute to knowledge-intensive activities [4].

The effectiveness and efficiency of the results provided by the crowd is strongly influenced by the quality of the microtask design [5], [6]. Few studies addressed the use of crowdsourcing for music scores transcription, and they typically focus the transcription of whole scores [7] instead of microtasks, or rely on experts [8], [9]. To the best of our knowledge, how to address the task of score transcription through microtask crowdsourcing remains an open research question [10].

In this paper we build upon our preliminary work [11] which shows the general feasibility of microtask crowdsourcing for error detection in music transcription. In this work, we now showcase how hybrid annotation systems that utilize both OMR processes and microtask crowdsourcing could be designed from a worker-data interaction point-of-view, and we discuss the feasibility of different approaches.

II. Related work

The topic of microtask crowdsourcing for music transcription is scarcely addressed in literature, with many relevant research questions left unanswered. In Burghardt et al. [7] the *Allegro* system was developed, a tool to allow the transcription of entire scores by a (single) human worker. However, *Allegro* has only been tested on a limited number of users, and it was not deployed on an online microtask crowdsourcing platform. The same limitation holds for the work in [8], one of the first attempts to study human input and how the task design can affect human input. This study focused on analysing segments which are one measure long, which is the smallest unit of analysis in our study as well. We expand this by studying also how the size of the segment shown to the crowd affect performance. OpenScore [12], up to now the largest-scale project to incorporate humans in music score transcription, is mainly carried out by seven community members with extensive musical background. Moreover they report different issues related to the management of data (done manually by the administrators of the platform) and user engagement (without any control they would focus on their preferred music score) admitting in the end that in their project "OMR (involving humans) is not currently a scalable solution".

So far, there is no literature that has targeted unknown crowds with varying skills for music transcription tasks, thus research questions on [10] what type of tasks users can perform and how to evaluate them still remain open. In this work we address this research gap by looking into similar crowdsourcing works in other domains. More specifically, in [13] it was found that for knowledge-intensive tasks involving artworks, a crowd with varying and unknown domain-specific knowledge found on online platforms can produce useful annotations when aided by good task design.

Research has shown that UI design is an important part of a microtask design [14]. Experiments with various designs such as spectogram visualisations for audio annotation [15] or the use of chat-bots to assist common types of microtasks [16] have yielded positive results on the worker performance.

III. Hybrid Music Transcription Workflows

In a hybrid annotation workflow, the goal is to effectively and efficiently combine automated methods with human work to achieve a result that couldn't be attained by either approach alone [17]. Individual steps of the workflow need to be identified early, and through careful design be allocated to the appropriate processing method. Automated methods and human input need to co-exist and complement each other. The complexity and niche knowledge required by a person to transcribe a music score but also the shortcomings of current automated OMR methods need to factored in. Specifically, hybrid annotation workflows cover three main components: Algorithms / Machine Learning, the Crowd, and a Quality Assessment mechanism.

A. Hybrid Workflow Patterns: Main Components

Algorithms / Machine Learning: A set of algorithms, typically machine learning algorithms, which process the input data into the desired output data. These algorithms typically have at least one of the following two shortcomings: 1) The results produced by the algorithms are of bad quality and insufficient for the desired use. 2) The algorithm relies on extensive training which is typically not or only partially available.

The first problem can manifest in different ways [17]. For instance, the algorithm could be good enough in most cases, but might fail in others. It would be necessary to identify when the algorithm fails (automatically or using crowdsourcing) and revert to using crowdsourcing fully redoing the failed task in these cases. The worst case would be a scenario in which the algorithm fails always, resulting in a pure crowdsourcing system. In another scenario, the algorithms generally works on all types of input data, but the output quality is slightly too low in nearly all cases. Here, all outputs need to be adjusted and fixed using crowdsourcing. The second problem requires the creation of training data that usually covers a large number of examples of correct input data / desired output data pairs. Crowdworkers can provide such training pairs upfront for initial training, or as part of the fixing measures introduced for the first problem. Then, this crowd-provided data can be used for incremental re-training.

The Crowd: The crowd can be used to execute cognitive Human Intelligence Tasks. The choice of crowdworkers and their incentivisation is a core challenge not addressed in this work, but could range from paying microtask workers on platforms like Amazon Mechanical Turk to motivating expert online communities using intrinsic incentives. In general, the crowd can be used to: 1) Check the correctness of an intermediate algorithm result. This can range from simple correct / incorrect checks to more complex checks which give a detailed overview of the location and nature of the error. 2) Produce results: Here, the crowd is used to perform the same task the algorithm was designed for: transform a given input data instance into the correct output data. This functionality is employed when an algorithm failed to process, or when that data is required for further/initial training. 3) Improve results: Here, a machine-produced result with sub-par quality is manually improved. Typically, this should be employed when improving slightly faulty outputs is easier and cheaper than creating a new output manually from scratch.

Quality Assessment: This is a core component to ensure the effectiveness of a good hybrid crowdsourcing process. The quality assessment is central in both judging the quality of algorithmic results in order to decide if and what kind of crowd treatment is needed, but also for judging the reliability and quality of crowd feedback in light of low-skill and/or malicious workers.

B. OMR Processes and Challenges

To better understand how hybrid annotation patterns could transfer to music transcription, we first identify OMR processes that are being used [18] with respect to the following three main categories:

- Image pre-processing
- Music symbol segmentation and recognition
- Semantic Reconstruction

During the image pre-processing, different techniques are applied to scanned images for reducing the computational cost and making the next OMR steps more efficient. One of the most important methods of image preprocessing is "Binarization" which is the process of converting the pixel image into a binary image (black and white), separating the foreground from the background. This is a common step for most of the OMR tools. Binarization eases the OMR tasks by reducing the amount of information the following steps need to process. For example, it is easier to detect a music symbol in a binary image than in a color image. However, binarization can also pollute the image, loosing relevant details [18]. Many music scores are ancient documents in poor condition due to paper degradation (yellowing, mold, and mildew, etc.), and this often introduces noise on the image, reducing the quality of the OMR tool output. Therefore, working with old music score sheet requires a specialized algorithm for image-cleaning and binarization to reduce the aforementioned problems [19].

Music symbol segmentation is the process of locating and isolating the music object. The main objective is to find the correct position of each symbol to be identified in the next OMR step. This is one of the most challenging OMR steps and highly error-prone. Most symbols on a music score are connected by staff lines. In order to isolate those symbols, staff lines must be detected and removed. Accurate staff line removal is challenging because symbols and staff lines have to be disconnected without removing pixels belonging to the symbols. Unfortunately, staff lines are not always perfectly horizontal, knowing the exact location of the staff line is required. This procedure can be even more complex due to low image quality (paper degradation, stains, etc), and zones with a high density of symbols [20]. After the segmentation stage, the segmented symbols need to be recognized and classified into predefined groups, such as notes, rests, accidentals, clefs, etc. Symbol identification is a hard task because of symbol variability. Each symbol can have different variations due to different score editors or the continuous evolution of music notation over time. However, variability can be also observed in the same music score, making it even more difficult to symbol ambiguity. In addition, the previous segmentation step may have cut or degraded the objects [20].

The last stage of the OMR framework is to reconstruct the music semantics from the recognized symbols, combining them the staff system to reproduce the meaning of the scanned music [18]. Unlike optical character recognition (OCR) which is predominantly one-dimensional and which can also rely on strong language pattern heuristics, OMR tools require an interpretation of two-dimensional relationships between music objects. As a consequence many errors may occur due to a symbol placed in a wrong position. For example, a slur symbol is a curved line generally located over the notes. If a slur is placed in a wrong position, it leads to a misinterpretation of the music score. Likewise, a dot has different meaning depending on where it is located (e.g., on top vs on the side). The last step of OMR systems is to export the final score into a machine-readable format. Several formats have been developed, such as MIDI (Musical Instrument Digital Interface), MusicXML, MEI (Music Encoding Initiative), NIFF (Notation Information File Format), etc. Generally, each tool has its own (set of) output formats. This lack of a commonly accepted representation imposes an obstacle for OMR tool assessment [21].

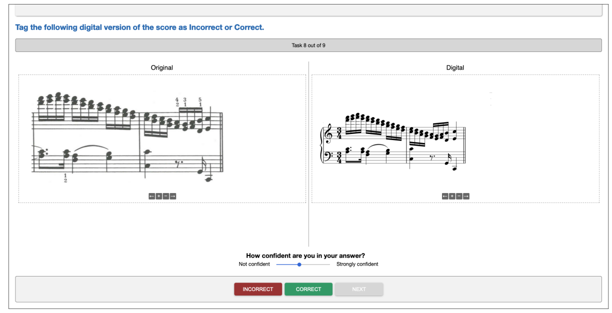
IV. Case Study: Error Detection

We will showcase the experiment conducted in [11]. Through that preliminary work, we researched how microtask crowdsourcing for music transcription can be implemented. In an hybrid annotation system, such workflows could fit during the training or evaluation step of an OMR algorithm. The main focus of that work was to study to what extent a general crowd can identify *errors* in a music score transcription (see section III-A). The experiment aimed at testing the ability of crowd workers to spot errors using interfaces having a combination of visual and audio components.

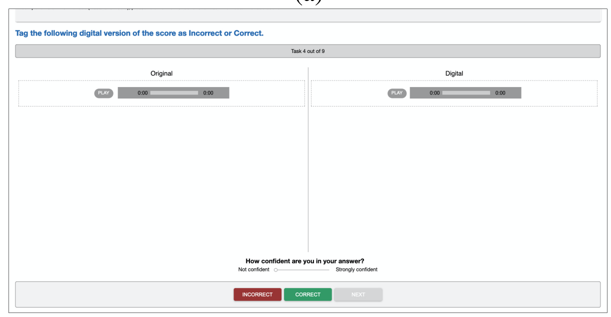
A. Task Design

This study aimed on how task design factors can influence the crowdworker performance, focusing on two aspects:

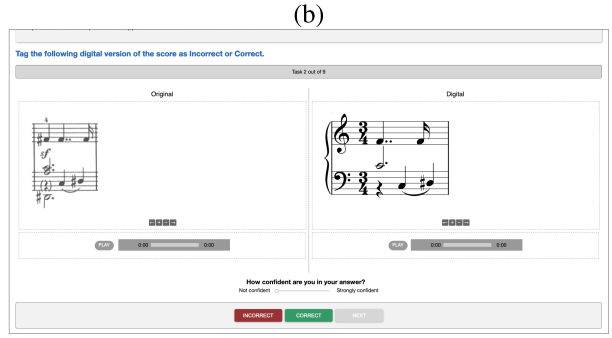
1) The *modality* (*visual* versus *audio*) used to spot errors: as music scores are complex artefacts, and music is primarily an auditory experience. Therefore, it was investigated how the score comparison *modality* affect the error detection performance in workers that are potentially not familiar with musical notation. Intuitively, the interest was if "hearing" errors is easier that "seeing" errors.
2) The score *size* offered to crowdworkers for annotation. The goal was to assess how the size (in terms of measures) of the score offered to worker affects their performance.



(a)



(b)



(c)

Fig. 1. Microtask User Interfaces: (a) Visual, (b) Audio and (c) Combination

B. Dataset Creation

A single classical music score was used to avoid additional confounding variables. The study used the Urtext of "*32 Variations in C minor*" by Ludwig van Beethoven. It is a piano piece and the music artifacts are all printed typeset forms. This is a slightly easier use case than hand-written scores. The score was retrieved from IMSLP as a PDF[1].

As a Gold standard transcription of that PDF we used an MEI[2] file that had been transcribed by an expert. This file was accepted as error free, and it allowed errors to be introduced in a controlled way for the experiments.

The music score was segmented in varying sizes to investigate how workers cope with shorter or longer tasks. We distinguish 1) *one measure* segments, 2) segments of *two measures* and 3) segments of *three measures*. Both digital

[1] https://imslp.org/wiki/32_Variations_in_C_minor%2C_WoO_80_(Beethoven%2C_Ludwig_van)

[2] https://music-encoding.org/

versions of the score (scanned PDF and the transcribed MEI file) were segmented using the aforementioned segment sizes.

The errors that were introduced to the MEI segments were derived from common errors that can occur in automatic OMR systems. The type of errors could impact the crowdworkers' ability to correctly identify them. Therefore, different types of errors focusing on the music notes and their accidentals were studied. Errors on performance annotations, clefs, finger numbers etc, were out of scope. The following types of error were introduced per MEI segment: 1) *Missing notes*; 2) *Wrong vertical position of a note*; 3) *Wrong duration of a note*; 4) *Wrong accidental*.

C. User Interface Design

These design considerations resulted in the following three interface designs. Each combination of interface with a segment size consists of a microtask:

- **Original Score** against **Correct/Incorrect MEI Render (Visual)**: This user interface, depicted in Figure 1(a), shows the segment of the original scanned score to the left, with the corresponding MEI render to the right. The user needs to compare the two images and spot differences related to the types of errors.
- **Correct MIDI** against **Correct/Incorrect MIDI (Audio)**: In this interface, as shown in Figure 1(b), we let the user listen to the correct MIDI extract on the left and the one generated from the MEI transcription to the right.
- **Original Score and Correct MIDI** against **Correct/Incorrect MEI and Correct/Incorrect MIDI (Combination)**: This final user interface, as shown in Figure 1(c), combines elements of the previous two. The user here has the option to either use the visual comparison, the audio comparison, or both to identify errors. The MEI render and MIDI extraction always originate from the same MEI transcription, therefore both will be either correct or not.

D. Results

In total, 144 workers executed our tasks on MTurk and we paid them according to the average US minimal hourly wage[3]. In order to minimize the effect of any biases or learning effect we randomized the order of the presentation of the different task designs (UI-segment size combination). One worker eluded the quality verification on task interface, which results in 143 unique workers.

As expected, people with some formal knowledge in music are very rare "in the wild". To still allow for microtask crowdsourcing, good task design is therefore of essence. We refer to [11] for more detailed analysis, but overall the results show that error detection is a task that could be successfully performed in a microtask crowdsourcing setting. Offering audio extracts of a target music score can positively affect the performance of the crowdworkers, especially for short segments of one or two measures. With larger segments, even though audio extracts are still yielding better results against to the textual measures of the score, a combination of the two modalities is more preferable. This result gives important indications for task splitting and scheduling purposes, as it suggests that it is possible to evaluate larger portions of scores without incurring accuracy penalties. This has strong implications in terms of overall transcription costs.

[3]We estimated an average task completion time of 15'; each crowdworker was awarded 2.5$ per task

V. Conclusion

Crowdsourcing and human computation are powerful tools which can be integrated in a data processing pipeline or information system to handle processing tasks which cannot easily be covered by current algorithmic approaches due to the involved semantic complexity. However, crowdsourcing is expensive: workers need to be incentivised (often with monetary incentives, or carefully engineered social incentives), and human work is of course often slower than automated algorithms. This gives a strong argument to strive for hybrid crowdsourcing workflows, where algorithms and humans work hand in hand. Such systems get the best of both worlds: the efficiency of algorithms and the cognitive power and insight of humans.

References

[1] B. Almeida and S. 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*, vol. 25, no. 23. New York, New York, USA: ACM Press, 2017, pp. 15–20. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3078081.3078101

[2] P. Bellini, I. Bruno, and P. Nesi, "Assessing Optical Music Recognition Tools," *Computer Music Journal*, vol. 31, no. 1, pp. 68–93, mar 2007. [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/comj.2007.31.1.68

[3] 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

[4] J. Oosterman, J. Yang, A. Bozzon, L. Aroyo, and G.-J. Houben, "On the impact of knowledge extraction and aggregation on crowdsourced annotation of visual artworks," *Computer Networks*, vol. 90, pp. 133 – 149, 2015, crowdsourcing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128615002315

[5] E. Law and L. v. Ahn, "Human computation," *Synthesis lectures on artificial intelligence and machine learning*, vol. 5, no. 3, pp. 1–121, 2011.

[6] A. Bozzon, M. Brambilla, S. Ceri, and A. Mauri, "Reactive crowdsourcing," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 153–164.

[7] M. Burghardt and S. 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*, ser. DATeCH2017. New York, NY, USA: ACM, 2017, pp. 15–20. [Online]. Available: http://doi.acm.org/10.1145/3078081.3078101

[8] L. Chen and C. Raphael, "Human-Directed Optical Music Recognition," *Electronic Imaging*, vol. 2016, no. 17, pp. 1–9, feb 2017. [Online]. Available: http://www.ingentaconnect.com/content/10.2352/ISSN.2470-1173.2016.17.DRR-053

[9] L. Chen, R. Jin, and C. Raphael, "Human-Guided Recognition of Music Score Images," in *Proceedings of the 4th International Workshop on Digital Libraries for Musicology - DLfM '17*. New York, New York, USA: ACM Press, 2017, pp. 9–12. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3144749.3144752

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

[11] I. Samiotis, S. Qiu, A. Mauri, C. Liem, C. Lofi, and A. Bozzon, "Microtask crowdsourcing for music score transcriptions: an experiment with error detection," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020.

[12] M. Gotham, P. Jonas, B. Bower, W. Bosworth, D. Rootham, and L. VanHandel, "Scores of scores: an openscore project to encode and share sheet music," in *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, 2018, pp. 87–95.

[13] J. Oosterman, A. Bozzon, G.-J. Houben, A. Nottamkandath, C. Dijkshoorn, L. Aroyo, M. H. Leyssen, and M. C. Traub, "Crowd vs. experts: nichesourcing for knowledge intensive tasks in cultural heritage," in *Proceedings of the 23rd International Conference on World Wide Web*, 2014, pp. 567–568.

[14] U. Gadiraju, A. Checco, N. Gupta, and G. Demartini, "Modus operandi of crowd workers: The invisible role of microtask work environments," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–29, 2017.

[15] M. Cartwright, A. Seals, J. Salamon, A. Williams, S. Mikloska, D. MacConnell, E. Law, J. P. Bello, and O. Nov, "Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations," *Proceedings of the ACM on Human-Computer Interaction*, vol. 1, no. CSCW, pp. 1–21, 2017.

[16] P. Mavridis, O. Huang, S. Qiu, U. Gadiraju, and A. Bozzon, "Chatterbox: Conversational interfaces for microtask crowdsourcing," in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, 2019, pp. 243–251.

[17] C. Lofi and K. El Maarry, "Design patterns for hybrid algorithmic-crowdsourcing workflows," in *2014 IEEE 16th Conference on Business Informatics*, vol. 1. IEEE, 2014, pp. 1–8.

[18] 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, no. 3, pp. 173–190, 2012.

[19] B. Gatos, I. Pratikakis, and S. J. Perantonis, "An adaptive binarization technique for low quality historical documents," in *International Workshop on Document Analysis Systems*. Springer, 2004, pp. 102–113.

[20] F. Rossant and I. Bloch, "Robust and adaptive omr system including fuzzy modeling, fusion of musical rules, and possible error detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–25, 2006.

[21] G. Jones, B. Ong, I. Bruno, and N. Kia, "Optical music imaging: music document digitisation, recognition, evaluation, and restoration," in *Interactive multimedia music technologies*. IGI Global, 2008, pp. 50–79.

Completing Optical Music Recognition with Agnostic Transcription and Machine Translation

Antonio Ríos-Vila, David Rizo, Jorge Calvo-Zaragoza, José M. Iñesta
U.I. for Computing Research, University of Alicante, Spain
{arios,drizo,jcalvo,inesta}@dlsi.ua.es

***Abstract*—Optical Music Recognition (OMR) research has been traditionally addressed from an image recognition perspective, as it is one of the most challenging steps to develop these systems. Despite the advances in the field, there are still open issues to bring OMR results into real-use cases. One of these issues is the final encoding step, where the graph-based outputs of the image recognition stages are converted into a score encoding format, which can be exported to other tools. In this paper, we address the issue from an implementation perspective, where we evaluate the performance of a recognition pipeline that uses three different Machine Translation techniques to perform this encoding step compared to a direct image-to-encoding approach. The results obtained show that the proposed approaches have an interesting accuracy in contexts where training data is scarce, which can be beneficial in scenarios such as music heritage.**

***Index Terms*—Optical Music Recognition, Convolutional Recurrent Neural Networks, Agnostic Encoding, Machine Translation**

I. Introduction

Optical Music Recognition (OMR) [1] is a research field that still has open problems, especially in the most complex contexts.

Recent advances in Machine Learning, namely Deep Learning (DL), have led the research field from dealing with subproblems, to a state in which complete results are attainable [2], [3].

Despite these advances, there are still many issues that hinder taking OMR results from research to practice, such as the development of end-user applications with the appropriate visual interfaces [4], the implementation of proper document indexation and retrieval, or output the recognition results in standard formats. This paper focuses on this last mentioned issue.

Typically, these DL-based approaches cover all the processes that involve the transcription of an input image, which is usually a music staff, into a sequence of recognized symbols. These sequences represent the glyphs and symbol positions in the given score, which are sometimes referred to in the literature as the *agnostic encoding* outputs. However, these graphically-based results cannot be used by an end-user or reproduced in a music editor or visualizer [5], as all of these available applications require standard semantically-based music file formats to work. The last step to achieve this semantic-encoded document from the recognition is named *encoding process* in the OMR pipeline [1], where the graphic-based recognition outputs are converted into a standard semantic encoding.

Unfortunately, this step has hardly been addressed in the DL-based OMR literature, due to the approaches required by the challenges of the previous steps. The few existent works on the topic point to the possibility of using Machine Translation (MT) to solve this issue.

A usual approach found in most commercial systems to convert from agnostic to semantic encoding is to use rule-based translation systems. However, this has been proved to be a challenging task in complex scores, such as polyphonic ones [6], [7]. This approach also presents significant problems both in terms of generalization for different types of notation and scalability, since it requires the design of new rules or systems when the source and/or the target encoding vary. This is hardly maintainable when the music encodings reach a significant size or when there are too many slight differences in the sources (e.g. the different graphical positions of symbols in handwritten manuscripts).

In [8] research was performed on retrieving the music semantics from the agnostic encoding with simple neural MT approaches. Despite proving hat it is possible to use these systems to perform an encoding step, research remains to be done, as the output of the system was not encoded in a standard semantic format. More in-depth research on the topic was performed in [9], where the authors discussed the use of advanced MT techniques to achieve the desired conversion. In that paper, the use of automatic translation systems between music encodings was proved to be feasible and three approaches were proposed to do so: one based in data-driven statistical techniques and two neural implementations.

Despite the advance presented in that paper, it was concluded that those systems had to be tested in a real-case scenario, as the article only proved that it was indeed possible to do the translation process using those tools. In the present paper, we focus on completing the pipeline by evaluating these models in a real-case scenario, where the agnostic output produced by the recognition sequence is thus translated into a standard semantic encoding format.

II. Methodology

In this section, we present and justify both the target semantic encoding used to carry out our experimentation and

This work was supported by the Spanish Ministry HISPAMUS project TIN2017-86576-R partially funded by the EU.

describe the implemented models used to complete the OMR pipeline.

A. Target Encoding Format

One relevant objective of this work is to find a suitable music notation format to be used as the target semantic encoding of the translation process.

The first options that may be considered are the most extended semantic encodings in music information retrieval and musicology contexts: MEI [10] and MusicXML [11], which represent music score components and metadata in XML-based languages. Despite being comprehensive formats, these semantic representations are highly verbose, which is inconvenient in a MT context. This means that the target language would require a huge number of tokens for even small music excerpts, thereby making sequence alignment unnecessarily complicated.

In the previously cited research, the use of *Humdrum **kern* [12] was proposed. This is a robust and widely-used semantic encoding for many musicological projects. Its benefits for our purpose lie in a simple vocabulary, a sequential-based format, and its compatibility with dedicated music software like Verovio Humdrum Viewer [13], which offers the possibility of automatically converting the output sequence into other formats.

For all the above, we selected **kern as our target semantic encoding language. Despite that, Humdrum **kern has some issues representing specific music symbols present in our corpora, such as multi-measure rests. So we use an extension of it, named ***kern**, proposed in [14], which solves these problems, maintains the benefits of the raw encoding, and is able to be directly converted to raw **kern and reverted with simple text processing instructions.

B. OMR pipeline

In this section, we describe the different systems implemented for the experimentation phase. The main idea, represented in Figure 1, is to combine a graphic recognition end-to-end model, which takes a music staff image as input and outputs an agnostic-encoded sequence with the recognized symbols. Then, this output sequence is translated by one of the proposed Machine Translation models into a ***kern** sequence. We implement one pipeline for each different translation system and a Direct Encoding pipeline that acts as a baseline.

1) Graphic recognition: This step is implemented with a state-of-the-art model OMR, which consists of a CRNN model trained with a CTC loss function [15]. The configuration specified in [16] has been followed.

2) Translation: For this step, we considered the three models presented in [9] and evaluated their performance in a real-case scenario. These models are the following:

- Statistical Machine Translation (**SMT**): The SMT approach [17] consists in a data-driven approach to MT in which several independent models are combined to produce a translation from a text in the source language into the target language. This system produces two main models: a Translation Model and a Language Model, optimised with statistical techniques (such as distortion, smoothing and length penalties) that permit the SMT models to produce both accurate and fluent target language predictions.
- Sequence-to-Sequence with Attention Mechanisms (**Seq2Seq-Att**): This neural approach is based on Recurrent Neural Networks (RNN) whose outputs are modelled by Attention matrices. Specifically, we resort to the "Global Attention" strategy proposed by Luong et al. [18], with a scoring method given by the scalar product between the encoder and the decoder outputs.
- The Transformer: This model currently represents the state-of-the-art in Neural MT. This model implements an encoder-decoder architecture, like in the previous approach, which replaces the recurrent layers with Attention-based ones, referred to in the literature as the multi-headed attention (MHA) neurons [19].

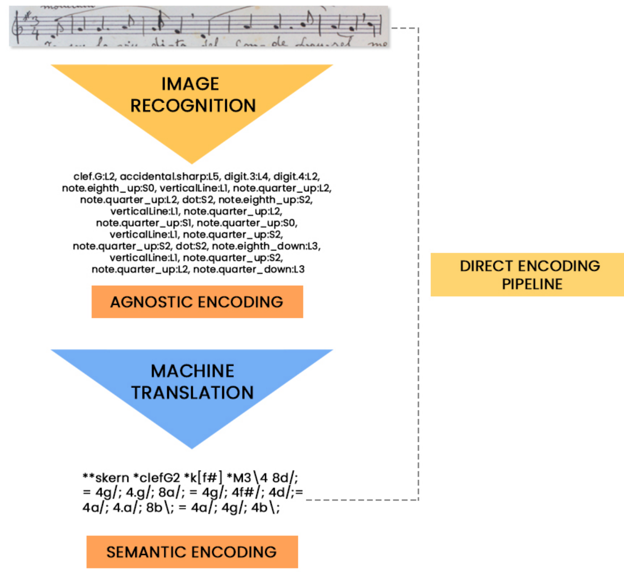


Fig. 1. Overview of the procedures proposed for complete OMR, receiving a staff-section image as input and predicting a semantic music encoding sequence as output.

3) Direct Encoding: For the sake of comparison, we implemented this approach as a simplification of an OMR pipeline, which directly outputs the encoded semantic sequences from the image input, avoiding any translation step. Specifically, we implemented the model presented in [14], which directly outputs the ***kern** representation of the recognized symbols of the analyzed music staff.

This implementation establishes a good comparison baseline, as it is the easiest and simplest model to implement and reduces the number of steps to one.

III. Experimental Setup

In this section, we present and characterize the datasets used to perform the evaluation of the previously proposed pipelines,

TABLE I
DETAILS OF THE CONSIDERED CORPORA.

	PrIMuS	FMT
Engraving	Printed	Handwritten
Size of the corpus (staves)	87,678	872
Agnostic vocabulary size ($\|\Sigma_a\|$)	862	266
Semantic vocabulary size ($\|\Sigma_s\|$)	1,421	206
Running symbols (agnostic)	2,520,245	18,329
Running symbols (semantic)	2,425,355	18,616

as well as present the evaluation metrics used to measure that generalization error when performing the recognition process.

A. Corpora

Two corpora of music score images, with varying features in printing style, have been used to assess and discuss the performance of the different pipelines.

The first considered corpus is the "Printed Images of Music Staves" (PrIMuS) dataset; specifically, the camera-based version [20]. It consists of $87,678$ music incipits[1] from the RISM collection [21]. They consist of music scores in common western modern notation, rendered with Verovio and extended with synthetic distortions to simulate the imperfections that may be introduced by taking pictures of sheet music in a real scenario, such as blurring, low-quality resolutions, and rotations.

The second considered corpus is a collection of four groups of handwritten score sheets of popular Spanish songs taken from the 'Fondo de Música Tradicional IMF-CSIC' (FMT),[2] that is a large set of popular songs manually transcribed by musicologists between 1944 and 1960.

The characterization of these corpora can be found in Table I, while representative examples are shown in Fig. 2 and Fig. 3 for PrIMuS and FMT, respectively, along with agnostic and semantic annotations.

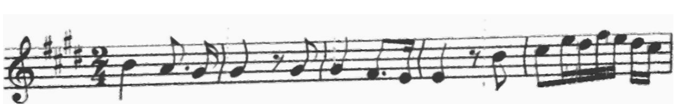


Fig. 2. Example music excerpt from the Camera PrIMUs dataset.

Fig. 3. Example music excerpt from the FMT dataset.

B. Evaluation metrics

To carry the evaluation of the models performance, we used the Sequence Error Rate (SER) metric [22], as it represents accurately the performance of the model in recognition tasks and correlates with the effort a user would have to expend to manually correct the results.

To obtain a more robust approximation of the generalization error, we followed a 5-fold cross-validation process, where the resultant SER is the average of the produced test error within the five data partitions.

IV. RESULTS

The experimentation results are given in Table II, comparing the proposed two-step approaches with the direct encoding approach, that serves as a baseline. We also report the individual results of the two stages involved, in order to provide more insights. In the case of the translation process, the partial results show the SER obtained when a ground-truth agnostic sequence is input for translation.

TABLE II
AVERAGE SER (%) OVER THE TEST SET. ERRORS PRODUCED IN THE RECOGNITION AND TRANSLATION STEPS (TRAINED AND TESTED SEPARATELY) AND THE COMPLETE PIPELINE ERROR RATE (FROM THE INPUT IMAGE TO A SEMANTIC SEQUENCE OUTPUT). THE BEST RESULTS FOR THE COMPLETE PIPELINE ARE HIGHLIGHTED.

	PrIMuS	FMT
Individual step results		
Graphical recognition (CRNN)	3.5	34.9
Translation w/ SMT	23.7	9.6
Translation w/ Seq2Seq-Attn	2.04	9.8
Translation w/ Transformer	0.53	15.4
Complete pipeline		
CRNN + SMT	28.1	42.0
CRNN + Seq2Seq-Attn	**4.3**	**36.8**
CRNN + Transformer	6.4	38.9
CRNN Direct encoding (baseline)	4.7	52.2

Concerning the individual stage results, it can be observed that the graphic recognition step performs well on the printed dataset and gets much worse results in the handwritten one, as might be expected in terms of their training set size and complexity. The Transformer is the best translation option when there is enough training data, while the SMT results are better in the case of limited training data. However, these facts do not extrapolate to the complete process.

If we analyze the complete pipeline, the combination of CRNN and Neural MT models outperform the direct encoding approach, both in the PrIMuS and the FMT dataset. The difference is especially significant in the handwritten corpus, where the difference between the Seq2Seq-Attn model and the baseline is about 20%. One interesting fact from these results is that the Neural MT models are able to deal reasonably well with the inconsistencies introduced during the graphics recognition, as we observe their SER does not increase as much as in the data-driven (SMT) approach.

Furthermore, it is interesting to note that the Transformer is the most accurate NMT model when translating from ground-truth data. However, if we pay attention to the complete

[1] Short sequence of notes, typically the first ones, used for identifying a melody or musical work.

[2] https://musicatradicional.eu

pipeline, it does not produce a model as robust to inconsistencies as the Seq2Seq-Attn one does. This scenario is the most frequent in OMR, where the graphical recognition step tends to make mistakes. Therefore, the Seq2Seq-Attn approach is, as far as our results generalize, the most suitable alternative for the translation process in the two-step pipeline.

Despite of the above evidences, some doubts may arise regarding the error fluctuation between the presented pipelines, as we observed a drastic change in the performance between the two datasets. In order to further analyze the situation, we repeated the same experimentation in reduced versions of the PrIMUS dataset, where we tried to find an intermediate point between FMT and this corpus complexities. The obtained results are reported in Fig. 4. It can be observed that the tendency described from the original PrIMUS results, where the CRNN+Transformer performed the worst, is maintained until dropping to $5,000$ samples, where the direct approach is then outperformed by it. In all cases, however, the CRNN+Seq2Seq-Attn is postulated as the best option by different margins, depending on the complexity of the dataset.

This new experiment summarized the behavior of all alternatives. On the one hand, a direct encoding pipeline —which acted as baseline— depends highly on the amount of training data, attaining competitive results in such case. On the other hand, the two-step process, especially when using the Seq2Seq-Attn as translation mechanisms, clearly represents the best option when training data is limited, also achieving the best performance when the training set is of sufficient size.

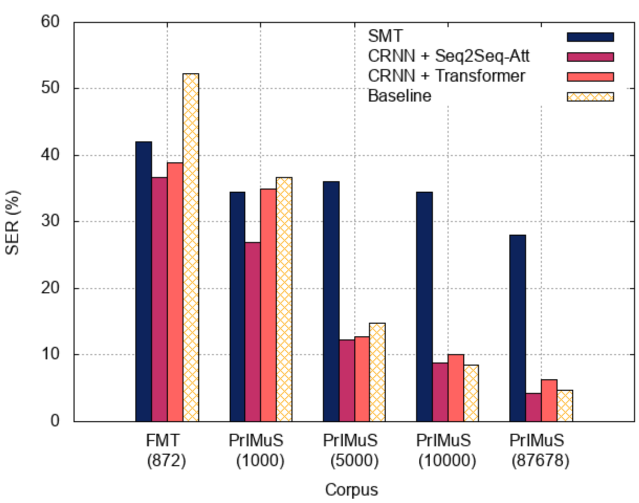


Fig. 4. Graphic bar plot comparison of the average SER produced by the proposed pipelines with the different corpora, which consists in the initially proposed datasets and two reductions on PrIMUS size in order to establish intermediate points between the handwritten and the printed corpus. The Baseline results refer to the Direct Encoding approach described in other sections.

V. Conclusions

In this paper, we studied the implementation of complete two-step OMR pipelines using Agnostic Transcription and Machine Translation. After the experimentation, we observed different aspects about the performance of these approaches on different corpora. We obtained a relevant idea that outlines this work: the two-step pipeline with Neural MT is a good option when the target corpus to digitize does not have enough labelled data.

This solution is interesting in practical scenarios, specifically in the case of early music heritage, as it is common to find situations where manual data labelling is required in order to constitute a corpus before using OMR tools, as there is no need to label a vast amount of data to start using them. Therefore, Neural MT-based OMR pipelines could be considered as an interesting helping tool for corpora labelling, as it could significantly speed up the process. However, the two-step pipeline has a considerable drawback: the corpus has to be labelled in two encoding languages (agnostic and semantic) in order to make it work. Despite this issue, there are possible workarounds, as the translation process does not depend on a specific manuscript; therefore, just one translation model could be used for many cases.

Despite the positive connotation of the paper, we believe that further research is required to maximize the potential benefits of this approach. This includes improving the Neural MT models consistency (especially for the Transformer) with data augmentation, the modelling of cohesive vocabularies to obtain more profit from the encoding models, or the study on how to integrate these systems to produce a single-step OMR pipeline with a dual training process.

References

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

[2] Jorge Calvo-Zaragoza and David Rizo. End-to-end neural optical music recognition of monophonic scores. *Applied Sciences*, 8(4), 2018.

[3] 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)*, pages 205–210, 2020.

[4] 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, page 52–56, New York, NY, USA, 2018. Association for Computing Machinery.

[5] Laurent Pugin, Rodolfo Zitellini, and Perry Roland. Verovio: A library for Engraving MEI Music Notation into SVG. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 107–112. ISMIR, October 2014.

[6] 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.

[7] Jan Hajic and Pavel Pecina. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. *ICDAR*, 2017.

[8] Martha E Thomae, Antonio Ríos Vila, Jorge Calvo-Zaragoza, David Rizo, and José M Iñesta. Retrieving music semantics from optical music recognition by machine translation. 2020.

[9] Antonio Ríos-Vila, Miquel Esplà-Gomis, David Rizo, Pedro J. Ponce de León, and José M. Iñesta. Applying automatic translation for optical music recognition's encoding step. *Applied Sciences*, 11(9), 2021.

[10] Perry Roland. The music encoding initiative (MEI). In *Proceedings of the First International Conference on Musical Applications Using XML*, pages 55–59, 2002.

[11] Michael Good and Geri Actor. Using MusicXML for file interchange. *Web Delivering of Music, International Conference on*, 0:153, 2003.

[12] David Huron. *Humdrum and Kern: Selective Feature Encoding*, page 375–401. MIT Press, Cambridge, MA, USA, 1997.

[13] Craig S. Sapp. Verovio humdrum viewer. *Proceedings of Music Encoding Conference (MEC), Tours, France*, 2017.

[14] Antonio Ríos-Vila, Jorge Calvo-Zaragoza, and David Rizo. Evaluating simultaneous recognition and encoding for optical music recognition. In *7th International Conference on Digital Libraries for Musicology*, pages 10–17, 2020.

[15] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the Twenty-Third International Conference on Machine Learning, (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 369–376, 2006.

[16] 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.

[17] Philipp Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

[18] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421. The Association for Computational Linguistics, 2015.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[20] 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, ISMIR 2018, Paris, France, September 23-27*, pages 248–255, 2018.

[21] Répertoire International des Sources Musicales (RISM) Series A/II: Music manuscripts after 1600 on CD-ROM. Technical report, 2005.

[22] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Citeseer, 2006.

Neural architectures for exploiting the components of Agnostic Notation in Optical Music Recognition

María Alfaro-Contreras, Jose J. Valero-Mas, Jose M. Iñesta
University of Alicante, Spain
{malfaro,jjvalero,inesta}@dlsi.ua.es

***Abstract*—State-of-the-art Optical Music Recognition (OMR) techniques follow a holistic approach, i.e., a sole stage for completely processing a single-staff section image and retrieving the symbols that appear therein. These approaches usually consider an agnostic music representation which characterizes music symbols by their shape and height (vertical position in the staff). Nevertheless, existing approaches neglect this two-sided nature in the learning process since these two features are gathered into single symbols. This work explores the possibility of individually exploiting this particularity of music notation in the context of end-to-end neural OMR approaches at the particular case of staff-line recognition. For that, we consider two Convolutional Recurrent Neural Network (CRNN) schemes trained to simultaneously extract the shape and height pieces of information and propose different policies for eventually merging them at the actual neural level. The results obtained for two corpora of monophonic early music manuscripts prove that our proposal decreases the recognition error in figures ranging between 14.4% and 25.6%, in the best-case scenarios, when compared to the baseline considered.**

***Index Terms*—Optical Music Recognition, Deep Learning, Connectionist Temporal Classification, Agnostic Music Notation, Sequence Labeling**

I. Introduction

Current state-of-the-art Optical Music Recognition (OMR) technologies, which are based on Convolutional Recurrent Neural Networks (CRNN), typically follow an end-to-end or holistic approach that operates at the staff level: they map the series of symbols that appear in an image of a single staff to a sequence of music symbol labels. Such recognition systems are characterized by not requiring an exact alignment between each staff and their corresponding labels, hence facilitating the creation and retrieval of labeled corpora.

When considering an *agnostic music representation* [1], i.e., a representation based on the graphical content rather than its musical meaning, symbols depict a two-dimensional nature [2]: the *shape*, which encodes the temporal duration of the event (sound or absence of it), and the *height*, which indicates the pitch of the event represented with its vertical position in the staff. However, when holistic OMR systems are trained, each possible combination of shape and height is represented as unique categories, which leads to music symbols being treated the same way as text characters [3], thus not exploiting the particularities of music notation. Based on this premise, recent research OMR works [4]–[7] have explored this topic, concluding that the individual exploitation of each dimension generally yields better recognition rates.

In this work we propose to further explore and exploit the two-dimensional nature of music symbols and, more precisely, we focus on symbol recognition at a staff-line level of monophonic early music documents. Considering the end-to-end neural-based framework by Calvo-Zaragoza et al. [2] as a starting point, we propose and discuss the separate extraction of shape and height features for then proposing several integration policies. The results obtained for two different corpora show that our proposal outperforms the recognition performance of the baseline considered, even in cases in which there is a considerably narrow room for improvement.

The rest of the paper is organized as follows: Section II introduces the proposed approach; Section III describes the experimental setup as well as the results and their analysis; finally, Section IV concludes the work and poses some ideas for future research.

II. Methodology

The proposed OMR recognition task works at the staff level, thus we assume that a certain preprocess (e.g., [8], [9]) has already segmented the different staves in a music sheet. In this sense, given an image of a single staff, our goal is retrieving the series of symbols that appear therein, i.e., our recognition model is a *sequence labeling* task [10].

Formally, let $\mathcal{X}$ represent a space of music staves. Also let Σ represent a symbol vocabulary and $\mathcal{Z} = \Sigma^*$ the complete set of possible sequences which may be obtained from that vocabulary. We assume the existence of a set $\mathcal{T} = \{(x_i, \mathbf{z}_i) : x_i \in \mathcal{X},\ \mathbf{z}_i \in \mathcal{Z}\}_{i=1}^{|\mathcal{T}|}$ which relates a given staff x_i to the sequence of symbols $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iN})$ assuming the existence of an underlying function $g : \mathcal{X} \rightarrow \mathcal{Z}$.

In this work we consider the introduced Convolutional Recurrent Neural Network (CRNN) scheme together with the Connectionist Temporal Classification (CTC) training algorithm [11] for approximating the underlying function as $\hat{g}(\cdot)$. Based on this premise we shall derive different neural designs for performing the recognition task.

The rest of the section further develops the idea of the two-dimensional natural nature of music symbols when considering agnostic notation and its application in our case as well as the different neural architectures considered.

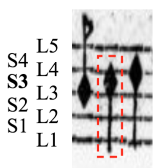
This research work was partially funded by the University of Alicante through project GRE19-04, the "Programa I+D+i de la Generalitat Valenciana" through grant APOSTD/2020/256, and the Spanish Ministerio de Universidades through grant FPU19/04957.

A. Symbol representation

Agnostic music notation allows defining each music symbol by its individual shape and height graphical components. It must be noted that this duality can be applied to all symbols, even to those that indicate the absence of sound, i.e. rests, as they may also appear at different vertical positions.

Let Σ_S and Σ_H be the spaces for the different shape and height labels, respectively. Formally, $\Sigma_T = \Sigma_S \times \Sigma_H$ represents the set of all possible music symbols in which a given ith element is denoted as the 2-tuple $\langle s_i, h_i \rangle : s_i \in \Sigma_S,\ h_i \in \Sigma_H$. However, while all aforementioned combinations are theoretically possible, in practice some pairs are very unlikely to appear, being hence $\Sigma_T \subset \Sigma_S \times \Sigma_H$. In a practical sense, for facilitating the converge of the model, we restrict the Σ_T vocabulary to the 2-tuples elements present in the corpus, i.e., $\Sigma_T = \{\Sigma_S, \Sigma_H\}^{\mathcal{T}}$.

Figure 1 shows a graphical example of the commented agnostic representation for a given symbol in terms of its shape (Σ_S), height (Σ_H), and combined labels (Σ_T), respectively.



Shape label: *note.quarter_down*
Height label: *S3*
Combined label: *note.quarter_down:S3*

Fig. 1: Agnostic representation of a handwritten music symbol, showing its shape, height, and combined labels. Note that Ln and Sn respectively denote the staff line or space on which the symbol may be placed.

B. Recognition architectures

The architecture of both recognition frameworks, baseline and proposed, are respectively described below.

1) Baseline approach: Systems based on CRNN trained using a CTC scheme, which are considered the state of the art in terms of end-to-end OMR systems, model the posterior probability of generating a sequence of output symbols given an input image. These networks are formed by an initial block of *convolutional* layers meant to learn the adequate features for the case at issue followed by another group of *recurrent* stages which model the temporal or spatial dependencies of the elements from the initial feature-learning block [12]

As commented, the CTC function allows training the CRNN scheme using unsegmented sequential data. In our case this means that, for a given staff image $x_i \in \mathcal{X}$, we only have its associated sequence of characters $\mathbf{z}_i \in \mathcal{Z}$ as its expected output, without any correspondence at pixel level or similar input-output alignment. Due to its particular training procedure, CTC requires the inclusion of an additional "*blank*" symbol within the Σ vocabulary, i.e., $\Sigma' = \Sigma \cup \{blank\}$.

At inference, CTC estimates a frame-wise posteriogram using a fully-connected network with $|\Sigma'|$ outputs and a *softmax* activation. This posteriogram is decoded using a *greedy* policy which retrieves the label that maximizes the probability in each frame. Eventually, a squash function which merges consecutive repeated symbols and removes the *blank* label is applied, hence obtaining the predicted sequence $\mathbf{z}'$.

In this work we consider as baseline the particular CRNN configuration proposed in the work by Calvo-Zaragoza et al. [2]. This neural model comprises four convolutional layers for the feature extraction process followed by two recurrent units for the dependency modeling. As commented, the output of the last recurrent layer is connected to a dense unit with $|\Sigma'|$ output neurons.

2) Proposed approach: This section presents the different neural architectures proposed for exploiting the individual shape and height properties when considering an agnostic music notation. For that, we modify the base CRNN architecture introduced in Section II-B1 by adding different layers for adequately exploiting such pieces of information with the aim of improving the overall recognition rate.

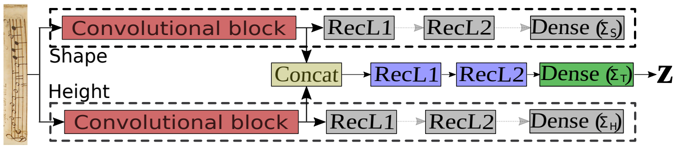
More precisely, our hypothesis is that having two CRNN models which are respectively specialized on retrieving the shape and height features may be beneficial with respect to having a unique system that deals with the task as a whole. The input staff image is individually processed by each model and the different characteristics obtained may be gathered at some point of the model before the actual classification process.

Based on that premise, we propose three different end-to-end architectures which basically differ on the point in which the two CRNN models are joined: (i) the *PreRNN* one, which joins the extracted features by each model right before the recurrent block, (ii) the *InterRNN* one, which does this process after the first recurrent layer, and (iii) the *PostRNN* one, which gathers both sources of information after the recurrent block. These proposals are graphically shown in Figure 2.

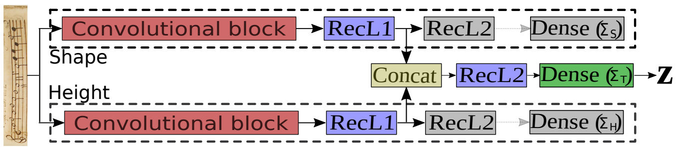
As it may be noted, all models depict three differentiated parts. Out of these three parts, two of them constitute complete CRNN models specialized on a certain type of information (either shape or height) while the third one is meant to join the previous sources of information for the eventual classification.

Note that all branches are separately trained using the same set of staves $\mathcal{T}$ with the CTC learning algorithm, simply differing on the output vocabulary considered. This way we somehow bias the different *Shape* and *Height* CRNN branches to learn specific features for those pieces of information, whereas in the case of the *Combined* branch the training stage is expected to learn how to properly merge those separate pieces of information.

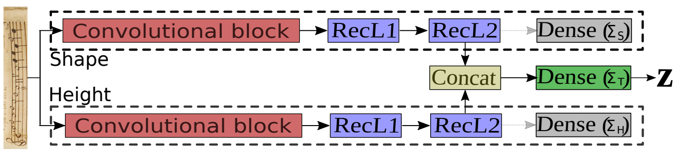
In this work we consider two different policies for training the introduced models: a first scenario in which the parameters of the entire architecture are learned from scratch, i.e., we sequentially train the different branches without any particular initialization; and a second case in which the shape and height branches are separately trained and, after their convergence, the same procedure of the first scenario is reproduced. In this regard, we may assess how influential the initial training



(a) **PreRNN** model: Features extracted by each specialized model are joined after the convolutional block.

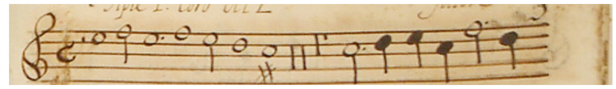


(b) **InterRNN** model: Features extracted by each specialized model are joined after first recurrent layer.



(c) **PostRNN** model: Features extracted by each specialized model are joined after the recurrent block.

Fig. 2: Graphical description of the three CRNN-based architectures proposed. The *Concat* block concatenates the input features from each branch. Gray layers represent the ones that are only considered during the training stage of the model.



(a) Handwritten music staff extracted from the *Capitan* set.

(b) Staff from the *Il Lauro Secco* corpus.

Fig. 3: Music excerpts of the corpora used in the experiments.

stage of the different branches of the scheme is on the overall performance of the system.

While these new architectures suppose an increase in the network complexity with respect to the base model considered, the separate exploitation of the graphic components of the commented agnostic music notation is expected to report an improvement in terms of recognition. Nevertheless, this increase does not imply a need for using more data since each part of the network specializes in a set of image features. Finally, note that, while the training stage may require more time until convergence, the inference phase spans for practically the same time-lapse as in the base network since the different branches work concurrently before the merging phase.

III. Experimentation

This section introduces the different corpora considered for assessing the goodness of our proposal as well as the evaluation protocol contemplated. The obtained results are finally provided together with a brief discussion about them.

A. Corpora

We consider two corpora of music scores depicting Mensural notation with varying printing style:

- *Capitan* corpus [13]: Manuscript of ninety-six pages dated from the 17th century of *missa* (sacred music). An example of a particular staff from this corpus is depicted in Figure 3a.
- *Il Lauro Secco* corpus [14]: Collection of one hundred and fifty-five typeset pages corresponding to an anthology of Italian madrigals of the 16th century. Figure 3b shows an staff example of this set.

For comparative purposes with the reference work, we reproduce the exact experimentation conditions. In this sense, we resize each image to a height of 64 pixels, maintaining the aspect ratio (thus, each sample might differ in width) and convert them to grayscale, with no further pre-processing. In terms of data partitioning, we also reproduced their train, validation, and test divisions with the same 5-fold Cross Validation policy.

B. Evaluation protocol

The performance of the proposal is measured with the Symbol Error Rate (Sym-ER), computed as the average number of elementary editing operations (insertions, deletions or substitutions) necessary to match the sequence predicted by the model with the ground truth sequence, normalized by the length of the latter. Mathematically, this is represented as:

$$\text{Sym-ER}\ (\%) = \frac{\sum_{i=1}^{|\mathcal{S}|} \text{ED}\left(\mathbf{z}_i,\ \hat{h}\left(x_i\right)\right)}{\sum_{i=1}^{|\mathcal{S}|} |\mathbf{z}_i|} \qquad (1)$$

where $\mathcal{S} = \{(x_i, \mathbf{z}_i) : x_i \in \mathcal{X},\ \mathbf{z}_i \in \mathcal{Z}\}_{i=1}^{|\mathcal{S}|}$ is a set of test data and $\text{ED}\left(\cdot,\cdot\right)$ represents the string Edit distance [15]. Note that we resort to this metric for comparative purposes with the reference work by Calvo-Zaragoza et al. [2], but other alternatives could be further considered [16].

C. Results

This section presents and discusses the results obtained. Since the experiments have been performed in a cross-validation scheme, the figures provided constitute the average values obtained for each of the cases considered. Note that these values constitute that of the test data partition for the case in which the validation data achieves its best performance.

The results obtained in terms of the Symbol Error Rate (Sym-ER) for the base neural configuration and the three different proposed architectures for each data corpus considered are shown in Table I and Figure 4. Note that these recognition models consider the Σ_T vocabulary case, i.e., each detected element is represented as the 2-tuple which defines both the shape and height of the symbol.

TABLE I: Results obtained in terms of the Symbol (Sym-ER) comparing the base neural model with the different architectures proposed for the Σ_T vocabulary case. *Pretrained* and *Raw* denote the cases in which shape and height branches have been trained or not before considering the joint model, respectively.

	Baseline	**PreRNN**		**InterRNN**		**PostRNN**	
		Raw	**Pretrained**	**Raw**	**Pretrained**	**Raw**	**Pretrained**
Capitan	10.32	11.12	8.62	7.68	8.18	8.30	8.39
Il Lauro Secco	4.87	4.54	4.70	4.35	4.17	4.36	4.42

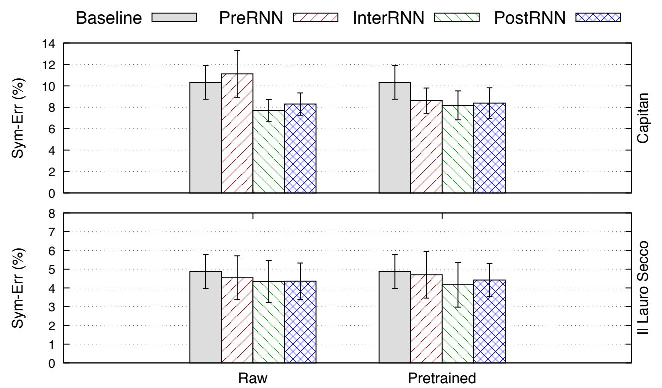


Fig. 4: Comparison of the base neural model and the different architectures proposed for the Σ_T vocabulary case in terms of the Symbol Error Rate — Sym-ER (%)— and their respective standard deviations. *Pretrained* and *Raw* denote the cases in which shape and height branches have been trained or not before considering the joint model, respectively.

An initial remark to begin with is that, as it can be appreciated both in Figure 4 and Table I, all proposed architectures improve the results obtained by their respective baselines except for the case of the *PreRNN* architecture in the *Raw* scenario for the *Capitan* corpus. We may also check that the *InterRNN* model is the one that consistently achieves the best error rates for all scenarios considered only tying with the *PostRNN* proposal for the *Raw* case of the *Il Lauro Secco* set.

It must be also noted that, for all cases, the error rates obtained with the *Capitan* corpus are remarkably higher than those obtained with the *Il Lauro Secco* set. This is a rather expected result since the graphical variability inherent to handwritten data compared to the typeset format supposes a drawback to the recognition algorithm.

As a last point to comment from these figures is that, while differences between the *Raw* and *Pretrained* do not generally differ in a remarkable sense, convergence is always faster in the latter approach than in the former one.

IV. Conclusions

Holistic symbol recognition approaches have proved their usefulness in the context of Optical Music Recognition (OMR) since, given that no alignment is required between the input score and the sequence of output elements, corpora are relatively easy to create. In such context, these sets are commonly labeled using either a *semantic* notation, which codifies the actual musical meaning of each element in the score, or an *agnostic* representation which encodes the elements as a combination of a shape tag and the vertical position (height) in the score. While this latter representation lacks the underlying music sense of the former, it has the clear advantage of perfectly suiting an image-based symbol recognition task as the vocabulary is defined directly on visual information. However, it is still unclear how to take advantage of the fact that each symbol is actually a combination of two individual primitives representing the shape and height of the element.

This work presents an end-to-end approach that exploits this two-dimensional nature of the agnostic music notation to solve the OMR task at a staff-line level. We have considered two Convolutional Recurrent Neural Network (CRNN) schemes to concurrently exploit the shape and height pieces of information to then merge them at the actual neural level. Three different integration policies are empirically studied: (i) the *PreRNN* one, which joins the shape and height features right before the recurrent block, (ii) the *InterRNN* one, which does the merging process after the first recurrent layer, and (iii) the *PostRNN* one, which collects both sources of information after the recurrent block. The results obtained confirm that the gathering point impacts the performance of the model, exhibiting an error reduction which ranges between 14.4% and 25.6% referred to the base neural model depending on this configuration.

In light of the obtained conclusions, this work opens new research points to address. In that sense, future work considers extending the approach from this monophonic context to a homophonic one [17]. Also, given the variability of music notation and the relative scarcity of existing labeled data, we aim at exploring *transfer learning* and *domain adaptation* techniques to study different strategies to properly exploit the knowledge gathered from a given corpus on a different one. Besides, we also consider that other network architectures may provide some additional insights to the ones obtained in this work as well as more competitive recognition rates. Finally, since the *greedy* decoding strategy considered does not take advantage of the actual Language Model inferred in the Recurrent layer of the network, our premise is that more sophisticated decoding policies may report improvement in the overall performance of the proposal.

Acknowledgment

This research work was done in the context of the Spanish Ministry HISPAMUS project (TIN2017-86576-R, partially funded by the EU).

REFERENCES

[1] D. Rizo, J. Calvo-Zaragoza, J. M. Iñesta, and I. Fujinaga, "About agnostic representation of musical documents for optical music recognition," in *Music Encoding Conference*, May 2017.

[2] 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.

[3] F. J. Castellanos, J. Calvo-Zaragoza, and J. M. Inesta, "A neural approach for full-page optical music recognition of mensural documents," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020, pp. 23–27.

[4] E. van der Wel and K. Ullrich, "Optical music recognition with convolutional sequence-to-sequence models," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, S. J. Cunningham, Z. Duan, X. Hu, and D. Turnbull, Eds., 2017, pp. 731–737.

[5] A. Nuñez-Alcover, P. J. P. de León, and J. Calvo-Zaragoza, "Glyph and position classification of music symbols in early music manuscripts," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2019, pp. 159–168.

[6] 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.

[7] A. Ríos-Vila, J. Calvo-Zaragoza, and J. M. Inesta, "Exploring the two-dimensional nature of music notation for score recognition with end-to-end approaches," in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2020, pp. 193–198.

[8] D. Rizo, J. Calvo-Zaragoza, and J. M. Iñesta, "Muret: A music recognition, encoding, and transcription tool," in *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, 2018, pp. 52–56.

[9] J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga, "One-step detection of background, staff lines, and symbols in medieval music manuscripts with convolutional neural networks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, S. J. Cunningham, Z. Duan, X. Hu, and D. Turnbull, Eds., 2017, pp. 724–730.

[10] A. Graves, "Supervised sequence labelling," in *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, pp. 5–13.

[11] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 369–376.

[12] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[13] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, "Handwritten music recognition for mensural notation: Formulation, data and baseline results," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1081–1086.

[14] E. Parada-Cabaleiro, A. Batliner, and B. W. Schuller, "A diplomatic edition of il lauro secco: Ground truth for omr of white mensural notation." in *ISMIR*, 2019, pp. 557–564.

[15] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet physics doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[16] J. Hajič Jr, "A case for intrinsic evaluation of optical music recognition," in *Proceedings of the 1st International Workshop on Reading Music Systems*, 2018, pp. 15–16.

[17] M. Alfaro-Contreras, J. Calvo-Zaragoza, and J. M. Iñesta, "Approaching end-to-end optical music recognition for homophonic scores," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2019, pp. 147–158.

Implementation and evaluation of a neural network for the recognition of handwritten melodies

Nils Wenzlitschke
Universität Leipzig
Leipzig, Germany
nw20hewo@studserv.uni-leipzig.de

***Abstract*— Today, digitalization affects all areas of life, including research. However, many historical documents exist only in physical form, preventing the use of computer-aided methods for preservation and analysis. Therefore, this paper deals with the examination of the sheet music collection of the University Library of Regensburg using deep learning techniques. It addresses the research question of whether current state-of-the-art deep learning architectures are able to recognize handwritten monophonic sheet music in order to digitize the music collection. For this purpose, an optical music recognition system is implemented and evaluated. This full-page optical music recognition system consists of two neural networks: a staff recognition using selection autoencoders and an end-to-end note recognition using convolutional recurrent neural networks. Additionally, a pilot corpus is formed from the Regensburger Liedblattsammlung and annotated for the corresponding tasks.**

I. Introduction

The University Library of Regensburg owns a song sheet collection of 140,000 folk songs. These were handed down either orally or in handwriting and originate from the entire German-speaking area. Within the scope of a *German Research Foundation* (DFG) funded project, three large collections were digitized. One of these collections is the previously mentioned *Regensburger Liedblattsammlung* (RegLS). This corpus is available with scanned images and respective metadata. However, neither lyrics nor sheet music could be transcribed satisfactorily [16]. Hajič et al. [12] state that current *optical music recognition* (OMR) systems are proving useful for digital libraries. They mention that many musical compositions have neither been recorded nor digitized. For the most part, these musical pieces exist only as manuscripts, as typesetting music has been an expensive endeavor in the past. Following this statement by Hajič et al. [12], this paper will address the research question of whether it is possible to use a state-of-the-art OMR neural network to transcribe the melodies of the RegLS.

II. Related Work

The OMR framework proposed by Bainbridge and Bell [1] and refined by Rebelo et al. [18] had its problems due to the complex multi-steps, suboptimal results in staff-line retrieval, and symbol classification [8]. Moreover, the traditional approach requires a separate, customized OMR system for each different type of sheet music, which subsequently requires a high level of effort and provides little room for generalization of an OMR system.

Finally, advancing developments in machine learning brought tools that changed the OMR workflow. In particular, the development of CNN marked a decisive advance for computer vision research [15]. This shift has led to a generalization of OMR systems; consequently, systems can be designed that are trained with enough training data for different manuscripts.

Following the deep learning technology, different approaches have evolved. On the one hand, the traditional workflow achieving significant improvements in certain steps, i.e. staff-line removal [11] and symbol classification [17]. On the other hand, a different approach has evolved with the intent to enable a holistic end-to-end-workflow [6, 23]. This end-to-end-workflow involves an input image identified by a deep learning architecture and converted directly into an output representation. These approaches have been successfully applied to printed sheet music in the past [6, 23]. Calvo-Zaragoza et al. [8] demonstrate that the end-to-end approach can also be applied to handwritten sheet music.

Creating the individual staves of the respective music documents corresponds to an enormous amount of manual work. This is reduced using layout detection. Castellanos et al. [10] proposes a layout detection using *selectional auto-encoders* (SAE). These two approaches combined provide a method to perform full-page recognition, as proposed by Castellanos et al. [9]. This approach is used in this paper to examine the given dataset of the University Library of Regensburg.

To evaluate the results of the study, I use the *Camera Printed Images of Music Staves* (Camera-PrIMuS) dataset as a comparison corpus. Calvo-Zaragoza and Rizo [7] created this dataset intending to provide a corpus for experimenting and evaluating heterogeneous OMR systems. It consists of 87,678 real music incipits.

III. Data

Since 2001 the University Library of Regensburg has been responsible for several extensive collections and folk music research sources. These collections include printed texts, video recordings, folk songs with sheet music, lyrics, and audio tracks. The 140,000 music sheets form the heart of the RegLS. These have been handed down orally or handwritten over the years. A large part of the folk song collection dates to 1914. At that time, the German Folk Song Archive was founded in Freiburg im Breisgau to systematically collect German folk songs [3]. For this paper, I only examine 1000 pages of scanned sheet music with respective cover and back. To recognize the sheet music, only the staves themselves are relevant. Due to layout detection, there is no need to delete the non-sheet pages because the system looks for staff in the first place. If it does not find any, it moves to the next page without distorting the results. Figure 1 shows one example sheet music of the RegLS. The monophonic melodies of the folk songs are handwritten on the respective song sheet. Below the melody, there is the text of the song written with a typewriter. Additional meta-information is located at the bottom of the sheet, such as the number of the song sheet within the collection and the archive´s name from which this song sheet originally came.

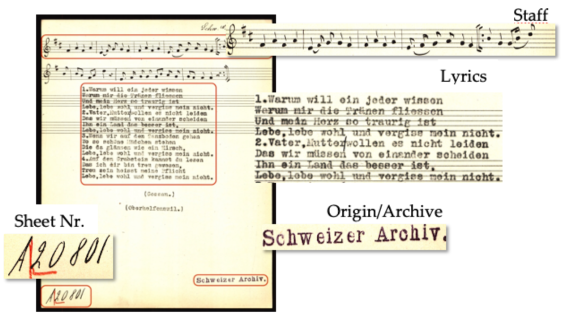


Figure 1: Music sheet A20801

3.1 Music notation formats

Ríos-Vila et al. [19] show that the well-known music notation format (MusicXML, MEI) poses a problem for the end-to-end approach, which requires a separate encoded word for each note [19]. Considering this requirement, Humdrum **kern [13] seems to be a suitable format, as stated by Ríos-Vila et al. [19]. **kern offers the advantage of being a format that describes individual note symbols by corresponding labels.

3.2 Camera-PrIMuS

A comparative corpus is needed to evaluate the results obtained by the OMR system based on the RegLS. For this purpose, I use the Camera-PrIMuS developed by Calvo-Zaragoza and Rizo [7].

I use this dataset for both training and evaluation. The RegLS data should first be trained and evaluated without the effect of other data on its own. In a second step, the model will be trained with the Camera-PrIMuS dataset only. The RegLS data will be used for evaluation. In a third step, the model will be trained using the Camera-PrIMuS dataset, load the resulting model and perform a so-called transfer learning, i.e., a fine-tuning using the RegLS data, which in turn will be evaluated using the RegLS data.

IV. Methodology

The architecture used is based on that proposed by Castellanos et al. [9]. This approach consists of a layout detection with a following end-to-end recognition. Following this approach, initially, the staves must be detected in the first place, for that, SAE are used. Second, the symbols in the detected staff need to get recognized and interpreted, therefore CRNN are used. To implement these procedures deep neural networks are trained one for each task. Castellanos et al. [9] provide a solution that is based on layout detection by Castellanos et al. [10] and the end-to-end recognition refined by Calvo-Zaragoza et al. [8]. These two essential components are combined to enable music sheet recognition on a full page. This improves the workflow in the sense that the preparation of the notes into single staves does not have to happen, since the first model recognizes, extracts, and passes them on to the second model, which then recognizes the notes in the staff. Different to the method used by Castellanos et al. [9], this paper is using the music notation format **kern proposed by Ríos-Vila et al. [19] to automate the encoding into a digital score. This avoids another manual step to convert the semantic representation used by Calvo-Zaragoza et al. [8] into a digital music format.

4.1 Data preparation

Data are essential in deep learning tasks. It is equally crucial to understand the nature of the data and how it must be prepared for the task. In this case, the goal is to digitize the notes but omit the text and metadata. The architecture consists of layout detection and note recognition using SAE and CRNN. Therefore, two data preparation steps are needed. The first workflow is made up of labeling the sheet music with LabelImg [1]. LabelImg is an annotation tool for labeling image data for computer vision tasks. The second workflow deals with working out the sequential representation in the form of the **kern format from the sheet music. This is about transferring each staff with its corresponding characters into a separate file with the file extension .krn.

4.2 Staff Retrieval

The staff retrieval stage aims to detect and extract the single individual staves. The layout detection is done using SAE. Castellanos et al. [10] introduce SAEs to classify documents into different levels of information. They distinguish background, staff, musical symbols, and lyrics. For this work, however, only the distinction between background and staff has to be made since an analysis of the lyrics or the individual notes does not occur.

The staff retrieval is done by converting sheet music into an image that is black except for the areas where staves are visible, these are displayed in white, as shown in Figure 2. This image serves as the ground truth on which the loss function calculates its error and adjusts the neurons' values in the neural network by backpropagation.

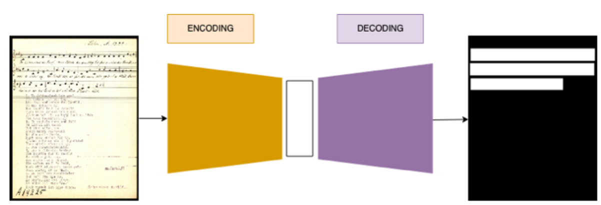


Figure 2: Graphic representation of the SAE model applied to sheet music. Background in black and staves in white.

4.3 Recognition of monophonic scores

Castellanos et al. [9] build their CRNN architecture on the existing state-of-the-art sequence recognition proposed by Shi et al. [22]. Because they have achieved successful results with this approach on the Camera-PrIMuS dataset and the notation is similar both in complexity and task requirements, the decision has been made to use this approach.

The architecture of the note recognition consists of a CRNN and a CTC loss function. A CRNN is composed of a CNN and an RNN. This composition offers the decisive advantage that learned features refer to each other and influence each other. Thus, the neural network learns both the visual recognition of notes and the possible semantic relationship. It consists of multiple convolutional layers, which are

[1] https://github.com/tzutalin/labelImg

extracting certain features of an input image. These features are handed over to several recurrent layers, which are implemented by bidirectional long short-term memory (BLSTM) layers.
The CRNN is followed by the CTC loss function, which evaluates the probability of various possible sequence labels. The note recognition output is the decoded output matrix of the CTC in the form of a **kern sequence. The training of the CRNN model was done in several different combinations through transfer learning. For this purpose, it was trained with the corpora RegLS, fmt, and Camera-PrIMuS.
The existing RAM of the Goolge Colab reached its limits when training model with the Camera-PrIMuS. The final version the transfer learning could only be trained with 8732 of the 87.679. This should be taken into account with regard to the results and further work.

V. Experiments

Experiments were run in a Google Colab instance under Keras v2.4 and Tensorflow v2.4.1. The data was split into three data partitions for the SAE training: 60% training, 20% validation, and 20% testing. The training for note recognition were done with two partitions: 90% training and 10% validation.

5.1 Staff Retrieval

The table notation in this chapter reads as follows: Conv2D(n, h x w, 'act') for a two-dimensional convolutional layer with n filters and a kernel size of height h and width w with the activation function act; MaxPool(h x w) for a max-pooling function to down-sample the input with a window of dimensions height h and width w; UpSamp(h x w) for an up-sampling operator with h rows and w columns; BLSTM(n) represents the BLSTM layer, where n corresponds to the number of units; Dropout(p) corresponds to the ratio of dropout operations ; Dense(n, 'act') describes the Dense layer with n neurons and the act activation function act [9].
The staff retrieval's architecture consists of the encoding and the decoding steps. These steps are composed of six convolutional layers with the non-linearity rectified linear unit (ReLU) activation function, followed by max-pooling layers in the encodings step and up-sampling layers and another convolutional layer with a sigmoid activation function in the decoding step as can be seen in table 1.
The sigmoid activation function calculates the concluding approximation of layout detection. The input size is 512x512 pixels. This is based on informal tests done by Castellanos et al. [9]. The low resolution is sufficient because the model does not need to recognize small details. The images were rescaled to this size through the OpenCV library [9]. For the training, the binary cross-entropy was used as the loss function, and the Adam optimizer [14] were used to train the model. The training was conducted with a batch size of four over 100 epochs.

5.2 Recognition of monophonic scores

The first end-to-end architecture (E2E_1) follows the best practices from the work of Calvo-Zaragoza et al. [8] as can be seen in Table 2a., but, in contrast, has fewer layers and units on each layer. The second end-to-end architecture (E2E_2) is based on the first but extended by an additional convolutional layer and more units per layer, as can be seen in detail in Table 2b. This adjusts the network what was initially created for the recognition of printed sheet music to the additional complexity of the handwritten recognition task. It consists of an input layer that has a height of 64 pixels and an arbitrary width. The given staves are of different lengths, and the CTC loss function does not specify a fixed length because it determines the time steps itself. The input layer is followed by four convolutional layers with ReLU activation function and subsequent max-pooling layers. This CNN is followed by two BLSTM layers with dropout and a Dense layer on which the softmax activation function is applied to retrieve the final output sequence. The softmax function converts the output into a probability for each symbol, including the blank symbols. The model outputs the symbol with the highest probability.

Input	Encoding	Decoding	Output
$[0,255]^{512X512}$	Conv2D(128,3x3,ReLU) MaxPool(2,2)	Conv2D(128,3x3,ReLU) UpSamp(2,2)	$[0,1]^{512X512}$
	Conv2D(128,3x3,ReLU) MaxPool(2,2)	Conv2D(128,3x3,ReLU) UpSamp(2,2)	
	Conv2D(128,3x3,ReLU) MaxPool(2,2)	Conv2D(128,3x3,ReLU) UpSamp(2,2)	
		Conv2D(1,3x3,Sigmoid)	

Table 1: Detailed representation of the architecture for staff retrieval using SAE adjusted for my purposes based on [9].

In several experiments the architectures E2E_1 and E2E_2 were first trained on the RegLS dataset for 250 epochs with a batch size of 4. Every fifth epoch the evaluation metric was compared with the best error rate value so far, if the current model gives a lower error value, the model was saved with these configurations. In the next experiment the fmt dataset, was trained for 125 epochs at a time and the resulting output, i.e. the weights and bias, was then loaded and trained on the RegLS dataset for 125 epochs. These experiments were performed in the different combinations (RegLS only, RegLS+fmt, RegLS+Camera-PrIMuS, RegLS+fmt+Camera-PrIMuS and Camera-PrIMuS only) for the E2E_1 and E2E_2 architectures.

Input: $[0, 255]^{64 \times W}$
Conv2D(16,3x3,ReLU) MaxPool(2,2) Conv2D(32,3x3,ReLU) MaxPool(2,1) Conv2D(64,3x3,ReLU) MaxPool(2,1)
BLSTM(32, Dropout(0.25)) BLSTM(32, Dropout(0.25)) Dense(,softmax')

Input: $[0, 255]^{64 \times W}$
Conv2D(32,5x5,ReLU) MaxPool(2,2) Conv2D(64,3x3,ReLU) MaxPool(2,1) Conv2D(64,3x3,ReLU) MaxPool(2,1) Conv2D(128,3x3,ReLU) MaxPool(2,1)
BLSTM(256, Dropout(0.25)) BLSTM(256, Dropout(0.25)) Dense(,softmax')

Table 2: a) First Architecture for note recognition as proposed by Calvo-Zaragoza et al. [8], with fewer layers and units. b) Second Architecture for note recognition. Extends the first by another convolutional layer and more units per layer.

5.3 Evaluation metrics

To evaluate the architecture, I used the evaluation metrics of the based paper by Castellanos et al. [9]. The evaluation needs

to be done for the two steps individually in order to see where further adjustments are needed in terms of model settings or training data. A common metric to evaluate layout detection is the *Intersection over Union* (IoU). IoU indicates the overlap of the ground truth data and the staves extracted by the model. Thus, a high IoU reveals that the prediction strongly aligns with the ground truth, a low one that this diverges widely. Accordingly, the layout detection training tries to maximize the IoU.
Researchers who have used end-to-end recognition calculate the edit distance to retrieve the quality of a model in different disciplines [19, 22]. In terms of this work, this means how many operations a user theoretically needs to perform to correct the errors in the **kern output of the model. This number of operations corresponds to the *symbol error rate* (SER). The performance of note recognition can be measured by the SER. A high SER, in this case, means that many recognized symbols in the note sequence are incorrect and require manual editing, while a low SER means that the note sequence predicted by the model is more similar to the ground truth. The SER must be minimized during training.

VI. Results & Discussion

The layout detection results, as can be seen in Table 3, show that the SAE model learned the note structure well. The experiments were performed on 1028 staves. It was able to retain all 1028 staves of the 328 sheet music: 1028 true positives staves were detected, while 14 false positives were recognized. A staff is considered to be correctly detected if the IoU is greater than 55%.

	Staff Retrieval
Nr. Of sheet music	328
Nr. Of Staves	1028
True Positives	1028
False Positives	14
False Negatives	0
Average IoU	91,93%
FScore	0.992

Table 3: Results of the SAE layout detection

Castellanos et al. [9] obtained an average IoU of 86.3% for the CAPITAN dataset with 737 staves and one of 79.9% for the SEILS dataset with 1278 staves. In this work on the RegLS, this approach achieved 91% average IoU on the validation data. These are good results, which could be achieved with a pilot dataset of 328 images. However, the predicted staves showed that partial notes or parts of notes such as head, stem, flag, were outside the staves and thus not detected with. This raises issues for the following step. For note recognition, the two end-to-end architectures were compared directly. This was done as shown in Table 4 for all possible training combinations, including transfer learning. The training, which was done on the 180 staves of the RegLS, shows that the extended architecture (E2E_2) is superior to the simple one (E2E_1). The adapted architecture for handwritten sheet music learned the sequential representation of the notes partially. Ríos-Vila et al. [19] achieved with the first tested architecture on the Camera-PrIMuS dataset a SER of 5.31%. The Camera-PrIMuS dataset contains 87,679 incipits. However, in this work there are not 87,679 staves, but 169 sheet music of the RegLS. This gap in the amount of data makes it difficult to compare the results with the one Ríos-Vila et al. [19] achieved. However, it shows what large datasets state-of-the-art application uses and might explain a limitation in the results.

	RegLS	RegLS+fmt	RegLS+CP	RegLS+fmt+CP	CP
E2E_1	63.25	56.02	42.97	44.98	80.72
E2E_2	47.49	46.99	40.76	44.38	81.12

Table 4: Results of note recognition given in SER in %. Comparison of the transfer learning approach on different combinations of the RegLS, fmt and Camera-PrIMuS (CP) corpora and the different end-to-end architectures: The architecture provided by Calvo-Zaragoza et al. [8] for the recognition of music documents is denoted as E2E_1 and the extended one as E2E_2 in the table.

The research question of whether a state-of-the-art deep learning architecture of OMR can be used to recognize and thereby digitize a handwritten music collection of a university library can be confirmed with these results under some limitations. While the layout detection task already achieves good results with 328 annotated note pages, the results of the note recognition task showed that although it partially learned certain musical symbols, on average, 47.49% would still need to be corrected manually. The consequence of this result is that more annotated data are needed.
A broader approach was explored in this work that requires less specially annotated data. Instead, already annotated datasets served as a foundation for the model. The transfer learning significantly improved the results in the E2E_1 architecture. There, SER improved by up to 20%, while E2E_architecture merely benefited by 7%. In addition, transfer learning done exclusively on the pre-training through the Camera-PrIMuS dataset was found to provide the best results of 40.76% SER.

VII. Conclusion

In this paper, I have presented an approach to recognize the sheet music of the RegLS. To implement this, an annotated pilot dataset is provided in this paper. This includes 328 sheet music of the RegLS for layout detection and 180 staves for note recognition. The presented approach combines the architecture of Castellanos et al. [9] extended for the task of handwritten music recognition and the findings of Ríos-Vila et al. [19] on the output format.The results show that full-page sheet music recognition works with trade-offs. The staff retrieval on the sheet music works in the first part of the full-page recognition. However, the note recognition results indicate that the detection of note sequences in the second step works only partially. The main limitations are the amount of data. I tried to benefit from existing annotated datasets via transfer learning. Thus, the obtained results of the bare training using RegLS data could be improved by training with the fmt and the Camera-PrIMuS dataset. Transfer learning with the Camera-PrIMuS corpus gave the best result and should be considered for further work.
This paper set out to validate the research question of whether it is feasible to recognize RegLS scores using deep learning techniques and has shown ways in which this can be used in future work to digitize the RegLS dataset.

References

[1] Bainbridge, D., & Bell, T. (2001). The challenge of optical music recognition. Computers and the Humanities, 35(2), 95–121. https://doi.org/10.1023/A:1002485918032

[2] Baró, A., Riba, P., Calvo-Zaragoza, J., & Fornés, A. (2019). From Optical Music Recognition to Handwritten Music Recognition: A baseline. Pattern Recognition Letters, 123, 1–8. https://doi.org/10.1016/j.patrec.2019.02.029

[3] Burghardt, M. (2018). Digital Humanities in der Musikwissenschaft – Computergestützte Erschließungsstrategien und Analyseansätze für handschriftliche Liedblätter. Bibliothek Forschung Und Praxis, 42(2), 324–332. https://doi.org/10.1515/bfp-2018-0032

[4] Calvo-Zaragoza, J., Castellanos, F. J., Vigliensoni, G., & Fujinaga, I. (2018). Deep neural networks for document processing of music score images. Applied Sciences (Switzerland), 8(5). https://doi.org/10.3390/app8050654

[5] Calvo-Zaragoza, J., Hajič, J., & Pacha, A. (2019). Understanding Optical Music Recognition. August. http://arxiv.org/abs/1908.03608

[6] Calvo-Zaragoza, J., & Rizo, D. (2018a). End-to-end neural optical music recognition of monophonic scores. Applied Sciences (Switzerland), 8(4), 1–23. https://doi.org/10.3390/app8040606

[7] Calvo-Zaragoza, J., & Rizo, D. (2018b). Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores. Proceedings of the 19th International Society for Music Information Retrieval Conference, 248–255. http://imslp.org

[8] Calvo-Zaragoza, J., Toselli, A. H., & Vidal, E. (2019). Handwritten Music Recognition for Mensural notation with convolutional recurrent neural networks. Pattern Recognition Letters, 128(August), 115–121. https://doi.org/10.1016/j.patrec.2019.08.021

[9] Castellanos, F. J., Calvo-Zaragoza, J., & Inesta, J. M. (2020). A Neural Approach for Full-Page Optical Music Recognition of Mensural Documents. Proc. of the 21st Int. Society for Music Infor-Mation Retrieval Conf., October.

[10] Castellanos, F. J., Calvo-Zaragoza, J., Vigliensoni, G., & Fujinaga, I. (2018). Document analysis of music score images with selectional auto-encoders. Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, 256–263.

[11] Gallego, A. J., & Calvo-Zaragoza, J. (2017). Staff-line removal with selectional auto-encoders. Expert Systems with Applications, 89, 138–148. https://doi.org/10.1016/j.eswa.2017.07.002

[12] Hajič, J., Dorfer, M., Widmer, G., & Pecina, P. (2018). Towards full-pipeline handwritten OMR with musical symbol detection by U-NETS. Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, 225–232. http://music-encoding.org/

[13] Huron, D. (2002). Music information processing using the humdrum toolkit: Concepts, examples, and lessons. Computer Music Journal, 26(2), 11. https://doi.org/10.1162/014892602760137158

[14] Kingma, D. P., & Lei Ba, J. (2015). ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION.

[15] Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[16] Meier, F., Bazo, A., Burghardt, M., & Wolff, C. (2015). A Crowdsourced Encoding Approach for Handwritten Sheet Music. Music Encoding Conference Proceedings 2013 and 2014, 127–130.

[17] Pacha, A., & Eidenberger, H. (2018). Towards a Universal Music Symbol Classifier. Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2, 35–36. https://doi.org/10.1109/ICDAR.2017.265

[18] Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marcal, A. R. S., Guedes, C., & Cardoso, J. S. (2012). Optical music recognition: state-of-the-art and open issues. International Journal of Multimedia Information Retrieval, 1(3), 173–190. https://doi.org/10.1007/s13735-012-0004-6

[19] Ríos-Vila, A., Calvo-Zaragoza, J., & Rizo, D. (2020). Evaluating simultaneous recognition and encoding for optical music recognition. ACM International Conference Proceeding Series, October, 10–17. https://doi.org/10.1145/3424911.3425512

[20] Rizo, D., Calvo-Zaragoza, J., & Inesta, J. M. (2018). MuRET: A music recognition, encoding, and transcription tool. DLfM. https://doi.org/10.4000/adlfi.11021

[21] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673–2681. https://doi.org/10.1109/78.650093

[22] Shi, B., Bai, X., & Yao, C. (2017). An End-to-End Trainable Neural Networkfor Image-Based Sequence Recognition andIts Application to Scene Text Recognition. History of Economics Society Bulletin, 39(11), 2298–2304. https://doi.org/10.1017/S1042771600010292

[23] Van Der Wel, E., & Ullrich, K. (2017). Optical Music Recognition with Convolutional Sequence-to-Sequence models. Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, 731–737.

DoReMi: First glance at a universal OMR dataset

1st Elona Shatri
Centre for Digital Music
Queen Mary University of London
London, UK
e.shatri@qmul.ac.uk

2nd György Fazekas
Centre for Digital Music
Queen Mary University of London
London, UK
g.fazekas@qmul.ac.uk

***Abstract*—The main challenges of Optical Music Recognition (OMR) come from the nature of written music, its complexity and the difficulty of finding an appropriate data representation. This paper provides a first look at DoReMi, an OMR dataset that addresses these challenges, and a baseline object detection model to assess its utility. Researchers often approach OMR following a set of small stages, given that existing data often do not satisfy broader research. We examine the possibility of changing this tendency by presenting more metadata. Our approach complements existing research; hence DoReMi allows harmonisation with two existing datasets, DeepScores and MUSCIMA++. DoReMi was generated using a music notation software and includes over 6400 printed sheet music images with accompanying metadata useful in OMR research. Our dataset provides OMR metadata, MIDI, MEI, MusicXML and PNG files, each aiding a different stage of OMR. We obtain 64% mean average precision (mAP) in object detection using half of the data. Further work includes re-iterating through the creation process to satisfy custom OMR models. While we do not assume to have solved the main challenges in OMR, this dataset opens a new course of discussions that would ultimately aid that goal.**

***Index Terms*—optical music recognition, deep learning, dataset, sheet music**

I. Introduction

Despite improvements in music notation software, writing music on paper or distributing music in print is still very common. However, written music needs to be digitised for further editing, preparation for printing, sharing or auditioning on a computer, or creating playable musical demos. The process of manually digitising scores is slow and tiresome. New work is not the only subject of this process. The abundance of undigitised sheet music in archives and libraries is an essential motivation of OMR as well. Scores are often scanned and become part of library archives which advances their accessibility only partially. For example, the search of scanned music is limited to metadata and excludes musical content or patterns. Moreover, scanned scores do not allow for plagiarism check. An automatic process of converting handwritten scores, prints and scans to documents that a machine can read and interpret is the essence of Optical Music Recognition (OMR). OMR research typically divides the problem into distinct stages. Conventionally, four main stages are considered, starting with image pre-processing, followed by the detection of primitive musical objects, reconstructing musically meaningful composites, and finally, encoding using a machine-readable format [24]. By progressing one stage at a time, inconsistencies were created, mainly in the datasets used by researchers. Often, new work does not align with current work creating issues in evaluation and comparison.

Recently, researchers have been working on bringing their datasets together, namely MUSCIMA++ [4] and DeepScores [9], to improve compatibility. MUSCIMA++ contains handwritten music, while DeepScores is a typeset music dataset, both of which can be used in object detection. Harmonising them aids the goal of creating a reference dataset. With the motivation to support this goal and compare and use these datasets in future experiments, DoReMi is easily harmonised with MUSCIMA++ and DeepScores. Another limitation of several existing OMR datasets is that only one research stage is supported primarily, while other stages, such as reconstruction, are typically understudied. DoReMi addresses this issue by also providing semantic information of the written music. Given that DoReMi was generated using music notation software, we could retrieve musical information that otherwise would be impossible to obtain. Such data is available for playable notes (also grace notes), stave lines, clefs, slurs and ties. Depending on the nature of the element, different degrees of semantic information is provided. This data is integrated with visual data, such as bounding boxes and pixel information of each element. We expect this to facilitate further research in end-to-end deep learning systems in OMR, which are quickly becoming state-of-the-art. Furthermore, DoReMi includes different types of representations, including MusicXML, MIDI, MEI and PNG images of sheet music alongside OMR metadata.

II. Handwritten and Typeset OMR Datasets

Most of the existing OMR datasets target different approaches, stages and objectives within OMR research. Differences between datasets primarily lie in the annotations that accompany images of sheet music. There are also variations in the type of notation, such as CWMN or mensural notation, in primary focus, and differences in the printing type, i.e., typeset or handwritten scores. A list of the main datasets, engraving type, number of symbols and A4 images, formats provided and usage is given in Table I. Conditional on the final objective of the task, there are also different music representations used in datasets. For instance, we only need the MIDI file as the ground truth to retrieve a replayable-only file. If the graphical

Funded by UKRI, EPSRC and Steinberg.

elements and an editable score are needed, other more complex formats such as MusicXML and MEI are essential. The use case determines the optimal music representation. MEI and MusicXML are score focused, while MIDI focuses on the musical content and synthesiser control.

HOMUS (Handwritten Online Musical Symbols) [6] takes the approach of recording pen-based (online) compositions. They present the strokes drawn by pen and the image generated after drawing the symbols. The authors then propose a baseline on the recognition of these two modalities. To conclude the recognition accuracy, they use Nearest Neighbor (NN) technique and Hidden Markov Models (HMM).

CVC-MUSCIMA [8] is the root dataset of MUSCIMA++. CVC-MUSCIMA was originally designed to perform stave line removal. A total of 50 different musicians were asked to write 20 pages of identical sheet music, having the same pen and style. This dataset creates a link between low-level such as noteheads and higher-level symbols such as key and time signatures and is mainly used and best suited for musical object detection. Baseline experiments for object detection using MUSCIMA++ use detection algorithms such as Faster R-CNN [3], Single Shot Detectors [22] and DeepWaterShed Detectors [23]. A version of MUSCIMA++ containing annotations for measures and staves only exists for bar measure detection.

Another dataset that assists in the detection stage is DeepScores [9] which is the largest OMR dataset that contains annotated images of typeset scores used for object classification, detection and segmentation. There is variation provided by rendering the sheets using five different fonts. This dataset assists work in recognising tiny objects in large images. A new version of DeepScores has detailed annotations, increased number of annotated symbols, while also providing oriented bounding boxes for the symbols, a higher level of rhythm and pitch information that includes onset beat for all symbols and line position for noteheads and finally compatibility with MUSCIMA++ dataset [19].

Printed Images of Music Staves (PrIMuS) [10] is one of the few datasets that satisfy use cases in training end-to-end object recognition models. It includes 87,678 real-music sequences of notes, typically the first ones, in five different formats: MIDI, PNG, MEI, semantic and an encoding that contains the symbols and their positions, disregarding their musical meaning. Another version Camera-PrIMuS [11] includes images with distortion to simulate real-world imperfections.

Universal Music Symbol Collection [7] is a dataset that collects and combines symbols from HOMUS, MUSCIMA++, Audiveris OMR dataset, the Printed Music Symbols dataset, OpemOMR dataset and two sets from the group of Rebelo et al. [15] and Fornes et al. [16] that can be used to train classifiers. The symbols amount to 74,000 handwritten and 16,000 printed symbols. The objective was to create a universal, harmonised dataset that could assist in building a written music classifiers.

Limitations of each dataset lie in their differences in data types, formats, and narrow objectives. DoReMi goes one step further by allowing harmonisation with MUSCIMA++ and DeepScores while adding semantic and graphical information about the symbols. Furthermore, DoReMi uses five file formats, XML (with positions), MusicXML, PNG, MEI and MIDI, with complementary information. PNG images are complemented by metadata in an XML metadata file. MEI, MusicXML and MIDI all representing possible encoded outputs of OMR, depending on the task.

III. Dataset Description

In this section, we present DoReMi, an OMR dataset with typeset symbols. This dataset is designed to be compatible with MUSCIMA++ and DeepScores and serve as a reference for research in other stages of OMR. Notably, it helps research in the reconstruction stage of music semantics such as notes pitch, duration, beats and their relations to other elements in the score. Furthermore, we expect DoReMi to aid work towards an end-to-end OMR system in non-monophonic scores.

Music used to generate this dataset comes from a software test set provided by the Dorico team [1]. This test set includes a wider number of objects, classes and various cases of notations not normally seen in real-world music. About 600 files of the underlying material used to generate this dataset is copyright protected; therefore, we only include openly distributable scores in the final published dataset [2]. However, pre-trained models trained in the whole dataset will also be published. DoReMi includes around 6432 images of sheet music with nearly a million annotated objects which is $\frac{1}{50}$th the size of DeepScores and 42 times the size of MUSCIMA++. Each object on the page is annotated with category labels from 94 different classes. However, there is an emphasised class imbalance; stems and noteheads make up half of the annotated objects in the dataset. We also provide prepared subsets fulfilling different requirements on the number of pages, number of classes and the number of staves. Most of the images include one system per page; depending on the number of voices, they have one or more staves per page.

Following the organisation of MUSCIMA++, DoReMi has an OMR metadata file which includes bounding boxes of each element: top, left, width and height. It also includes the pixel mask for each element giving each object's pixels inside the bounding box. Additionally, DoReMi provides the relationships between primitives. It vaguely follows the Music Notation Graph (MUNG), which creates a graph representation of music notations. Inlinks and outlinks reference back and forth to the ID of the objects they are related to—for instance, a notehead half outlinks to a stem or a slur or both of them. Conversely, the stem inlinks to the notehead half.

As opposed to other existing datasets, DoReMi provides semantic information on playable notes and interpretive elements, see Appendix A. Certain (playable) objects are also annotated with a Dorico event ID which is a unique event identifier that provides additional information on how some objects are linked. For instance, notes like noteheads have information

[1] Dorico is a music notation software - https://new.steinberg.net/dorico
[2] https://github.com/steinbergmedia/DoReMi/releases

TABLE I
COMPARISON OF MAJOR OMR DATASETS PUBLISHED TO DATE

Dataset	Engraving	Symbols	Images	Classes	Format	Usage
DoReMi	Typeset	911771	6432	94	✓ML metadata, images, MIDI, MEI, MusicXML	Object Detection, Reconstruction and Encoding, End-to-end
Handwritten Online Musical Symbols (HOMUS) [6]	Handwritten	15200	-	32	Text-File	Symbol Classification (online + offline)
Universal Music Symbol Collection [7]	Typeset + Handwritten	90000	-	79	Images	Symbol Classification (offline)
MUSCIMA ++ [4]	Handwritten	91255	140	110	Images, Measure Annotations, MuNG	Symbol Classification, Object Detection and Measure Recognition
DeepScores [9], [19]	Typeset	100m	255,386	135	Images, XML	Symbol Classification, Object Detection, Semantic Segmentation
PrIMuS [11]	Typeset	87678	-	-	Images, MEI, simplified encoding, agnostic encoding	End-to-End Recognition
Capitan collection [20]	Handwritten	-	10230	30	Images, Text-File	Symbol Classification
Bounding Box Annotations of Musical Measures [21]	Typeset	940	24,329	-	Images	CSV, plain JSON and COCO

on the duration beats, onset beats, pitch octave, midi pitch code, normalised pitch step and an event ID. For elements such as clefs, our dataset provides an event ID, clef type, clef hotspot, clef required stave lines and clef stave position. Clef hotspot identifies the midi pitch that clef denotes, i.e. for treble clef is G4, as that is the pitch of the second stave line from the bottom. Clef required stave lines shows how many stave lines the clef needs. Time signatures include the event ID and its description, for example, 3/2 (h, 1+1+1) [3]. Flags, if they are part of grace notes, have a boolean value set to True. Slurs and ties have their event IDs, while barlines, rests, accidentals, augmentation dots, stems do not have such information. Beams do not have their event ID; instead, they have a list of the event IDs their respective noteheads have. Other types of data given are Dorico project files, MIDI files, PNGs, MusicXML and MEI. PNG files are binarised and provided with a resolution of 300 DPI and dimensions of 2475x3504 pixels. There is a possibility of creating lower or higher resolution images depending on limitations in computational expense. One OMR XML metadata file may be pointing back to multiple images. Each image has a reference page ID in the XML file. MIDI files included can be used as ground truth to OMR tasks where MIDI is the desired output. MusicXML [17] and MEI [14] are two file formats that we desire to output after encoding the reconstructed information. They were conceived for two different reasons. MusicXML was first proposed as a file format to ease digital sheet exchange in the music publishing industry [18]. MEI was born in the music research world to aid the storage of diverse music manuscripts. Both share similarities in the objects they encode and their file format being XML. MEI, beyond functionality in notation and page layout, also encodes information about the notation in a more structured and semantic way. In other words, MusicXML was designed for software rendering, while MEI captures more music semantics [24].

IV. BASELINE EXPERIMENTS

Object detection is a crucial stage of optical music recognition. It is concerned with localising and identifying objects of certain classes in a sheet music image. Using the DoReMi dataset, we propose a baseline in object detection to assess the benefits brought forward by the richer data in this task. Based on previous work from Pacha et al. [2], [6], [9], we use Faster R-CNNs as our central architecture. We also compare the results with those reproduced using MUSCIMA++ using the same architecture.

Detecting objects in sheet music is considered more challenging than many general-purpose computer vision tasks, given that the number of tiny objects is very high. To detect such objects, we need to localise objects that assist classification. This localisation is defined by four values b_x, b_y, b_h and b_w, while the first two terms determine the centre of the bounding box that isolates that object, the latter ones provide its height and width. The network also provides a class name for classification purposes. When objects to be detected are not overlapping, it is reasonably easy to obtain and interpret their bounding boxes. When two objects overlap in the same grid cell, having more than one midpoint in a cell, the network needs to know which object to predict. This overlap of objects is often seen in sheet music. To deal with overlaps, anchor boxes can be used. Objects are assigned to the respective grid cell and an anchor box for that grid cell with the highest Intersection over Union (IoU). We get two predicted bounding boxes for each of the grid cells then omit low probability object predictions. Subsequently, for each class in our dataset, we run non-max suppression to generate predictions. Non-max suppression eliminates the bounding boxes that have a low probability, retaining the bounding box with the highest probability.

Fast R-CNN [3] serves as a base for almost all proceeding work in object detection. Fast R-CNNs use selective search to generate region proposals, which is expensive. After region CNNs were introduced, Faster R-CNN became the state-of-

[3] Beat division: h means half note, 1 + 1 + 1 means 3 equal beat, 1,2,3

TABLE II
OBJECT DETECTION BASELINE RESULTS IN DOREMI AND MUSCIMA++

Meta-Architecture	Feature Extractor	Classes	Training steps	Data%	mAP (%)
DoReMi					
Faster R-CNN	Inception-ResNet-v2	71	80K	90%	57.5067
Faster R-CNN	Inception-ResNet-v2 with MUSCIMA++	71	120K	90%	64.8614
Faster R-CNN	ResNet50	71	120K	90%	63.4910
Faster R-CNN	ResNet101	71	148K	90%	26.994
MUSCIMA++					
Faster R-CNN	Inception-ResNet-v2	110	80K	100%	82.4

the-art approach in object detection [3]. Most of the work subsequent to it follows a similar architecture by adding other valuable blocks such as Mask R-CNNs, which also output the object masks indicating the pixels where the object is in the bounding box. Faster R-CNNs are very similar to Fast R-CNN, with the most significant difference being how the region proposals are considered and CNNs are run on those, which cut the run time. Faster R-CNNs introduce region proposal network (RPN), which enables sharing full-image convolution features with the detection network [2]. RPN is a fully convolutional layer, trained to generate good region proposals used by the rest of the network.

The experiments II are performed in a subset of the DoReMi dataset consisting of 5832 A4 images of scores. These images are of the same dimension throughout the dataset with the limitation of one system per page. These systems very often consist of multiple staves. The maximum number of staves per page is six. Moreover, files that had less frequent objects were disregarded for this set of experiment.

A. Feature Extractors

We first apply a convolutional feature extractor in all input images, so the highest-level features are retrieved. Based on the number of parameters and layer types, the processing time is highly affected. We show three types of feature extractors and their training time for reference. We use open source Inception Resnets (v2) [13] and Resnet50 and Resnet101 [12]. All use Tensorflow implementations. Inception Resnets (v2) is a blended extractor with 164 layers which benefits from optimisations of the residual connections and efficiency of Inception units. It replaces the filter concatenation stage of the Inception architecture with residual connections. Resnet50 is another feature extractor with a depth of 50 layers, meaning that it is considerably lighter than Inception Resnet (v2), hence faster. Resnet101 belongs to the same residual connection family with Resnet50 with more depth.

B. Results

A common technique to boost detection results is to feed the data into a pre-trained network using the image representations in intermediate layers. We use the respective pre-trained models in the general COCO dataset for each extractor, except for a Faster R-CNN meta-architecture that uses Inception Resnet (v2) with a pre-trained model on MUSCIMA++. This model provides the best mAP score, which can be a result of the same domain pre-training. Resnet50 performs very well, yielding a mAP of 63% as shown in Table from where we can also see that Resnet50 produce better results. Furthermore training time using Resnet50 extractors is half of that using Inception Resnet (v2) II.

V. DISCUSSIONS

Baseline work in object detection with deep learning uses pre-trained models such as Faster R-CNN, Fast R-CNN, SSD [1], with some fine-tuning during training. However, while such pre-trained models can detect musical objects, classes that are not well-represented pose a significant challenge. Creating custom models for OMR is one of the main challenges for both the detection and note assembly stage, given the graphical and structured nature of music. Such models would improve the state-of-the-art work in OMR, but it would further facilitate and trigger new research paths. The DoReMi dataset is designed in a way that it can be harmonised with Deepscore and MUSCIMA++, which allows various engravings and takes steps towards creating standardised, universal OMR datasets. The design can further help to standardise evaluation at different stages of OMR and facilitate end-to-end approaches in the field. The dataset design follows the mung (Music Notation Graph), where nodes represent primitives such as noteheads, stems, beams, and their relations being stored. The DoReMi dataset is the first step towards overcoming some of the challenges mentioned above. DoReMi in baseline experiments does not perform as good as MUSCIMA++, resulting from MUSCIMA++ images being cut by staff, meaning the number of objects is often smaller and of a larger dimension. While this dataset aims to fill in existing gaps in OMR research, it is also prepared with a view to investigate a more extensive research question, i.e., whether deep learning can assist the OMR research field by providing data in richer structure, so common tasks considered in isolation may be gradually integrated into more complex architectures. What sets DoReMi apart from other existing datasets primarily is data richness. Given that this dataset was generated using Dorico, we could obtain more musical information and different data types. Finally, the pipeline developed for assembling this dataset allows us to extend the dataset, as well as iterate and re-design the structure if needed based on the course of new research.

Appendix A
Dataset Samples and Individual Class Attributes

```xml
<Nodes>
	<Node>
		<Id>49</Id>
		<ClassName>noteheadBlack</ClassName>
		<Top>1017</Top>
		<Left>2242</Left>
		<Width>28</Width>
		<Height>24</Height>
		<Mask>0: 12 1: 11 0: 14 1: 16... </Mask>
		<Outlinks>160 161 194</Outlinks>
		<Data>
			<DataItem key="duration_beats" type="float">0.250000</DataItem>
			<DataItem key="onset_beats" type="float">5.750000</DataItem>
			<DataItem key="pitch_octave" type="int">4</DataItem>
			<DataItem key="midi_pitch_code" type="int">60</DataItem>
			<DataItem key="normalized_pitch_step" type="str">C</DataItem>
			<DataItem key="dorico_event_id" type="int">1890</DataItem>
			<DataItem key="staff_id" type="int">1</DataItem>
		</Data>
	</Node>
	<Node>
		<Id>206</Id>
		<ClassName>gClef</ClassName>
		<Top>896</Top>
		<Left>357</Left>
		<Width>56</Width>
		<Height>147</Height>
		<Mask>0: 35 1: 5 0: 50 1: 7 0:... </Mask>
		<Data>
			<DataItem key="dorico_event_id" type="int">1257</DataItem>
			<DataItem key="clef_type" type="str">kGClef</DataItem>
			<DataItem key="clef_hotspot" type="str">G4</DataItem>
			<DataItem key="clef_required_stave_lines" type="int">5</DataItem>
			<DataItem key="clef_stave_position" type="int">2</DataItem>
			<DataItem key="staff_id" type="int">1</DataItem>
		</Data>
	</Node>
	<Node>
		<Id>160</Id>
		<ClassName>stem</ClassName>
		<Top>905</Top>
		<Left>2267</Left>
		<Width>3</Width>
		<Height>121</Height>
		<Mask>0: 0 1: 363 </Mask>
		<Inlinks>49</Inlinks>
		<Data>
			<DataItem key="staff_id" type="int">1</DataItem>
		</Data>
	</Node>
</Nodes>
```

Fig. 1. Snippet of three different nodes in OMR XML data files

TABLE III
LIST OF 94 CLASSES SHOWING FREQUENCY OF APPEARANCE FOR EACH CLASS AND THEIR DATA ATTRIBUTES

Class	Freq	DE id	clef type	clef hs	clef RSL	clef SP	staff id	DE ids	grace note	DB	OB	PO	MPC	NPS	TSD	text
accidentalDoubleFlat	244						✓									
accidentalDoubleSharp	330						✓									
accidentalFlat	12705						✓									
accidentalKomaFlat	5						✓									
accidentalKomaSharp	5						✓									
accidentalNatural	11137						✓									
accidentalQuarterToneFlatStein	162						✓									
accidentalQuarterToneSharpStein	191						✓									
accidentalSharp	12908						✓									
accidentalThreeQuarterTonesFlatZimmermann	1						✓									
accidentalThreeQuarterTonesSharpStein	27						✓									
accidentalTripleFlat	1						✓									
accidentalTripleSharp	1						✓									
articAccentAbove	1477						✓									
articAccentBelow	1916						✓									
articMarcatoAbove	278						✓									
articMarcatoBelow	34						✓									
articStaccatissimoAbove	536						✓									
articStaccatissimoBelow	414						✓									
articStaccatoAbove	5018						✓									
articStaccatoBelow	6108						✓									
articTenutoAbove	823						✓									
articTenutoBelow	735						✓									
augmentationDot	2762															
barline	28142															
beam	52539						✓	✓								
cClef	1161	✓	✓	✓	✓	✓	✓									
dynamicFF	242						✓									
dynamicFFF	88						✓									
dynamicFFFF	4						✓									
dynamicForte	777						✓									
dynamicFortePiano	60						✓									
dynamicForzando	35						✓									
dynamicMF	1803						✓									
dynamicMP	185						✓									
dynamicPiano	3987						✓									
dynamicPP	1807						✓									
dynamicPPP	93						✓									
dynamicPPPP	13						✓									
dynamicRinforzando2	5						✓									
dynamicSforzato	236						✓									
dynamicSforzatoFF	6						✓									
dynamicText	110						✓									✓
fClef	3543	✓	✓	✓	✓	✓	✓									
flag16thDown	430						✓		✓							
flag16thUp	18245						✓		✓							
flag32ndDown	31						✓		✓							
flag32ndUp	7264						✓		✓							
flag64thUp	10						✓		✓							
flag8thDown	3672						✓		✓							
flag8thUp	8113						✓		✓							
gClef	12278	✓	✓	✓	✓	✓	✓									
gradualDynamic	7084						✓									
kStaffLine	160426						✓									
mensuralNoteheadMinimaWhite	9	✓					✓			✓	✓	✓	✓	✓		
noteheadBlack	247741	✓					✓		✓	✓	✓	✓	✓	✓		
noteheadDiamondWhole	8	✓					✓			✓	✓	✓	✓	✓		
noteheadDoubleWholeSquare	3	✓					✓			✓	✓	✓	✓	✓		
noteheadHalf	8648	✓					✓			✓	✓	✓	✓	✓		
noteheadTriangleUpBlack	27	✓					✓			✓	✓	✓	✓	✓		
noteheadTriangleUpHalf	3	✓					✓			✓	✓	✓	✓	✓		
noteheadWhole	1502	✓					✓			✓	✓	✓	✓	✓		
noteheadXBlack	81	✓					✓			✓	✓	✓	✓	✓		
ornamentMordent	7	✓					✓									
ornamentTrill	58	✓					✓									
ornamentTurn	6	✓					✓									
rest	4															
rest16th	29411															
rest32nd	10218															
rest64th	4															
rest8th	24124															
restHalf	1376															
restQuarter	13223															
restWhole	14382															
slur	13928	✓					✓									
stem	227889						✓									
systemicBarline	2078															
tie	8626	✓					✓									
timeSig1	2	✓					✓								✓	
timeSig2	1084	✓					✓								✓	
timeSig3	1203	✓					✓								✓	
timeSig4	2674	✓					✓								✓	
timeSig5	507	✓					✓								✓	
timeSig6	246	✓					✓								✓	
timeSig7	125	✓					✓								✓	
timeSig8	1661	✓					✓								✓	
timeSig9	79	✓					✓								✓	
timeSigCommon	206	✓					✓								✓	
timeSigCutCommon	113	✓					✓								✓	
timeSignatureComponent	443	✓					✓								✓	
tupletBracket	4970	✓					✓									
tupletText	4900	✓					✓									✓
unpitchedPercussionClef1	223	✓	✓	✓	✓	✓	✓									
wiggleTrill	45	✓					✓									

Acknowledgement

The authors acknowledge the support of the AI and Music CDT, funded by UKRI and EPSRC under grant agreement no. EP/S022694/1 and our industry partner Steinberg Media Technologies GmbH for their continuous support especially during the dataset creation.

References

[1] A. Pacha, K. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi and H. Eidenberger, "Handwritten Music Object Detection: Open Issues and Baseline Results," 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), Vienna, Austria, 2018, pp. 163-168, doi: 10.1109/DAS.2018.51.

[2] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

[3] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497.

[4] J. Hajič and P. Pecina, "The MUSCIMA++ Dataset for Handwritten Optical Music Recognition," 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017, pp. 39-46, doi: 10.1109/ICDAR.2017.16.

[5] J. Calvo-Zaragoza, J. Hajič Jr., and A. Pacha. 2020. Understanding Optical Music Recognition. ACM Comput. Surv. 53, 4, Article 77 (September 2020), 35 pages. DOI:https://doi.org/10.1145/3397499

[6] J. Calvo-Zaragoza and J. Oncina, "Recognition of Pen-Based Music Notation: The HOMUS Dataset," 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 2014, pp. 3038-3043, doi: 10.1109/ICPR.2014.524.

[7] A. Pacha and H. Eidenberger, "Towards a Universal Music Symbol Classifier," 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017, pp. 35-36, doi:

[8] Fornés, A., Dutta, A., Gordo, A., and Lladós, J. (2012). CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal. International Journal on Document Analysis and Recognition (IJDAR), 15(3), 243-251.

[9] Tuggener, L., Elezi, I., Schmidhuber, J., Pelillo, M., Stadelmann, T. (2018, August). DeepScores-a dataset for segmentation, detection and classification of tiny objects. In 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 3704-3709). IEEE.

[10] Calvo-Zaragoza, J., Rizo, D. (2018). End-to-end neural optical music recognition of monophonic scores. Applied Sciences, 8(4), 606.

[11] Calvo-Zaragoza, J., and Rizo, D. (2018, September). Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores. In ISMIR (pp. 248-255).

[12] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[13] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261, 2016.

[14] Roland, P. (2002, September). The music encoding initiative (MEI). In Proceedings of the First International Conference on Musical Applications Using XML (Vol. 1060, pp. 55-59).

[15] A. Rebelo, G. Capela, and J. S. Cardoso, "Optical recognition of music symbols" International Journal on Document Analysis and Recognition (IJDAR), vol. 13, no. 1, pp. 19–31, 2010.

[16] A. Fornes, J. Llados, and G. Sanchez, "Old Handwritten Musical Symbol Classification by a Dynamic Time Warping Based Method". Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 51–60.

[17] Good, M. (2001, December). MusicXML: An internet-friendly format for sheet music. In Xml conference and expo (pp. 03-04).

[18] Cunningham, S. (2004). Suitability of musicxml as a format for computer music notation and interchange. In Proceedings of IADIS Applied Computing 2004 International Conference, Lisbon, Portugal.

[19] Tuggener, L., Satyawan, Y. P., Pacha, A., Schmidhuber, J., and Stadelmann, T. (2020). The DeepScoresV2 dataset and benchmark for music object detection. In 25th International Conference on Pattern Recognition 2020 (ICPR'20), Online, 10-15 January 2021. IAPR.

[20] Calvo-Zaragoza, J., Rizo, D., and Quereda, J. M. I. (2016, August). Two (Note) Heads Are Better Than One: Pen-Based Multimodal Interaction with Music Scores. In ISMIR (pp. 509-514).

[21] 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". Late Breaking/Demo at the 20th International Society for Music Information Retrieval, Delft, The Netherlands, 2019.

[22] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., and Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[23] Tuggener, L., Elezi, I., Schmidhuber, J., and Stadelmann, T. (2018). Deep watershed detector for music object recognition. arXiv preprint arXiv:1805.10548.

[24] Shatri, E. and Fazekas, G., "Optical music recognition: State of the art and major challenges," in Proceedings of the International Conference on Technologies for Music Notation and Representation TENOR'20/21, R. Gottfried, G. Hajdu, J. Sello, A. Anatrini, and J. MacCallum, Eds. Hamburg, Germany: Hamburg University for Music and Theater, 2020, pp. 175–184.

Unsupervised Neural Document Analysis for Music Score Images

Francisco J. Castellanos, Antonio-Javier Gallego
Department of Software and Computing Systems
University of Alicante, Alicante, Spain
{fcastellanos, jgallego}@dlsi.ua.es

***Abstract*—Document analysis is a key step within typical Optical Music Recognition (OMR) systems. It processes the input image to obtain a layered version by extracting different sources of information. Recently, this task has been formulated as a supervised machine learning problem, where Convolutional Neural Networks perform particularly well. However, the requirement for a large amount of labeled training data makes this task a not straightforward problem. Domain Adaptation (DA) is the research field that aims to adapt in an unsupervised fashion the knowledge learned from domains where there is labeled data available to other domains for which labels are not available. In this work, we study the use of a well-known DA approach based on adversarial training, which we combine with Selectional Auto-Encoders to propose an unsupervised document analysis framework. The results show a relevant improvement when the layers' features are very different from those of other domains, whereas, when there are certain similarities between domains, such improvement is not achieved. Note, however, that in the latter case, it is possible to obtain good results without DA.**

***Index Terms*—Document Analysis, Neural Networks, Domain Adaptation, Music Score Images**

I. Introduction

Optical Music Recognition (OMR) is an automatic process that aims to read the musical notation of scanned music documents and export their content to a structured digital format [1]. Given the complexity of OMR, it typically is divided into a series of sequential tasks with specific processing goals. Document analysis is usually one of the most important tasks, in which the relevant elements that make up the image content are recognized and split into different layers, e.g. by classifying each pixel into a set of categories such as staff lines, music notes, lyrics, or background [2], as shown schematically in Figure 1.

Recent advances in machine learning, and particularly in Deep Neural Networks (DNNs), have opened opportunities to carry out OMR processes efficiently [3]. However, despite their high performance and the good results shown in multiple tasks, these approaches bring with them a new handicap: the need for training data. Indeed, this is a drawback commonly associated with machine learning, which requires the use of labeled data to learn from it. However, the large amount of musical manuscripts contributes to this not being an easy task, as labeling is a tedious and error-prone process.

Domain adaptation (DA) is the research field that studies how to adapt the knowledge learned from a labeled collection of data—the source domain—to apply it in an unsupervised manner to another related but different one—the target domain. In this way, a model is capable of processing target images without ground-truth information for that domain, so it is therefore not necessary to annotate images.

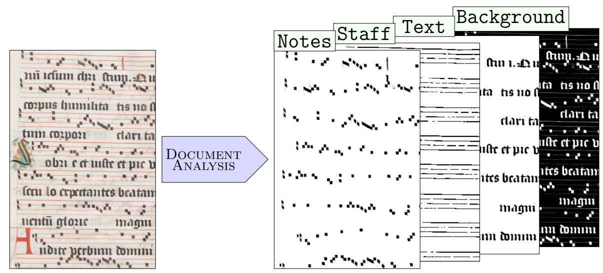


Fig. 1. Scheme of the document analysis task.

We propose an unsupervised layout analysis approach based on Selectional Auto-Encoders (SAEs) and adversarial training by means of Gradient Reversal Layer (GRL), which is inspired by a successful binarization approach [4]. We follow the idea of the framework presented in a previous work [5] to recognize different layers of information—staff lines, notes, lyrics, and background—but in this case without using labeled images to carry out the training process. The proposed approach is assessed through experiments with several corpora, obtaining a significant improvement over the previous method.

II. Related Work

Document analysis is a well-known stage within OMR [6], so that we can find multiple proposals for this task. Traditionally, this problem was addressed through smaller and consecutive processes. An example is binarization—used to classify foreground and background information—for which we can also find several proposals, including traditional algorithms [7]–[9] or even specific approaches for music documents [10], [11]. There are also works in which staves and lyrics are split so that they can be processed separately, such as [12], [13]. Another common step is the staff-line removal, where staff lines are eliminated to isolate the music symbols and make easier their classification. Dalitz et al. [14] reviews traditional methods, however, this is an active research field in which new work continually appears [15], [16].

More recently, document analysis has been formulated as a machine learning problem. Given its high performance and efficiency, the SAE architecture has been explored in several works. Although it was originally proposed to remove staff lines from musical manuscripts [17], it has been extended to perform binarization [18], and more recently, for layout analysis [5]. This latest work proposes a framework that detects the different layers of information by training a set of SAE models to recognize each layer separately. However, although these approaches are often aligned with high performance and generalizability, they entail a significant drawback derived from supervised learning: the need for labeled data.

DA aims to alleviate this issue by adapting the knowledge learned with a labeled manuscript to another but related unlabeled one in an unsupervised fashion. Within this field, adversarial training is a strategy in which different neural networks—or parts of them—are trained adversely to learn a common representation. A relevant example is Domain-Adversarial Neural Network (DANN) [19], which proposes a neural network based on Gradient Reversal Layer (GRL) for classification. This work has been recently extended to binarization [4], a task closely related to document analysis, and evaluated with several documents including some of music scores. However, the similarity between domains was decisive in the viability of applying DA. For this they compared the probability distributions provided by SAE.

In this work, we propose an extension to DA of the SAE-based framework previously proposed in [5]. Our method integrates the GRL into the SAE architecture in order to learn domain-invariant features, similarly to the proposal in [4]. The experiments focus on the study of how the similarity of domains can be an important aspect to use a DA strategy, which, as we will see, revealed that the result depends on the layer and the information to be detected.

III. Method

Let $\mathcal{S}$ be an annotated or *source* domain composed of a set of images with their corresponding ground truth $(\mathcal{X}_{\mathrm{S}}, \mathcal{Y}_{\mathrm{S}})$, where $\mathcal{X}_{\mathrm{S}}$ contains scanned document images $\mathcal{X}_{\mathrm{S}}^i = [0, 255]^{h_{\mathrm{s}}^i \times w_{\mathrm{s}}^i \times c}$, being $\mathcal{X}_{\mathrm{S}}^i$ the i-th image within $\mathcal{X}_{\mathrm{S}}$ with height h_{s}^i px., width w_{s}^i px. and c channels, and $\mathcal{Y}_{\mathrm{S}}$ standing for a pixel-wise binary class annotation of each $\mathcal{X}_{\mathrm{S}}^i$, with $\mathcal{Y}_{\mathrm{S}}^i = \{0, 1\}^{h_{\mathrm{s}}^i \times w_{\mathrm{s}}^i}$, where 1 means the considered layer and 0 the rest.

Let $\mathcal{T}$ be a non-annotated or *target* domain that consists of a set of images $\mathcal{X}_{\mathrm{T}}$, whose k-th image is defined as $\mathcal{X}_{\mathrm{T}}^k = [0, 255]^{h_{\mathrm{t}}^k \times w_{\mathrm{t}}^k \times c}$ for which no labeling is available.

Our proposal addresses the problem of how to perform document analysis for the target domain since, as the images are not labeled, it has to be done in an unsupervised way. Specifically, the proposed method performs a pixel-wise classification of the images in the set $\mathcal{T}$, thus separating the image pixels into different information layers. The categories considered are the same as in [5]: `staff lines`, `notes`, `text` and `background`. In this previous work, a framework composed of a set of SAE was proposed to process each of these categories separately, with the aim of creating specialized classifiers that detect each layer of information to eventually be processed or even combined.

We extend the method previously proposed in [5] for layout analysis, to allow its use with new target domains without labeled data. Specifically, this method is modified to integrate the DA technique proposed in [4] for binarization, with the aim of adapting the knowledge learned with the source domain to the target domains. This approach uses adversarial training by means of a GRL in order to penalize those domain-specific features that allow differentiating the domains [19]. The goal is to obtain a neural network model capable of processing images from $\mathcal{S}$ or $\mathcal{T}$ indistinctly. Note that GRL includes the hyper-parameter λ to adjust the contribution of the domain classifier during the training process, which will be studied empirically.

The scheme of Figure 2 shows an example of the architecture proposed for document analysis for a single layer—music symbols. This same architecture is repeated for the rest of layers. The idea is to use independent SAEs trained with the ground truth of a specific layer, thus creating specialized classifiers per layer. In line with the previous work, we therefore perform layout analysis using four SAE models, one for each information layer. Note that this architecture allows the addition of new categories easily.

The SAE models are trained by patches, so that each image is split into chunks of $h \times w \times c$ px. This achieves better results by not scaling the image, since each chunk can be processed at the original scale. To balance the $\mathcal{S}$ and $\mathcal{T}$ training sets, N samples are randomly selected for each domain. Note that SAE provides a probabilistic map with each pixel indicating the probability of belonging to the layer. Therefore, a threshold—calculated with $\mathcal{S}$—is necessary to obtain the binary image.

In addition, as previously argued, depending on the similarity of the domains, it may not be necessary to apply the DA process. To assess this similarity, we use the metric proposed in [4], that compares the probabilistic maps returned by the SAE model for $\mathcal{S}$ and $\mathcal{T}$. These maps are converted into histograms as two one-dimensional vectors with a precision of 0.1, and normalized according to the number of pixels. The comparison is finally made by the Pearson's correlation [20].

IV. Experiment

A. Corpora

For the experiments, we selected three corpora manually labeled for the considered layers.

- SALZINNES: set of 10 music score images in Neumatic notation with an average resolution of $5\,847 \times 3\,818$ px., of Salzinnes Antiphonal (CDM-Hsmu2149.14)[1]
- EINSIEDELN: collection of 10 music documents in Neumatic notation, specifically those of Einsiedeln, Stiftsbibliothek, Codex 611(89)[2] with an average size of $6\,496 \times 4\,872$ px.

[1] https://cantus.simssa.ca/manuscript/133/
[2] http://www.e-codices.unifr.ch/en/sbe/0611/

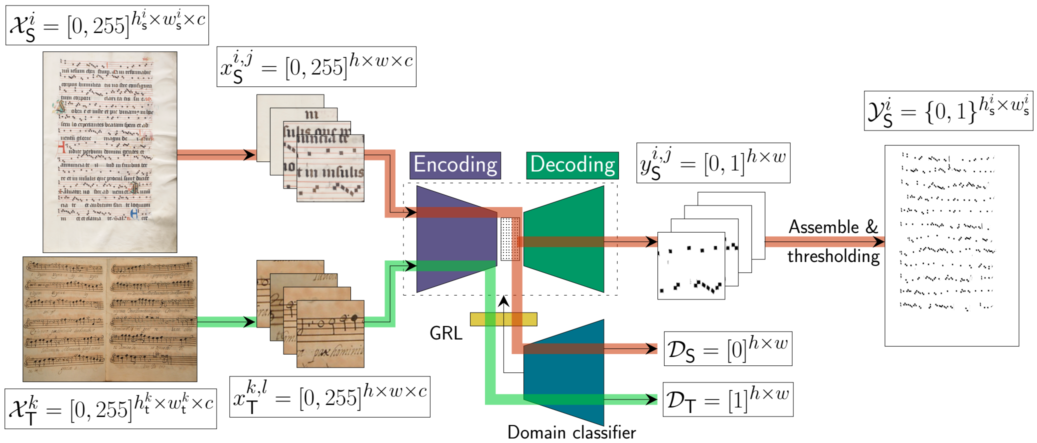


Fig. 2. Scheme of the DA approach for layout analysis. The figure shows the architecture used for the classification of "notes", this same scheme is repeated for the rest of the categories considered.

- CAPITAN: 10 images from a complete *Missa* of the second half of the 17th century [21] in Mensural notation with an average size of $2\,126 \times 3\,065$ px.

B. Metrics

Since the distribution of the classes is unbalanced, F-score (F_1) was considered for the evaluation of the method. In a two-class problem, this metric is defined as:

$$F_1 = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}}, \tag{1}$$

where TP, FP, and FN stand for *True Positives* or correctly classified elements, *False Positives* or type I errors, and *False Negatives* or type II errors, respectively. However, since this is multi-class problem, we report the results in terms of F_1 for each class and macro F_1 [22] for a global evaluation, which is calculated as the average of the F_1 obtained for each class.

C. Hyper-parameterization

The specific topology of the SAE and the layers used for DA can be very varied. For this reason, we will use a configuration based on those proposed in [4] and [5]. Table I details the topology considered for each SAE. Note that, also for a fair comparison with the previous methods, the input image is provided in grayscale, although the color space could be used in this approach as well. According to the proposed architecture, GRL is connected before the last convolutional block of the decoder (see Table I) using $\lambda = 0.01$ with increments of 0.001 per epoch. We consider to extract $10\,000$ grayscale samples per domain of 256×256 px.

As explained above, SAE models are trained with a set of patches extracted from the images to obtain a probabilistic map whose pixels indicate the probability of belonging to a specific class. These patches are randomly cropped from the original image before each training epoch for both $\mathcal{S}$ and $\mathcal{T}$. However, as it is an unsupervised task, we only have ground truth for $\mathcal{S}$. This ground truth is provided as output in the decoder part of the SAE (see Figure 2), while to train the other part of the network (the domain classifier, according to this same figure), it is only necessary to provide the domain of the input image, that is, source domain or target domain.

D. Results

The experiments were conducted with the aim of studying the benefits of using a DA strategy for document analysis. The corpora considered contain different music notations: Neumatic (SALZINNES and EINSIEDELN) and Mensural (CAPITAN). Therefore, we will focus on evaluating the adaptation process between these two types of notations, considering the following combinations of domains: SALZINNES→CAPITAN, EINSIEDELN→CAPITAN, CAPITAN→SALZINNES, and CAPITAN→EINSIEDELN.

Table II shows the average results for each information layer of the different combinations of domains considered. In addition to the result of the proposed method (SAE-DANN column), it also shows that of the state of the art (SAE), and the value of the upper bound that would be obtained by selecting the best option between SAE and SAE-DANN for each sample. As seen, our approach achieves the best results for almost all layers. For the staff layer the result is slightly worse compared to not applying DA, which may be due to the fact that the staff lines are very similar in all domains. As we argued before, when the domains are very similar, it may not be appropriate to apply DA, since it is based on an unsupervised process that searches domain-invariant features and that can hinder the training in the case discussed. In fact, if we check the result of this layer, we can see that the SAE model obtains a 76.0%, which is very close to the upper bound (77.8%). This means that both domains have similar features, since the SAE method, without applying the adaptation, obtains a result very close to the theoretical maximum that could be obtained by training in a supervised way.

Once analyzed the benefits of using the GRL in combination with SAE, we shall now report the experiments in which the domain similarity metric is used to determine whether DA

TABLE I

DESCRIPTION OF THE SAE ARCHITECTURE CONSIDERED, IMPLEMENTED AS A FULLY-CONVOLUTIONAL NETWORK (FCN). CONCERNING THE NOTATION, CONV(f,h,w,a) REPRESENTS A CONVOLUTION OPERATOR OF f FILTERS, A KERNEL WITH A SIZE OF $h \times w$ PIXELS, AND AN a ACTIVATION FUNCTION; MAXPOOL(h,w) INDICATES A MAX-POOLING OPERATOR WITH A $h \times w$ KERNEL; UPSAMP(h, w) STANDS FOR AN UP-SAMPLING OPERATOR OF $h \times w$ PX.; *ReLU* AND *Sigmoid* DENOTE RECTIFIER LINEAR UNIT AND SIGMOID ACTIVATIONS, RESPECTIVELY.

Input	Encoding	Decoding	Output
$[0, 255]^{256\times256}$	Conv(64,3,3,ReLU) MaxPool(2,2)	Conv(64,3,3,ReLU) UpSamp(2,2)	$[0, 1]^{256\times256}$
	Conv(64,3,3,ReLU) MaxPool(2,2)	Conv(64,3,3,ReLU) UpSamp(2,2)	
	Conv(64,3,3,ReLU) MaxPool(2,2)	Conv(64,3,3,ReLU) UpSamp(2,2)	
		Conv(1,3,3,Sigmoid)	

TABLE II

AVERAGE RESULTS IN TERMS OF F_1 (%) FOR EACH INFORMATION LAYER. THESE EXPERIMENTS WERE CONDUCTED FOR ADAPTATION BETWEEN DOMAINS WITH NEUMATIC AND MENSURAL NOTATIONS. THE BEST RESULTS ARE MARKED IN BOLD TYPE.

Layer	SAE	SAE-DANN	Upper bound
Staff	**76.0**	72.1	77.8
Note	37.5	**42.7**	46.2
Text	10.5	**25.8**	25.8
Background	67.2	**68.0**	73.1
Avg.	47.8	**52.1**	55.7

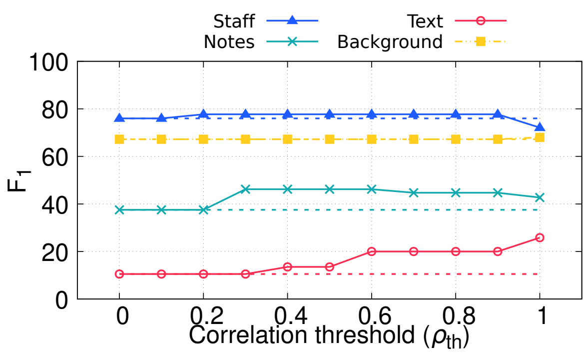


Fig. 3. Study of the impact of the correlation threshold to determine whether to apply the DA process. The dashed curves indicate the results of the state-of-the-art model (SAE).

should be applied. Figure 3 shows the impact of the correlation threshold $\rho_{S,T}$ for the document analysis process. The figure shows the result obtained (F_1, vertical axis) when applying this process with the different possible threshold values (horizontal axis). As seen, for low thresholds, the result coincides with that of SAE, but as the threshold increases and the proposed method is applied, this result improves. In some cases (staff and notes), when the threshold is very high, the result worsens slightly. This is because, as we have already argued, when trying to apply DA for very similar domains, we force an unsupervised training process that worsens the result.

For some classes, the improvement obtained by applying DA is minimal, such as for background and staff lines. However, the proposed process improves the average result obtained by the state of the art (as shown previously in Table II). Besides, the threshold $\rho_{S,T}$ could be set between 0.6 and 0.9 (the range with the best results for all classes according to Figure 3), thus automatically selecting whether to apply the adaptation process.

V. CONCLUSIONS

The growing interest of the musicology community in automatically reading music notation from image scores, either to preserve or disseminate this valuable heritage, has promoted the development of systems for music retrieval. Document analysis is one of the most common steps in the traditional OMR workflow that aims to process the images to facilitate the next steps within the pipeline.

This work presents an extension of a previous layout analysis proposal [5] that aims to allow its use in an unsupervised way. In our case, the method is combined with a domain adaptation technique with which to process new documents without using labeled data. Besides, the similarity between domains is also measured using the Pearson's correlation to determine the appropriateness of carrying out the adaptation.

The results obtained show that the proposed method improves the average result of the state of the art. These results show that, depending on the type of layer, it may not be appropriate to carry out the adaptation process. This is because the unsupervised adaptation process is based on the similarity of the domains, since it searches domain-invariant features. So when these are very similar, the adaptation process actually hinders the training, obtaining worse results. When evaluating the results based on this similarity, it is observed that it is possible to set a threshold to automatically determine whether to apply the adaptation process or to directly use the method without adaptation.

Given the good results obtained, as future work we intend to extend these experiments by carrying out a much more complete study, with more domains, and also evaluating the similarity at the sample level rather than at the domain level.

ACKNOWLEDGMENT

This work was supported by the Spanish Ministry HISPAMUS project TIN2017-86576-R, partially funded by the EU, and by the University of Alicante project GRE19-04. The first author also acknowledges support from the "Programa I+D+i de la Generalitat Valenciana" through grant ACIF/2019/042.

REFERENCES

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

[2] Jorge Calvo-Zaragoza, Francisco J. Castellanos, Gabriel Vigliensoni, and Ichiro Fujinaga. Deep neural networks for document processing of music score images. *Applied Sciences*, 8(5):654, 2018.

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

[4] Francisco J. Castellanos, Antonio-Javier Gallego, and Jorge Calvo-Zaragoza. Unsupervised neural domain adaptation for document image binarization. *arXiv preprint arXiv:2012.01204*, 2020.

[5] Francisco J. Castellanos, Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. Document analysis of music score images with selectional auto-encoders. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 256–263, 2018.

[6] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, André R. S. Marçal, 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.

[7] Jaakko Sauvola and Matti Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[8] Basilios Gatos, Ioannis Pratikakis, and Stavros J. Perantonis. Adaptive degraded document image binarization. *Pattern Recognition*, 39(3):317–327, 2006.

[9] Nicholas R. Howe. Document binarization with automatic parameter tuning. *International Journal on Document Analysis and Recognition*, 16(3):247–258, 2013.

[10] Telmo Pinto, Ana Rebelo, Gilson A. Giraldi, and Jaime S. Cardoso. Music score binarization based on domain knowledge. In *5th Iberian Conference on Pattern Recognition and Image Analysis, Las Palmas de Gran Canaria, Spain*, pages 700–708, 2011.

[11] Quang Nhat Vo, Soo Hyung Kim, Hyung Jeong Yang, and Gueesang Lee. An MRF model for binarization of music scores with complex background. *Pattern Recognition Letters*, 69(Supplement C):88–95, 2016.

[12] John Ashley Burgoyne and Ichiro Fujinaga. Lyric extraction and recognition on digital images of early music sources. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 723–728, 2009.

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

[14] Christoph Dalitz, Michael Droettboom, Bastian Pranzas, and Ichiro Fujinaga. A comparative study of staff removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):753–766, 2008.

[15] Jaime dos Santos Cardoso, Artur Capela, Ana Rebelo, Carlos Guedes, and Joaquim Pinto da Costa. Staff detection with stable paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1134–1139, 2009.

[16] Thierry Géraud. A morphological method for music score staff removal. In *International Conference on Image Processing*, pages 2599–2603, 2014.

[17] Antonio-Javier Gallego and Jorge Calvo-Zaragoza. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications*, 89:138–48, 2017.

[18] Jorge Calvo-Zaragoza and Antonio-Javier Gallego. A selectional auto-encoder approach for document image binarization. *Pattern Recognition*, 86:37 – 47, 2019.

[19] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[20] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*, pages 1–4. Springer, 2009.

[21] Jorge Calvo-Zaragoza, Alejandro Héctor Toselli, and Enrique Vidal. Handwritten music recognition for mensural notation: Formulation, data and baseline results. In *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15*, pages 1081–1086, 2017.

[22] Arzucan Özgür, Levent Özgür, and Tunga Güngör. Text categorization with class-based and corpus-based keyword selection. In *International Symposium on Computer and Information Sciences*, pages 606–615, 2005.

Handwritten Historical Music Recognition through Sequence-to-Sequence with Attention Mechanism

Arnau Baró*, Carles Badal†, Pau Torras* and Alicia Fornés*
*Computer Vision Center - Computer Science Department
†Art and Musicology Department
Universitat Autònoma de Barcelona, Bellaterra, Catalonia, Spain
Email: abaro@cvc.uab.cat, carles.badal@uab.cat, pau.torras@e-campus.uab.cat, afornes@cvc.uab.es

***Abstract*—Despite decades of research in Optical Music Recognition (OMR), the recognition of old handwritten music scores remains a challenge because of the variabilities in the handwriting styles, paper degradation, lack of standard notation, etc. Therefore, the research in OMR systems adapted to the particularities of old manuscripts is crucial to accelerate the conversion of music scores existing in archives into digital libraries, fostering the dissemination and preservation of our music heritage. In this paper we explore the adaptation of sequence-to-sequence models with attention mechanism (used in translation and handwritten text recognition) and the generation of specific synthetic data for recognizing old music scores. The experimental validation demonstrates that our approach is promising, especially when compared with long short-term memory neural networks.**

I. Introduction

The recognition of music scores is a classical research field within the document image analysis and recognition community. Optical Music Recognition (OMR) [1], [2], [3] consists of converting images of music scores into a digital format, such as MEI, MusicXML, MIDI, etc. for further processing, study or publication, among others.

OMR has reached very good performance on scanned printed music scores recently, especially for monophonic scores. However, the recognition of handwritten scores is still a challenge due to the high degree of variability in handwriting styles, which becomes even worse in the context of historical scores due to paper degradation, the frequent appearance of touching elements and the lack of a standard notation system. Besides, the availability of labelled datasets of old handwritten music scores is scarce, which hinders the training of deep-learning based architectures. Given the amount of historically relevant music scores stored in old archives and churches, it becomes essential to have robust tools to transcribe them in order to guarantee their conservation and study.

For the above reasons, in this paper we propose an OMR system for old handwritten scores. Our method is based on a Sequence-to-Sequence model with an attention mechanism, which has been successfully applied to translation and handwritten text recognition. Also, and since the lack of available transcribed scores for training deep learning systems pose a challenge, we also generate specific synthetic data that emulates the particularities of old scores (e.g. lyrics touch the stave or even the musical symbols.). The experiments demonstrate the suitability of our approach, especially when compared to Long Short-Term Memory Recurrent Neural Networks.

The contributions are: 1) The adaptation of a sequence-to-sequence model with attention for historical music recognition. 2) A novel synthetic data generation, emulating old handwritten scores. 3) The historical handwritten music dataset is made available [1].

II. Related Work

In this section we describe the most relevant approaches in Optical Music Recognition related to our work.

For decades, the problem of OMR has been tackled through traditional techniques. For example, and since monophonic scores follow a sequential structure, Hidden Markov models have been applied [4], [5]. Other works are based on symbol segmentation and recognition [6], [7], [8]. Since errors or ambiguities are frequent, grammars or syntactic rules[9], [10] are used to minimize them.

However, during the last years OMR performance has significantly improved thanks to Deep Learning architectures. We can mention the attention-less Sequence to Sequence model of Van der Wel and Ullrich[11], the long short-term memory recurrent neural networks (BLSTMs) of Calvo-Zaragoza *et.al.*[12], or the segmentation and classification models proposed by Wen *et.al.*[13], which have been successfully applied to printed music scores.

Nevertheless, there is little research for handwritten scores, and the few existing approaches are focused on Western music notation, such as the well-known MUSCIMA++ dataset [14], [15]. Methods using this dataset include Pacha *et.al.*'s [16] detection of music primitives through deep convolutional neural networks (CNNs), Baró *et.al.*'s [17] CNN and BLSTM approach, and Tuggener *et.al.*'s use of ResNets[18]. Not many fully annotated datasets exist, which can be palliated with the use of synthetic samples for training [17].

For these reasons, and inspired by the success of Sequence to Sequence models that incorporate attention mechanisms, we explore their adaptation to the recognition of handwritten historical scores, which we believe has never been attempted before.

[1] http://www.cvc.uab.es/people/abaro/datasets.html

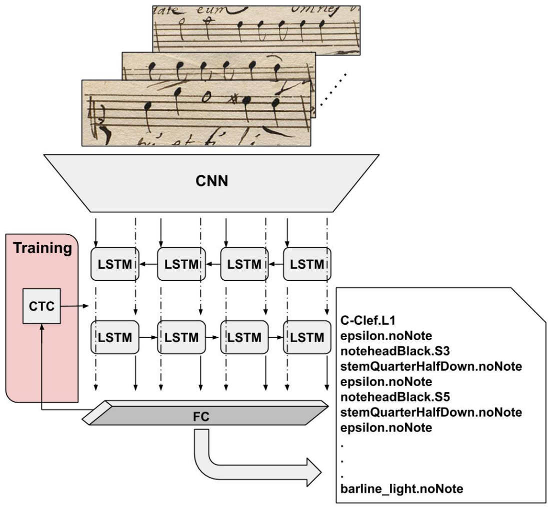


Fig. 1. Convolutional Neural Network and Bidirectional Long Short-Term Memory model.

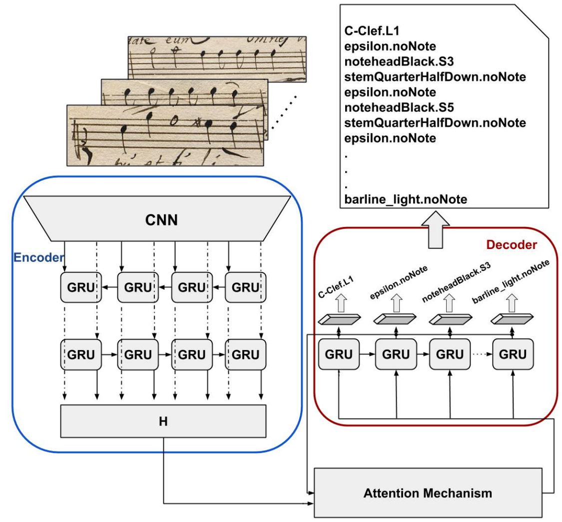


Fig. 2. Sequence-to-sequence model with attention mechanism.

III. Baseline: Long Short-Term Memory Model

Before describing our Seq2Seq OMR architecture, we first describe our baseline, based on Long Short-Term Memory Recurrent Neural Networks [12], [17]. Note that our baseline is based on recurrent models because of the sequentiality of monopohonic music staves.

Long short-term memory networks can work on the raw image directly. Nevertheless, performance improves when adding a Convolutional Neural network (CNN) as a feature extractor. Thus, our baseline is composed of a Convolutional Neural Network and a bidirectional Long Short-Term Memory neural network (BLSTM) with Connectionist Temporal Classification (CTC) loss. Figure 1 shows the model architecture. The modules are described next.

- **Convolutional Network:** This layer extracts the features that will be used in further steps. The convolutional network is composed of the first three layers of the ResNet18 [19], consisting of convolution, batch normalization and ReLU activation.
- **Bidirectional LSTM:** The BLSTM gets as input the features from the CNN. We use a LSTM to reduce the vanishing gradient problem since LSTMs can remember information for longer time. We use bi-directonal LSTMs to increase context information (from left and right sides in the image) and reduce the number of ambiguities.
- **Fully connected layers:** The results obtained by the BLSTM network are passed to a fully connected layer to return the final result.
- **Connectionist Temporal Classification:** This step helps to evaluate the output and check that the predictions are correct. As a loss function we use the Connectionist Temporal Classification (CTC) [20], which is trained using Stochastic Gradient Descent (SGD) optimizer with Momentum.

IV. Architecture: Sequence to Sequence model

As explained before, music scores are written on staves following a sequence, so our approach is also based on recurrent models. Concretely, our method is based on the sequence-to-sequence (seq2seq) text recognition method [21], and adapted to music scores.

A. Sequence-to-Sequence model with attention mechanism

This methodology makes use of an attention-based encoder-decoder framework consisting of 3 components: the encoder, the attention mechanism and the decoder. Figure 2 depicts our proposed architecture for optical music recognition.

- **Encoder.** Given an input image, the encoder extracts high-level features encoding the contents of the image. In this work, the proposed encoder is implemented with a VGG-19-BN network [22] without the last max-pooling layer and with pre-trained weights from ImageNet. Finally, the VGG features are reshaped into a two-dimensional feature map that is fed into a multi-layered BGRU for extra positional information.
- **Attention Mechanism.** As an attention module, we use a location-based attention as proposed by Chorowski *et.al.* [23]. The attention mechanism is in charge of aligning our feature representations with our decoding steps.
- **Decoder.** The decoder module is formed by a one-directional multi-layered GRU. At each time step the decoder GRU receives the concatenation of its previous embedding vector (in step $i-1$) and the current context vector (defined by the encoded features and our attention mechanism) in order to predict an output symbol. To

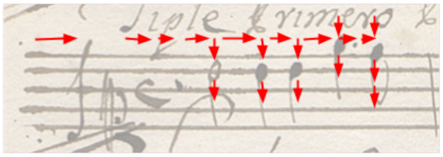


Fig. 3. Example of the labelling of the groundtruth, creating a 1D sequence. The transcription is written reading each measure from left to right, and from top to bottom if the symbol is divisible into primitives.

enhance the decoder's performance we have used, on the one hand, a multinomial layer that takes into account several decoding paths to obtain the final prediction and, on the other hand, label smoothing that allows better generalization by preventing overconfident predictions.

B. Adaptation to music scores

Music is a two-dimensional notation system. As such, a flattening step needs to be performed for musical data to be used within the seq2seq model. For this work, music scores have been annotated at primitive level (i.e., note heads, stems, beams, flags, rests, etc.). The problem of serialization can then be solved by defining a reading order, from left to right and from top to bottom, as illustrated in Figure 3. When moving horizontally through the score one step we use the epsilon (ϵ) symbol as a separator, as denoted by horizontal arrows. Otherwise, if primitives belong to the same symbol, they appear back-to-back as denoted using vertical arrows.

V. Dealing with the lack of data

Deep learning methods need a lot of labelled data to train. Since the amount of historical labelled data is scarce, we must look for alternatives. Therefore, we have generated two synthetic datasets using Lilypond [2]. Each one contains about 30,000 bar images, and are divided into 60% train, 20% validation and 20% test. These two datasets are complementary: one simulates the particularities of historical scores, whereas the other provides examples of a large diversity of symbols, including polyphony. These datasets are described next:

- Old synthetic (Fig. 4): This dataset tries to imitate the texture and degradation of the paper of historical scores adding a background. Thus, it is monophonic, and the type and diversity of symbols is limited to those seen in real historical samples.
- Modern synthetic (Fig. 5): This dataset contains polyphonic symbols written in one staff. This data will allow our model to generalize to any kind of historical music score, either monophonic or poliphonic.

[2]https://lilypond.org/

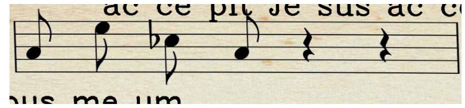


Fig. 4. Example image from the old synthetic dataset.

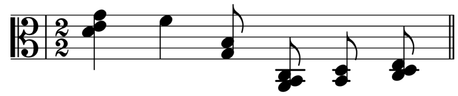


Fig. 5. Example image from the modern (polyphonic) dataset.

These synthetic datasets are used to pretrain our system. We train our model using curriculum learning [24] to improve performance: we initially train with a high proportion of modern polyphonic samples over old synthetic ones, and as training progresses we increase the proportion of the latter up to a 100% over the total.

VI. Experimental Validation

A. Historical Dataset

The historical data used in the experimental validation is a motet composed by Pau Llinàs, a catalan musician who worked as chapel master in Santa Maria del Pi of Barcelona between 1709 and 1749, time during which this work was most likely written. This religious motet (psalm number 148: Laudate Domine - Praise the Lord) is preserved in 12 separate parts, instead of a full score. It belongs to the *Fons Musical de la Catedral de Barcelona* and has been incorporated in the *Biblioteca Nacional de Catalunya* (BNC) catalogue [3].

For our experimental validation, we have manually labeled 40 music staves, containing 245 measure images. These are divided into 147 measures for training, 49 for validation and 49 for test. Figure 6 shows a page from this historical dataset, illustrating their main difficulties. On the first staff we can observe that the lyrics are touching the staff and in some cases even the symbols. Also, at the end of the third staff or at the beginning of the fifth staff the are ink stains.

B. Results on historical music scores

We have used the Symbol Error rate (SER) metric to evaluate our approach, which consists of the sum of insertions, removals and substitutions required to get the groundtruth sequence from the predicted sequence divided by the length of the groundtruth sequence. As it is a metric that evaluates error, lower means better.

Next, we evaluate our sequence-to-sequence architecture, and compare with the baseline described in Section III.

Table I shows the comparison between our Seq2Seq model and the baseline model using Convolutional Neural Network and Bidirectional Long Short Term Memory Neural Networks with Connectionist time classification (CNN+BLSTM) [17]. The first column indicates the method used, the second column

[3]http://mdc.csuc.cat/cdm/landingpage/collection/musicatedra

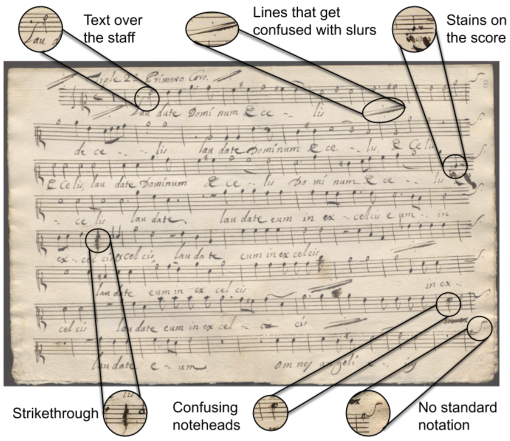


Fig. 6. Page example from the historical dataset.

TABLE I
QUANTITATIVE RESULTS COMPARING THE CNN+BLSTM MODEL AND OUR SEQUENCE-TO-SEQUENCE (SEQ2SEQ) MODEL.

Architecture	**Dataset Train**	**Test SER (%)**
CNN+ BLSTM	Historical	56.20
	Modern Synthetic	96.20
	Old Synthetic	75.20
	Modern + Old Synthetic	74.40
Seq2Seq	Historical	40.39
	Modern Synthetic	83.80
	Old Synthetic	61.89
	Modern + Old Synthetic	60.69

indicates which dataset has been used for training and the third column indicates the percentage of Symbol Error Rate (SER). From the Table I, we can observe that, in all setups, the Seq2Seq outperforms the BLSTMs by a large margin. As expected, the best result is obtained when training with real historical data, even though the amount of real labelled data is very low. We also observe that training with the modern synthetic dataset leads to a very low performance. However, if we train with the old synthetic dataset, we can reduce the SER by 20 points. Finally, if we combine both synthetic datasets (50% modern and 50% old), there is more varied data during training, so the methods obtain a slightly better SER.

Given that the best results are obtained using our proposed Seq2Seq approach combining both synthetic datasets, we have performed a second experiment using both real and synthetic samples for training. As explained in Section V, we use curriculum learning to train with easy examples first, and gradually incorporate more difficult ones. Table II shows how we have modified the percentage of historical and synthetic data at training time. The first four columns of the table shows the percentage of measures used for training and validation for each dataset, whereas the last column shows the SER on the real historical test set. We start the first epochs (see the first row) with few historical samples and a high percentage of synthetic ones. Every 10 epochs we increase the percentage of real data, while decreasing the amount of synthetic samples.

TABLE II
RESULTS USING OUR SEQ2SEQ MODEL WITH CURRICULUM LEARNING. WE SHOW THE AMOUNT OF DATA OF EACH KIND USED DURING TRAINING.

Percentage in Training (%)		**Percentage in Validation (%)**		**Test SER (%)**
Historical	**Modern+Old Syn.**	**Historical**	**Modern+Old Syn.**	
10	90	100	0	60.03
40	60	70	30	66.20
60	40	50	50	43.38
80	20	30	70	37.86
90	10	20	80	34.56
100	0	10	90	**31.79**

To minimize overfitting and given that the amount of synthetic scores is much higher than the number of historical ones, in the validation set, we do exactly the opposite: we started with a high percentage of historical samples, which decreases during training. At the end of the training phase, the training set has mainly historical data whereas the validation set has mainly synthetic one.

From the results reported in Table II, we can conclude that training with real and synthetic data highly benefits the overall system performance. Indeed, the obtained SER of 31.79% is significantly lower than the SER of 40.39% that was obtained when training with historical data only, as shown in Table I.

VII. CONCLUSIONS AND FUTURE WORK

In this work we have proposed a sequence-to-sequence architecture with attention mechanism for recognizing historical handwritten music scores. We have experimentally demonstrated that our model obtains promising results, especially compared to Bidirectional Long Short-Term Memory networks. We have also shown that the generation of specific synthetic data that simulates old scores is beneficial. In this sense, we have demonstrated that curriculum learning can gain leverage from the combination of real and synthetic data, improving the overall performance.

Nevertheless, the difficulties of historical scores in terms of paper degradation, touching lyrics and music symbols as well as the lack of annotated data still pose a challenge for optical music recognition. Concerning this last issue, we believe that the research community can benefit from our three labelled datasets, which will be publicly available.

As future work we plan to tackle polyphonic scores and improve our Seq2Seq architecture by exploring the incorporation of language models and domain adaptation techniques.

ACKNOWLEDGEMENT

This work has been partially supported by the Spanish projects RTI2018-095645-B-C21 and PGC2018-098986-B-C33, the Ramon y Cajal Fellowship RYC-2014-16831, the CERCA Program/ Generalitat de Catalunya, and the FI fellowships AGAUR 2020 FI_B2 00021 and FI_B2 00149 (with the support of the Secretaria d'Universitats i Recerca of the Generalitat de Catalunya and the Fons Social Europeu). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] 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." *IJMIR*, vol. 1, no. 3, pp. 173–190, 2012.

[2] A. Fornés and G. Sánchez, "Analysis and recognition of music scores," in *Handbook of Document Image Processing and Recognition*. Springer-Verlag London, 2014, pp. 749–774.

[3] J. Calvo-Zaragoza, J. H. Jr, and A. Pacha, "Understanding optical music recognition," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–35, 2020.

[4] L. Pugin, "Optical music recognition of early typographic prints using hidden markov models," in *ISMIR*, 2006, pp. 53–56.

[5] L. Pugin, J. A. Burgoyne, and I. Fujinaga, "Map adaptation to improve optical music recognition of early music documents using hidden markov models," in *ISMIR*, 2007, pp. 513–516.

[6] A. Fornés, J. Lladós, G. Sánchez, and D. Karatzas, "Rotation invariant hand drawn symbol recognition based on a dynamic time warping model," *IJDAR*, vol. 13, no. 3, pp. 229–241, 2010.

[7] A. Rebelo, G. Capela, and J. S. Cardoso, "Optical recognition of music symbols: A comparative study," *IJDAR*, vol. 13, no. 1, pp. 19–31, 2010.

[8] S. Escalera, A. Fornés, O. Pujol, P. Radeva, G. Sánchez, and J. Lladós, "Blurred Shape Model for binary and grey-level symbol recognition," *Pattern Recognition Letters*, vol. 30, no. 15, pp. 1424–1433, 2009.

[9] B. Coüasnon and B. Rétif, "Using a grammar for a reliable full score recognition system," 1995.

[10] A. Baró, P. Riba, and A. Fornés, "Towards the recognition of compound music notes in handwritten music scores," in *ICFHR*, 2016, pp. 465–470.

[11] E. van der Wel and K. Ullrich, "Optical music recognition with convolutional sequence-to-sequence models," in *ISMIR*, 2017, pp. 731–737.

[12] J. Calvo-Zaragoza and D. Rizo, "End-to-end neural optical music recognition of monophonic scores," *Applied Sciences*, vol. 8, pp. 1–23, 2018.

[13] C. Wen, A. Rebelo, J. Zhang, and J. Cardoso, "A new optical music recognition system based on combined neural network," *Pattern Recognition Letters*, vol. 58, pp. 1 – 7, 2015.

[14] A. Fornés, A. Dutta, A. Gordo, and J. Lladós, "CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal," *IJDAR*, vol. 15, no. 3, pp. 243–251, 2012.

[15] J. Hajič jr. and P. Pecina, "The MUSCIMA++ Dataset for Handwritten Optical Music Recognition," in *ICDAR*, 2017, pp. 39–46.

[16] A. Pacha, K.-Y. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi, and H. M. Eidenberger, "Handwritten music object detection: Open issues and baseline results," in *DAS*, 2018, pp. 163–168.

[17] A. Baró, P. Riba, J. Calvo-Zaragoza, and A. Fornés, "From optical music recognition to handwritten music recognition: A baseline," *Pattern Recognition Letters*, vol. 123, pp. 1 – 8, 2019.

[18] L. Tuggener, I. Elezi, J. Schmidhuber, and T. Stadelmann, "Deep watershed detector for music object recognition," in *ISMIR*, 2018, pp. 271–278.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[20] A. Graves, S. Fernández, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006, pp. 369–376.

[21] L. Kang, J. Toledo, P. Riba, M. Villegas, A. Fornés, and M. Rusiñol, "Convolve, attend and spell: An attention-based sequence-to-sequence model for handwritten word recognition," in *GCPR*, 2019, pp. 459–472.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[23] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *NIPS*, p. 577–585, 2015.

[24] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*, 2009, pp. 41–48.