TOWARDS FULL-PIPELINE HANDWRITTEN OMR WITH MUSICAL SYMBOL DETECTION BY U-NETS

Jan Hajič jr.[1] **Matthias Dorfer**[2] **Gerhard Widmer**[2] **Pavel Pecina**[1]
[1] Institute of Formal and Applied Linguistics, Charles University
[2] Institute of Computational Perception, Johannes Kepler University
hajicj@ufal.mff.cuni.cz

ABSTRACT

Detecting music notation symbols is the most immediate unsolved subproblem in Optical Music Recognition for musical manuscripts. We show that a U-Net architecture for semantic segmentation combined with a trivial detector already establishes a high baseline for this task, and we propose tricks that further improve detection performance: training against convex hulls of symbol masks, and multichannel output models that enable feature sharing for semantically related symbols. The latter is helpful especially for clefs, which have severe impacts on the overall OMR result. We then integrate the networks into an OMR pipeline by applying a subsequent notation assembly stage, establishing a new baseline result for pitch inference in handwritten music at an f-score of 0.81. Given the automatically inferred pitches we run retrieval experiments on handwritten scores, providing first empirical evidence that utilizing the powerful image processing models brings content-based search in large musical manuscript archives within reach.

1. INTRODUCTION

Optical Music Recognition (OMR), the field of automatically reading music notation from images, has long held the significant promise for music information retrieval of making a great diversity of music available for further processing. More compositions have probably been written than recorded, and more have remained in manuscript form rather than being typeset; this is not restricted to the tens of thousands of manuscripts from before the age of recordings, but holds also for contemporary music, where many manuscripts have been left unperformed for reasons unrelated to their musical quality. Making the content of such manuscript collections accessible digitally and searchable is one of the long-held promises of OMR, and at the same time OMR is reported to be the bottleneck there [17]. On printed music or simpler early music notation, this has been attempted by the PROBADO [17, 28] or SIMSSA/Liber Usualis [3] projects. However, for manuscripts, results are not forthcoming.

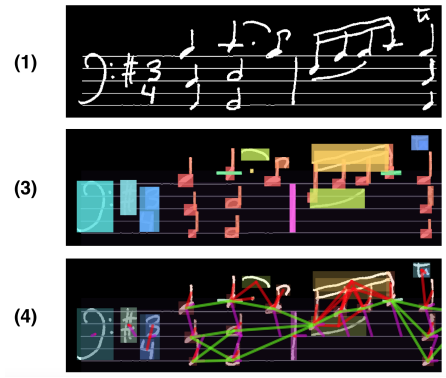

Figure 1. OMR pipeline in this work. Top-down: (1) input score, (3) symbol detection output, (4) notation assembly output. Obtaining MIDI from output of notation assembly stage (for evaluating pitch accuracy and retrieval performance) is then deterministic. Our work focuses on the symbol detection step (1) → (3); notation reconstruction is done only with a simple baseline.

The usual approach to OMR is to break down the problem into a four-step pipeline: (1) preprocessing and binarization, (2) staffline removal, (3) symbol detection (localization and classification), and (4) notation reconstruction [2]. Once stage (4) is done, the musical content — pitch, duration, and onsets — can be inferred, and the score itself can be encoded in a digital format such as MIDI, MEI [1] or MusicXML. We term OMR systems based on explicitly modeling these stages *Full-Pipeline OMR*.

Binarization and staff removal have been successfully tackled with convolutional neural networks (CNNs) [4, 11], formulated as semantic segmentation. Symbol classification achieves good results as well [12, 13, 33]. However, detecting the symbols on a full page remains the next major bottleneck for handwritten OMR. As CNNs have not been applied to this task yet, they are a natural choice.

Full-Pipeline OMR is not necessarily the only viable approach: recently, *end-to-end OMR* systems have been proposed. [16, 24]. However, they have so far been limited to short excerpts of monophonic music, and it is not clear how to generalize their output design from MIDI equivalents to

[1] http://music-encoding.org/

lossless structured encoding such as MEI or MusicXML, so full-pipeline approaches remain justified.

Our work mainly addresses step (3) of the pipeline, applied in the context of a baseline full-pipeline system, as depicted in Fig. 1. We skip stage (2): we treat stafflines as any other object, since we jointly segment and classify and do not therefore have to remove them in order to obtain a more reasonable pre-segmentation. We claim the following contributions:

(1) U-Nets used for musical symbol detection. Applying fully convolutional networks, specifically the U-Net architecture [38], for musical symbol segmentation and classification, without the need for staffline removal. We apply improvements in the training setup that help overcome OMR-specific issues. The results in Sec. 5 show that the improvements one expects from deep learning in computer vision are indeed present.

(2) Full-Pipeline Handwritten OMR Baseline for Pitch Accuracy and Retrieval. We combine our stage (3) symbol detection results with a baseline stage (4) system for notation assembly and pitch inference. This OMR system already achieves promising pitch-based retrieval results on handwritten music notation; to the best of our knowledge, its pitch inference f-score of 0.81 is the first reported result of its kind, and it is the first published full-pipeline OMR system to demonstrably perform a useful task well on handwritten music.

2. RELATED WORK

U-Nets. U-Nets [38] are fully convolutional networks shaped like an autoencoder that introduce skip-connections between corresponding layers of the downsampling and upsampling halves of the model (see Fig. 2). For each pixel, they output a probability of belonging to a specific class. U-Nets are meant for semantic segmentation, not instance segmentation/object detection, which means that they require an ad-hoc detector on top of the pixel-wise output. On the other hand, this formulation avoids domain-specific hyperparameters such as choosing R-CNN anchor box sizes, is agnostic towards the shapes of the objects we are looking for, and does not assume any implicit priors on their sizes. This promises that the same hyperparameter settings can be used for all the visually disparate classes (the one neuralgic point being the choice of receptive field size). Furthermore, U-Nets process the entire image in a single shot — which is a considerable advantage, as music notation often contains upwards of 500 symbols on a single page. A disadvantage of U-Nets (as well as most CNNs) is their sensitivity to the training data distribution, including the digital imaging process. Because of the variability of musical manuscripts, it is likely real-world applications will require case-specific training data, and data augmentation would therefore be used to mitigate this sensitivity; fortunately, fully convolutional networks are known to respond well to data augmentation over sheet music [30] as well as over other application scenarios [9, 23]. Therefore, we consider this choice reasonable, at the very least to establish a strong baseline for handwritten musical symbol detection with deep learning.

Object Detection CNNs. A standard architecture for object detection is the Regional CNN (R-CNN) family, most notably Faster R-CNN [40] and Mask R-CNN [26]). These networks output probabilities of an object's presence in each one of a pre-defined sets of anchor boxes, and make the bounding box predictions more accurate with regression. In comparison, the U-Net architecture may have an advantage in dealing with musical symbols that have significantly varying extents, such as beams or stems, as it does not require specifying the appropriate anchor box sizes, and it is significantly faster, requiring only one pass of the network (the detector then requires one connected component search). Furthermore, Faster R-CNN does not output pixel masks, which are useful for archival- and musicology-oriented applications downstream of OMR, such as handwriting-based authorship attribution. Mask R-CNN, admittedly, does not have this limitation, but still requires the same bounding box setup.

Another option is the YOLO architecture [25], specifically the most recent version YOLOv3 [36], which predicts bounding boxes and confidence degrees without the need to specify anchor boxes. A similar approach was proposed in [22], achieveing a notehead detection f-score of 0.97, but only with a post-filtering step.

Convolutional Networks in OMR. Convolutional networks have been applied in OMR to symbol classification [33], indicating that they can in principle handle the variability of music notation symbols, but not yet in also finding the symbols on the page. Fully convolutional networks have been successfully applied to staff removal [4], and to resolving the document to a background, staff, text, and symbol layers [11]. However, these are semantic segmentation tasks; whereas we need to make decisions about individual symbols. The potential of U-Nets for symbol detection was preliminarily demonstrated on noteheads [22, 31], but compared to other symbol classes, noteheads are "easy targets", as they look different from other elements, have constant size, and appear only in one pose (as opposed to, e.g., beams).

OMR Symbol Detection. Localizing symbols on the page has been previously addressed with heuristics rather than machine learning, e.g. with projections [8, 18], Kalman Filters [14], Line Adjacency Graphs [37], or other combinations of low-level image features [39]. On handwritten music, due to its variability, more complex heuristics such as the algorithm of [1] that consists of 14 interdependent steps have been applied.

OMR for Content-Based Retrieval. The idea of using imperfect OMR for retrieval is not new, although originally OMR was attempted in the context of transcribing individual scores. In the PROBADO project [17, 28], an off-the-shelf OMR system was applied to printed Common Western Music Notation (CWMN) scores, allowing retrieval and measure-level score following in a database of 1200 printed scores. The Liber Usualis project at SIMSSA is another such project, on square plainchant notation; it operates at a more fine-grained level that allows for ex-

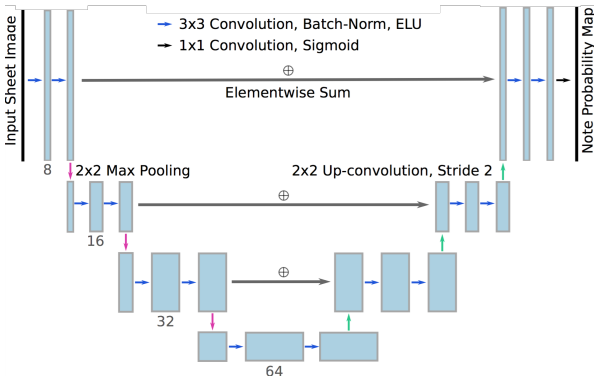

Figure 2. Baseline U-Net model architecture.

ample accurate motif retrieval [3]. However, for CWMN *manuscripts*, we are not aware of similar experiments.

3. MODEL

For all experiments, we use as a basis the same fully convolutional network architecture [38] as shown in Figure 2. There are three down-sampling blocks and three corresponding up-sampling blocks. Each down-sampling block consists of two convolutional layers with batch normalization using the same number of filters; down-sampling is done through 2x2 Max Pooling. After each downsizing step, we use twice the number of filters. The output layer uses sigmoid activation; otherwise, ELU nonlinearity is used. Additionally, we add element-wise-sum residual connections between symmetric layers of the encoder and decoder part of the network.

In the rest of this section, we propose modifications for both architecture and training strategy for symbol detection in handwritten sheet music.

3.1 Convex Hull Segmentation Targets

Our first proposal is to use the convex hull region of individual symbols as a target for training instead of the original segmentation masks. Figure 3 shows an example of the modified training targets. This simple adaptation is an elegant way of dealing with symbols such as f-clefs or c-clefs, which by definition consist of multiple components. As we employ a connected components detector for recognizing the symbols in our experiments in Section 4 we circumvent the need for treating these symbol classes in any special way. This advantage also holds "pre-emptively" for complex symbols which for example contain "holes" and might break up into multiple components after imperfect automatic segmentation, or may be disconnected due to handwriting style (e.g., flats).

3.2 Multichannel Training

Our second proposal is to train multichannel U-Nets predicting the segmentation simultaneously for multiple symbol classes. This design choice has two advantages over

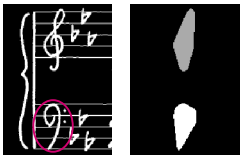

Figure 3. Training on convex hulls circumvents detection problems for symbols consisting of multiple connected components (see f-clef).

training separate detectors for each class. Firstly, at runtime we can predict the segmentation for multiple symbols with a single forward pass of the network. Furthermore, by simultaneously training on multiple symbols at the same time, we allow the model to share low-level feature maps for a certain symbol group (i.e., noteheads, beams, flags and stems), and on the other hand force the model to learn upper-layer features that discriminate well between the various symbols, which – because the capacity of the model stays fixed, and the output layer only uses 1x1 convolutions – could lead to more descriptive representations of the image. In other words, due to the strong correlations across classes induced by music notation syntax, whatever features are learned for one output channel will at the same time be relevant for a different channel; the 1x1 convolution will simply weigh them differently.

However, this setup presents an optimization problem due to imbalanced classes: both in terms of how many foreground pixels there are (i.e. beams vs. duration dots), and with respect to how often they occur on an "average" page of sheet music (noteheads vs. clefs). We address the first issue by splitting the multichannel model into groups of symbols with roughly similar amount of foreground pixels across the dataset. To overcome the second issue, as the training setup operates on randomly chosen windows of the input image (see Sec. 4), we use oversampling: when drawing the random window when a training batch is being built, we check whether the window contains at least one pixel of the target class, and we retry up to five times if there is none. If no target class pixel is found in five tries, we concede and use the last sampled window, even though no pixel of target class is in it. (As opposed to this oversampling, adjusting the weights of the output channels did not lead to improvements.)

Furthermore, if model capacity becomes a limiting factor, we can opt out of sharing the up-sampling part of the model and keep a separate "decoder" for each output channel. This is a compromise that retains some of the speed, space and feature-sharing advantages, but at the same time does not so severely restrict the capacity of the model.

4. EXPERIMENTAL SETUP

We restrict ourselves to the subset of symbol classes that are necessary for pitch inference and basic duration inference (we currently do not detect tuplets – detecting handwritten digits is straightforward enough, the difficulty with tuplets lies in the notation assembly stage). Already this

selection contains symbols with heterogeneous appearance: constant-size, trivial shape (specifically, noteheads, ledger lines, whole and half rests, duration dots), constant-size, non-trivial shape (clefs, flags, accidentals, quarter-, 8th- and 16-th rests), and symbols that have simple shapes, but varying extent (stems, beams, barlines). [2] We assume binary inputs, not least because large-scale OMR symbol detection ground truth is only available for binary images; however, binarization can be done with the same model.

Dataset. We use the MUSCIMA++ dataset, version 1.0 [20]. This is the only publicly available dataset of handwritten music notation with ground truth for symbol detection at a scale that is feasible for machine learning. The dataset contains over 90 000 manually annotated symbols with pixel masks. We use the designated writer-independent test set from MUSCIMA++.

Training Details. We set the network input size to a 256×512 window and randomly sample crops of this size as training samples. We train all our models using the Adam update rule with an initial learning rate of 0.001 [27] and a batch size of 2 (with the 256×512 input window, this is equivalent to batches of a single 512×512 image of [38]). After there is no update on the validation loss for 25 epochs, we divide the learning rate by 5 and continue training from the previously best model. This procedure is repeated two times.

5. RESULTS

As there is no work to which we can compare directly, we first gather at least related OMR solutions, in order to provide whatever context we can for the reader. Then, we report results for symbol detection, and evaluate it in context of downstream tasks: pitch inference in a baseline full-pipeline OMR scenario, as well as first experiments applying our models in retrieval settings.

5.1 Comparison to Existing Systems

Comparison to existing systems is hard, because there are few symbol detection results reported, and even fewer full-pipeline OMR results. Direct comparison is not possible, as the MUSCIMA++ dataset we use has been released only very recently, and previous OMR pipelines (see Sec. 2) generally do not have publicly available code. Furthermore, earlier literature on OMR rarely provides evaluation scores, most of previous work on OMR has (sensibly) focused on printed music rather than manuscripts, and there are few established evaluation practices in OMR anyway [15, 21]. We do our best to at least gather literature where some results on related tasks are given, in order to provide context for our work.

Pitch accuracy, printed music. In printed music, results for pitch accuracy have been consistently very good, when reported. Already in [32], the GAMUT system is said to correctly recover 96 % of pitches in printed music. The complex fuzzy system of [39] achieves near-perfect pitch accuracy (98.7 %). Similarly, the CANTOR system evaluated in [5] achieves 98 % semantic accuracy — this time, including polyphonic music. On printed square notation, [19] achieves 95 % pitch accuracy. A combination of systems in [42] achieves over 85 % joint pitch and duration accuracy.

Symbol detection, handwritten music. The most extensive evaluation of symbol detection in handwritten music has been carried out in [1]. Using a complex combination of robust heuristics for segmentation and machine learning for classification, they achieve an average symbol detection f-score of 0.75. These results seem ripe to be surpassed with CNNs: in [31], 98 % handwritten notehead detection accuracy has been reported. For staff detection, a similar architecture has been used in [4] with over 97 % pixel-wise f-score, and similar results are available with a ConvNet pixel classification approach for semantic segmentation into background, text, staffs, and notation symbols [11]. At the same time, [33] reports symbol *classification* (without localization) accuracy over 98 %, indicating that CNNs are well capable of generalizing over the variety of handwritten musical symbols. However, we are *not* aware of pitch accuracy results reported on handwritten CWMN scores.

OMR for Retrieval. For retrieval, it is even harder to find comparable results, since evaluation metrics for retrieval depend on the test collection, and there is no such established collection for OMR. Using the open-source Audiveris [3] OMR software, [7] matches 9803 printed monophonic fragments from *A Dictionary of Musical Themes* to their electronic counterparts, using a comparable DTW alignment that also (mostly) ignores note duration, reporting a top-1 accuracy of 0.44; however, the collection of themes is a difficult one, since it often contains very similar melodies.

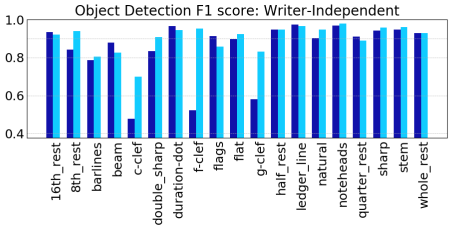

Figure 4. Results for binary segmentation models for individual symbols. Blue: baseline training with mask output; green: training with convex hulls.

5.2 Symbol Detection

We report detection f-scores for the chosen subset of symbols. Aggregating the results is not too meaningful: some rare symbols have an outsized impact on downstream processing (clefs). In Fig. 4, we show the baseline results and compare them to the convex hull setup. Training against

[2] There are also notation symbols that can have non-trivial shape *and* varying extent, such as slurs or hairpins; however, these are not required for neither pitch, nor duration inference, and we therefore leave them out.

[3] https://github.com/audiveris/audiveris

Method	c-clef	g-clef	f-clef
single channel – no convex hull	0.48	0.58	0.52
single channel	0.70	0.83	**0.95**
multi-channel – all	0.16	0.37	0.49
multi-decoder – clefs, oversampling	**0.77**	**0.96**	**0.93**

Table 1. Comparison of detection performance (F-score) of clefs using different segmentation strategies.

convex hulls of objects does address the issue of detecting otherwise disjoint symbols using connected components; otherwise it achieves mixed results.

Compressing the detector with multichannel training without a loss of performance was possible on correlated sub-groups of symbols that bypass the class imbalance problem, such as training together noteheads, stems, beams, and flags; the results worsened when all classes were trained at once. The clefs were most affected by all the changes to the model described in Section 3: improved by convex hull training, neglected when the multichannel model was trained to predict all symbol classes at once, and then drastically improved again when trained as a group with separate decoders and the oversampling strategy. Table 1 summarizes the results for clef detection. Clefs are critical for useful OMR, since they affect the pitch inferred from all subsequent noteheads.

6. APPLICATION SCENARIO: FULL-PIPELINE HANDWRITTEN OMR IN RETRIEVAL

We now explore the utility of the symbol detectors within an OMR pipeline. It is known in OMR that low-level errors can lead to effects on recognition of wildly different magnitudes [15, 35]; in the presence of detection errors, one should therefore see how severely they impact downstream applications. We choose a *retrieval* scenario as the application context for evaluating symbol detection. As opposed to applications where we produce the transcribed score [15, 21, 41], this is straightforward to evaluate.

To verify that our symbol detection approach can yield useful results in an application context, we add a simple notation assembly and pitch inference system on top of the symbol detection results. We choose *retrieval* as the most feasible application of handwritten OMR: there are music manuscript archives with thousands of scores that contain manual copies, and matching them cannot be done without their musical content.

For inferring pitch, we must re-introduce stafflines. However, we can safely assume they have been detected correctly: both [4] and our replication of their experiments with stafflines on this dataset exhibit extremely few errors, and these can be filtered away with a trivial projection heuristic such as that of [18].

6.1 Notation Assembly and Music Inference

Symbol detection alone is not sufficient for decoding musical information: meaningful units are *configurations* of symbols rather than the symbols themselves [6, 20]. The notation assembly stage is the step where these configurations are recovered (step (4) in the OMR pipeline: see 1). In the MUSCIMA++ dataset, they are represented as an oriented graph; once this graph is recovered, one can perform deterministic pitch inference. [4]

Symbol detection outputs vertices of the notation graph; we therefore need to recover graph edges. Replicating the baseline established in [20], we train a binary classifier over ordered symbol pairs. While this classifier achieves an f-score of 0.92, it makes embarrassing errors: noteheads connected to irrelevant ledger lines in chords, to beams that belong to an entirely different staff, and sometimes to multiple adjacent stems. We discard these obviously wrong edges using straightforward heuristics. We also discard detected objects that are entirely contained within another detected object. The last step is recovering *precedence* edges: we just order rest and noteheads on each staff left-to-right; noteheads connected to the same stem are considered simultaneous, but actual polyphony is ignored.

Once the pitches, durations, and onsets are inferred for the detected noteheads, we then export them as a MIDI file. MIDI is appropriate for retrieval, since it presents straightforward ways of computing similarity. This file then can serve as both the query and the database key for the given score. To compute the similarity of two MIDI files, we align them using Dynamic Time Warping [29] (DTW) over sequences of time frames that contain onsets. The DTW score function for a pair of frames is 1 minus the Dice coefficient of the onset pitch sets in the frames. Then, we match individual pitches within the frame sets that are aligned by DTW and measure the f-score of predicted pitches. DTW is used as the similarity function in [7]; however, we do not reduce polyphonic music to its upper pitch envelope.

6.2 Results

We now report how the full-pipeline baseline on top of the object detection U-Nets predicts pitches, and how it can be used to retrieve related scores.

Pitch accuracy. We use the DTW alignment to directly evaluate pitch classification. [5] Performing DTW on the inference outputs for page images, we achieve a (micro-)average F-score of only 0.59. Rather than due to errors in symbol detection, this is mostly due to the polyphony de-synchronization effects of bad duration inference; indeed, on (mostly) monophonic music, pitch F-score jumps to 0.78. In order to bypass de-synchronization problems that in fact obscure correct pitch recognition, we split the scores into individual staffs (118 in total) and evaluate pitch accuracy on these. The results for the test set staffs are reported in Fig. 5. On average, we obtain pitch F-score 0.81, with 0.83 for monophonic staffs (and ignoring clef errors, 0.88).

Finally, we evaluate our detector in the context of a retrieval application. We run experiments both on gold-

[4] A proof-of-concept implementation: `https://github.com/hajicj/muscima`.

[5] We could evaluate duration classification as well, but due to errors by the notation assembly baseline, this is too low to be worth reporting.

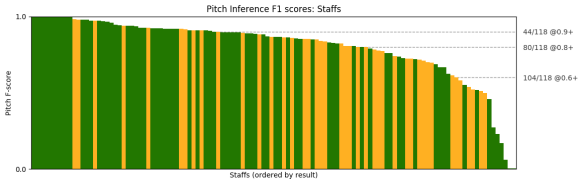

Figure 5. Pitch F-score after DTW alignments on the 118 individual staffs in the writer-independent test set, ordered by result. Monophonic staffs (darker green) predictably score better than staffs with multiple voices or chords (yellow). We found no clear relationship between pitch accuracy and handwriting style.

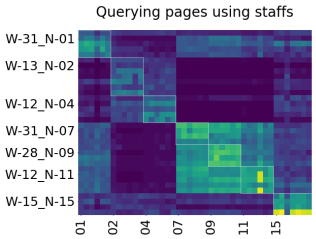

Figure 6. Pitch f-score between predictions on test set staffs and (predictions on) training set pages. Notice the pages 07, 09 and 11: these are three movements from J. S. Bach's Cello suite no. 1, which contain musically highly related material.

standard MIDI retrieval and duplicate score retrieval, using the predicted scores; since the similarity metric is pitch f-score, all retrieval experiments work in both directions. Experiments with ground truth MIDI correspond to cross-modal retrieval, where the modalities are a symbolic representation, and the score projected into the MIDI modality using the OMR system; queries with predictions correspond to a simpler scenario where we are querying scores with scores, using the OMR system as a hash function.

Retrieving gold MIDI with scores. Given how small the test set is, retrieving the correct ground truth page — and even staff — should be near-perfect. For staff-to-staff retrieval, Prec@1 is 0.93; for page-to-page and staff-to-page retrieval, this is 1.0, indicating that with our U-Net object detection stage, retrieving gold-standard MIDI using handwritten scores (and vice versa, as the similarity metric is symmetrical) is feasible.

Retrieving scores with scores. The next scenario is to run retrieval not against the ground truth, but against MIDIs predicted from different versions of the test set scores. While errors related to differences in handwriting get compounded, the rest of the pipeline imposes consistent limitations on both the database and query recognition outputs and may make the *same* errors on both query and database scores, making the task actually easier. Therefore, we select a *confuse-retrieval* subset of 7 scores from MUSCIMA++ that are as similar to each other as possible: mostly monophonic, and with 0 – 2 sharps. Some of these pieces are musically closely related. For these experiments, our database consists of recognition outputs computed from all *confuse-retrieval* pages in the *training* subset of MUSCIMA++. Queries are taken from predictions on the writer-independent test set: we use both the 7 entire pages and individual staffs (34 of those).

The system achieves perfect Prec@1 when pages are used as queries, and 0.94 when using staff queries (2 staff queries did not return the right piece as the top result). The retrieval scores are plotted in Fig. 6. We checked this score also with ground truth queries; this system made only 2 errors as well, but in different queries, which we take as circumstantial evidence that the ground truth MIDI has different issues when matching against a predicted MIDI than a different prediction. When measuring MAP with the cutoff k=6 (as there are 7 versions of each page in MUSCIMA++ and one of them is used for querying), it drops to 0.86.

7. DISCUSSION & CONCLUSIONS

We consider our work a successful step towards enabling applications of hitherto problematic handwritten OMR. The retrieval scenario results are an indication that U-Nets are a workable solution to the handwritten symbol detection bottleneck in the context of full-pipeline OMR. (Here, we must re-state that these results should *not* be interpreted as more than supporting evidence that our object detection method is viable for such scenarios!)

However, U-Nets are still in principle limited by the size of the receptive field: for instance the middle of a long stem looks exactly the same as a barline. We could further leverage syntactic properties of music notation: e.g., the self-attention layer of [34] allows building up the final output from partial recognition results. Fragmenting of long symbols could be overcome with instance segmentation embeddings [10].

To the best of our knowledge, this is also the first time OMR was done with a machine-learning method for notation assembly. We in fact consider this the most interesting line of follow-up work. Recovering the notation graph itself seems like the next bottleneck, especially for duration inference. The non-independent nature of the edges poses an interesting structured prediction challenge, and one could also work towards models that jointly detect symbols and recover their relationships.

Despite their limitations, U-Nets can be used to detect handwritten music notation symbols. They establish a new CNN-based baseline for the object detection task, and we believe the results in pitch inference and a proof-of-concept retrieval scenario indicate that a significant step has been taken towards full-pipeline OMR systems, so that the content of musical manuscripts can become accessible digitally.

8. ACKNOWLEDGMENTS

Jan Hajič jr. and Pavel Pecina acknowledge support by the Czech Science Foundation grant no. P103/12/G084, Charles University Grant Agency grants 1444217 and 170217, and by SVV project 260 453.

9. REFERENCES

[1] Ana Rebelo. *Robust Optical Recognition of Handwritten Musical Scores based on Domain Knowledge*. PhD thesis, 2012.

[2] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso. Optical Music Recognition: State-of-the-Art and Open Issues. *Int J Multimed Info Retr*, 1(3):173–190, Mar 2012.

[3] Andrew Hankinson, John Ashley Burgoyne, Gabriel Vigliensoni, Alastair Porter, Jessica Thompson, Wendy Liu, Remi Chiu, and Ichiro Fujinaga. Digital Document Image Retrieval Using Optical Music Recognition. In *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012*, pages 577–582. FEUP Edições, 2012.

[4] Antonio-Javier Gallego and Jorge Calvo Zaragoza. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications*, 89:138–148, 2017.

[5] David Bainbridge. Extensible optical music recognition. page 112, 1997.

[6] David Bainbridge and Tim Bell. A music notation construction engine for optical music recognition. *Software - Practice and Experience*, 33(2):173–200, 2003.

[7] Stefan Balke, Sanu Pulimootil Achankunju, and Meinard Müller. Matching Musical Themes based on noisy OCR and OMR input. pages 703–707, 2015.

[8] P. Bellini, I. Bruno, and P. Nesi. Optical music sheet segmentation. In *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC 2001*, pages 183–190. Institute of Electrical & Electronics Engineers (IEEE), 2001.

[9] Avi Ben Cohen, Idit Diamant, Eyal Klang, Michal Amitai, and Hayit Greenspan. Fully Convolutional Network for Liver Segmentation and Lesions Detection. In Gustavo Carneiro, Diana Mateus, Loïc Peter, Andrew Bradley, João Manuel R. S. Tavares, Vasileios Belagiannis, João Paulo Papa, Jacinto C. Nascimento, Marco Loog, Zhi Lu, Jaime S. Cardoso, and Julien Cornebise, editors, *Deep Learning and Data Labeling for Medical Applications*, pages 77–85, Cham, 2016. Springer International Publishing.

[10] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic Instance Segmentation with a Discriminative Loss Function. *CoRR*, abs/1708.02551, 2017.

[11] Jorge Calvo Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. A machine learning framework for the categorization of elements in images of musical documents. In *Third International Conference on Technologies for Music Notation and Representation*, A Coruña, 2017. University of A Coruña.

[12] Sukalpa Chanda, Debleena Das, Umapada Pal, and Fumitaka Kimura. Offline Hand-Written Musical Symbol Recognition. *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 405–410, sep 2014.

[13] Cuihong Wen, Jing Zhang, Ana Rebelo, and Fanyong Cheng. A Directed Acyclic Graph-Large Margin Distribution Machine Model for Music Symbol Classification. *PLOS ONE*, 11(3):e0149688, mar 2016.

[14] V.P. d'Andecy, J. Camillerapp, and I. Leplumey. Kalman filtering for segment detection: application to music scores analysis. In *Proceedings of 12th International Conference on Pattern Recognition*. IEEE Comput. Soc. Press, 1994.

[15] 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.

[16] Eelco van der Wel and Karen Ullrich. Optical Music Recognition with Convolutional Sequence-to-Sequence Models. *CoRR*, abs/1707.04877, 2017.

[17] Christian Fremerey, Meinard Müller, Frank Kurth, and Michael Clausen. Automatic mapping of scanned sheet music to audio recordings. *Proceedings of the International Conference on Music Information Retrieval*, pages 413–418, 2008.

[18] Ichlro Fujinaga. Optical Music Recognition using Projections. Master's thesis, 1988.

[19] Gabriel Vigliensoni, John Ashley Burgoyne, Andrew Hankinson, and Ichiro Fujinaga. Automatic Pitch Detection in Printed Square Notation. In *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011*, pages 423–428. University of Miami, 2011.

[20] Jan Hajič jr. and Pavel Pecina. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. In *14th International Conference on Document Analysis and Recognition*, pages 39–46, New York, USA, November 2017. Dept. of Computer Science and Intelligent Systems, Graduate School of Engineering, Osaka Prefecture University, IEEE Computer Society.

[21] Jan Hajič jr., Jiří Novotný, Pavel Pecina, and Jaroslav Pokorný. Further Steps towards a Standard Testbed for Optical Music Recognition. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 157–163, New York, USA, 2016. New York University, New York University.

[22] Jan Hajič Jr. and Pavel Pecina. Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression. *CoRR*, abs/1708.01806, 2017.

[23] Jesus Munoz Bulnes, Carlos Fernandez, Ignacio Parra, David Fernandez Llorca, and Miguel A. Sotelo. Deep fully convolutional networks with random data augmentation for enhanced generalization in road detection. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, oct 2017.

[24] Jorge Calvo-Zaragoza, Jose J. Valero-Mas, and Antonio Pertusa. End-to-End Optical Music Recognition Using Neural Networks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, pages 472–477, 2017.

[25] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *CoRR*, abs/1506.02640, 2015.

[26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[27] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR) (arXiv:1412.6980)*, 2015.

[28] F. Kurth, M. Müller, C. Fremerey, Y. Chang, and M. Clausen. Automated synchronization of scanned sheet music with audio recordings. *Proc. ISMIR, Vienna, AT*, pages 261–266, 2007.

[29] Lawrence R. Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice Hall signal processing series. Prentice Hall, 1993.

[30] Matthias Dorfer, Andreas Arzt, and Gerhard Widmer. Learning Audio-Sheet Music Correspondences for Score Identification and Offline Alignment. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, pages 115–122, 2017.

[31] Matthias Dorfer, jr. Jan Hajič, and Gerhard Widmer. On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection. In *Proceedings of the 12th IAPR International Workshop on Graphics Recognition*, pages 53–54, New York, USA, 2017. IAPR TC10 (Technical Committee on Graphics Recognition), IEEE Computer Society.

[32] Michael Droettboom and Ichiro Fujinaga. Symbol-level groundtruthing environment for OMR. *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 497–500, 2004.

[33] Alexander Pacha and Horst Eidenberger. Towards a Universal Music Symbol Classifier. In *Proceedings of the 12th International Workshop on Graphics Recognition*, 2017.

[34] N. Parmar, A. Vaswani, J. Uszkoreit, Ł. Kaiser, N. Shazeer, and A. Ku. Image Transformer. *ArXiv e-prints*, February 2018.

[35] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. Assessing Optical Music Recognition Tools. *Computer Music Journal*, 31(1):68–93, Mar 2007.

[36] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. Technical report, 2018.

[37] K. T. Reed and J. R. Parker. Automatic computer recognition of printed music. *Proceedings - International Conference on Pattern Recognition*, 3:803–807, 1996.

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, pages 234–241, Cham, 2015. Springer International Publishing.

[39] Florence Rossant and Isabelle Bloch. Robust and Adaptive OMR System Including Fuzzy Modeling, Fusion of Musical Rules, and Possible Error Detection. *EURASIP Journal on Advances in Signal Processing*, 2007(1):081541, 2007.

[40] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.

[41] Mariusz Szwoch. Using MusicXML to Evaluate Accuracy of OMR Systems. *Proceedings of the 5th International Conference on Diagrammatic Representation and Inference*, pages 419–422, 2008.

[42] Victor Padilla, Alan Marsden, Alex McLean, and Kia Ng. Improving OMR for Digital Music Libraries with Multiple Recognisers and Multiple Sources. *Proceedings of the 1st International Workshop on Digital Libraries for Musicology - DLfM '14*, pages 1–8, 2014.