

Hybrid AI Approaches for Improving 4-part Harmony Learning. Applications to Handwritten Optical Music Recognition.

Elia Pacioni^{1,2}, Francisco Fernández de Vega²

¹HES-SO Valais-Wallis, 3960 Sierre, Switzerland

²University of Extremadura, 06800 Mérida, Spain

Abstract

This paper presents a hybrid AI approach to handwritten Optical Music Recognition, aimed at supporting music students in learning four-part harmony. In previous work, we introduced Sharpmony, an AI-based mobile and web application adopted by more than 5,000 users, which has automatically checked over 21,000 harmony exercises against a set of classic harmony rules since 2020. Students had to create every one of those exercises in a music notation editor first, even though harmony is written by hand in a music notebook. That step is a barrier to adoption and a departure from the practice teachers value. We remove it. The student writes the exercise by hand, and a single photo produces the digital score that Sharpmony checks. A detector proposes a bounding box for each note. Our pipeline reconstructs the pre-printed staff from the photograph, locates the note head inside each box, and reads the pitch from the fitted geometry, exporting a MusicXML score ready for automatic checking. No part of that reading is learned from data. The geometry also states where it stops. Beyond the last printed line the student draws the ledger lines by hand, and that boundary is measurable from the staff itself, so the notes at risk are known before any pitch is assigned and can be flagged for confirmation. On held-out pages the pipeline recovers 90.3% of the notes with both type and pitch correct, against 87.6% for an end-to-end detector trained to predict the two jointly. Practice stays on paper and the feedback stays immediate.

Keywords

Hybrid AI, YOLO, Optical Music Recognition, Music Teaching Support, Music Information Retrieval

1. Introduction

Optical Music Recognition (OMR) is a well-known problem in the area of Music Information Retrieval. Although good results have been achieved in digital score recognition, and some professional software tools embody the required technologies, offering good capabilities to professional musicians, the opposite happens when dealing with handwritten scores. Novotný and Pokorný [1] describe the four-phase OMR pipeline: preprocessing, symbol segmentation, object recognition, and semantic reconstruction. They describe the main challenges and also the lack of large, heterogeneous ground-truth corpora that would allow defining standard evaluation metrics, reproducibility, and benchmarking.

In recent years, the Sharpmony [2, 3, 4] project was launched to offer music students and their teachers software tools to improve the teaching-learning experience in the area of 4-part harmony. This is a core subject in European music conservatories, required of every student pursuing a professional music degree. Sharpmony offers an editor that students can use to write 4-part harmony scores (uploading scores edited with other tools is also possible). Once the score is ready, Sharpmony checks it according to a set of well-defined classic harmony rules that have been encoded in cooperation with music teachers.¹

Given the traditional way teachers and students work, using the music notebook to write their scores, a natural next step for Sharpmony would be to allow students to continue using their notebook to write their score, and then take a photo of the score to be checked by Sharpmony. This is an interesting next step that aligns with the new trend aimed at reducing the use of digital devices at school [5, 6].

AI-LEAP 2026 - 1st International Workshop on AI Learning, Education And Practice, co-located with AIXIA 2026, University of Perugia, October 2026

✉ elia.pacioni@hevs.ch (E. Pacioni); fcofdez@unex.es (F. F. d. Vega)

ORCID 0000-0002-1557-4870 (E. Pacioni); 0000-0002-1086-1483 (F. F. d. Vega)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹<https://sharpmony.unex.es/index.php?r=tutorials/harmonic-manual>

Although a device is still needed to take the photo, OMR would allow students to work mainly in the traditional way, the music notebook, and only use the device when the exercise is ready to be checked. Moreover, this is an easy task that could be performed by the teacher if students are not allowed to use their own device at the conservatory. Still, Sharpmony could be easily used at home.

2. Optical Music Recognition

OMR refers to the technology and process of automating the interpretation and conversion of musical notation found in scanned images, printed scores, or manuscripts into a structured digital format. OMR is particularly relevant when applied to handwritten scores. The difficulty of the problem is greater than with Optical Character Recognition, because a musical symbol carries two independent pieces of information. A staff is a grid of five equally spaced lines, and a note head sits on a line or in a space between two, so a staff offers nine vertical positions, consecutive positions being consecutive degrees of the scale. The shape of a note says how long it lasts, whole, half or quarter; its vertical position alone says how high it sounds, once the clef is known. Notes beyond the five lines are written on ledger lines, short strokes that continue the grid one position at a time, while accidentals, flat, sharp and natural, sit to the left of a head and alter its pitch without moving it on the grid (Figure 3a). In four-part harmony the four voices share two staves and notes sounding together are aligned into chords, so several heads share a column. Recognition must therefore recover both a shape and a location, and handwriting degrades them differently: shapes vary from writer to writer, while heads drift off the grid that gives them their meaning. Several considerations follow: staff detection, noise reduction, binarization, and the overlap of simultaneous voices. Ornamentation, such as trills, mordents and accents, is not considered here. Handwriting adds inconsistent stroke thickness, variable spacing, ink smudges and text annotations. A final natural step once the score has been processed is to convert it into a standard music file format, such as MusicXML.

OMR has already been addressed from a deep learning point of view [7], and Calvo-Zaragoza et al. [8] survey the field and its open problems. Public corpora such as MUSCIMA++ [9] have made handwritten OMR tractable, but they consist of single-voice manuscripts on unruled paper, whereas the exercises we address are four-voice and written on pre-printed staves. Recently, Baró et al. [10] describe a Convolutional and Recurrent Neural Network approach that is applied to single staves. They apply data augmentation to increase the number of samples for the training process. We instead deal with 4-part harmony, where two staves are required to describe the whole score.

In the area of deep learning, YOLO-based approaches have been validated for their high precision in detecting objects as outlined in studies that report excellent detection metrics [11]. In our previous work, we analyzed the capabilities of YOLO when addressing OMR to support digital and handwritten score recognition [12]. Three main steps were performed: i) dataset collection, combining a small set of handwritten scores (100) with 1000 digital ones. ii) Automatic annotation of digital scores together with manual labeling of handwritten ones. iii) Application of the YOLO algorithm to produce the MusicXML file corresponding to the score in the photo. Although results were positive, the accuracy found after the training process was below 70%. Moreover, although the quality was significantly higher for digital scores, on handwritten ones it was below a usable level. Therefore, new methodologies and techniques are required if we want to provide automatic conversion of handwritten scores into the standard MusicXML format.

3. Methodology

The methodology described below addresses a problem that OMR poses beyond object detection: recognizing a score requires not only finding the symbols but also locating them, since the pitch of a note is given by its vertical position on the staff. Three ways of dividing that work have been considered in this project. In previous work [13], two separate detectors were trained, one for the symbol and one for the pitch, and the difficulty proved to be establishing the correspondence between the two sets of

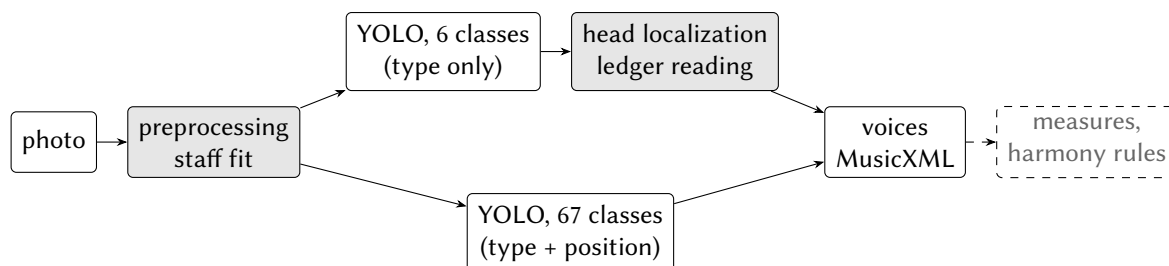


Figure 1: The two paths compared in this paper. Shaded boxes hold knowledge specified by hand, unshaded boxes knowledge learned from data; input and output are neither. The dashed box is future work.

detections. The second option, which we reimplement here as our baseline [12], is a single detector whose classes are the observed pairs of type and position, so that one prediction carries both. The third, which we propose, keeps the detector responsible for the symbol type alone and recovers the pitch geometrically from the staff. This paper compares the last two on the same images, the same frozen split, and the same detector architecture, so that the only variable is where the pitch comes from, as Figure 1 shows.

The idea behind reading the pitch geometrically rests on a simple fact: the staff is what gives a note its pitch, so detecting the staff lines accurately should be enough to assign it. Exercise notebooks help here, because the staff is pre-printed and therefore geometrically exact. Our working hypothesis was that this would make geometric reconstruction more reliable than a learned pitch predictor. It holds only where the staff is printed. Beyond the last printed line the student supplies the grid, and the exactness the method relies on is no longer there.

3.1. Corpus and detection

The corpus consists of 135 images and scans of four-part harmony exercises written in pencil on pre-printed staff paper, collected from conservatory students and annotated with 4,803 bounding boxes. Each note carries its type and its step, the vertical position counted in half spacings from the bottom printed line, so that steps 0 to 8 are the nine positions spanned by the five printed lines, and steps -1 and 9 are the spaces immediately below and above them. Anything beyond those two requires hand-drawn ledger lines. The class distribution is dominated by whole notes (2,469), followed by half notes (1,339), quarter notes (584), and accidentals (409). The corpus contains no eighth notes and only two rests, which are therefore out of scope. The pages are split once, stratified by whether the image is flat or visibly curved, into 95 training, 20 validation, and 20 test images. Key and time signature are supplied per page rather than recognized from the image, since a student knows the key of the exercise they were asked to write.

Symbol detection uses YOLO with six classes, namely whole, half, and quarter note heads plus flat, sharp, and natural, trained for 300 epochs at 1600 px with early stopping on the validation mAP. Since the apparent size of a note depends on the distance at which the photo was taken, every image is rescaled so that one staff space measures 24 pixels before it reaches the network, using the spacing estimated by the preprocessing stage; the same normalization is applied at inference time. Augmentation is modeled on the way the photographs are actually taken, with small rotations, mild perspective, and scale and lighting changes. No horizontal or vertical flip is used, because musical notation is not symmetric. Detection is tuned for recall rather than F1, since a missed note cannot be recovered downstream whereas a spurious one can be filtered by symbolic validation. The baseline we compare against is trained identically, from the same weights and with the same hyperparameters, on the same boxes. The only difference is that its classes are the 67 observed pairs of type and step, so that the network predicts type and pitch at once. Thirteen of those classes have fewer than ten training examples, and eight have fewer than three.

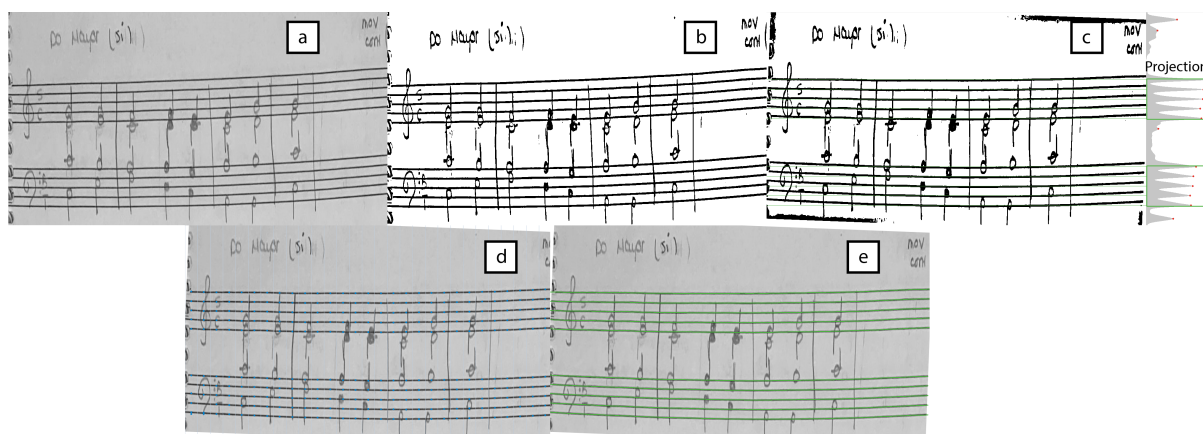


Figure 2: Staff detection: (a) the original photograph; (b) the page after Sauvola binarization; (c) the horizontal projection, whose peaks locate the printed lines and are grouped into staves; (d) the candidate points sampled along those lines; (e) the fitted curves from the shared parametric model with a robust Huber estimator, which absorbs the slope introduced by the angle of the photograph.

3.2. From the printed staff to a pitch

The score is first binarized with Sauvola [14] thresholding, and the two scale parameters, line thickness and staff space, are estimated from it as the modes of the vertical run lengths. The global skew is then estimated by a Hough [15] transform and corrected, after which the image is binarized a second time with a window sized from the spacing just measured, and the two parameters are re-estimated on the corrected image. Staff line candidates are extracted once, independently of any model, by a horizontal opening that suppresses stems, note heads, text and barlines, followed by a thinness test on the vertical runs that survive. The surviving rows are accumulated into a horizontal projection, whose peaks are grouped into sets of five to identify the individual staves of the page. Figure 2 skips the intermediate stages of this cleaning and shows the projection they produce. Within each staff, the candidate rows are then sampled column by column, giving a set of points that trace the actual course of each printed line across the page. Those candidates feed a parametric model shared by the five lines of a staff, $y_i(x) = f(x; \theta) + i s(x) + \varepsilon_i$ for $i = 1, \dots, 5$, with y_5 the bottom line, where f is the shape of the staff, $s(x)$ varies linearly to absorb the perspective of a photograph taken at an angle, and ε_i is a bounded per-line deviation. The parameters are estimated by a robust Huber IRLS fit, followed by a local refinement on the residuals that recovers the folds near the page border. Figure 2 follows the whole sequence on a page whose staves are visibly sloped.

Turning a detected box into a pitch then requires locating the head inside it. The predicted box includes the stem, so its center is not the center of the note head, and using it directly biases the pitch by more than one step; Figure 3b follows one note through the whole chain. Two details of the extraction turned out to matter more than the choice of estimator. The first is that the holes of the note heads are filled before the staff lines are removed inside the crop, since half and whole notes are drawn as rings, and removing a line that crosses the ring opens it, so that the hole to be filled no longer exists. The second is that staff removal is made aware of crossings, because clearing every pixel within a tolerance of a fitted line also clears the middle of a head that sits on that line, which displaces the estimated center by half a step. This is a systematic bias in the direction that matters most, since notes on a line are as frequent as notes in a space. A column is therefore kept whenever there is ink both above and below the line band over a run of columns as wide as a head, which distinguishes a head crossing the line from the line itself or from a stem crossing it. Whole notes are treated separately: having no stem, their box contains the head alone, and its center is the best available estimate with no crop analysis at all.

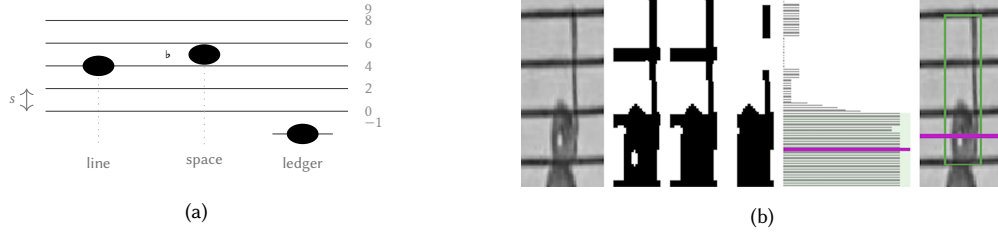


Figure 3: (a) Notation: a head sits on a line or in a space, one step, half a spacing s , apart; steps 0–8 are the printed positions, -1 and 9 the spaces just beyond, which need no ledger line. (b) Head localization: crop, binarization, holes filled, staff lines removed, row-width profile with the selected band, and the center found. The box includes the stem, so its center lies above the head.

3.3. Reading the notes outside the staff

Inside the staff, the step follows directly from the vertical distance between the located head center and the fitted bottom line, measured in half spacings and rounded to the nearest integer, $\text{step} = \text{round}(2(y_5(x) - c_y)/s(x))$, where c_y is the head center and x its horizontal position. Outside it, the grid has to be continued beyond the last printed line, and this is where the assumption breaks, because students draw the ledger lines by hand and do not reproduce the printed spacing, so a note sitting on a hand-drawn line is not where the extrapolated grid expects it. Whether a note lies outside the staff is decided before any pitch is assigned, from the same quantity that the reading uses: the vertical distance between the located head center and the fitted bottom line, measured in half spacings. A note is treated as inside whenever that distance rounds to a step between -1 and 9 , that is, to one of the nine positions of the printed lines or to the space immediately beyond either end, neither of which requires a ledger line.

The system evaluated here counts the strokes drawn beside the note. A ledger line sticks out of the bounding box on both sides and is thin, so the rows of the two lateral strips that carry ink on the left and on the right of the box are grouped into strokes, and the number of strokes between the edge of the staff and the head gives the level of the note. The head then sits either on the outermost stroke or in the space beyond it. When no stroke can be resolved, which happens on roughly half the notes because faint pencil and neighboring chords interfere, the extrapolated grid is used as a declared fallback.

We also tried to remove the extrapolation altogether, by detecting the ledger strokes once per page rather than once per note, and grouping them into levels whose index comes from the notation rather than from the printed spacing. Pooling the evidence across a chord column improves the reading of ledger notes on the validation set but does not transfer to the test set, and the corpus is too small to resolve a difference of that size. The configuration evaluated in the rest of the paper is therefore the per-crop counter.

Downstream, the notes are grouped into voices and written out as MusicXML, the format Sharpmony consumes. Segmenting the exercises into measures and validating each measure against the harmony rules extends the same hand-specified layer upward and is the next stage of this work rather than a contribution of this paper.

4. Results

Every rate below is computed over the 476 notes of the test set, so a note the detector never found counts as an error exactly as one whose pitch was misread: the output of the system is a MusicXML score, and a missing note is as harmful there as a wrong one. Spurious detections lie outside this accounting, since the stage that would reject them, by discarding notes the four voices cannot accommodate, is not evaluated here.

Both models use the YOLO26 [16] backbone, are trained from the same weights with the same hyperparameters on the same boxes, and are evaluated on the same test pages, which were used for final

Table 1

Reading accuracy over all 476 annotated notes of the test set.

	hybrid	end-to-end
heads found	471 (98.9%)	444 (93.3%)
correct type	468 (98.3%)	439 (92.2%)
correct pitch	433 (91.0%)	421 (88.4%)
inside the staff (386)	367 (95.1%)	355 (92.0%)
on ledger lines (90)	66 (73.3%)	66 (73.3%)
type and pitch	430 (90.3%)	417 (87.6%)

evaluation only. Pitch outside the staff is read by the per-crop stroke counter described in Section 3.3. The pages illustrated in the figures below belong to that set and were selected for display after the evaluation, not to guide any development decision.

4.1. Reading accuracy

The hybrid approach reads 90.3% of the notes with both type and pitch correct, against 87.6% for the end-to-end baseline, which is the criterion the exported MusicXML requires. Symbol type is close to saturation for both, so those figures are set almost entirely by the pitch reading, and the ordering is the same if only the pitch is scored.

With 476 notes drawn from twenty scores, on which per-score accuracy varies widely, we do not read the 2.7-point difference as establishing an ordering between the two approaches, and the more interesting observation is that no ordering is needed. The geometric reader inside the hybrid approach was never trained to assign a pitch. It matches a detector trained on sixty-seven type-and-position classes while its own detector only ever learns what a note head looks like, so the reading itself carries over to any pre-printed staff without further annotation. In a domain where annotated data is scarce, a reading that needs none of it is worth as much as a small advantage in accuracy. What those figures describe is a complete path from a photograph to a checkable score. On the same pages the staff fit succeeds on all but one, the detector proposes 98.9% of the note heads, and 90.3% of them reach the exported MusicXML with the right type and the right pitch, well above the sub-70% accuracy we reported on a smaller and partly digital corpus in earlier work [12]. The remaining errors are concentrated rather than spread, which is what the rest of this section examines.

4.2. Operating point

Before decomposing the result, one bound on what the recall figures can support. The two detectors were run at the same confidence threshold, but not at matched operating points. For the 476 notes the six-class detector emitted 663 boxes and the sixty-seven-class one 490, so the former operates at a substantially lower precision. Precision and recall trade against each other along the curve of any single detector, so a detector operating further towards recall will find more heads. The gap of 5.6 points in heads found is therefore measured at one pair of operating points and is not a threshold-independent property of the two label spaces. Establishing that would require comparing recall at equal precision, which we have not done. Figure 4 shows what the difference looks like on a score. The notes the end-to-end baseline never proposes are not scattered: they fall in groups, often covering an entire chord column, which is the pattern that makes per-score accuracy vary so widely. Figure 5 shows the same comparison on a page whose staves are strongly sloped, where the errors of both approaches fall in part beyond the printed lines.

4.3. Where each method fails

Inside the printed staff, the hybrid pipeline is ahead by 3.1 points. Outside it, the two are indistinguishable: both read 66 of the 90 notes that require hand-drawn ledger lines. The limit in that region is therefore a property of the region rather than of the method. Neither reading strategy handles a grid

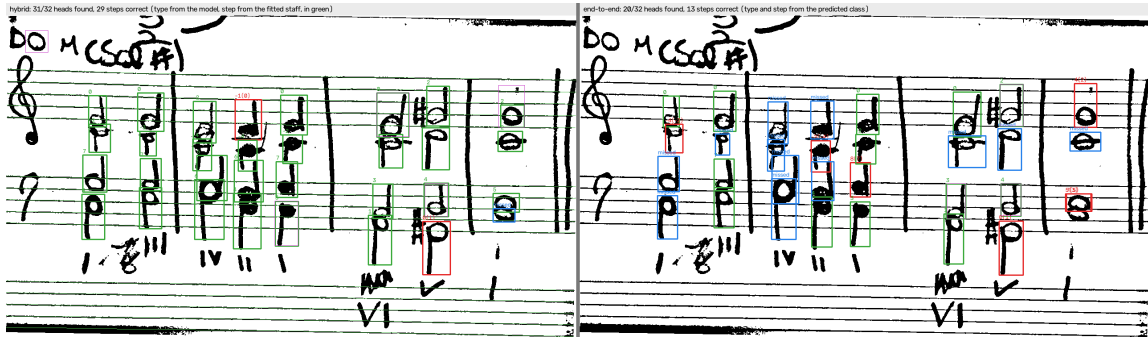


Figure 4: The two approaches on one system of a test page. Left, the **hybrid approach**; right, the **YOLO baseline** detector. Green marks a note read correctly, red a note whose pitch is wrong, labeled predicted (annotated), and blue an annotated note the detector never proposed. The detector that must commit to a position as well as a type leaves whole chord columns undetected, and a note that is never proposed cannot be recovered by any later stage. Two of the notes marked as errors here are cases where the annotation itself is doubtful, which is discussed among the limitations.

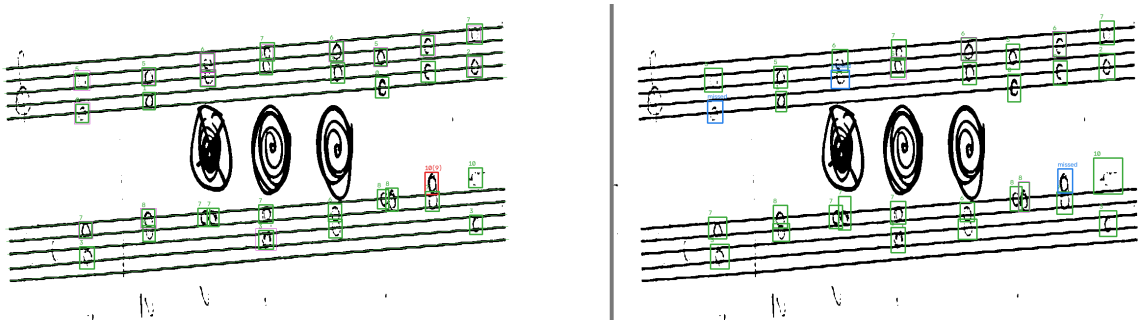


Figure 5: A page whose staves are strongly sloped. Colors as in Figure 4. The fitted curves follow the slope across the full width, and the notes inside the staff are read reliably. The errors of both approaches, one misreading a note and the other not proposing it, fall in part on the notes that sit beyond the printed lines.

the writer supplies by hand, and the printed notation stops constraining the image exactly where the printed lines stop.

Those notes are 90 of 476, under a fifth of the total, and they carry 56% of the hybrid pipeline’s errors against 44% of the end-to-end baseline’s. What distinguishes them is not their difficulty but the fact that they can be identified in advance. Whether a note lies outside the staff comes from the same measurement that reads it, but it asks far less of it: the decision only requires the head to fall on one side of a boundary set by the printed lines themselves, where the grid has barely been extrapolated, whereas the reading has to resolve an exact step two or three ledger lines away, where the error accumulates. A system built on this decomposition can therefore mark those notes for confirmation instead of reporting a pitch it has no reliable way to assign. We did not find a comparable signal for the errors of the end-to-end baseline; whether its own confidence provides one is untested.

The fit that the whole reading depends on is accurate. Against a manually annotated ground truth of staff curves on 40 pages, the fitted curves have a median error of 0.93 px, or 0.19 times the thickness of a printed line, and they meet the acceptance threshold on 39 pages out of 40. The single failure is a faded photograph whose staff detection fails upstream. The degradation observed beyond the printed lines is therefore not a consequence of a poor fit.

4.4. Limitations

Several limitations bound these results. The corpus comes from a single institution and is split by image deformation rather than by writer, so a learned predictor may exploit handwriting regularities shared between training and evaluation pages while a geometric method cannot. The evaluation sets are small,

and the notes are not independent, since errors cluster by page, as Figure 4 illustrates. The detectors are compared at a single operating point and trained with one seed, and the preprocessing steps are not ablated. The annotations are not error-free either: inspecting the pages shown here revealed a small number of notes whose recorded pitch is doubtful, and since we did not audit the corpus systematically we cannot say how many such cases it contains, nor in which direction they bias the two approaches. Both are measured against the same annotations, so the comparison is unaffected, but the absolute rates should be read as lower bounds. Finally, measure segmentation and harmony validation are not evaluated here.

5. Conclusions

We set out to remove the last manual step between a 4-part harmony exercise written by hand and the automatic feedback Sharpmony can give on it, and we tested a specific hypothesis along the way: that reconstructing the pre-printed staff would be enough to read the pitch of a handwritten note more reliably than a detector trained to predict it. The hypothesis holds only inside the printed staff, and the boundary where it stops holding is the result of this paper.

The pipeline recovers nine notes in ten with the right type and the right pitch, enough for the exported score to be checked, and a detector trained to predict both at once does no better. Where the two differ is not in how often they fail but in where. Inside the printed staff the geometric reading is the more accurate; outside it, where the writer supplies the grid by hand, neither strategy holds. What matters is that this boundary is visible in advance, so the notes at risk can be marked for confirmation rather than reported with a pitch the system has no reliable way to assign. For a student the difference is concrete: an unflagged misreading reaches the harmony check as an error of their own, a flagged one costs a confirmation. The score does not have to be perfect either, since Sharpmony already provides the editor in which it lands, so the student corrects the few notes the system flags instead of entering the exercise from scratch. Underneath it is a division of labor between two kinds of knowledge, one learned because it could not be specified, the other specified once and exactly, and it is the second that states its own domain and therefore predicts where it will fail.

The decomposition points to a natural next design: a single system that reads geometrically inside the staff and hands the notes beyond it to a learned classifier working on the same crops. Reading the hand-drawn ledger strokes rather than extrapolating the printed grid is the harder half of that, and a larger corpus is needed before either can be settled. The pipeline currently stops at transcription. What turns it into feedback is the stage above: grouping the notes into measures, which is not trivial when many exercises carry no barlines at all, and checking each measure against the harmony rules. Both are rules that can be written down, like the ones the reading already applies.

In the classroom the exercise stays on paper. The student writes it in the notebook, photographs the finished page, and gets it back transcribed in the Sharpmony editor with the notes beyond the printed staff marked for confirmation. A page carries around thirty notes, four or five of them outside the staff, so confirming a handful replaces several minutes of typing, and that typing is the step students skip. The teacher sees what the system read and what it declined to read. We will test this with one conservatory class, measuring confirmations per exercise and how many exercises each student submits compared with the editor-based workflow. The second number decides whether the feature is useful. Sharpmony has checked over 21,000 exercises since 2020, and every one was entered in an editor first. Growing the corpus is the constraint on everything else, and the exercises that would grow it are written by students in their own notebooks. We therefore plan to explore federated learning over that data, so that the detector improves from handwriting it never has to leave the device to see.

Acknowledgments

This activity was 85% co-financed by the European Union, the European Regional Development Fund, and the Regional Government of Extremadura through its Managing Authority, Ministry of Finance,

Project GR24142. It was also supported by the Spanish Ministry of Economy and Competitiveness (PID2023-147409NB-C22), funded by MCIN/AEI/10.13039/501100011033, and by HES-SO R&I MTG (140012/RI-PSTRAT25-06) and ELITE (140090/RI-PSTRAT25-05).

Declaration on Generative AI

The authors used ChatGPT, Claude, and Grammarly for language editing. All generated suggestions were reviewed and edited by the authors, who take full responsibility for the content of this publication.

References

- [1] J. Novotný, J. Pokorný, Introduction to optical music recognition: Overview and practical challenges, in: CEUR Workshop Proceedings, 2015.
- [2] F. Fernández de Vega, J. Alvarado, A. Sánchez, M. Serrano, E. Pacioni, Evolutionary algorithms: A new hope for the future of music teaching, in: GECCO '23 Companion, ACM, 2023. doi:10.1145/3583133.3596945.
- [3] J. Alvarado, E. Pacioni, F. Fernández de Vega, Beneficios mutuos de la enseñanza-aprendizaje máquina-humano, 2025. URL: <https://openreview.net/forum?id=mSgy8kYjxl>.
- [4] E. Pacioni, F. Fernández de Vega, Advancing 4-part evolutionary harmony through analysis of human-machine approaches to teaching-learning, Complexity (2025). doi:10.1155/cplx/3086287.
- [5] Q. Gao, Z. Yan, C. Zhao, Y. Pan, L. Mo, To ban or not to ban: Differences in mobile phone policies at elementary, middle, and high schools, Computers in Human Behavior (2014). doi:10.1016/j.chb.2014.05.011.
- [6] M. Antoninis, et al., Global education monitoring report 2023: Technology in education: A tool on whose terms?, 2023.
- [7] A. Pacha, J. Hajič Jr., J. Calvo-Zaragoza, A baseline for general music object detection with deep learning, Applied Sciences (2018). doi:10.3390/app8091488.
- [8] J. Calvo-Zaragoza, J. Hajič Jr., A. Pacha, Understanding optical music recognition, ACM Computing Surveys (CSUR) (2020). doi:10.1145/3397499.
- [9] J. Hajič Jr., P. Pecina, The MUSCIMA++ dataset for handwritten optical music recognition, in: 14th IAPR ICDAR, 2017. doi:10.1109/ICDAR.2017.16.
- [10] A. Baró, P. Riba, J. Calvo-Zaragoza, A. Fornés, From optical music recognition to handwritten music recognition: A baseline, Pattern Recognition Letters (2019). doi:10.1016/j.patrec.2019.02.029.
- [11] C.-Y. Wang, I.-H. Yeh, H.-Y. Mark Liao, YOLOv9: Learning what you want to learn using programmable gradient information, in: ECCV, Springer, 2025. doi:10.1007/978-3-031-72751-1_1.
- [12] A. Marselletti, E. Pacioni, F. Fernández de Vega, D. Calvaresi, Decentralized optical music recognition using YOLO and FedGP for music education, in: MetroXRaine, IEEE, 2025. doi:10.1109/MetroXRaine66377.2025.11340382.
- [13] F. Fernández de Vega, J. Alvarado, J. V. Cortez, Optical music recognition and deep learning: An application to 4-part harmony, in: 2022 IEEE CEC, 2022. doi:10.1109/CEC55065.2022.9870357.
- [14] J. Sauvola, M. Pietikäinen, Adaptive document image binarization, Pattern Recognition (2000). doi:10.1016/S0031-3203(99)00055-2.
- [15] R. O. Duda, P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM (1972). doi:10.1145/361237.361242.
- [16] R. Sapkota, R. H. Cheppally, A. Sharda, M. Karkee, YOLO26: Key architectural enhancements and performance benchmarking for real-time object detection, 2026. URL: <https://arxiv.org/abs/2509.25164>.