From Notes to Keys: A VR Learning Environment for Sheet Music Interpretation

Sandeep Khanna[1,3][0000−0001−7272−6124]⋆, Atanu Saha[2,3][0000−1111−2222−3333]⋆⋆, Rahul Kumar Ray[3][0000−0002−6050−3158], Rakesh Patibanda[4][0000−0002−2501−9969], and Chiranjoy Chattopadhyay[3][0000−0002−3431−0483]

[1] Indian Institute of Technology Jodhpur, Jodhpur, India
[2] Ramakrishna Mission Vivekananda Educational and Research Institute, Belur, India
[3] FLAME University, Pune, India
[4] Monash University, Australia
{sandeep.khanna, atanu.saha,,rahul.ray,chiranjoy.chattopadhyay}@flame.edu.in, rakesh@exertiongameslab.org

Abstract. Learning to interpret sheet music and play musical instruments (piano) remains a significant challenge for beginners, often requiring extensive practice and guidance. Existing platforms lack real-time feedback and seamless sheet music interpretation, creating inefficiencies that require a system for accurate note recognition, intuitive guidance, and reduced cognitive load. To address this, we propose a novel system that integrates optical music recognition (OMR) and virtual reality (VR) to create an immersive piano learning environment. The proposed approach improves the detection of musical notes by making it scale and rotation invariant. The detected notes are converted into the corresponding piano keys and sequential instructions. These instructions are then visualized in a VR environment in Meta Quest 3, where a virtual piano highlights keys dynamically to guide the user. The experimental results demonstrate high accuracy in note detection and significant improvements in the learning curve for beginners, reducing cognitive load, and bridging the gap between sheet music interpretation and piano playing. This work highlights the potential of combining document image analysis and VR technologies to revolutionize music education, as well as other related fields, offering a scalable and accessible solution.

Keywords: Optical Music Recognition · Document Recognition· Virtual Reality · Symbol spotting · Immersive Learning · Music Education.

1 Introduction

Learning to interpret staff notation is one of the most challenging aspects of music education, particularly for beginners. Unlike textual reading, which pro-

⋆ Equal contribution
⋆⋆ Equal contribution

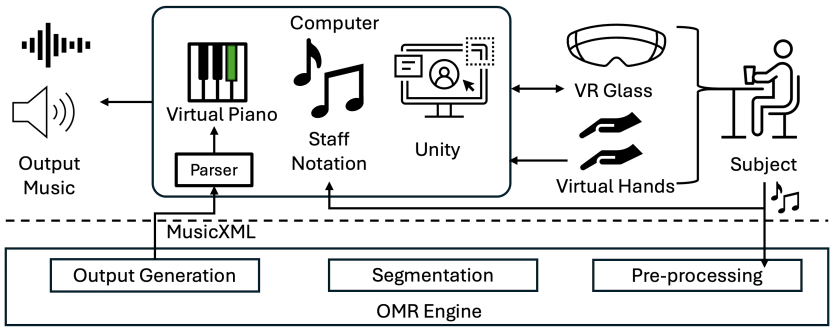

Fig. 1: Schematic representation of the proposed system for immersive music learning. The Optical Music Recognition (OMR) engine processes staff notation through pre-processing, segmentation, and output generation to produce a MusicXML file. This output musicXML file is parsed and is integrated into a Unity-based virtual piano environment, where keys are highlighted and played interactively using Meta Quest 3 and virtual hands, enabling users to learn and practice music in real time.

gresses linearly, music notation simultaneously conveys multiple dimensions of information, including pitch, rhythm, and expression [28]. For novices, this multidimensionality can be overwhelming. They must memorize note locations on the staff, understand rhythmic values, and internalize key signatures, all while coordinating the physical act of playing an instrument. Furthermore, sight reading introduces another layer of complexity, requiring rapid recognition and real-time response to notation [1]. This disconnect between theoretical understanding and practical application often delays progress and discourages learners [7]. Studies have suggested that many beginners struggle to associate musical symbols with their corresponding actions on an instrument, leading to frustration and slower learning outcomes [10,15]. This paper proposes a novel approach that integrates optical music recognition (OMR) and virtual reality (VR) in a seamless manner (see Fig. 1) to bridge this gap and ensure a seamless learning experience.

OMR offers a potential solution by automating the interpretation of musical notation and converting sheet music to machine-readable formats such as MIDI or MusicXML [11]. OMR can help beginners by reducing the cognitive burden associated with manual interpretation, enabling them to focus more on developing and performing skills [6,24,33]. Recent advances in deep learning and pattern recognition have significantly enhanced the accuracy and reliability of OMR, particularly through convolutional neural networks (CNNs), which have shown high precision in recognizing musical symbols [13]. However, OMR still faces challenges in handling handwritten notations, degraded sheet music, and complex polyphonic compositions [4], highlighting the need for further advancements in robustness and adaptability [9,31].

Table 1: Summary of Recent works in various related categories.

Category	Description	Datasets	References
Preprocessing	Binarization, noise reduction, deskewing, and staff-line removal for improved readability in scanned or camera-based images.	CVC-MUSCIMA, MUSCIMA++, camera-captured photos	[26], [3], [18]
Segmentation	Identifying graphical objects using connected components or deep learning-based object detection (e.g. Mask R-CNN, stave-aware methods).	MUSCIMA++, DeepScores, CPMS	[12], [20], [27]
Symbol Recognition	Classifying musical symbols (notes, clefs, rests) using CNNs, template matching, or feature-based methods.	MUSCIMA++, Universal Music Symbol Collection	[26], [12], [29]
Deep Learning in OMR	End-to-end sequence models (CNN+RNN, Transformers) for full-score transcription; includes multi-dataset training for broader generalization.	DeepScores, Camera-PrIMuS, OpenScore Lieder corpus, synthetic GrandStaff dataset	[8], [5], [20], [17], [16]
Post-processing & Assembly	Reconstructing musical notation using rule-based methods, graph approaches, or sequence models.	Outputs from earlier datasets	[21], [2]

The document analysis and recognition (DAR) community has looked at this research problem extensively and proposed techniques, such as binarization, edge detection, and symbol segmentation, form the backbone of OMR workflows, facilitating the detection and classification of musical symbols [24,31]. Deep learning approaches, particularly CNNs, have further enhanced OMR by learning hierarchical characteristics, allowing improved recognition of musical structures in both printed and handwritten notation [6, 9]. Beyond this, immersive technologies such as augmented reality (AR) and virtual reality (VR) are transforming education across various domains [14]. These technologies have shown promise in education by increasing participation, retention, and skill acquisition [19]. In music learning, virtual reality environments can simulate instruments, provide real-time feedback, and create interactive sight-reading exercises without the need for physical instruments. By integrating VR with OMR in conjunction with tactile feedback, we can bridge the gap between sheet music interpretation and practical performance, offering a seamless and intuitive learning experience [22, 23, 25].

Table 1 provides a comparison of recent OMR approaches, highlighting the data sets used, the methodologies, the evaluation metrics, and the key findings. OMR is challenged by a lack of standardized terms and formats, which complicates system interoperability and result comparison. Encoding musical information is complex due to the need to interpret a wide range of interdependent

symbols such as notes, dynamics, and annotations in a two-dimensional space. Overlapping elements, such as beams and slurs, further complicate segmentation, potentially leading to transcription errors. Despite advances in machine learning and deep learning that improve accuracy, issues such as data scarcity and domain adaptation keep OMR a challenging research field.

This paper presents a document analysis system that integrates OMR and VR to improve the learning experience of music through the following contributions:

- **Methodological:** An improved scale and rotation invariant OMR recognition approach is proposed. In addition, a structured pipeline is also introduced to map OMR-extracted data to real-time VR interactions. This workflow can be used by HCI and AI researchers to design interactive learning environments by linking document recognition outputs with dynamic virtual representations.
- **Artifact:** We develop a prototype software that converts sheet music into an interactive VR learning interface. This system can be used by developers and music technologists to create OMR-driven music applications by incorporating real-time visual feedback in VR.
- **Empirical:** We evaluate the impact of OMR-VR integration on learning engagement and skill acquisition. Educators and researchers can use this evaluation framework to assess immersive music learning in user studies and classroom settings.

The rest of the paper is organized as follows. Section 2 presents the framework for music recognition and the algorithm for generating MusicXML. Section 3 illustrates the results and provides a discussion on the final results. Lastly, Section 4 wraps up the paper and outlines directions for future research.

2 System Design and Implementation

This section describes the architecture and components of the VR learning environment, detailing how each element contributes to the overall user experience. The design and implementation of our proposed system integrates OMR and VR to create an immersive music learning environment. This modular approach seamlessly transitions from sheet music interpretation to practical instrument learning in a VR environment.

2.1 Proposed Optical Music Recognition Approach

This methodology, shown in Fig. 2, includes four stages: preprocessing, deep learning detection, post-processing, and output generation. It integrates classical image processing (rotation, affine corrections) with deep learning models Real-ESRGAN [32] for enhancement and dual U-Nets for the segmentation of staff lines and symbols. By separating staff-line and symbol detection, it utilizes the music notation structure to convert sheet music into MusicXML for compatibility and piano key generation for auditory or visual feedback. The subsystems

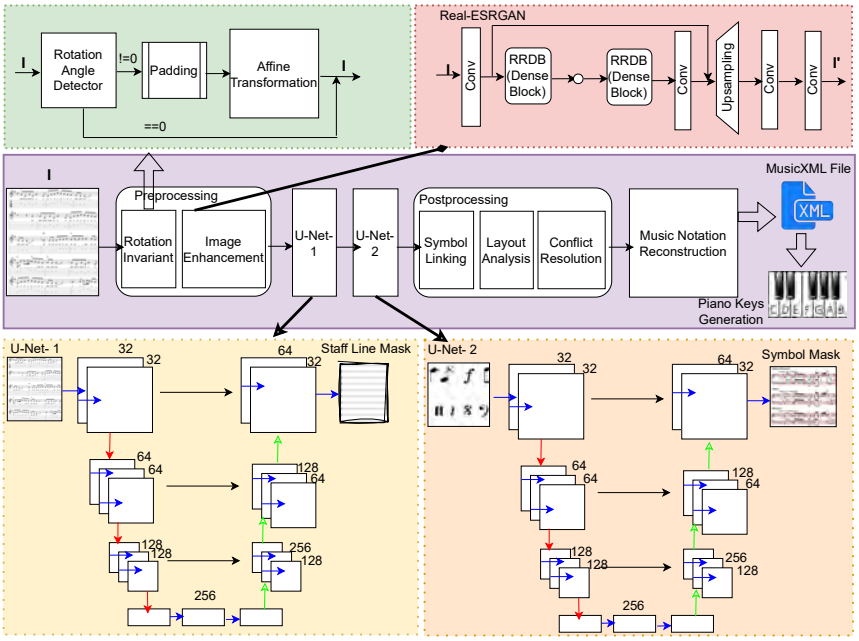

Fig. 2: A unified framework for music sheet symbol segmentation, illustrating the preprocessing stage (including rotation correction and Real-ESRGAN enhancement), separate U-Net models for staff line and symbol segmentation, and a postprocessing pipeline that converts the output into MusicXML, enabling automatic piano key generation.

handle challenges such as scanning imperfections, improving notation details, and accurate layout analysis.

Preprocessing

Rotation Angle Detector Music scores are frequently scanned at slight angles, which can degrade the accuracy of recognition. Small rotation errors significantly hamper staff line and symbol segmentation. Automatic detection ensures that subsequent convolutional networks see a properly oriented image and can learn staff/symbol features more accurately. The system first estimates a rotation angle θ that best aligns the staff lines horizontally.

We have employed the Hough transform, which is applied on detected staff lines or edges and then searches for the global maximum in the Hough accumulator space. For image I, E is an edge map, the system solves for

$$\theta^* = argmax_\theta \sum_{(x,y)} \mathcal{H}(f(x, y, \theta) \quad (1)$$

where $\mathcal{H}$ is the Hough accumulator and $f(.)$ the function mapping (x,y) and θ is the parameter space.

Padding To preserve border regions and avoid losing staff lines near edges, the image is padded uniformly on all sides. A simple zero-padding method is used. For I with dimension (H, W), we created a padded image I_p with dimensions $(H + 2p, W + 2p)$.

$$I_p(x, y) = \begin{cases} 0, & \text{if } x \text{ or } y \text{ falls outside } I, \\ I(x - p, y - p), & \text{otherwise.} \end{cases} \tag{2}$$

Padding ensures that transformations and subsequent downsampling/upsampling in U-Nets do not cut off relevant musical symbols near the image boundary.

Affine Transformation If the detected rotation angle $\theta^*! = 0$ an affine rotation correction. Proper alignment of staff lines is essential so that the neural networks (e.g., U-Nets) can focus on learning relevant shapes of staves and symbols rather than dealing with random rotations.

Image Enhancement Low-quality scans suffer from blur, noise, and compression artifacts. Real-ESRGAN (Enhanced Super-Resolution Generative Adversarial Network) aims to improve image clarity and resolution before recognition. Let G_{ESRGAN} be the be the generator network, parameterized by θ_G. The enhanced image I' is produced by

$$I' = G_{ESRGAN}(I|\theta_G) \tag{3}$$

Internally RRDB (Residual-in-Residual Dense Blocks) provide deeper context and more robust feature extraction for super-resolution. The details of Real-ESRGAN can be found in this paper [32]. Clearer staff lines and sharper note heads are crucial for accurate segmentation. Removal of artifacts also reduces false positives in symbol detection.

U-Net–Based Segmentation The framework introduces two unnet architectures inspired by [34], where U-Net-1 detects staff lines, whereas U-Net-2 detects musical symbols (notes, clefs, rests, etc.).

U-Net-1 performs the binary segmentation of the image into staff-line vs. background. I' the enhanced image be the input to U-Net-1, the output is the staff-line mask $M_{\mathcal{S}}(x, y) \in \{0, 1\}$

U-Net-1 is a fully convolutional encoder–decoder $M_{\mathcal{S}} = f_{U1}(I')$, where f_{U1} denotes the parameterized function learned by U-Net-1. The loss function typically combines a pixelwise cross-entropy and the Dice coefficient to balance class imbalance.

$$L_{U1} = -\sum_{x,y} \Big[W_{\mathcal{S}} y_{\mathcal{S}} \log M_{\mathcal{S}}(x, y) + w_{bkg}(1 - y_{\mathcal{S}}(x, y)) \log(1 - M_{\mathcal{S}}) \Big] + \lambda \mathcal{L}_{Dice} \tag{4}$$

U-Net-2 for Symbol Mask: U-Net-2 is similar to an encoder-decoder network $M_{Sym} = f_{U2}(I')$ where $M_{Sym}(x, y) \in \{0, 1\}$ indicates whether a pixel belongs to a music symbol.

Separating staff-line and symbol detection enhances performance by allowing networks to specialize in distinct feature sets. After recognizing staff lines, a second pass isolates music symbols like notes, clefs, rests, and accidentals.

- **(1) Notehead Extraction and Classification:** Each notehead is detected, its bounding box measured, and its vertical position mapped to a pitch. This yields symbol type (τ_i) for example, note or rest—as well as pitch and accidental information ($\mathcal{A}_\rangle$). Determining whether a notehead is solid or hollow (quarter vs. half/whole) sets an initial idea of duration (d_i).
- **(2) Note Grouping (Chords) and Stem Analysis:** The pipeline groups noteheads that occur together vertically into chords (e.g., multiple noteheads on one stem) and identifies stem directions. This step pinpoints the time-overlap among notes and corrects multi-voice or multi-track layouts.
- **(3) Symbol Extraction (Accidentals, Clefs, Barlines, etc.):** Beyond noteheads, the pipeline extracts clefs, accidentals, barlines, and other notation. Accidentals become attributes (A_i), barlines help the algorithm infer measure boundaries, and clefs confirm the pitch mapping.
- **(4) Rhythm Refinement (Dots, Beams, Flags)**: Finally, the pipeline analyzes beams/flags and dots to determine exact note durations. By counting the number of beams or flags in each group, the system refines τ_i (symbol type) and d_i (duration). Once durations are assigned, the set S (of all symbols) is complete: each entry s_i comes with a track t_i, horizontal position x_i, symbol type τ_i, duration d_i, and an attributes set A_i.

PostProcessing: After obtaining $M_{\mathcal{S}}$ and $M_{\mathcal{S}\dagger\Updownarrow}$ the symbol components are linked, the layout is analyzed, and the conflicts are resolved.

Symbol Linking: Each recognized symbol must be linked as a single unit so that the system can classify it (e.g., as a note head, rest, clef, etc.). Many symbols may appear fragmented in multiple connected components. We group the connected pixels in M_{sym} into coherent symbol objects. Contour detection is used to gather clusters of pixels. Centroids or bounding boxes are computed for each symbol cluster. $C = \{C_1, C_2, .., C_n\}$, where each C_i is a set of pixels that belong to the i-th symbol.

Layout Analysis: Music notation is highly dependent on positioning i.e.,, the line or space in a stave. Layout analysis ensures that each symbol is correctly associated with its pitch and timing context. To determine how symbols are arranged relative to the staff line, to which staff lines a symbol belongs, to vertical alignment of simultaneous notes, and to measure boundaries. We map the bounding box $(x_{min}, y_{min}, x_{max}, y_{max})$, of each symbol, to its nearest staff lines in M_S. The vertical position relative to staff lines is used to infer pitch, while horizontal grouping can align notes in the same chord.

Conflict Resolution Sheet music is rich with overlapping notation like ledger lines, accidentals, ties/slurs, so a cleanup step is necessary to refine the final set of musical symbols. When multiple symbols overlap or ambiguous segmentations occur, the system must remove or correct spurious classifications. Simple heuristics (e.g., no two note heads can occupy the same staff position at the same time) or learned rules can be applied. A conflict is flagged when the bounding boxes overlap beyond a threshold or if the staff position is inconsistent with the recognized notes.

Music Notation Reconstruction Optical Music Recognition is not complete until the symbolic representation is generated. MusicXML is a standard, interoperable format used by many notation softwares. After segmentation and conflict resolution, the recognized symbols and their positions must be converted into an actual music-notation data structure. Each detected symbol is classified as note head, rest, clef, accidental, etc., by morphological shape matching. The position on the staff yields pitch; shape (filled, unfilled, presence of a stem/flag) yields duration. The system assembles the recognized notes, measures, and other musical directives into a valid MusicXML file.

Output Generation To store the recognized notation in an industry-standard format for further editing, rendering, or MIDI playback. We employ a rule-based parser that writes out staves, measures, notes, and directions into MusicXML tags. The final output is an XML file with complete music information. To produce a quick reference or basic playback mechanism by mapping detected pitches to piano key numbers. We convert each note's pitch class (C, D, E, etc.) and octave to a piano key index. The detected keys were fed into the virtual reality system to provide a real-world experience of playing and learning piano.

Algorithm 1 offers a detailed and systematic approach to sorting and aligning musical symbols, adjusting their timing, and ultimately exporting the entire composition into a MusicXML format. The procedure is as follows:

- **Sort Symbols:** Begin with a collection of musical symbols (such as notes or rests) that are categorized into one or more groups (like layers, staves, or voices). Each symbol is labeled with a specific position on the x-axis and a group identifier.
  - First, sort these symbols based on their group allocation (e.g., all symbols in Track 1 followed by Track 2, and so on).
  - Then, sort within each group according to their x-axis position from left-most to right-most.
- **Initialize the First Measure:** Introduce the necessary symbols (clef, key signature, and time signature if applicable) at the start of the measure based on the user-chosen or detected clef and key. Set up internal structures to handle measures, such as creating a list of measures per track.
- **Determine the Alignment Across Tracks:** Identify corresponding symbols at each unique x-axis position on different tracks. Develop an alignment

Algorithm 1 MusicXML Generation Algorithm

Input: Set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ where $s_i = (t_i, x_i, \tau_i, d_i, \mathcal{A}_i)$ represents a musical symbol
    $t_i \in Z^+$: track identifier
    $x_i \in R^+$: horizontal position
    $\tau_i \in \{1, 2, \ldots, \Gamma\}$: symbol type (note, rest, etc.)
    $d_i \in Q^+$: duration in quarter notes
    $\mathcal{A}_i$: set of attributes (pitch, accidental, etc.)
Output: MusicXML document $\mathcal{X}$
Phase I: Symbol Ordering and Partitioning
Define total ordering relation $\prec$ on $\mathcal{S}$ where $s_i \prec s_j \iff (t_i < t_j) \vee ((t_i = t_j) \wedge (x_i < x_j))$
Sort $\mathcal{S}$ according to $\prec$ to obtain ordered set $\mathcal{S}'$
Partition $\mathcal{S}'$ into equivalence classes $\mathcal{T}_k = \{s_i \in \mathcal{S}' | t_i = k\}$ for $k \in \{1, 2, \ldots, m\}$ where $m = \max_i\{t_i\}$
Phase II: Temporal Quantization
Define quantization function $\mathcal{Q} : R^+ \to Q^+$ mapping positions to rational time points
For each $\mathcal{T}_k$, apply $\mathcal{Q}$ to create time-quantized sets $\hat{\mathcal{T}}_k = \{(s_i, \mathcal{Q}(x_i)) | s_i \in \mathcal{T}_k\}$
Define time signature $\sigma = (n, 2^p)$ where $n \in Z^+$ and $p \in \{0, 1, 2, 3, 4\}$
Let measure duration $\mu = \frac{n \cdot 4}{2^p}$ in quarter notes
Phase III: Track Alignment
Define alignment function $\mathcal{A} : R^+ \times Z^+ \to \mathcal{P}(\mathcal{S})$ mapping (position, track) to subsets of $\mathcal{S}$
Define temporal grid $\mathcal{G} = \{(g_i, \delta_i)\}$ where $g_i \in Q^+$ are alignment points and $\delta_i \in Q^+$ are durations
For each $(x, t) \in R^+ \times \{1, 2, \ldots, m\}$, compute $\mathcal{A}(x, t) = \{s_i \in \mathcal{S} | (t_i = t) \wedge (|\mathcal{Q}(x_i) - \mathcal{Q}(x)| < \epsilon)\}$
Construct bipartite graph $\mathcal{G}_B = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E})$ where:
    $\mathcal{V}_1 = \{(x, t) | \exists s_i \in \mathcal{S} : (t_i = t) \wedge (x_i = x)\}$
    $\mathcal{V}_2 = \{s_i \in \mathcal{S}\}$
    $\mathcal{E} = \{((x, t), s_i) | s_i \in \mathcal{A}(x, t)\}$
Compute maximum bipartite matching $\mathcal{M} \subset \mathcal{E}$
Phase IV: Rhythmic Normalization
Define measure map $\mathcal{M}_{\text{meas}} : Q^+ \to Z^+ \times Q^+$ mapping time to (measure number, offset)
For each $\hat{\mathcal{T}}_k$, compute beat positions:
    $\beta_i^k = \sum_{j=1}^{i-1} d_j^k \mod \mu$ for $s_j^k \in \hat{\mathcal{T}}_k$
Apply correction function $\mathcal{C} : Q^+ \times Q^+ \to Q^+$ where:
    $$\mathcal{C}(d, \beta) = \begin{cases} d & \text{if } \beta + d \leq \mu \\ \mu - \beta & \text{if } \beta + d > \mu \end{cases}$$
Initialize rest insertion set $\mathcal{R} = \emptyset$
For each measure $l \in \{1, 2, \ldots, L\}$ and track $k \in \{1, 2, \ldots, m\}$:
    Let $\mathcal{S}_{l,k} = \{s_i \in \hat{\mathcal{T}}_k | \mathcal{M}_{\text{meas}}(\mathcal{Q}(x_i)) = (l, \cdot)\}$
    If $\sum_{s_i \in \mathcal{S}_{l,k}} d_i < \mu$, add rest $r_{l,k}$ to $\mathcal{R}$ with $d(r_{l,k}) = \mu - \sum_{s_i \in \mathcal{S}_{l,k}} d_i$
Phase V: MusicXML Synthesis
Define transformation $\mathcal{T}_{\text{XML}} : \mathcal{S} \cup \mathcal{R} \to \mathcal{X}_{\text{elem}}$ mapping musical symbols to XML elements
Let $\mathcal{X}_{\text{meas}} = \{(l, \mathcal{X}_l) | l \in \{1, 2, \ldots, L\}\}$ be the set of measure XML elements
For each $s_i \in \mathcal{S} \cup \mathcal{R}$:
    $(l, \beta_i) = \mathcal{M}_{\text{meas}}(\mathcal{Q}(x_i))$
    Add $\mathcal{T}_{\text{XML}}(s_i)$ to $\mathcal{X}_l$ with offset β_i
Define XML document structure $\mathcal{X} = (\mathcal{X}_{\text{head}}, \mathcal{X}_{\text{part}})$ where:
    $\mathcal{X}_{\text{head}}$ contains part list, metadata, etc.
    $\mathcal{X}_{\text{part}} = \{(\mathcal{X}_{\text{part},k}, \mathcal{X}_{\text{meas}}) | k \in \{1, 2, \ldots, m\}\}$
Return $\mathcal{X}$

mapping where identical x-axis positions across tracks are deemed aligned (i.e., occurring simultaneously in the music timeline).
- **Adjust Rhythms and Insert Rests:** To ensure that aligned symbols initiate simultaneously in the musical score, certain tracks may need to be divided or extended by inserting rests to match the 'beat position' of other tracks. Modify duration or add rests as required, ensuring that the aligned symbols share the same temporal slot in the final notation.
- **Decode and Generate MusicXML:** Transform the synchronized collection of notes and rests, now correctly aligned by beat position, into MusicXML elements, including measures, attributes (like clef, key, and time signatures), notes (encompassing pitch, duration, voice), and rests.

2.2 Piano Interface Design in Virtual Reality

The VR design methodology integrates the OMR output with an immersive virtual piano environment developed in Unity 3D. The system leverages the Meta Quest 3 headset for visualization and interaction, allowing users to play the piano with virtual hands. This section details the implementation of the virtual piano interface, hand tracking, and interaction mechanisms that enable seamless music learning in VR.

The virtual piano is modeled as a realistic 3D object in Unity, with each key having individual colliders and animations. The piano keys are tagged as "PianoKey" to facilitate interaction and are linked to corresponding audio clips that represent their sounds. The PianoController script manages key interactions, sound playback, and animations. It uses a dictionary (keyMapping) to map keyboard inputs or hand interactions to specific keys, enabling dynamic octave switching and precise note identification. When a key is pressed, its animation is triggered using Unity's Animator component, simulating the physical movement of a piano key.

The Meta Quest 3 provides advanced hand-tracking capabilities, which are utilized to interact with the virtual piano. Each fingertip of the user's virtual hands is equipped with sphere colliders tagged as "FingerTip". These colliders detect proximity to piano keys and trigger interactions when entering or exiting the key's collider. A dedicated script (PianoKeyInteraction) handles these events, playing the corresponding sound, and animating the key when pressed. This mechanism mimics real-world piano playing by responding to natural hand movements.

A key highlighting feature is implemented using the PianoHighlightManager script to assist users in learning sheet music. This script dynamically changes the color or material of specific keys based on instructions derived from the OMR output (MusicXML). Keys are highlighted sequentially before being played, providing visual guidance for learners. The highlighting sequence synchronizes with the timing of the song, thereby ensuring that users can follow along intuitively.

The system uses spatial audio to enhance immersion. Piano sounds are played at the location of each keypress using Unity's AudioSource.PlayClipAtPoint method. This creates a realistic auditory experience that matches the visual

interaction. Sound fading techniques are also applied to simulate natural decay in piano notes. To ensure smooth operation in Meta Quest 3, several optimization techniques are employed:

- **Collider Management:** Efficient use of colliders ensures accurate detection without excessive computational overhead.
- **Animation Triggers:** Coroutine-based animations minimize performance impact while maintaining responsiveness.
- **Rendering Pipeline:** Unity's lightweight rendering pipeline is used to optimize graphics for VR.

When users wear the VR headset, they enter an immersive environment featuring a virtual piano and sheet music display. The OMR engine processes input sheet music into MusicXML format, which is fed into Unity to generate instructions for key highlighting and playback. Users interact with the piano using their virtual hands by pressing highlighted keys sequentially. The system provides real-time feedback through animations and audio cues, facilitating an engaging learning experience.

3 Experiments and Results

This section provides both the qualitative and quantitative outcomes of our proposed system. Our experiments have been performed using the DeepScores V2 dataset [30], which focuses on Music Object Detection and includes digitally generated images of sheet music along with their respective ground truth annotations.

3.1 Rotation Invariant

Fig. 3 illustrates the ability of our system to automatically correct for slight rotations in scanned music sheets. The top row shows the original image (left) along with versions that were intentionally rotated by $\pm 5^{\circ}$ and $\pm 15^{\circ}$. The bottom row displays the corresponding 're-rotated' outputs, where the system has detected each rotation angle and applied an inverse rotation to restore the music sheet to a proper upright orientation. This demonstrates the rotation invariance of our approach: if an end user places a sheet at a slight tilt, the system accurately detects and corrects it, ensuring that the final image is properly aligned for reading or further processing.

3.2 Image Enhancement and Segmentation

Fig. 4 presents a three-stage process to transform a blurred music sheet into a clean and well-segmented score. On the left, the "Blurred Image" shows a typical degraded scan caused by issues such as low-resolution scanning, camera shake, or paper wear. These factors obscure fine-notation details, including staff lines, note heads, and textual markings, making the sheet difficult to read or analyze.

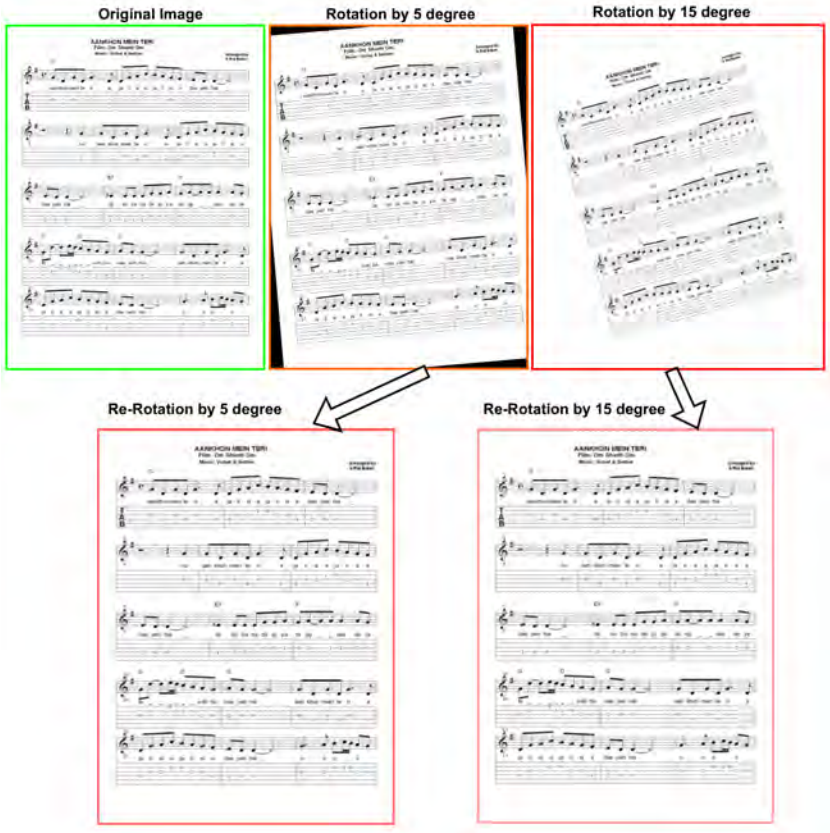

Fig. 3: Detection and correction of rotational skew in sheet music. The top row shows the original scan (left) and images rotated by +5° and +15°; the bottom row displays the respective 're-rotated' outputs restored to proper alignment.

In the middle, the "Enhanced Image" illustrates the results of applying Real ESRGAN, a generative adversarial network designed to infer and reconstruct lost image details. By reducing noise and sharpening edges, Real ESRGAN significantly improves the legibility of the notation. Staff lines and musical symbols become more defined, bridging the gap between a low-quality scan and a near-original quality sheet. This enhanced clarity benefits both human readers and automated systems, such as optical music recognition (OMR), which rely on clear notation to accurately extract musical information.

Finally, on the right, the "Segmented Image" demonstrates the output of two U-Net segmentation models: one trained to isolate staff lines and the other trained to detect symbols. This dual segmentation approach allows for a more precise separation of notation components, ensuring that important elements, such as noteheads, rests, and chord names, are correctly identified. Building on

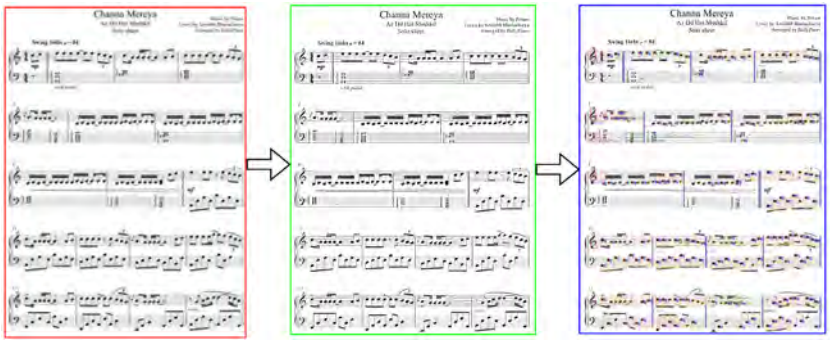

Fig. 4: Enhanced and Segmented Results: From left to right: (1) the original blurred music sheet, (2) the high-resolution enhanced sheet obtained via Real-ESRGAN, and (3) the segmented music sheet produced by two U-Net models.

the sharply enhanced image produced by Real ESRGAN, the segmentation step achieves higher accuracy, establishing a robust foundation for further tasks such as transcription, editing, or analysis of the musical score.

3.3 Quantitative Evaluation

Despite advances, handling degraded images remains a challenge in DAR. Noise, blur, low resolution, and compression artifacts arise from poor lighting, motion, low-quality sensors, or transmission errors. These challenges hinder traditional models, which are usually trained in high-quality images, leading to poor performance in degraded images. For quantitative evaluation, we used DeepScores V2 [30], which contains musical sheet images and segmentation masks. We tested the proposed approach with Gaussian blur levels from 13 to 39, in increments of 2, to assess robustness against perturbations. Table 2 shows that the proposed method consistently outperforms the OEMER method, achieving higher IoU scores and accurately segmenting objects in blurred images. It maintains high IoU scores (above 0.90) up to blur level 19, while OEMER's performance declines as blur increases. For higher blur levels (21 and above), the proposed method remains reliable, whereas OEMER struggles or fails to detect results (marked 'ND'). This illustrates the superior ability of the proposed method to manage severe image degradation, offering a more robust solution for blurred images in the DeepScoresV2 dataset, maintaining accuracy and reliability under challenging conditions.

Similarly, we also experimented by adding Gaussian, salt and pepper, and speckle noise at different levels and observed and compared the IOU with the existing method. Table 3 shows that the proposed method demonstrates superior noise robustness compared to OEMER for all noise types tested (Gaussian, Speckle). At lower noise levels (0.01 to 0.1), the proposed method consistently achieves higher IoU scores greater than 0.91, while OEMER shows a gradual

Table 2: Comparison of Intersection over Union (IoU) scores between the OEMER and the proposed method across varying levels of image blur on the DeepScoresV2 dataset. "ND" indicates no detectable result for the given blur level.

Blur Level	OEMER [34]	Our Proposed	Blur Level	OEMER [34]	Our Proposed
13	0.91	0.96	27	0.74	0.89
15	0.89	0.95	29	0.71	0.88
17	0.87	0.94	31	ND	0.62
19	0.84	0.94	33	ND	0.50
21	0.83	0.91	35	ND	ND
23	0.81	0.90	37	ND	ND
25	0.80	0.90	39	ND	ND

Table 3: Comparison of Intersection over Union (IoU) scores between the OEMER and the proposed method under varying noise types (Gaussian, Salt_Pepper) and noise levels on the DeepScoresV2 dataset. “ND” denotes no detectable result.

Noise Level	OEMER [34]		Our Proposed	
	Gaussian	salt_pepper	Gaussian	salt_pepper
0.02	0.94	0.94	0.96	0.96
0.01	0.93	0.92	0.96	0.95
0.1	0.90	0.91	0.93	0.93
0.2	0.85	0.89	0.91	0.92
0.5	ND	ND	0.78	0.74

decline in performance (IoU from 0.94 to 0.87). In particular, at higher noise levels (0.2–0.5), the proposed method maintains usable IoU results greater than 0.73), while OEMER fails entirely (ND) under severe noise (level 0.5).

3.4 Virtual Environment

Figure 5 refers to the typical scenario of how the virtual world is able to get the required information from the real world and generate the audio signal, and highlights the keys to better human-machine interaction. As indicated, the MusicXML file generated by the proposed approach gives the cue (rel-time instructor) to which keys to be highlighted at what pitch. In the VR environment, such cues are used to highlight the key which the user plays. The interactive learning environment created by our VR system has shown promising initial results. The dynamic highlighting of piano keys, derived from the OMR output, provides an intuitive guide for users to follow sheet music accurately. This feature significantly reduces cognitive load by visually directing users to the correct keys in real time. Preliminary tests with basic songs such as "Happy Birthday," "Jingle Bells," demonstrate that the system effectively bridges the gap between sheet music interpretation and practical piano playing. We have also experimented with music sheets of popular songs from the interent and ob-

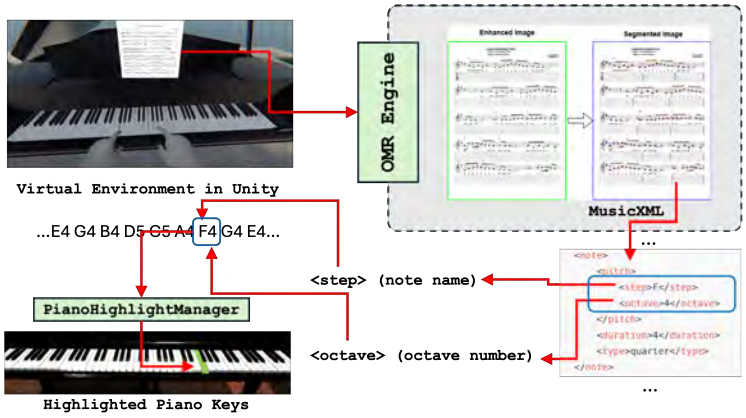

Fig. 5: Virtual Piano with a highlighted key.

served similar results. The audio files played using highlighted keys closely match the original compositions, demonstrating the system's ability to support precise musical performance. Initial impressions indicate that the VR piano learning system is both engaging and user-friendly. Its combination of visual guidance and auditory feedback offers an immersive, interactive music learning experience. Highlighted keys serve as a real-time instructor, allowing beginners to play melodies without prior training. This feature enhances accessibility and supports future studies on learning outcomes, user engagement, and system usability.

4 Conclusion and Future Work

The proposed approach integrates Optical Music Recognition (OMR) with Virtual Reality (VR) to create an immersive music learning environment, making twofold contributions. Firstly, the OMR engine employs advanced preprocessing techniques, including rotation correction and Real-ESRGAN-based image enhancement, alongside dual U-Net architectures for staff line and symbol detection. The post-processing steps refine recognition, ensuring adherence to the musical conventions in the reconstructed MusicXML files with varying resolution. This makes the proposed approach for the rotation and scale recognition of OMR invariant. Secondly, the VR component translates this output into an interactive Unity-based virtual piano environment using the hand-tracking capabilities of Meta Quest 3. Preliminary tests demonstrate its effectiveness in bridging theoretical sheet music knowledge with practical piano skills. Future directions include expanding OMR to support handwritten music, developing mobile VR applications for accessibility, conducting formal user studies to measure learning outcomes, and incorporating adaptive learning algorithms and gamification elements. These advancements aim to establish the system as a scalable tool for interactive music education.

References

1. Arthur, P., Khuu, S., Blom, D.: Music sight-reading expertise, visually disrupted score and eye movements. Journal of Eye Movement Research **9** (10 2016). https://doi.org/10.16910/jemr.9.7.1
2. Baró, A., Riba, P., Calvo-Zaragoza, J., Fornés, A.: Optical music recognition by long short-term memory networks. In: Graphics Recognition. Current Trends and Evolutions: 12th IAPR International Workshop, GREC 2017, Kyoto, Japan, November 9-10, 2017, Revised Selected Papers 12. pp. 81–95. Springer (2018)
3. Calvo-Zaragoza, J., Gallego, A.J., Pertusa, A.: Recognition of handwritten music symbols with convolutional neural codes. pp. 691–696 (11 2017). https://doi.org/10.1109/ICDAR.2017.118
4. Calvo-Zaragoza, J., Jr, J.H., Pacha, A.: Understanding optical music recognition. ACM Computing Surveys (CSUR) **53**(4), 1–35 (2020)
5. Calvo-Zaragoza, J., Rizo, D.: End-to-end neural optical music recognition of monophonic scores. Applied Sciences **8**(4) (2018). https://doi.org/10.3390/app8040606, https://www.mdpi.com/2076-3417/8/4/606
6. Castellanos, F.J., Gallego, A.J., Fujinaga, I.: Deep learning for optical music recognition: A review. TechRxiv (February 2025). https://doi.org/10.36227/techrxiv.174077177.78767136/v1
7. Chiu, S.C., Chen, M.S.: A study on difficulty level recognition of piano sheet music. 2012 IEEE International Symposium on Multimedia pp. 17–23 (2012), https://api.semanticscholar.org/CorpusID:36545038
8. van Der Wel, E., Ullrich, K.: Optical music recognition with convolutional sequence-to-sequence models. arXiv preprint arXiv:1707.04877 (2017)
9. Dos Santos, J., Zanlorensi, L.A., Oliveira, L.A.: Deepomr: A deep learning approach for optical music recognition. In: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). pp. 291–298 (2018). https://doi.org/10.1109/SIBGRAPI.2018.00044
10. Fourie, E.: The processing of music notation: Some implications for piano sight-reading. Journal of the Musical Arts in Africa **1** (01 2004). https://doi.org/10.2989/18121000409486685
11. Fujinaga, I., Hankinson, A., Pugin, L.: Automatic score extraction with optical music recognition (omr). Springer Handbook of Systematic Musicology pp. 299–311 (2018)
12. Hajic Jr, J., Dorfer, M., Widmer, G., Pecina, P.: Towards full-pipeline handwritten omr with musical symbol detection by u-nets. In: ISMIR. pp. 225–232 (2018)
13. Ju, Y., Wang, X., Chen, X.: Research on omr recognition based on convolutional neural network tensorflow platform. In: 2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). pp. 688–691. IEEE (2019)
14. Lal, S.A., Chattopadhyay, C., Ray, R.K.: An approach for effective cpr trainings in virtual reality with multimodal feedback. In: IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). pp. 326–331 (2025)
15. Lehmann, A., Ericsson, K.: Sight-reading ability of expert pianists in the context of piano accompanying. Psychomusicology: A Journal of Research in Music Cognition **12**, 182–195 (10 1993). https://doi.org/10.1037/h0094108
16. Martinez-Sevilla, J.C., Rosello, A., Rizo, D., Calvo-Zaragoza, J.: On the performance of optical music recognition in the absence of specific training data. In: ISMIR. pp. 319–326 (2023)

17. Mayer, J., Straka, M., Hajič, J., Pecina, P.: Practical end-to-end optical music recognition for pianoform music. In: International Conference on Document Analysis and Recognition. pp. 55–73. Springer (2024)
18. Ng, W., Nguyen, X.T.: Improving deep-learning-based optical music recognition for camera-based inputs. In: 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS). pp. 118–121 (2022). https://doi.org/10.1109/AICAS54282.2022.9869934
19. Oueida, S., Awad, P., Mattar, C.: Augmented reality awareness and latest applications in education: A review. International Journal of Emerging Technologies in Learning (iJET) **18**, 21–44 (07 2023). https://doi.org/10.3991/ijet.v18i13.39021
20. Pacha, A., Calvo-Zaragoza, J.: Optical music recognition in mensural notation with region-based convolutional neural networks. In: ISMIR. pp. 240–247 (2018)
21. Pacha, A., Calvo-Zaragoza, J., Hajic Jr, J.: Learning notation graph construction for full-pipeline optical music recognition. In: ISMIR. pp. 75–82 (2019)
22. Patel, P., Ray, R.K., Manivannan, M.: Power law based "out of body" tactile funneling for mobile haptics. IEEE transactions on haptics **12**(3), 307–318 (2019)
23. Pietrzak, T., Ray, R.K.: Comparing apparent haptic motion and funneling for the perception of tactile animation illusions on a circular tactile display. IEEE Transactions on Haptics (2025)
24. Pugin, L.: Optical music recognitoin of early typographic prints using hidden markov models. pp. 53–56 (01 2006)
25. Ray, R.K., Kumar Vasudevan, M., Manivannan, M.: Electrotactile displays: taxonomy, cross-modality, psychophysics and challenges. Frontiers in Virtual Reality **5**, 1406923 (2024)
26. Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marçal, A., Guedes, C., Cardoso, J.: Optical music recognition: State-of-the-art and open issues. International Journal of Multimedia Information Retrieval **1** (10 2012). https://doi.org/10.1007/s13735-012-0004-6
27. Shatri, E., Fazekas, G.: Knowledge discovery in optical music recognition: Enhancing information retrieval with instance segmentation. arXiv preprint arXiv:2408.15002 (2024)
28. Tan, S.L., Wakefield, E., Jeffries, P.: Musically untrained college students- interpretations of musical notation: Sound, silence, loudness, duration, and temporal order. Psychology of Music - PSYCHOL MUSIC **37**, 5–24 (08 2008). https://doi.org/10.1177/0305735608090845
29. Tuggener, L., Elezi, I., Schmidhuber, J., Stadelmann, T.: Deep watershed detector for music object recognition. arXiv preprint arXiv:1805.10548 (2018)
30. Tuggener, L., Satyawan, Y.P., Pacha, A., Schmidhuber, J., Stadelmann, T.: Deepscoresv2. Tech. rep., Zenodo (2020)
31. Vo, M., Shin, A.: Handwritten music symbol recognition using convolutional neural networks. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 419–424 (2016). https://doi.org/10.1109/ICFHR.2016.0091
32. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1905–1914 (2021)
33. Wu, F.H.F.: An evaluation framework of optical music recognition in numbered music notation. In: 2016 IEEE International Symposium on Multimedia (ISM). pp. 626–631 (2016). https://doi.org/10.1109/ISM.2016.0134
34. Yoyo, Liebhardt, C., Samuel, S.: Breezewhite/oemer: v0.1.7 (Oct 2023)