applied sciences

MDPI

Article

Design of a Semantic Understanding System for Optical Staff Symbols

Fengbin Lou, Yaling Lu * and Guangyu Wang

School of Electrical and Electronic Engineering, Wuhan Polytechnic University, Wuhan 430023, China; lou_fengbin@163.com (F.L.); hnsqwgy@163.com (G.W.)
* Correspondence: luyl@whpu.edu.cn

Abstract: Symbolic semantic understanding of staff images is an important technological support to achieve "intelligent score flipping". Due to the complex composition of staff symbols and the strong semantic correlation between symbol spaces, it is difficult to understand the pitch and duration of each note when the staff is performed. In this paper, we design a semantic understanding system for optical staff symbols. The system uses the YOLOv5 to implement the optical staff's low-level semantic understanding stage, which understands the pitch and duration in natural scales and other symbols that affect the pitch and duration. The proposed note encoding reconstruction algorithm is used to implement the high-level semantic understanding stage. Such an algorithm understands the logical, spatial, and temporal relationships between natural scales and other symbols based on music theory and outputs digital codes for the pitch and duration of the main notes during performances. The model is trained with a self-constructed SUSN dataset. Experimental results with YOLOv5 show that the precision is 0.989 and that the recall is 0.972. The system's error rate is 0.031, and the omission rate is 0.021. The paper concludes by analyzing the causes of semantic understanding errors and offers recommendations for further research. The results of this paper provide a method for multimodal music artificial intelligence applications such as notation recognition through listening, intelligent score flipping, and automatic performance.

Keywords: semantic understanding; neural networks; optical music recognition; YOLOv5; digital code

check for updates

Citation: Lou, F.; Lu, Y.; Wang, G. Design of a Semantic Understanding System for Optical Staff Symbols. *Appl. Sci.* **2023**, *13*, 12627. https://doi.org/10.3390/app132312627

Academic Editors: Lorenzo J. Tardón, Isabel Barbancho and Dimitris Mourtzis

Received: 5 July 2023
Revised: 17 November 2023
Accepted: 20 November 2023
Published: 23 November 2023

1. Introduction

Our project seeks to develop an "intelligent score flipping" that automatically turns sheet music, allowing performers to focus on their performance, teaching, and practice without the need to use their hands to flip pages. Currently, using staff paper is one of the primary methods musicians use to annotate music. Therefore, our research in this paper seeks to enable the device to recognize each symbol on the staff and convert the image of the staff into the corresponding pitch and duration for each note that should be played during a performance. These values are then compared with the actual pitch and duration of the notes being played to determine whether the device should trigger the page flip. The findings of this research have significant implications for various fields, such as notation recognition through listening [1,2], intelligent score flipping, music information retrieval [3,4], and more.

It is difficult to understand the pitch and duration of each note in the staff because of the complex composition of the symbols, the strong semantic correlation between the symbols, and the complexity and cohesiveness of the notes. Semantic understanding of the optical staff is closely related to optical music recognition (OMR) [5–7]. This area has been an important application in machine learning since the middle of the last century. The extent to which optical music recognition can be achieved has varied according to technology development and different needs. In the low period of deep learning, OMR

went from separating and extracting symbolic primitives (lines, heads, stems, tails, beams, etc.) to using correlations between primitives and related rules of musical notation and recognizing notes [8,9]. With gradual improvements in the deep learning ecological system, various types of research based on deep learning have provided new ideas for OMR and put forward new recognition requirements.

This paper aims to achieve codes for pitch and duration of the notes in a complex staff image during a performance, so an end-to-end optical staff semantic understanding system is designed. The system consists of YOLOv5 as the Low-Level Semantic Understanding Stage (LSUS) and the Note-Encoding Reconstruction Algorithm (NERA) as the High-Level Semantic Understanding Stage (HSUS). In the LSUS, the whole optical staff is the input of the system. The model is then trained with the self-constructed SUSN dataset to output digital codes for the pitch and duration of the main note under the natural scales as well as for other symbols that affect the pitch and duration of the main note. The NERA, which takes the output of the LSUS of the staff as the input and applies music theory and MIDI encoding rules [10], resolves the natural scale and other symbol semantics as well as their mutual logical, spatial, and temporal relationships, which results in the output of the staff symbol relationship structure of the given symbols, realizes the HSUS of the main notes through calculation, outputs the pitch-duration codes of the main notes during the performance, and provides an end-to-end optical staff symbol semantic understanding encoding for notation recognition through listening, intelligent score flipping, and music information retrieval. The dataset, code, pre-trained models, and experimental results covered in this paper are open source (https://github.com/Luyledu/Semantic-Understanding-System-for-Optical-Staff-Symbols, accessed on 18 November 2023).

The innovations of this paper mainly include the following:

- The LSUS pre-trained model has the ability to recognize the pitch and duration of symbols and notes in staff images, even when they exhibit varying levels of complexity. The test results demonstrate precision and recall rates of 0.989 and 0.972, respectively, fulfilling the requirements for "intelligent score flipping" with high accuracy.
- According to the specific application of "intelligent score flipping", a comprehensive end-to-end system model has been developed and implemented. This model converts graphical staff symbols into performance coding.
- A note-encoding reconstruction algorithm has been developed, which establishes the relationship between individual symbols based on the notation method. This algorithm outputs the pitch and duration of each note during performance.
- The SUSN dataset has been created. This dataset innovatively includes relative positional information of symbols in the staff without increasing the length of the label field. The dataset is suitable for end-to-end-type algorithm models.

2. Related Work

This section describes key references to optical music recognition using deep learning and related datasets relevant to the present work.

2.1. Optical Music Recognition

With the rapid development of computer vision, research into OMR based on deep learning has brought new breakthroughs and improvements to traditional music score recognition and analysis methods. Object detection is an important problem in the field of computer vision. The YOLO [11–14] algorithm family adopts a one-stage detector structure, which combines classification and localization tasks into a regression problem, using a neural network model to directly predict the class and bounding box of each object. It is known for its speed, accuracy, and lightweight design, making it one of the state-of-the-art object detection algorithms. Compared to other object detection algorithms, such as the Fast R-CNN family [15,16] and SSD [17], YOLOv5 [18,19] has faster detection speed, higher precision, and performs well with small objects. Moreover, YOLOv5 is implemented

using the PyTorch framework, which provides advantages such as ease of use, extensive community support, and seamless integration with other deep learning tools. Additionally, YOLOv5 supports efficient deployment on various hardware platforms, making it an ideal choice for practical applications. Considering these advantages, YOLOv5 is well-suited for application in the field of OMR.

Pacha et al. [20] proposed a region-based convolutional neural network for staff symbol detection tasks and used the Faster R-CNN neural network model to locate and classify single-line staff symbols. Both semantic segmentation methods for staff symbols, which include the U-Net [21] neural network model applied by Hajič, Jr. et al. [22] and the deep-water detector algorithm proposed by Tuggener et al. [23], fail to detect pitch and duration. Huang et al. [24] proposed an end-to-end network model for staff symbol recognition by modifying YOLOv3 to detect pitch and duration separately. OMR algorithms based on sequence modeling mainly target monophonic music sheets and cannot completely understand the meaning of all symbols; e.g., Van der Wel et al. [25] used a sequence-to-sequence [26] model, while Baró et al. [27] used a convolutional recurrent neural network consisting of CNN and LSTM [28].

2.2. OMR Dataset

In the past, several OMR datasets have been published that address one or more of the following problems.

DeepScore [29]: DeepScore is a large-scale, comprehensive dataset for music symbol recognition. This dataset collects staff notation images from the classical music domain and manually annotates the music symbols contained within them, including symbol category, position, and size information. The symbols in the dataset include common music symbols such as notes, rests, clefs, and key signatures. For note symbols, the dataset adopts an annotated symbol primitive approach, which requires reassembling the primitives during application. This undoubtedly increases the difficulty of post-processing and prevents end-to-end recognition of the pitch and duration of the notes.

MUSCIMA++ [30]: This dataset contains thousands of handwritten music symbol images and their corresponding annotations. The real annotations are defined as a symbol graph; in addition to individual symbols, the relationships between them are annotated to infer semantics such as pitch, duration, and start time. It is possible to train a complete OMR pipeline on this dataset, but recognizing the pitch and duration of the notes still poses significant challenges.

HOMUS [31]: HOMUS is a large dataset designed for music symbol classification. It contains 15,000 isolated samples of music symbols, each with recorded individual strokes used to draw them. This unique feature allows for online symbol classification to be performed.

2.3. Summary

In summary, thus far, deep learning algorithms are able to detect and recognize the locations and classes of some symbols in staff images with low complexities (i.e., low symbol density, small span, and few varieties), achieving partial semantic understanding. Therefore, this paper uses YOLOv5 as an LSUS to recognize the pitch and duration of symbols and notes in staff images of different complexity. Since YOLO is based on an end-to-end object detection algorithm, the datasets that are applied to YOLO should be those with the detected target as the object, while datasets that do not consider the relationship between symbols' spatial locations in the staff cannot be applied to this algorithm. In this paper, the SUSN dataset is constructed by taking the musical note and control symbol as the recognition object, fully considering the spatial position of the note in the staff, and providing accurate information on pitch and duration.

3. Materials and Methods

3.1. Dataset

In this paper, the overall goal of the Semantic Understanding of Staff Notation (SUSN) dataset is to encode the pitch and duration of the main notes during the performance. (https://github.com/Luyledu/Semantic-Understanding-System-for-Optical-Staff-Symbols/tree/master, accessed on 18 November 2023; The staff images in the dataset are the open-license staffs provided by the International Music Score Library Project (IMSLP). No copyright issues are involved.) In addition to single notes, there are numerous other forms of notes in the score, such as appoggiaturas, overtones, and harmonies. Aurally, appoggiaturas (shown in Figure 1h) and overtones increase the richness of musical frequencies but do not change the fundamental frequency of the main melody; generally, all the notes in harmony except the first one are weak-sounding. Therefore, we define single notes and the first note of the harmony in the score as the main note. When labeling the dataset, the notes are only labeled with the category of the main note and its related information. In this context, the annotated information and the method of labeling used for this dataset are as follows:

- The main notes are labeled with information about the note position, pitch, and duration in the natural scale. The labeling method has two steps. Firstly, draw the bounding box: the bounding box should contain the complete note (head, stem, and tail) and the specific spatial information of the head. In other words, the bounding box is supposed to contain the 0th line to the 5th line of the staff as well as the position of the head. Then, annotate the object: the format of the label is the 'duration_pitch' code under the natural scale (as shown in Figure 1f,g).
- Label the categories of symbols that affect the pitch and duration of the main notes as well as position information. In the score, the clef, key signature, dot, and pitch-shifting notation (sharp, flat, and natural) are the main control symbols that affect the pitch and duration of the main note, and Table 1a–c lists the control symbols identified and understood in this paper. Each of these kinds of symbols is labeled with a minimum external bounding box containing the whole symbol and category information, as shown in Figure 1a–c,e.
- Label the categories and position of the symbols of the rest. The rest is used in a score to express stopping performance for a specified duration. The symbol of the rest is labeled with a minimum external bounding box that contains the rest entirely as well as information about its category and duration. The rests identified and understood in this paper are listed in Table 1d, while the rests in the staff are labeled as shown in Figure 1d.

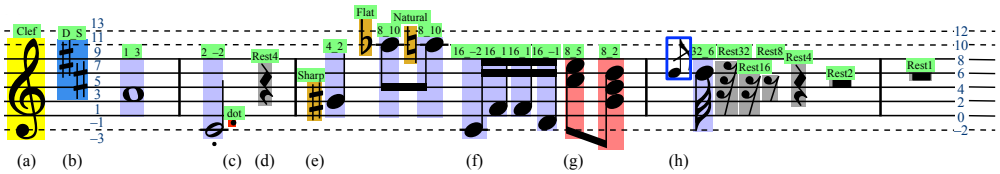
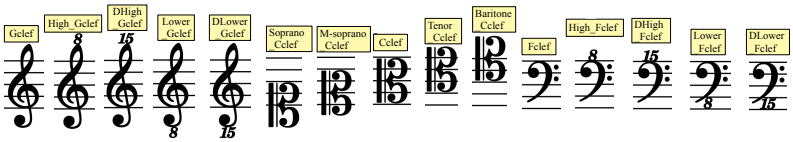
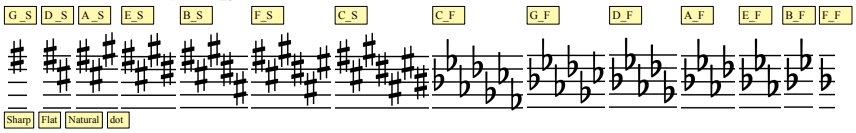
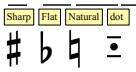
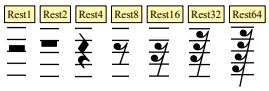


Figure 1. Dataset labeling method. (**a**) Labeling of the treble clef: the yellow minimum external bounding box labeled as 'Gclef'; (**b**) Labeling of D major: the blue minimum external bounding box labeled as 'D_S'; (**c**) Labeling of the dot: the red minimum external bounding box labeled as 'dot'; (**d**) Labeling of the quarter rest: the gray minimum external bounding box labeled as 'Rest4'; (**e**) Labeling of the sharp symbol: the orange minimum external bounding box labeled as 'Sharp'; (**f**) Labeling of the single note C in the main note: purple bounding box from the note head (including the lower plus 1 line) to the 5th line labeled as '16_−1'; (**g**) Harmony in the main note: the rose colored minimum external bounding box, labeling only the first note and annotated as '8_5'; (**h**) Appoggiatura: the appoggiatura's size is smaller than the main note's in the staff.

Table 1. Images with labels of the note control symbols and rests.

Classes	Images/Labels
(**a**) clefs	
(**b**) key signatures	
(**c**) accidentals	
(**d**) rests	

In this paper, the system trained by the SUSN dataset achieves a great result. This proves that it meets the requirements of the system in this paper.

3.2. Low-Level Semantic Understanding Stage

YOLOv5 is used to implement the LSUS of the staff notation. The symbols in the staff images belong to the category of small object graphics in image recognition, and the multi-scale (1 × 1, 2 × 2, 3 × 3) convolutional neural network is used as the backbone network structure for feature extraction. The multi-scale convolutional network uses convolutional kernels of different sizes to obtain different types of features at different scales, thus extracting richer symbolic features to address the small object, multiple poses, and complexity of the staff symbols. The backbone network is composed of the convolutional layers, the C3 modules, and an SPPF module [32]. The C3 module of the backbone network, mainly composed of a convolutional layer and X ResNet blocks, is the main module for learning the residual features of the staff, which divides the feature mapping into two parts: one goes through multiple stacked residual modules and a convolutional layer, while the other goes through one convolutional layer. They were then merged through a cross-stage hierarchy to reduce the computational effort while ensuring accuracy. The SPPF module passes the staff symbol feature map sequentially through three maximum pooling layers, each with a 5 × 5 network structure, which extracts spatial features of different sizes and improves the model's computational speed and robustness to the spatial layout.

The neck network uses a pathway aggregation network [33] for feature fusion. The neck network is composed of the convolutional layers, the upsampling layers, the connection layers, and the C3 modules without the residual structure. The staff features generated by the feature extraction network contain more symbol location information in the bottom features and more symbol semantic information in the top features. The Feature Pyramid Network (FPN) [34] is introduced to communicate symbolic semantic features from top to bottom. Path enhancement is achieved by the bottom-up feature pyramid structure to convey the localized features. The entire feature hierarchy is enhanced by using localized information at the lower levels, which also shortens the information path between the bottom-level and top-level features.

The LSUS implements the mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ of the input staff image $\mathcal{X}$ to the output set $\mathcal{Y}$ of digital codes corresponding to the symbols (https://github.com/Luyledu/Semantic-Understanding-System-for-Optical-Staff-Symbols/tree/main/result/LSUS/*/labels, accessed on 18 November 2023), where $\mathcal{Y} = \{y_1, y_2, \ldots, y_N\}$ denotes all symbols in the staff, and each symbol y_i ($i \in [1, N]$) has positional coordinates and semantic information.

3.3. High-Level Semantic Understanding Stage

The NERA is designed to convey music theory and the method of staff notation in a system model. Using this algorithm, the resulting set $\mathcal{Y}$ is preprocessed to construct a structure of notation relations for the given symbols and, using music theory and MIDI encoding rules, the pitch and duration of each note are parsed to achieve the HSUS of the optical staff notation. The general rules of the music theory targeted by the NERA are as follows:

- The clefs are the symbols used to determine the exact pitch position of a natural scale in the staff. It is recorded at the leftmost end of each staff, and there is also a flag that indicates the mth line in the staff. Meanwhile, it is also the first symbol considered by the NERA when encoding the pitch;
- The key signature located after the clef is the symbol used to mark the ascending or descending pitch of the corresponding notes and is expressed as a value in the NERA. The clefs and key signatures are effective within one line of staff notation;
- In accidentals, the pitch-shifting notation changes the pitch. It raises, lowers, or restores the pitch of the note on which it is applied. The dot extends the original duration of the note by half.

3.3.1. Data Preprocessing

Data preprocessing using the preprocessing part of the NERA for numeric encoding set $\mathcal{Y}$ has the purpose and functions as follows:

- Removal of invalid symbols. The task of this paper is to implement the encoding of the pitch and duration of staff notes during the performance. Among the numerous symbols that affect the pitch and duration of notes are the clefs, the key signatures, the accidentals, and the natural scales, while other symbols are considered invalid symbols within this article. In the preprocessing stage, invalid symbols are removed, and valid symbols are retained. We define the set of valid symbols as $\mathbb{E}$. The relationships among clefs, key signatures, accidentals, natural scales, the valid symbol set, and the dataset are shown in Equation (1):

$$\mathbb{C}, \mathbb{L}, \mathbb{T}, \mathbb{S} \subseteq \mathbb{E} \subseteq \mathcal{Y}, \tag{1}$$

where clef, key signature, accidental, and natural scale are denoted by $\mathbb{C}$=$\{Gclef, Fclef, \ldots, Cclef\}$, $\mathbb{T} = \{0, D\_S, A\_S, \ldots, C\_F,\}$, $\mathbb{L} = \{sharp, flat, natural, dot\}$, and set $\mathbb{S} \in [P, Du]$, respectively. Specifically, P is the space spanned by the natural scale (C, D, E, F, G, A, B), and Du is spanned by the duration $(1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64)$. Further, the element 0 in set $\mathbb{T}$ means there is no key signature and implies that the signature in this line is C major. Each natural scale $s \in \mathbb{S}$ has two pieces of information that indicate the pitch and duration, respectively. Table 2 shows the relationship between categories and sets.

Table 2. The relationship between categories and sets.

Sets			Categories
$\mathcal{Y}$	$\mathbb{E}$	$\mathbb{C}$	Gclef, High_Gclef, DHigh_Gclef, Lower_Gclef, DLower_Gclef, Soprano_Cclef, M-soprano_Cclef, Cclef, Tenor_Cclef, Baritone_Cclef. Fclef, High_Fclef, DHigh_Fclef, Lower_Fclef, DLower_Fclef
		$\mathbb{T}$	0, G_S, D_S, A_S, E_S, B_S, F_S, C_S, C_F, G_F, D_F, A_F, E_F, B_F, F_F
		$\mathbb{L}$	Sharp, Flat, Natural, Dot
		$\mathbb{S}$	**P** C, D, E, F, G, A, B
			Du $1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/6$
	other		Rest1, Rest2, Rest4, Rest8, Rest16, Rest32, Rest64

- Sorting of valid symbols. The YOLOv5 algorithm in the LSUS outputs the objects, and each object y_i is unordered with the information (cls, X, Y, W, H), where $'cls'$ denotes the symbol's class; X, Y denote the Cartesian coordinate values of the center point of the object bounding box; and W, H denote the width and height. The clef is the first element of each row in the staff. Let its center point coordinate be (X_C, Y_C). Denote $\Delta = D/2$, where D is the distance between two adjacent clefs' center points. If the symbol y_i is $Y \in [Y_C - \Delta, Y_C + \Delta]$, then it goes to the same line. Next, the symbols in the same row are sorted in order by X from small to large. By this method, all valid symbols are rearranged in a new order, which is the exact order of the symbols when reading the staff.

After the preprocessing, the digital information of the staff with M lines is represented as M vectors, and each vector has J_m elements. The specific implementation of the preprocessing part is shown in Algorithm 1 (https://github.com/Luyledu/Semantic-Understanding-System-for-Optical-Staff-Symbols/tree/main/result/LSUS/*/csv, accessed on 18 November 2023).

Algorithm 1 Algorithm for the NERA Preprocessing Part.

Input: The output of the LSUS.
Output: The staff digital information in the right order. //With M vectors and J_m elements in each vector.

```
Initialize: N ← len(𝒴); i ← 0; m ← 0; //f : 𝒳 ↦ 𝒴
while (i ≤ N) do
    i ← i + 1
    if (y_i ∉ 𝔼) then
        continue; //To determine whether the current symbol is a valid symbol.
    end if
    if (y_i ∈ ℂ) then
        m ← m + 1;
        j ← 0; //If the input symbol belongs to the clefs, a new vector is created.
    else
        j ← j + 1; //If the valid symbols are not clefs, then continue.
    end if
end while
return Output
```

3.3.2. Note Reconstructing

In the process of constructing the staff symbol relationship structure, the understanding of the semantic information of the symbols and the interrelationship between the symbols are what we should focus on. We define $e(m, j)$ as the semantics of the element j in the mth vector, $e(m, j) \subseteq \mathbb{E}$. As the symbol acts directly or indirectly on the natural scale, it affects the played pitch and duration. As for the entire staff image, we define the global variables $e(m, 0)$ and $e(m, 1)$ for the mth line, where $e(m, 0) \subseteq \mathbb{C}$, $e(m, 1) \subseteq \mathbb{T}$. In addition, we define the local variables v_{1mn} and v_{2mn} that affect the nth note in the mth line. The variable v_{1mn} indicates whether the note is transposed or not and how it should be transposed, i.e., sharp, flat, or natural. The variable v_{2mn} indicates whether the note's duration is extended to 1.5 times.

In this context, when YOLOv5 outputs the symbol class $'cls'$ as a note, the duration and pitch information of the symbol are expressed as (p_{mn}, du_{mn}). Thus, the note information in line m is represented as $[\mathbf{p_m}, \mathbf{du_m}]$, where vectors $\mathbf{p_m} = [p_{m1}, p_{m2}, p_{m3}, \cdots, p_{mNm}]^T$ and $\mathbf{du_m} = [du_{m1}, du_{m2}, du_{m3}, \cdots, du_{mNm}]^T$, and where Nm, the number of notes in each line, varies depending on the line. The control information for pitch and duration is expressed as $[\mathbf{v_{1m}}, \mathbf{v_{2m}}]$, where vector $\mathbf{v_{1m}} = [v_{1m1}, v_{1m2}, v_{1m3}, \cdots, v_{1mNm}]^T$ and vector $\mathbf{v_{2m}} = [v_{2m1}, v_{2m2}, v_{2m3}, \cdots, v_{2mNm}]^T$.

The variables v_{1mn} and v_{2mn} are calculated as shown in Equations (2) and (3):

$$v_{1m(n+1)} = \begin{cases} 0 & e(m,j) \notin \{sharp, flat, natural\} \\ 1 & e(m,j) = sharp \\ -1 & e(m,j) = flat \\ -e(m,1) & e(m,j) = natural \end{cases} \tag{2}$$

$$v_{2mn} = \begin{cases} 0 & e(m,j) \neq dot \\ 1/2 & e(m,j) = dot \end{cases} \tag{3}$$

In Equation (2), $n+1$ is the update of the note index. If $e(m,j)$ is natural, the corresponding note performs the opposite control of the key signature, e.g., F in G major has been raised a semitone, but when there is a natural before an F, v_{1mn} controls the note to perform a descending semitone operation. In Equation (3), if $e(m,j)$ is a dot, the duration of the corresponding note is extended by 1/2 of the original duration; otherwise, it is not extended. The specific implementation of the note reconstructing part is shown in Algorithm 2 (https://github.com/Luyledu/Semantic-Understanding-System-for-Optical-Staff-Symbols/tree/main/result/HSUS, accessed on 18 November 2023).

Algorithm 2 Algorithm for the NERA Notation Reconstructing Part.

Input: Vector data.
Output: Staff notation relationship structure.

Initialize: $M \leftarrow len(\mathbb{C} \in \mathbb{E}); m \leftarrow 0; J_m \leftarrow len(e(m,))$; //Maximum value of the line index according to the clef number; initializes the row index and symbolic index.
while $(m \leq M)$ **do**
  $m \leftarrow m+1; j \leftarrow 0; n \leftarrow 0$; //Initializes index j for symbols and index n for notes
  $e(m,0), e(m,1), j \leftarrow 1$; //Gets the value of the line clef and key signature.
  while $(j \leq J_m - 1)$ **do**
    $j \leftarrow j+1$; //Loop through all valid symbols in the mth line.
    if $(e(m,j) \in \mathbb{L}$ **then**
      $v_{1m(n+1)} \leftarrow value$; //Assign pitch-shifting notation to the v_{1mn} of the next note.
    else
      if $(e(m,j) \in \mathbb{S})$ **then**
        $n \leftarrow n+1; (p_{mn}, du_{mn})$; //If it is a note, then calculate its value of pitch and duration.
        $v_{2mn} \leftarrow 0$; //Assign the note duration.
        $v_{1m(n+1)} \leftarrow 0$; //Assign the pitch of the next note.
      else
        $v_{2mn} \leftarrow value$; //If it is a dot.
      end if
    end if
  end while
end while
return Output

3.3.3. Note Encoding

In note-encoding part, the encoding strategy is as follows:

- Pitch Encoding
  According to the clef, key signature, and MIDI encoding rules, the pitch code p_{mn} of the natural scale is converted to a code that includes the function of clef $e(m,0)$ and key signature $e(m,1)$ in the mth line one by one. We define $f(\cdot)$ as the mapping of this strategy and obtain the converted code $f(p_{mn}, e(m,0), e(m,1))$. The encoding process is shown in Figure 2. Then, the pitch encoding part obtains the pitch code PP_{mn} for

each note played using the MIDI encoding rules after scanning the note control vector $\mathbf{v_{1m}}$, as shown in Equation (4):

$$PP_{mn} = f(p_{mn}, e(m,0), e(m,1)) + v_{1mn}. \tag{4}$$

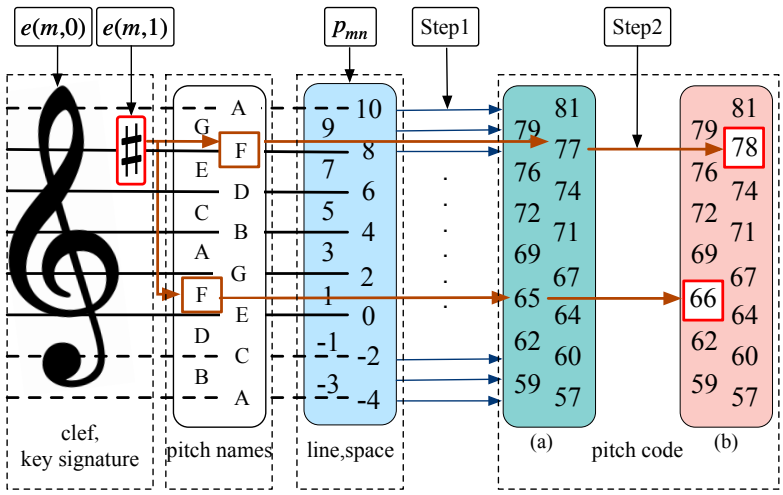


Figure 2. The mapping between the clef, key signature, and the pitch code. In the diagram, the clef $e(m,0)$ is a treble clef. Step1 means the clef's mapping and the MIDI encoding rules. After passing Step1, p_{mn} is converted to (**a**). The key signature $e(m,1)$ is G major. Each note F in the mth line is raised a half tone correspondingly; i.e., the upper F is 78, and the lower is 66. Then, (**a**) is converted to (**b**). The mapping relationship is shown in Step2 in the figure.

- Duration Encoding
  Scan each duration control vector $\mathbf{v_{2m}}$ and corresponding note duration vector $\mathbf{du_m}$, define the individual performance style coefficient as ω, and apply the MIDI encoding rule; then, the duration encoding strategy is shown in Equation (5):

$$PD_{mn} = \frac{1}{du_{mn}} * (1 + v_{2mn}) * \omega \tag{5}$$

  where ω varies according to the different performers and $\omega = 1$ means that the performers' characteristics are not considered.

3.4. System Structure

The structure of the optical staff symbol semantic understanding system is shown in Figure 3. The system consists of two parts: the LSUS (shown in Figure 3b) and the HSUS (shown in Figure 3d). The LSUS is YOLOv5. The model outputs information of the object y_i expressed as (cls, X, Y, W, H) when it is fed a staff image, and the visualization of its results is shown in Figure 3c. The HSUS takes three steps to produce the code for pitch and duration during the performance. To begin with, data preprocessing removes invalid symbols from the disordered $\mathcal{Y}$, then sorts the valid symbols. Additionally, the staff symbol structure vector set is obtained by using note reconstructing. Last but not least, the note encoding part outputs the final results according to the encoding strategies of pitch and duration, as shown in Figure 3e.

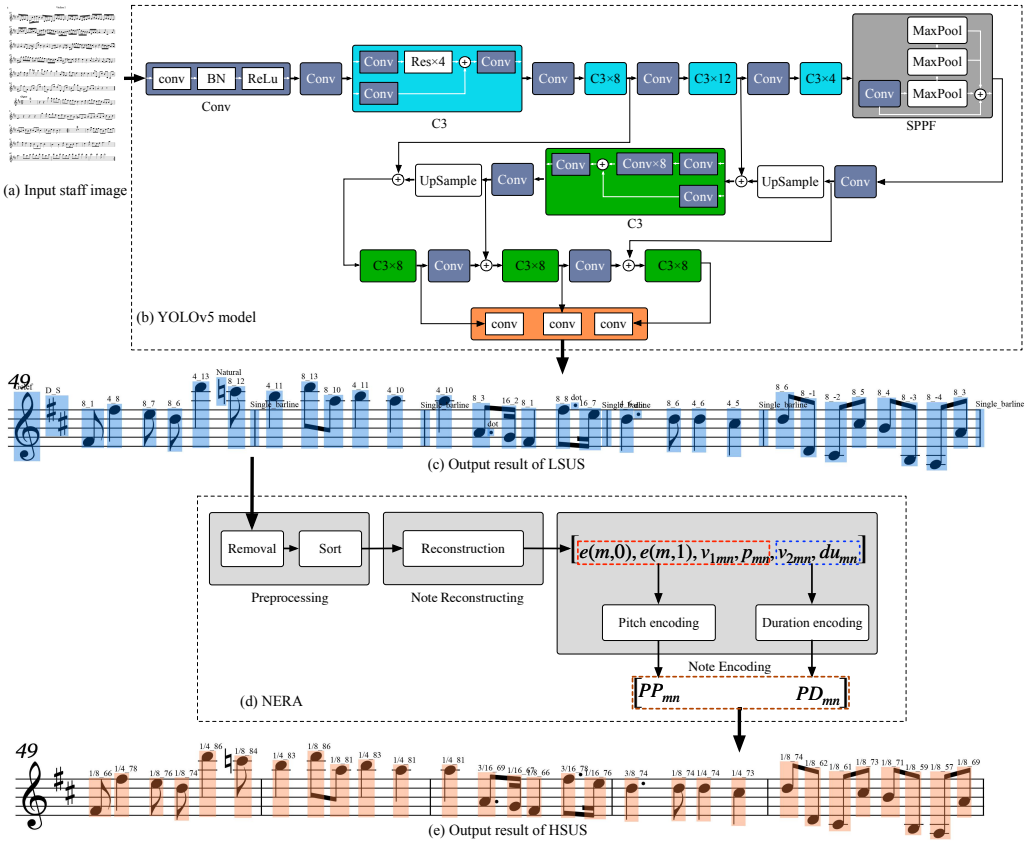


Figure 3. Structure of the optical staff symbol semantic understanding system. The system has two parts: the LSUS (**b**) and the HSUS (**d**). The natural scale (**c**) is the response of the (**b**) module when the optical staff image (**a**) is the system's excitation; (**c**) then acts as the excitation of the module (**d**), which eventually outputs the performance code (**e**).

4. Results

4.1. Data

The dataset used in this article includes a self-built SUSN dataset and an independent test set. The SUSN dataset contains 130,000 labeled samples. In training, a random partitioning strategy is adopted, with 90% of the dataset divided into the training set, and the remaining 10% as the validation set. The test set contains 47 pages of staff images from 10 different tracks with varying complexities (see Appendix A) for a comprehensive evaluation of the system's performance. Among them, the complexities of the staff image of a track are defined by attributes. The key attributes in this paper are the number of symbol types, interval span, symbol density, external note density, and image file size.

4.2. Training

The hardware platform used for training is a workstation with an AMD 32-Core Processor CPU with 80 GB of memory and an NVIDIA RTX 3090 graphics card. The system model is built on the PyCharm platform using the PyTorch framework. In training, the system adopted the Stochastic Gradient Descent (SGD) algorithm to optimize the loss function. In the experiments, the model was trained in 300 epochs, and the first 3 epochs contained linear warm-up, which linearly increases the learning rate from 0 to the initial learning rate. The Cosine Annealing algorithm was adopted to update the learning rate. After 11 training epochs, we determined the initial learning rate as 0.01 by the maximum mAP.

4.3. Evaluation Metrics

In this paper, precision and recall are used to evaluate the performance of the model and recognition effect. The precision reflects the ability of the model to accurately classify symbols:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

The recall shows the ability of the model to recognize symbols:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

where TP is the number of symbols whose categories are correctly identified, FP is the number of incorrectly identified symbols, and FN is the number of symbols that are not identified.

4.4. Experiment and Analysis

4.4.1. Experiment with LSUS

In this paper, the type of symbols, the span of the interval, the density of symbols, the density of external notes, and the file size of staff images are defined as staff complexity variables. Table 3 presents statistics for the complexity properties of the test set of 10 track staves and calculates the precision and recall after LSUS.

Table 3. Performance evaluation of LSUS with different complexity quintiles.

Staff		Complexity Variables					Evaluation	
Name	Page	Type	Span	Density (Symbols)	Density (External Notes)	File Size (kb)	Precision	Recall
Staff 1	2	16	19	484	78	1741	0.968	0.930
Staff 2	5	17	19	679	146	2232	0.996	0.988
Staff 3	3	13	19	319	95	1673	0.997	0.992
Staff 4	12	20	20	478	80	1741	0.994	0.981
Staff 5	7	19	24	530	145	200	0.980	0.958
Staff 6	5	19	20	367	63	435	0.992	0.970
Staff 7	5	15	19	350	62	854	0.996	0.993
Staff 8	3	13	20	441	40	1536	0.990	0.969
Staff 9	3	11	20	424	160	2389	0.986	0.966
Staff 10	2	17	18	315	86	1780	0.987	0.976

The following analysis is from three perspectives:

(1) Evaluation metrics

The average precision of the test set is 0.989, and the recall is 0.972. It is verified that the model has good generalization and robustness to the LSUS of staffs with different complexity. The recall of the model is lower than the precision for all staff images, which shows that the model misses a lot of symbols, especially the external note. For the semantic understanding of line notation, both missed and wrong checks affect the pitch and duration of the corresponding notes, especially the clef and key signatures that determine the pitch and duration of all the notes in a line. Table 4 shows the precision and recall of all the clefs and key signatures in the test set.

Table 4. Precision and recall of clef and key signatures.

	Precision	Recall
clef	1.0 00	0.993
key signature	0.992	0.990

(2) Complexity variables

By analyzing the complexity and evaluation in Table 3, we found that, in the test set of this paper, the main reasons for the error and omission of symbols in the LSUS process are as follows:

- In Staff 1, many complex note beams along with the high density of symbols result in relatively high rates of error and omission, as shown in Figure 4a;
- Staff 2 has the highest density of symbols, and its recall is relatively low, as shown in Figure 4b;
- Staff 3 has a lower complexity for each item, and its performance evaluation is better;
- The error and omission of notes in Staff 4 are mostly concentrated in the notes with longer note stems, as shown in Figure 4c;
- Staff 5 has a higher complexity for each item and very low image file size (200 kb), and its evaluation is worse than others;
- Staff 6 has a lower image file size (435 kb) and, similar to Staff 1, its notes with common note beams are tedious, as shown in Figure 4d;
- Staff 7 has a lower image file size (835 kb), but its performance evaluation is better due to the lower complexities of other attributes;
- In Staff 9, the error detection notes are those located in the higher positive line on the staff, as shown in Figure 4e.

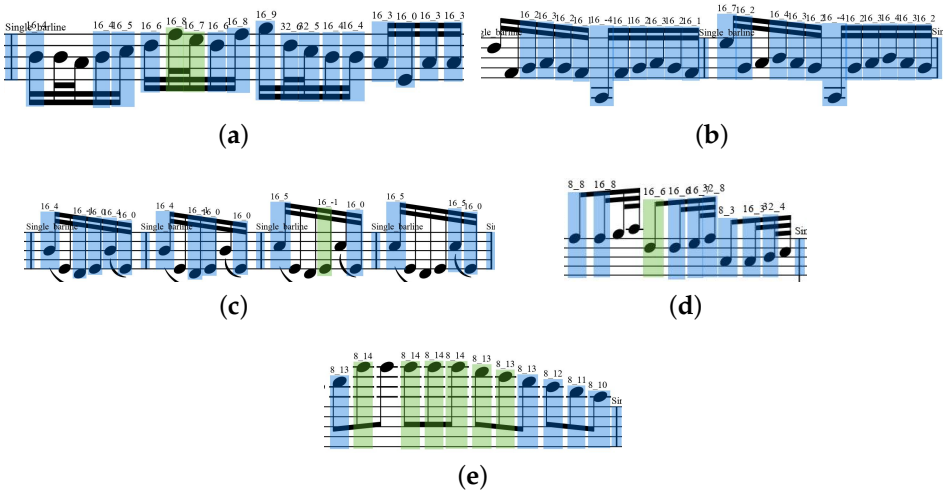


Figure 4. The partial error causes of LSUS. The blue boxes are the correctly identified symbols; the green boxes are the incorrectly identified symbols; the characters in the boxes are the identification results; and the symbols without boxes are the missed notes.

(3) Correlation analysis

The different complexity of staffs is a factor that affects the accuracy of symbol recognition. Using the Pearson correlation coefficient to calculate the correlation between the precision and recall of each of the complexity variables can eliminate the magnitude of the complexity variables and provide a more direct observation of the correlation between performance evaluation and complexity. The Pearson correlation coefficient is calculated as shown in Equation (8):

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma(X) * \sigma(Y)} \tag{8}$$

where, by calculating covariance $Cov(X,Y)$, the strength of linear correlation between complexity variable X and the precision and recall Y is obtained. By calculating the standard deviation $\sigma(X)$ of each complexity variable, as well as the standard deviation $\sigma(Y)$ of recall and precision, we ensure that the calculation of correlation coefficients is not affected by the scale of each variable. The computed complexity variable correlation coefficients are shown in Figure 5.

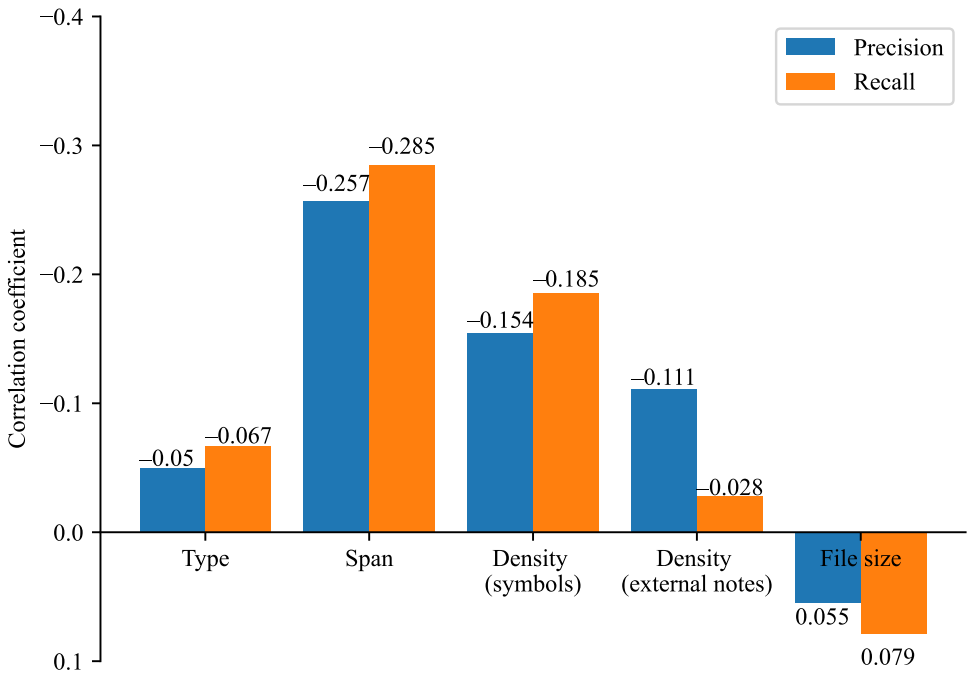


Figure 5. Complexity variable correlation coefficient.

As seen in Figure 5, the type of symbol, the span of interval, the density of symbols, and the density of external notes have a negative correlation with the performance evaluation of the recognition model, and the file size of images has a positive correlation with the model's ability. Take the example of Staff 3 and Staff 10. With a high similarity in other complexity variables, Staff 10 has more types of symbols, and its performance evaluation is lower than Staff 3. Staff 6 has a higher image file size, resulting in higher performance evaluation compared to Staff 4.

A visualization of the output results from the LSUS is shown in Figure 6.

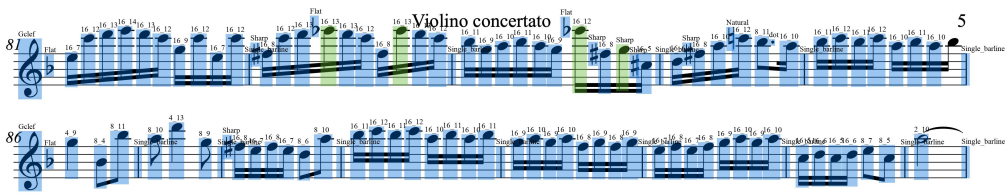


Figure 6. For LSUS visualization, the diagram shows the staff of *Oboe String Quartet in C Minor, Violin Concerto* (JS BACH BWV 1060), page 5, lines 1 and 2. The characters in each box indicate the semantics of the corresponding symbol. In the diagram, green boxes indicate incorrectly identified symbols, and the symbols without boxes are the missed notes.

4.4.2. Experiment with HSUS

The accuracy of the HSUS is related to the accuracy of the output of the LSUS and the stability and accuracy of the NERA. To verify the accuracy of the NERA, using the ideal data (manual annotation) and the practical data (the output of the LSUS) as inputs—named ideal input and practical input, respectively—the error rate and the omission rate of the output results are calculated as follows:

- When the input is ideal, the error rate and the omission rate of the output result are the performance indexes of the NERA;
- The error and omission rates are the performance indexes of the whole system when the output is practical.

Tests were conducted on the test set, and the experimental results are shown in Table 5. The specific analysis is as follows:

Table 5. Experimental results for HSUS.

Staff	Ideal Input		Practical Input	
	Error Rate	Omission Rate	Error Rate	Omission Rate
Staff 1	0.006	0.000	0.052	0.044
Staff 2	0.011	0.000	0.016	0.010
Staff 3	0.010	0.000	0.020	0.006
Staff 4	0.019	0.000	0.027	0.020
Staff 5	0.013	0.000	0.044	0.014
Staff 6	0.005	0.000	0.020	0.008
Staff 7	0.000	0.000	0.004	0.010
Staff 8	0.020	0.000	0.055	0.053
Staff 9	0.022	0.000	0.037	0.021
Staff 10	0.000	0.000	0.036	0.019

(1) HSUS output error

As shown in Table 4, the ideal data as the input of HSUS have a zero omission rate, which indicates that the NERA has performed the HSUS for each note input, thus proving its stability. The error is mainly caused by the deviation of the range of accidentals. The accidentals are defined in music theory to work for the notes with the same height within a bar; however, in this paper, the note reconstruction does not extend the effective range of accidentals to other symbols within the bar, which leads to the error in the HSUS, as shown in Figure 7.

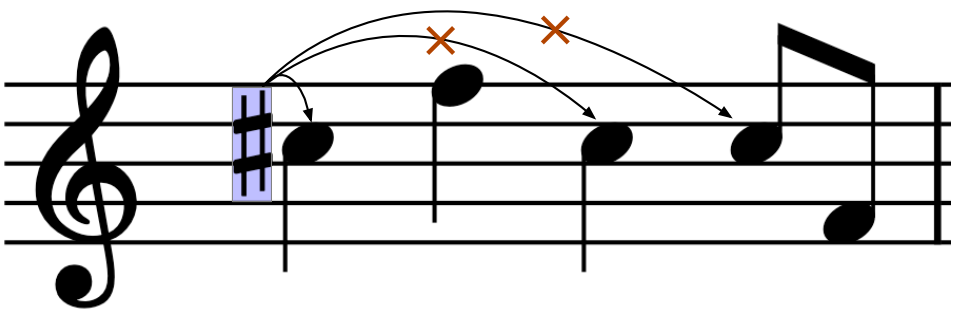


Figure 7. The pitch-shifting notation leads to HSUS errors. The sharp should be applied to notes of the same height in the bar, but the NERA only applies it to the first note after the sharp.

(2) Scope of application of NERA

The NERA proposed in this paper can only be applied to the general rules of staff notation whose expression is notation, i.e., the rules described in Section 2.3. Additionally, due to the ambiguous restriction that the author may want to express the content using symbols, there will be some special rules of note expressions [5], and then the output of the HSUS will be very different from what the author expresses, as in the example shown in Figure 8.

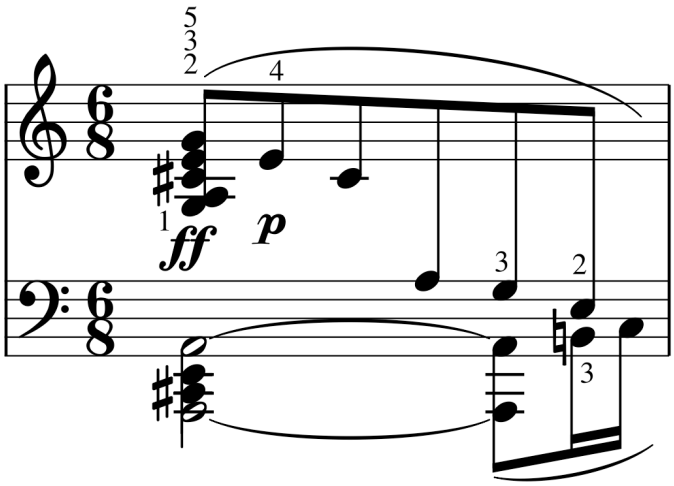


Figure 8. This excerpt from Beethoven's Piano Sonata illustrates some of the characteristics that distinguish musical notation from ordinary notation. The chord in the lower left contains a mixture of half- and quarter-notes of the mixed note head, yet the musical intent is that the two quarter notes in the middle of the chord are actually played as eighth notes, adding to the thickness of the first beat. (Excerpted from *Understanding Optical Music Recognition* by Calvo-Zaragoza et al. [5]).

(3) LSUS as input

The average error rate of the HSUS is 0.031, and the omission rate is 0.021 when the input is the system's practical data. Among numerous replicate experiments, we found that, despite the high overall accuracy of the system output, some errors with very low probability still occur. After analysis, we found that these errors are caused by LSUS errors, as shown in Figure 9, mainly as follows:

- Misidentification of the pitch and duration of natural scales can lead to errors during HSUS;
- Misidentification or omission of accidentals (sharp, flat, natural, dot) acting on natural scales can lead to errors during HSUS;
- Omission of a note affects the HSUS of the note or the preceding and following notes. There are three cases: (1) when the note is preceded and followed by separate notes, the omission of the note does not affect the semantics of the preceding and following notes; (2) when a note is preceded by a pitch-shifting notation (sharp, flat, natural) and followed by another note, the omission of the note will cause the pitch-shifting notation originally used for the note to be applied to the latter note, resulting in a pitch error at the HSUS of understanding of the latter note; (3) when the note is preceded by a note and followed by a dot, the omission of the note will cause the appendage originally used for the note to act on the preceding note, and thus the HSUS of the preceding note will be incorrectly timed;
- Misidentification or omission of the key signature will result in a pitch error in the HSUS for some notes in this line. There are three cases: (1) when the key signature is missed, the pitch of the note in the key signature range is incorrect at the HSUS; (2) when the key signature is misidentified as a key with the same mode of action, i.e., when both modes of action are the same, making the natural scale ascending (or descending) but with a different range of action, the HSUS of some of the notes will be wrong in terms of pitch; (3) when the key signature is incorrectly identified as a key with a different mode of action, the pitch will be incorrect when the note is semantically understood;
- When the clef is missed, all natural scales in this row are affected by the clef of the previous line. When the clef is incorrectly identified, an error occurs at the HSUS of all natural scales in this row.

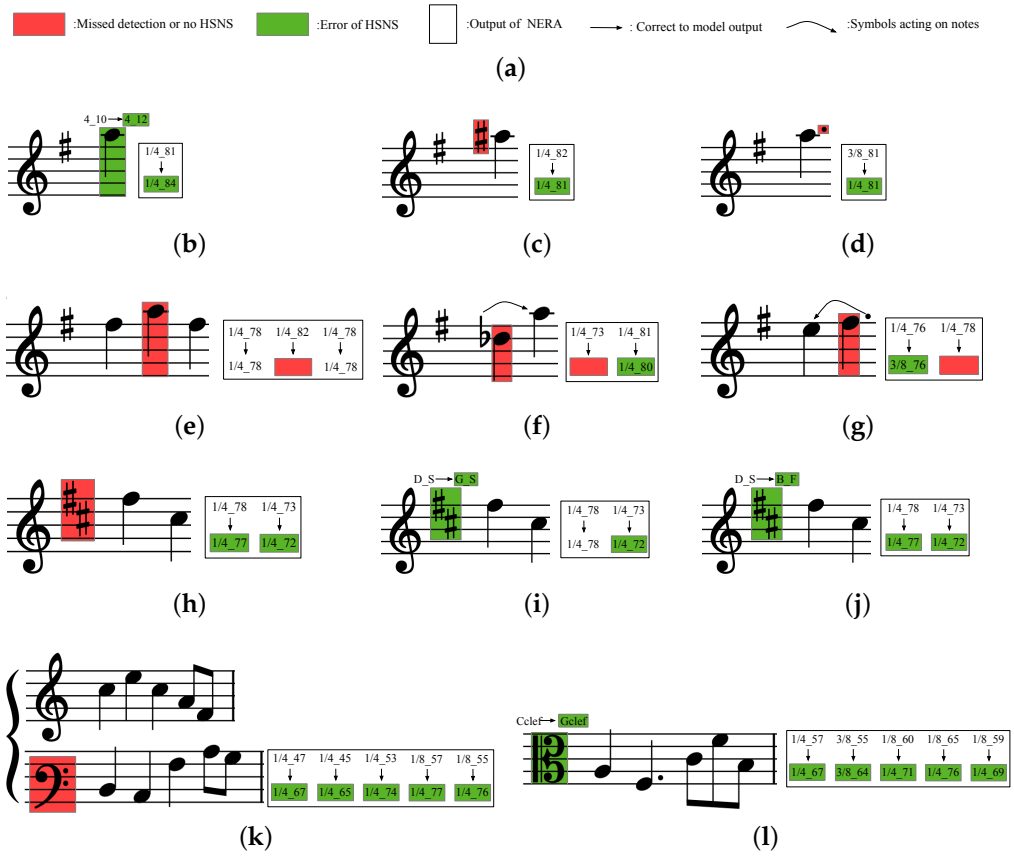


Figure 9. Impact on HSUS in case of symbolic errors or omissions at LSUS. (**a**) indicates the meaning of the corresponding symbol in the figure below. (**b**) has a wrong note pitch identification of 12, so the HSUS has a pitch error. (**c**) missed a sharp, so the note pitch of the action is wrong. (**d**) omitted the dot, so the note duration of the action is wrong. (**e**) has a note omission that does not affect the semantics of the preceding and following notes. (**f**), however, has an omission of a note, which causes the flat to act on the pitch of the next note, and the pitch of the next note is incorrect. (**g**) has an omission of a note, which causes the dot to act on the duration of the preceding note, and the duration of the preceding note is incorrect. (**h**) has missed the key signature of D major, and the notes in the natural scale roll call of "Do" and "Fa" will not be raised. (**i**) has incorrectly identified D major as G major, and the pitches of the notes in the range of action are raised (D major acts on natural scales with the roll call of "Fa" and "Do", while G major acts on natural scales with the roll call of "Fa"); when performing the HSUS, natural scales with a roll call of "Fa" in this line of the staff are not subject to error, while natural scales with a roll call of "Do" are subject to error. (**j**) has recognized D major as F major, and the mode of action is different, which causes the pitch of all the notes in the range to be incorrect. (**k**) missed the bass clef, and the pitch of all the notes in this line of the staff is determined by the clef of the previous line. (**l**) identified the alto clef incorrectly as the treble clef, and all the pitches in this line of the pitch are incorrect.

The pitch and duration codes of the played notes output by the staff notation semantic understanding system after the HSUS are shown in Figure 10.

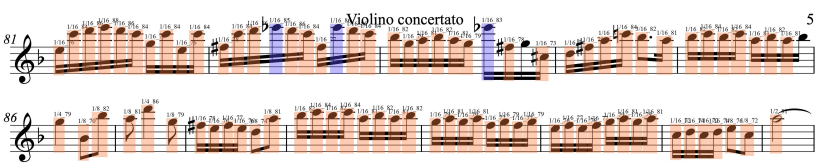


Figure 10. Visualization of the results of HSUS.

5. Conclusions and Outlooks

5.1. Conclusions

This paper aims to solve the problem of semantic understanding of the main notes of the optical staff as pitch and duration during performances in the field of music information retrieval. The SUSN dataset is constructed using the basic properties of the staff as a starting point, and the YOLO object detection algorithm is used to achieve LSUS of the pitch and duration of the natural scale and other symbols (such as clefs, key signatures, accidentals, etc.) that affect the natural scale. Experimental results of the LSUS show that the precision is 0.989 and the recall is 0.972. We analyze the causes of error and omissions in the LSUS due to the differences in the complexity of the staff.

The HSUS is based on the NERA proposed by music theory, which parses the low-level semantic information according to the semantics of each symbol and its logical, spatial, and temporal relationships with each other, constructs the staff symbol relationship structure of the given symbols, and calculates the pitch and duration of each note played. The NERA has limitations in modeling the staff image system and can only realize the encoding of the pitch and duration of the notes whose staff symbols are defined according to a version of the rules of notation. The accuracy of notes in the process of HSUS depends on the accuracy of LSUS, and once there are symbol errors and omissions, it will lead to incorrect pitch and duration encoding of the corresponding notes in the process of HSUS. In this paper, we summarize the different cases of HSUS errors caused by symbol errors and the omission of different symbols of the staff scale during the LSUS. The optical staff notation semantic understanding system implements the input staff images and outputs the encoding of the pitch and duration of each note when it is played.

5.2. Outlooks

The main problems with LSUS are as follows:

- The staff notation in this paper is mainly related to the pitch and duration of musical melodies. The recognition of other symbols, such as dynamics, staccatos, trills, and characters related to the information of the staff is one of the future tasks to be solved;
- The accurate recognition of complex natural scales such as chords is a priority;
- The recognition of symbols in more complex staff images, e.g., those with larger intervals, denser symbols, and more noise in the image.

For the HSUS, the following problems still need to be solved:

- It is important to improve the scope of accidentals, so that they can be combined with bar lines and repetition lines, etc;
- The semantic understanding of notes is based on the LSUS and, after solving the problem of the types of symbols recognized by the model, each note can be given richer expression information;
- In this paper, rests are recognized, but the information is not utilized in semantic understanding. In the future, this information and the semantic relationships of other symbols can be used to generate a complete code of the staff during performances.

The system provides an accurate semantic understanding of optical staff symbols for multimodal music artificial intelligence applications such as notation recognition through listening, intelligent score flipping, and automatic performance.

Author Contributions: Conceptualization, F.L. and Y.L.; methodology, F.L. and Y.L.; software, F.L.; validation, F.L.; formal analysis, F.L. and Y.L.; data curation, F.L. and G.W.; writing—original draft preparation, F.L.; writing—review and editing, F.L., Y.L. and G.W.; visualization, F.L.; supervision, Y.L.; project administration, F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in https://github.com/Luyledu/Semantic-Understanding-System-for-Optical-Staff-Symbols, accessed on 18 November 2023.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LSUS	Low-Level Semantic Understanding Stage
HSUS	High-Level Semantic Understanding Stage
NERA	Note-Encoding Reconstruction Algorithm

Appendix A

The ten staffs selected for the test set are shown below:

- Staff 1: *Canon and Gigue in D Major* (Pachelbel, Johann)
- Staff 2: *Oboe String Quartet in C Minor, Violin Concerto* (J.S. Bach BWV 1060)
- Staff 3: *Sechs ländlerische Tänze für 2 Violinen und Bass (Woo15), Violino 1* (Beethoven, Ludwig van)
- Staff 4: *Violin Concerto RV 226, Violino principale* (A. Vivaldi)
- Staff 5: *String Duo no. 1 in G for violin and viola KV 423* (Wolfgang Amadeus Mozart)
- Staff 6: *Partia à Cembalo solo* (G. Ph. Telemann)
- Staff 7: *Canon in D, Piano Solo* (Pachelbel, Johann)
- Staff 8: *Für Elise in A Minor WoO 59* (Beethoven, Ludwig van)
- Staff 9: *Passacaglia* (Handel Halvorsen)
- Staff 10: *Prélude n°1 Do Majeur* (J.S. Bach)

References

1. Moysis, L.; Iliadis, L.A.; Sotiroudis, S.P.; Boursianis, A.D.; Papadopoulou, M.S.; Kokkinidis, K.-I.D.; Volos, C.; Sarigiannidis, P.; Nikolaidis, S.; Goudos, S.K. Music Deep Learning: Deep Learning Methods for Music Signal Processing—A Review of the State-of-the-Art. *IEEE Access* **2023**, *11*, 17031–17052. [CrossRef]
2. Tardon, L.J.; Barbancho, I.; Barbancho, A.M.; Peinado, A.; Serafin, S.; Avanzini, F. 16th Sound and Music Computing Conference SMC 2019 (28–31 May 2019, Malaga, Spain). *Appl. Sci.* **2019**, *9*, 2492 . [CrossRef]
3. Downie, J.S. Music information retrieval. *Annu. Rev. Inf. Sci. Technol.* **2003**, *37*, 295–340. [CrossRef]
4. Casey, M.A.; Veltkamp, R.; Goto, M.; Leman, M.; Rhodes, C.; Slaney, M. Content-Based Music Information Retrieval: Current Directions and Future Challenges. *Proc. IEEE* **2008**, *96*, 668–696. [CrossRef]
5. Calvo-Zaragoza, J.; Hajič, J., Jr.; Pacha, A. Understanding Optical Music Recognition. *ACM Comput. Surv.* **2020**, *53*, 1–35. [CrossRef]
6. Rebelo, A.; Fujinaga, I.; Paszkiewicz, F.; Marcal, A.R.S.; Guedes, C.; Cardoso, J.S. Optical music recognition: State-of-the-art and open issues. *Int. J. Multimed. Inf. Retr.* **2012**, *1*, 173–190. [CrossRef]
7. Calvo-Zaragoza, J.; Barbancho, I.; Tardon, L.J.; Barbancho, A.M. Avoiding staff removal stage in optical music recognition: Application to scores written in white mensural notation. *Pattern Anal. Appl.* **2015**, *18*, 933–943. [CrossRef]
8. Rebelo, A.; Capela, G.; Cardoso, J.S. Optical recognition of music symbols. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2010**, *13*, 19–31. [CrossRef]
9. Baró, A.; Riba, P.; Fornés, A. Towards the Recognition of Compound Music Notes in Handwritten Music Scores. In Proceedings of the 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 23–26 October 2016; pp. 465–470.
10. Huber, D.M. *The MIDI Manual: A Practical Guide to MIDI in the Project Studio*; Taylor & Francis: Abingdon, UK, 2007.
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
12. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
13. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.

15. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015.
17. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single shot MultiBox detector. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
18. Thuan, D. Evolution of Yolo algorithm and Yolov5: The State-of-the-Art Object Detention Algorithm. Ph.D. Thesis, Oulu University of Applied Sciences, Oulu, Finland, 2021.
19. Al-Qubaydhi, N.; Alenezi, A.; Alanazi, T.; Senyor, A.; Alanezi, N.; Alotaibi, B.; Alotaibi, M.; Razaque, A.; Abdelhamid, A.A.; Alotaibi, A. Detection of Unauthorized Unmanned Aerial Vehicles Using YOLOv5 and Transfer Learning. *Electronics* **2022**, *11*, 2669. [CrossRef]
20. Pacha, A.; Choi, K.Y.; Coüasnon, B.; Ricquebourg, Y.; Zanibbi, R.; Eidenberger, H. Handwritten music object detection: Open issues and baseline results. In Proceedings of the 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), Vienna, Austria, 24–27 April 2018; pp. 163–168.
21. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; pp. 234–241.
22. Hajič, J., Jr.; Dorfer, M.; Widmer, G.; Pecina, P. Towards full-pipeline handwritten OMR with musical symbol detection by U-nets. In Proceedings of the 19th International Society for Music Information Retrieval Conference, Paris, France, 23–27 September 2018; pp. 225–232.
23. Tuggener, L.; Elezi, I.; Schmidhuber, J.; Stadelmann, T. Deep Watershed Detector for Music Object Recognition. *arXiv* **2018**, arXiv:1805.10548.
24. Huang, Z.; Jia, X.; Guo, Y. State-of-the-Art Model for Music Object Recognition with Deep Learning. *Appl. Sci.* **2019**, *9*, 2645. [CrossRef]
25. Van der Wel, E.; Ullrich, K. Optical Music Recognition with Convolutional Sequence-to-Sequence Models. *arXiv* **2017**, arXiv:1707.04877.
26. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.
27. Baró, A.; Riba, P.; Calvo-Zaragoza, J.; Fornés, A. From Optical Music Recognition to Handwritten Music Recognition: A baseline. *Pattern Recognit. Lett.* **2019**, *123*, 1–8. [CrossRef]
28. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [CrossRef] [PubMed]
29. Tuggener, L.; Satyawan, Y.P.; Pacha, A.; Schmidhuber, J.; Stadelmann, T. The DeepScoresV2 Dataset and Benchmark for Music Object Detection. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milano, Italy, 10–15 January 2021; pp. 9188–9195.
30. Hajič, J., Jr.; Pecina, P. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; pp. 39–46.
31. Calvo-Zaragoza, J.; Oncina, J. Recognition of Pen-Based Music Notation: The HOMUS Dataset. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 3038–3043.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
33. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
34. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.