applied sciences

MDPI

Article

A Stave-Aware Optical Music Recognition on Monophonic Scores for Camera-Based Scenarios

Yipeng Liu, Ruimin Wu, Yifan Wu, Lijie Luo and Wei Xu *

School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China; liuyipeng@hust.edu.cn (Y.L.); ruiminwu@hust.edu.cn (R.W.); yifanwu@hust.edu.cn (Y.W.); M201972007@hust.edu.cn (L.L.)
* Correspondence: xuwei@hust.edu.cn

Abstract: The recognition of printed music sheets in camera-based realistic scenarios is a novel research branch of optical music recognition (OMR). However, special factors in realistic scenarios, such as uneven lighting distribution and curvature of staff lines, can have adverse effects on OMR models designed for digital music scores. This paper proposes a stave-aware method based on object detection to recognize monophonic printed sheet music in camera-based scenarios. By detecting the positions of staff lines, we improve the accuracy of note pitch effectively. In addition, we present the Camera Printed Music Staves (CPMS) dataset, which consists of labels and images captured by mobile phones under different angles and lighting conditions in realistic scenarios. We compare our method after training on different datasets with a sequence recognition method called CRNN-CTC on the test set of the CPMS dataset. The results show that the accuracy, robustness, and data dependency of our method perform better.

Keywords: optical music recognition; printed monophonic score; realistic scenarios; sheet music photos; stave-aware

Citation: Liu, Y.; Wu, R.; Wu, Y.; Luo, L.; Xu, W. A Stave-Aware Optical Music Recognition on Monophonic Scores for Camera-Based Scenarios. *Appl. Sci.* **2023**, *13*, 9360. https://doi.org/10.3390/app13169360

Academic Editor: Fan Zhang

Received: 5 July 2023
Revised: 13 August 2023
Accepted: 16 August 2023
Published: 17 August 2023

1. Introduction

Optical music recognition (OMR) is a research field investigating how to read musical symbols in documents [1] computationally. In other words, its purpose is to convert music sheet images into a machine-readable format. OMR has a broad impact across various domains, including music transcription, archival preservation, automated music analysis, music education, and so on. It has brought innovation in digitizing music resources and providing computer-assisted instruction.

Over the past several decades, researchers have proposed various OMR methods, which typically focus on recognizing printed [2–6] or handwritten [7–10] music scores. However, most optical printed music recognition methods were designed for digitally printed scores rather than printed score photos in realistic scenarios [11–13]. Score photos have many different features compared to digital scores. For example, variations in lighting due to the distribution of light sources and different camera angles can lead to uneven illumination in the images. Moreover, the curvature of staff lines in the music scores caused by paper curvature during capture or distortion characteristics of the camera lens poses a challenge to accurately discerning the pitch of the recognized notes.

Anyway, there is no open dataset of real music score photos. Specifically, there is only one dataset called Camera-PrIMuS [3] containing images distorted to simulate photos in realistic scenarios. The images were generated from digital scores and lacked features of photos in realistic scenarios such as uneven illumination. It means that the images in Camera-PrIMuS could not completely represent the score photos.

To address these challenges, this paper presents a staff-aware optical music recognition method. It utilizes an object detection model to extract the positions of staff lines, musical notes, and accidental within the music scores. These positional details are then assembled

into sequences of pitch and duration through a well-designed notation assembly module. The result shows that our method has a significant improvement in the accuracy and robustness of recognition. The second contribution of this paper is that we build a monophonic printed music score photos dataset called Camera Printed Music Staves (CPMS). All the photos are captured by different mobile phone cameras in different realistic scenarios.

The rest of the paper is organized as follows: Section 2 reviews related works about OMR. Section 3 introduces the design of our method. Section 4 provides details of our dataset and experiments. Section 5 analyzes our experimental results. In Section 6, we conclude our work.

2. Related Work

Based on various data sources, OMR tasks can be categorized into two primary domains: handwritten music score recognition and printed music score recognition. Handwritten music recognition focuses on extracting and interpreting musical notations from handwritten manuscripts, including complexities such as varying handwriting styles and potential ambiguities. In recent years, there have been some breakthroughs in the research of handwritten music recognition with deep learning methods [8–10,14,15]. In contrast, printed music score recognition involves analyzing and extracting musical symbols and annotations from printed music scores, typically characterized by standardized fonts and precise formatting. This paper focuses on printed music recognition.

Based on the different approaches, printed music score recognition methods can be categorized into two types: multi-stage method and end-to-end method.

As a past mainstream, multi-stage methods [16–21] used to decompose the task of OMR into multiple subproblems. The research of multi-stage methods mainly includes the steps of binarization [17], staff line removal [18,19], note detection, classification [20,21], and notation assembly [18]. Pinto et al. [17] proposed a binarization based on domain knowledge. Szwoch et al. [18] applied horizontal projection to remove staff lines. Chuanzhen Li et al. [21] solved the problem of note head recognition and pitch position by adopting template matching. Szwoch et al. [18] proposed a context-free attributed grammar for notation assembly.

In recent years, more and more scholars have been exploring deep learning methods for OMR. The object detection is used for multi-stage OMR pipeline to detect musical symbols in music sheets [22–24], particularly when dealing with certain musical symbols that are relatively small in size compared to the overall dimensions of the music score [25–27]. The methods based on object detection can be divided into two categories according to the research objectives: the detection of simple musical symbols and the detection of note pitch and note type. The majority of research only focuses on the detection of musical symbols' position rather than the pitch and type of notes. Pacha et al. [22] considered the detection performance of three state-of-the-art networks on the DeepScores dataset, which include Faster R-CNN [28], RetinaNet [29], and U-Net [30]. The results show that the U-Net is able to achieve higher detection accuracy with an average classification accuracy of barely 24.8%. Tuggener et al. [23] proposed the Deep Watershed Detector (DWD). It is an object detection network based on synthetic energy maps and the watershed transform and has the ability to predicate the confidence, position, and type of each musical symbol. In addition, some scholars have also preliminarily studied the detection of note pitch and type. For example, Huang et al. [24] used the YOLOv3 [31] network based on darnet53 to predict the pitch and type of notes in an end-to-end manner and achieved 92% type accuracy and 96% pitch accuracy on the MuseScore dataset.

The end-to-end method [2,4–6,11,32–34] has become the mainstream of OMR for its simplicity of data preprocessing. The vast majority of the end-to-end methods is sequence recognition, which means that it directly converts the stave images into symbol sequences with the pitch and type of each note. Sequence recognition methods come from the study of recognizing sequence-like objects in images. Shi B et al. [35] first proposed a Convolutional Recurrent Neural Network (CRNN), which is able to naturally handle sequences in arbitrary

lengths and achieve an accuracy of 84% for music score recognition on a closed-source dataset. Based on the CRNN, Calvo-Zaragoza et al. [2,3] proposed the CRNN-CTC by improving the network structure of CRNN and optimizing the mapping dictionary of musical symbol sequences, and achieved note type accuracy of 99.2% on printed dataset PrIMuS [3] and 96.6% on synthetically distorted dataset Camera-PrIMuS [11]. Qiong W et al. [4] improved the CRNN-CTC by replacing the CNN with a multi-scale residual CNN and changing the BilSTM unit with BiSRU. It achieved an accuracy of 99.7% on the PrIMuS dataset. Eelco et al. [32] applied sequence-to-sequence to OMR and achieved 81% of note pitch accuracy and 94% of note type accuracy on the author's self-built printed dataset. Ríos-Vila A et al. [33] argued two output encodings and found the split-sequence encoding with the two-dimensional nature of music symbols works better. Edirisooriya et al. [34] focused on decoders for polyphonic OMR and found RNNDecoder achieves the highest accuracy of note on self-built MuseScore polyphonic dataset. Ríos-Vila et al. [36] researched the application of Transformer [37] and Vision Transformer [38] in OMR and proposed CNNT for recognizing piano scores. Li et al. [5] proposed transformer-based TrOMR, which outperformed RNNDecoder [34] on polyphonic music scores. They also built a camera scene dataset that is not open source.

3. Method Details

3.1. Overall Design

Our stave-aware OMR method is the process of extracting metadata, such as the position and type, from the musical symbols in score photos to reassemble the sequence of notes in text format. Specifically, we preprocess the image and then utilize YOLOX-S to detect the position of the musical staff. Then, we employ the FCOS model to detect the positions of the notes and utilize a classification network to classify the duration of notes within their respective segments. Finally, we employ a well-designed notation assembly module to reassemble the notes. The stave-aware OMR method consists of two stages: symbol detection and symbol assembly. The structure is shown in Figure 1.

1. *Symbol detection.* In this stage, we detect symbols, which affect the pitch of notes, in the music sheet such as note heads, clefs, key signatures, and accidentals. Simultaneously, the detection of staff lines and the segmentation of notes are also conducted. The result will be utilized for recognizing the pitch and type of the notes.
2. *Symbol assembly.* This stage includes pitch assembly and duration assembly. In the pitch assembly, information about the type and position of accidentals, and the position of note heads and staff lines are fused. In duration assembly, we combine information about note head duration and note duration to obtain note type. Note head duration refers to the type of note heads through an object detection network, whereas note duration is determined by the stem type and whether it is dotted or not through a classification network.

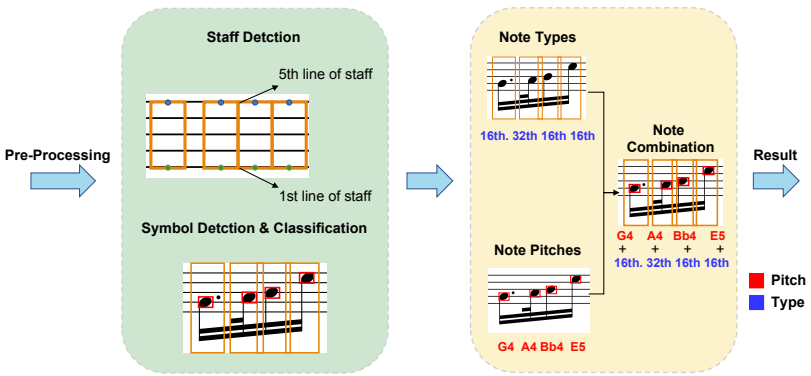


Figure 1. The structure of our two-stage OMR method.

3.2. Preprocessing

For subsequent stave position detection, it is necessary to preprocess original sheet music photos. We use the same preprocessing method as [39]. Specifically, the steps for preprocessing are as follows:

1. *Remove the background lighting.* All input images are converted into grayscale because color is an unnecessary element for OMR. Then, Gaussian blur [40] is applied to obtain blurred images. Finally, subtract the original grayscale image from the blurred image. The purpose of this step is to mitigate the brightness and contrast differences between different regions in the musical score image caused by uneven light distribution.
2. *Resize the image.* Firstly, we divide the image into a fixed number of columns, calculate the median gray scale value of each row within each column, and assign it as the value for that respective row. Next, convolve the modified image with a set of comb filters corresponding to different staff line spacing. We choose the spacing represented by the accumulated response of the most prominent comb filter as the distance between adjacent lines. Finally, we resize the image to ensure the lines' distance is fixed. The purpose of this step is to keep a constant distance between adjacent staff lines of the musical score.
3. *Morphological filtering.* We perform two rounds of morphological filtering on the image shown in Figure 2. The first round removes non-horizontal pixels, and the second round removes thin staff lines. Finally, subtracting the results of the two filtering steps provides the staff lines eliminated during the second round. The purpose of this step is to eliminate the musical notes from the image, preserving only the staff lines for stave detection.

Through preprocessing, we improve the image by fixing uneven lighting and minimizing the impact of musical notes affecting stave detection.

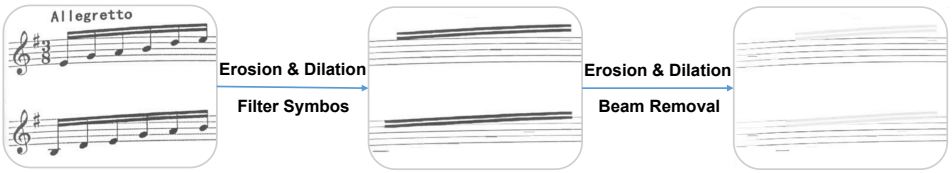


Figure 2. Flow chart of the morphological filter.

3.3. Stave Detection

Different from the general multi-stage OMR methods, our work concerns not only note detection but also staff line detection. As shown in Figure 3, stave features are fed into the stave detection model to obtain the position of staff lines. With the absolute position of the note head, we can estimate its relative position to the staff lines—i.e., which line or space it occupies—by linear interpolation.

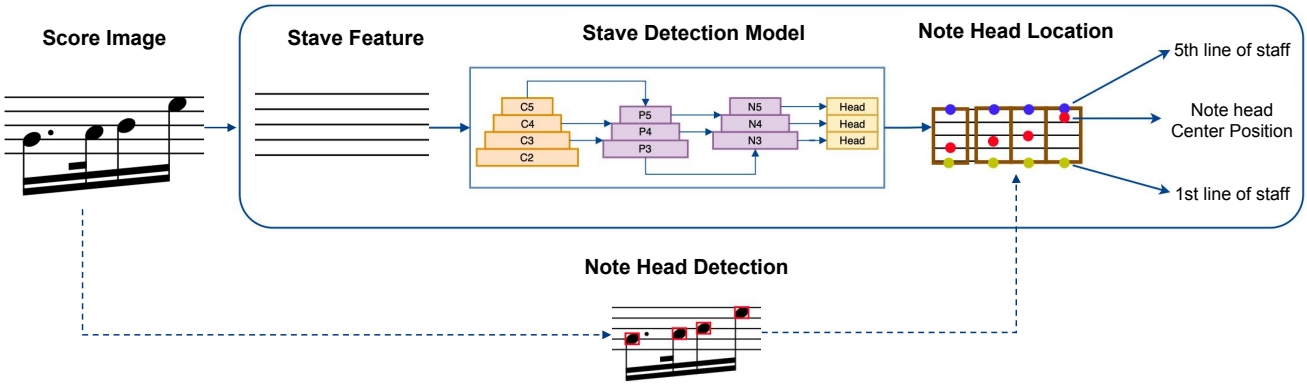


Figure 3. Overview of note pitch determination.

As for the stave detection model, we design a stave-aware network based on the YOLOX-S [41]. The complete network consists of three stages:

1. *Feature extraction.* We choose the CSPDarknet [42] backbone to extract the features of the staff lines (sized $576 \times 576 \times 3$). The CSPDarknet adopts the structure of YOLOv5s but replaces the LeakyReLU activation function with SiLU.
2. *Feature fusion.* The output of the backbone is an image with a size of $20 \times 20 \times 512$, which is then fed into PANet [43] for feature fusion. Finally, the network will obtain three feature branches with sizes of $20 \times 20 \times 512$, $40 \times 40 \times 256$, and $80 \times 80 \times 128$.
3. *Decoupling of prediction branch.* On the fused feature map, different channel feature maps are first unified to 128 dimensions with a 1×1 convolution. Then, two branches are used to perform decoupling on the detection head, and an IOU branch is added to the regression branch. Finally, the network merges the output of the three branches. Since the spectral-aware network used in this paper has a classification number of 1, the final network will output a two-dimensional vector of size 6×8400. Here, 8400 represents the number of predicted boxes, and 6 represents the regression and classification information for each predicted box.

On the basis of the original YOLOX network, we add the YIoU loss branch to strengthen the training of the stave-aware network. When predicting the position of staff lines, we only consider the vertical location of the prediction box rather than its horizontal width. This is because the vertical position plays a crucial role in determining the pitch arrangement. To enhance the network's perception of the height position of staff lines, we design a YIoU loss to measure the rectangular boxes' overlap degree in the ordinate direction, as shown in expression (1) and Figure 4.

$$YIoU = \frac{length(P \bigcap B)}{length(P \bigcup B)} = \frac{L1}{L2} \tag{1}$$

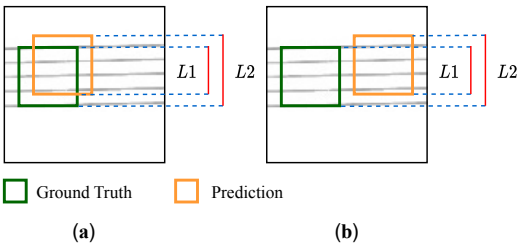


Figure 4. Two situations with the same YIoU loss: (**a**) bounding box and ground truth box intersecting. (**b**) bounding box and ground truth box not intersecting.

To avoid the situation shown in Figure 4b, we add YIoU loss to IoU loss in the original loss function. The total loss of the network and the loss of each branch are shown in expressions (2) and (3). In the expressions, α represents the weight coefficient, which controls the influence degree of IoU and YIoU loss. In our work, we set $\alpha = 5$. Both the classification branch and the regression branch use the BCE loss function.

$$L_{total} = \alpha \times L_{IoU} + \alpha \times L_{YIoU} + L_{cls} + L_{loc} \tag{2}$$

$$L_{YIoU} = \frac{1}{N_{pos}} \sum_{i \in pos}^{N} 1 - YIoU_i^2 \tag{3}$$

3.4. Musical Symbol Detection

Musical symbol detection is to determine the position and type of basic musical symbols, which include the notes, clefs, and accidentals shown in Figure 5a. The symbols in the music sheet are detected and split based on the positions of the staff lines. With the split symbols, especially split notes, we are able to identify the relative position of note

heads on the staff and then determine the note pitch with the type of clef and accidental in the same staff. We can obtain the note type after feeding the split note into the note type classification model. Specifically, the note type classification model is able to obtain the note duration and the note head duration, which determine the note type together from the split note. The note head duration shown in Figure 5b includes the whole note, half note, and quarter note, whereas the note duration shown in Figure 5c is classified into 10 types.

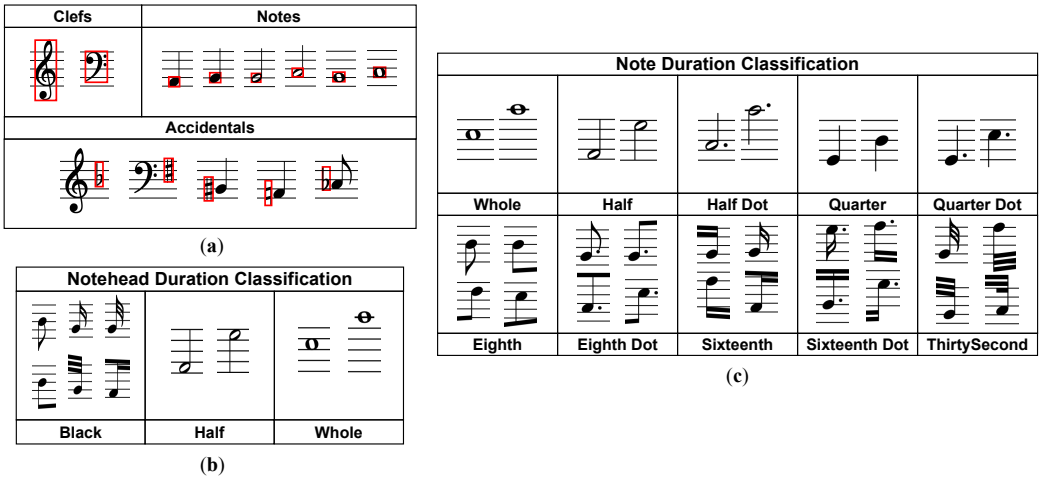


Figure 5. (**a**) Examples of basic musical symbols. (**b**) Examples of note head duration. (**c**) Examples of note duration.

To detect the symbols in the music sheet, we design a network based on the FCOS [44] and adopt the HRNetV2p-W18 [45] as the backbone network. The musical symbol detection model shown in Figure 6 consists of a backbone network, feature pyramid networks, and detection heads. After the input image (with a size of $600 \times 1200 \times 3$) is fused with multi-scale features through the feature pyramid, three sets of feature maps with sizes of 600×400, 300×200, and 150×100 will be generated. The detection heads will classify each pixel and output the classification, regression, and center-ness branches for each note. In addition, we modify the stride of the convolutional layer in the original input stem network to enhance the feature extraction of subsequent backbone networks, specifically for small objects. The input stem comprises two 3×3 convolutional layers with a stride of 2. We also modify the stride of the second convolutional layer from 2 to 1 for the feature extraction of small-sized detection objects in music sheets.

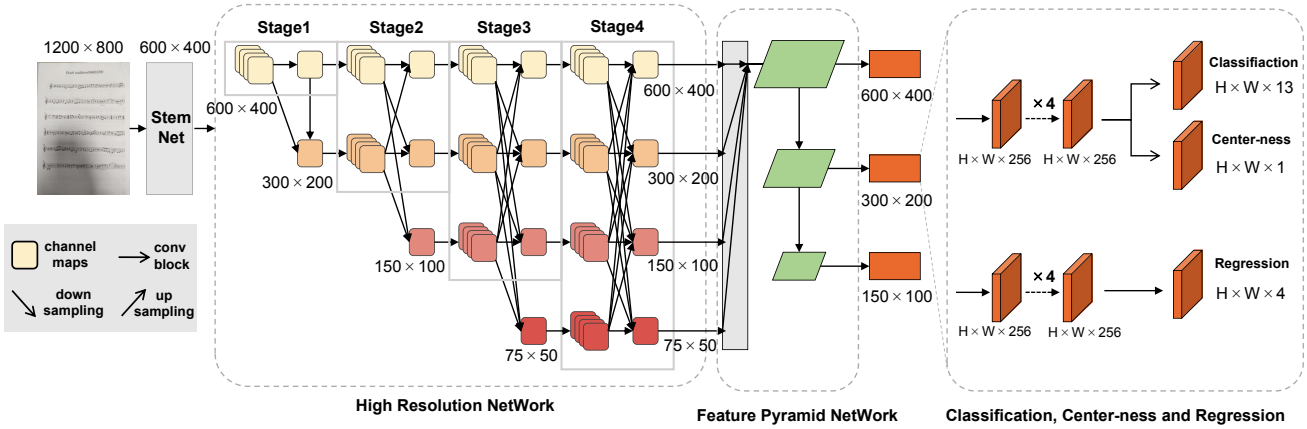


Figure 6. Musical symbol detection model.

To predict the note duration and note head duration of each note, we design a classification network based on RepVGG [46]. This network shown in Figure 7a is composed of 28 RepVGGBlocks, which are divided into five stages, each including 1, 4, 6, 16, and 1 block. The RepVGGBlock shown in Figure 7b consists of a 3×3 convolutional branch, a 1×1 convolutional branch, and an identity mapping branch. At the first block of each stage, the stride of the convolutional layer is set to 2 for downsampling, whereas the other convolutional layers' stride in the same stage is 1. The architecture of RepVGGBlock is equal to Figure 7c for inference. Following Stage 5, the network reduces dimensionality through a global average pooling layer. The scores of different note types are then outputted through a fully-connected layer with the Softmax function.

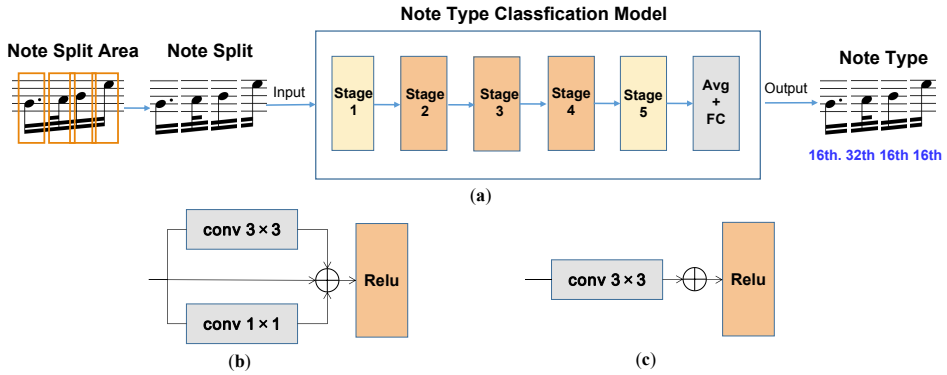


Figure 7. Overview of note type classification. (**a**) Note type classification model. (**b**) RepVGGBlock for training. (**c**) RepVGGBlock for inference.

3.5. Notation Assembly

The purpose of notation assembly is to calculate and combine the results of the stave detection model and musical symbol detection model according to the semantic rules of music sheets, and finally obtain the pitch and type of notes. Specifically, notation assembly includes three tasks:

1. *Determining the note pitch*. Note pitch is decided by the relative position of the note heads and the positions of accidental, clef, and staff lines.
2. *Determining the note type*. Note type is judged by the classification of note head duration and note duration.
3. *Combining output note sequences*. Identify the fused output sequence of notes by note pitch and note type.

The logic of notation assembly is shown in Figure 8. The (a) and (b), respectively, describe the five basic elements that serve the notation assembly: stave position, note head position, note head type, note head duration, and note duration. To determine the pitch of notes (as shown in Figure 8c), we interpolate the notes' relative position on the staff based on the absolute position of the staves and notes. Then, we obtain the note pitch according to the types of note heads, accidentals, and clefs. To determine the note type (as shown in Figure 8d), we compare and judge the classification results of note head duration and note duration. Specifically, we use three types as the duration of the note head: "Black" corresponds to a quarter (or shorter) note, "Half" corresponds to a half note, and "Whole" corresponds to a whole note. The types of note duration are divided into two categories: dotted notes and non-dotted notes, which can be used to judge the existence of dotted notes. The strategy for determining the type of note is as follows: when the note duration matches the note head duration, the note type is the same as the output of the network; when they do not match, we consider the final note type as a combination of the note head duration and the dotted note.

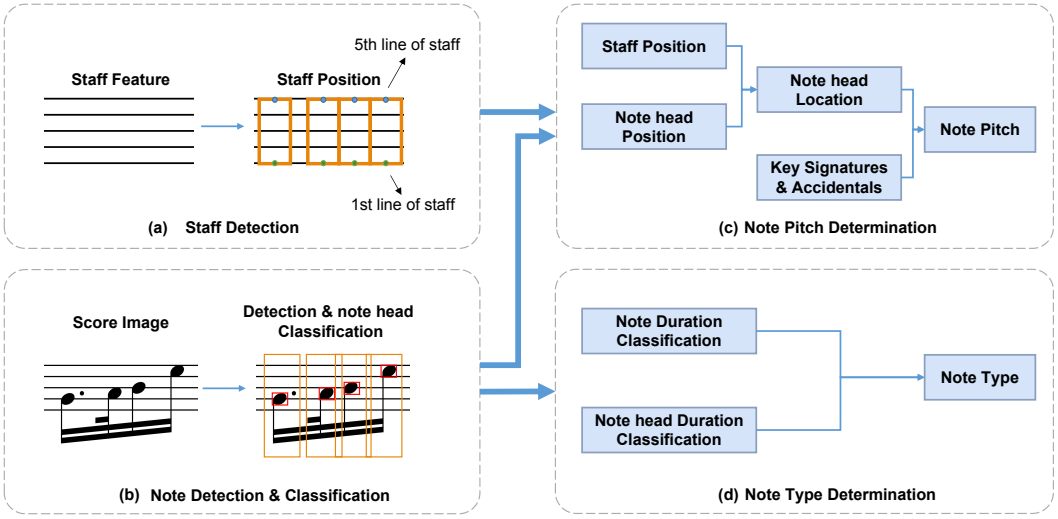


Figure 8. Overview of notation assembly.

4. Dataset and Experiment

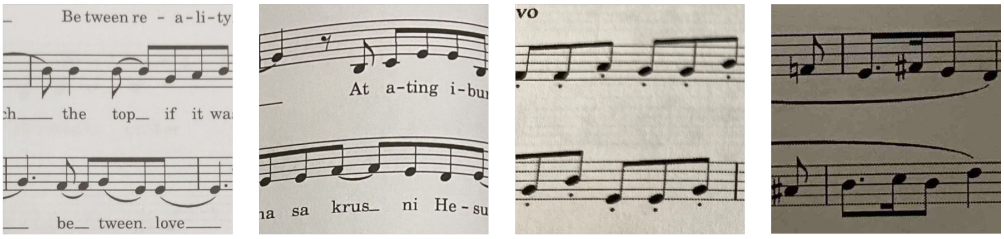
4.1. Dataset

PrIMuS. The PrIMuS [3] dataset is a monophonic printed music dataset for sequence recognition, which contains 87,678 monodic single-staff real scores in common Western notation. Five different formats are used to describe each record: a rendered PNG image, a MIDI file, an MEI file, and two label files in custom encodings (semantic encoding and agnostic encoding). We selected 15,062 scores from PrIMuS as a part of the training set for the non-photographic images. For object detection, every 10 images are concatenated vertically into one image, and each musical symbol in the music sheets is labeled with LabelMe software.

DeepScoresV2. The DeepScoresV2 dataset is the current mainstream printed dataset for music object detection [47]. It contains 255,386 images, 135 category annotations, and a total of 151 million musical symbols. The DeepScoreV2 dataset provides bounding boxes for 135 different classes of musical symbols, along with pitch and type information for notes. Additionally, DeepScoresV2 has a dense version, including 1714 of the most diverse music score images in total. We choose the dense version as a part of the training set.

CPMS (https://github.com/itec-hust/CPMS (accessed on 15 August 2023)). The Camera Printed Music Staves (CPMS) dataset is a monophonic printed single-staff music score dataset, differing from PrIMuS and CameraPrIMuS in that we provide photos captured by mobile phone cameras under various angles and lighting conditions in realistic scenarios for each record. In the CPMS, the training set consists of 910 lines containing 31,262 symbols, whereas the test set consists of 600 lines containing 15,542 symbols. The training set and the test set have different sources: the training set is sourced from the musescore-dataset (https://github.com/Xmader/musescore-dataset (accessed on 15 August 2023)) and the test set is sourced from the public repertoire of the 2020 sight-singing exam of the Wuhan Conservatory of Music in China (http://www.hbea.edu.cn/html/2019-09/12349.html (accessed on 15 August 2023)). Specifically, the training set is obtained by taking photos of A4-sized printed music scores, whereas the test set is sourced from real books. All the sheet music photos were captured from (1) flat (Figure 9a), (2) bend (Figure 9b), (3) keystone distortion (Figure 9c), and (4) uneven light distribution (Figure 9d) scenes that may appear in real books. For each image in the CPMS dataset, we labeled the position of all the symbols and provided a manually cropped PNG file and a semantic encoding file for each line, similar to the PrIMuS dataset. It means that the CPMS is available for both object detection and sequence recognition. We choose the test set of CPMS as the test set of the experiments behind because there is no other public OMR dataset that is also available for

both object detection and sequence recognition. That is also one of the reasons why we built the CPMS.



(**a**) (**b**) (**c**) (**d**)

Figure 9. Typical images from CPMS. (**a**) An example of flat sheets. (**b**) Image with curved lines. (**c**) Image with keystone distortion. (**d**) Image with uneven light distribution.

4.2. Evaluation Metric

We use the following three evaluation metrics to evaluate our method: pitch accuracy, type accuracy, and note accuracy. Their definitions are as follows:

1. *Pitch Accuracy*: the proportion of notes whose pitch is correctly predicted to the total number of notes.
2. *Type Accuracy*: the proportion of notes whose type is correctly predicted to the total number of notes.
3. *Note Accuracy*: the proportion of notes whose pitch as well as type is correctly predicted to the total number of notes.

For each note, when the predicted result is consistent with the corresponding ground truth, set it as a positive sample (PS), otherwise set it as a negative sample (NS). The accuracy is calculated as follows:

$$Accurary = \frac{PS}{PS + NS} \tag{4}$$

4.3. Implementation Details

We train the stave detection model, musical symbol detection model, and note type classification model with the following methods:

To train the stave detection model, we utilized an NVIDIA GTX 1080 GPU and employed the stochastic gradient descent (SGD) optimizer. The learning rate was set to 0.01, momentum to 0.9, and weight decay to 0.0005. Each iteration involved processing a batch of 16 images, and the total number of training epochs was set to 300. Additionally, we applied a learning rate warm-up strategy during the first five training epochs.

We trained the musical symbol detection model with two NVIDIA GTX 1080 GPUs and the optimization algorithm of SGD. The learning rate was set to 0.005, the momentum was set to 0.9, and the weight decay was set to 0.0001. Each iteration processed a batch of size 2, and we employed a learning rate decay strategy based on batches. The total number of training epochs was set to 1000, with learning rate adjustments made at the 500th and 750th epochs.

The note duration classification network was trained with an NVIDIA GTX 1080 GPU. We employed the SGD optimizer with a learning rate of 0.1, momentum of 0.9, and weight decay of 0.0001. The learning rate was reduced with a cosine annealing strategy, and the total number of training epochs was set to 120.

4.4. Experiment

4.4.1. Factors Affecting Pitch Accuracy

The purpose of this experiment is to analyze the impact of the stave detection model and the note head position on the accuracy of note pitch. We select the comb filter-based method proposed by Tsai et al. [39] as a baseline, which is a representative traditional method to obtain the position of note heads. With the training set of the DeepScoreV2

dataset and test set of the CPMS dataset, we design four experimental groups: (1) comb filter-based method; (2) comb filter-based method + note head position; (3) stave detection model; (4) stave detection model + note head position. The note head position is obtained by the musical symbol detection model.

4.4.2. Robustness Comparison

The purpose of this experiment is to figure out the impact of the photographic training data on detection accuracy. We select the CRNN-CTC [3] as a comparison for its high accuracy on non-photographic score images and representativeness of current open-source OMR sequence methods. In particular, we applied the preprocessing method described in the first step of Section 3.2 to remove background lighting from the input images. We compare the detection accuracy of our method and the CRNN-CTC model on the CPMS test set after being trained on two different combinations of datasets: one was only the PrIMuS dataset, and the other is a mixture of both the PrIMuS and CPMS datasets. Note that the PrIMuS dataset consists of printed staves and does not have any photographic features. We use these two datasets to compare the robustness of these two methods.

4.4.3. Data Dependence Proof

This experiment aims to argue the effect of the data dependence of our method. We compare the detection accuracy of our method on the CPMS test set after being trained on three different datasets, to clarify whether our method can get rid of the dependence on the photographic training set. The first two datasets are the same as the last experiment, and the third dataset is DeepScoresV2, which does not have any photographic features but has a much larger amount of scores than the other two.

5. Results and Discussion

5.1. Experimental Results

The results of Experiment 1 are shown in Table 1; both the stave detection model and the note head position significantly improve the note pitch accuracy. The note head position improves the note pitch accuracy of both the comb filter-based method and the stave detection model, which indicates that the relative position of note heads on the stave is an important factor affecting note pitch accuracy. Moreover, the note pitch accuracy of the stave detection model without note head position is higher than that of the comb filter-based method with note head position. The stave detection model with note head position achieves the highest accuracy of 99.23%. This demonstrates that our stave detection model can effectively detect the position of staff lines and works well in conjunction with the note head position.

Table 1. Pitch accuracy comparison between different methods.

Method	Note Pitch Accuracy on CPMS(%)
Comb filter-based method	85.54
Comb filter-based method + note head position	91.23
Stave detection model	95.49
Stave detection model + note head position	99.23

The results of Experiment 2 are shown in Table 2. It is clear that our method performs much better than the CRNN-CTC on recognition in camera-based scenarios when the training set consists of only the PrIMuS dataset. Even when the CPMS dataset, which includes sheet music photos, was added to the training set, our method still comprehensively outperformed the baseline. This indicates that our method is more robust and much better at learning photographic features on the training data than CRNN-CTC.

Table 2. Accuracy comparison between our method and the CRNN-CTC model.

Method	Training Set	Accuracy on CPMS(%)		
		Pitch	Type	Note
CRNN-CTC	PrIMuS	44.23	51.58	37.42
Ours		89.77	94.46	85.17
CRNN-CTC	PrIMuS + CPMS training set	95.07	96.74	91.95
Ours		97.10	97.15	94.40

The results of Experiment 3 are shown in Table 3. After training on DeepScoresV2, which does not contain sheet music photos, our method achieves higher pitch accuracy and note accuracy compared to the mixed dataset training of the PrIMuS and CPMS datasets, whereas the type accuracy is slightly lower than the best one. This indicates that our method is able to handle the complexities of sheet music photos in real-world scenarios through training on non-photographic data, demonstrating low data dependency and high robustness.

Table 3. Accuracy comparison of our method on different datasets.

Method	Training Set	Accuracy on CPMS(%)		
		Pitch	Type	Note
Ours	PrIMuS	89.77	94.46	85.17
	PrIMuS + CPMS training set	97.10	97.15	94.40
	DeepScoresV2	99.23	96.87	96.29

5.2. Distribution of Errors

Since the papers in books on camera-based realistic scenarios tend to have a great impact on the curvature of staff lines, we manually divide the test set of the CPMS dataset into 359 flat scores and 241 curved scores based on the degree of curvature.

The pitch accuracy comparison in Figure 10 shows that our method performs similarly for flat and curved scores, whereas the CRNN-CTC performs significantly better on flat scores than curved scores. This indicates that the effect of stave curvature on our method is much less than that on CRNN-CTC, which means that our stave-aware method effectively handles staff distortion and is more robust.

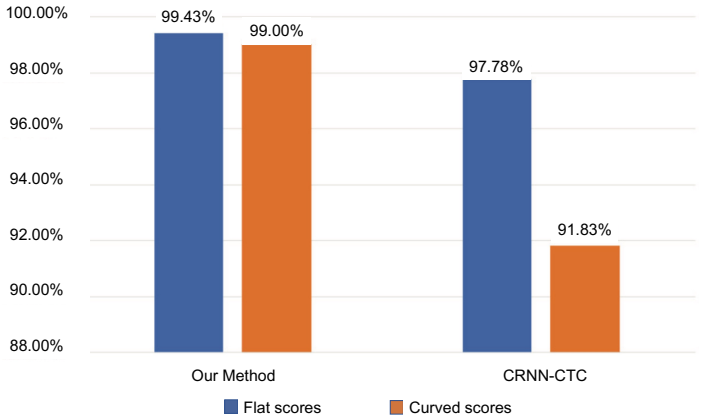


Figure 10. Pitch accuracy comparison histogram for different bending levels.

The comparison of type accuracy is presented in Figure 11. Both methods achieve similar type accuracy for both sets of scores. Moreover, the type accuracy of both methods is slightly higher for straight music scores than for curved music scores, suggesting that the

effect of music score curvature on type accuracy is minimal. This is because the curvature of staves primarily affects the position (pitch) of the notes rather than their shape (type).

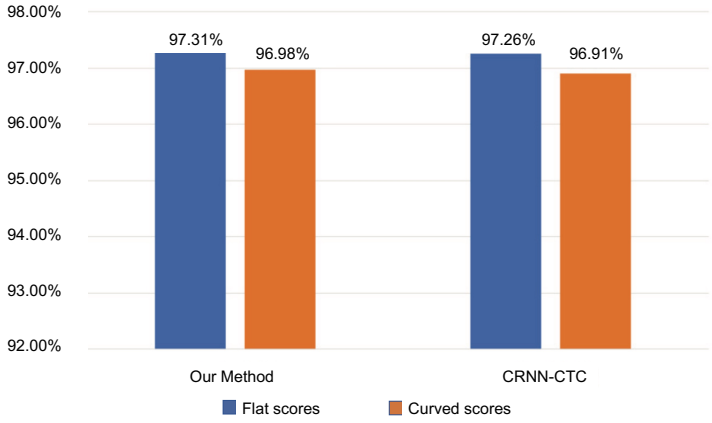


Figure 11. Type accuracy comparison histogram for different bending levels.

6. Conclusions

In this paper, we present a stave-aware OMR method based on object detection that aims to recognize sheet music photos captured by mobile phone cameras in real-world scenarios. Our method allows us to split curved stave into multiple segments by predicting the positions of staves and determine the pitch of notes by combining stave positions and note head positions, effectively improving the pitch accuracy on sheet music photos. Our method consists of two parts: symbol detection and notation assembly. Symbol detection is applied to locate note heads and classify note head duration, whereas notation assembly is used to determine note pitch and note type. Our method achieves higher accuracy and better robustness than the CRNN-CTC model in recognizing sheet music photos in real-world scenarios, particularly in handling the common issue of curved staves. Additionally, our method demonstrates less data dependence on training sets composed of sheet music photos.

Polyphonic music score recognition is an essential part of the OMR field. Currently, our work is limited to monophonic scores. In our future work, we intend to extend the application of our method to recognize more complex sheet music, including polyphonic scores and even piano scores. We are also going to expand our dataset with more photos of monophonic scores and polyphonic scores in various scenarios.

Author Contributions: Conceptualization, L.L.; methodology, Y.W.; software, Y.L.; validation, R.W. and Y.L.; investigation, Y.L.; data curation, R.W.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L. and R.W.; visualization, Y.W.; supervision, W.X.; funding acquisition, W.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62277019 and 61877060.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The CPMS dataset is available at https://github.com/itec-hust/CPMS (accessed on 15 August 2023) for further research purposes.

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

References

1. Shatri, E.; Fazekas, G. Optical Music Recognition: State of the Art and Major Challenges. *arXiv* **2020**, arXiv:2006.07885.
2. Calvo-Zaragoza, J.; Valero-Mas, J.J.; Pertusa, A. End-to-end optical music recognition using neural networks. In Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR, Suzhou, China, 23–27 October 2017; pp. 23–27.
3. Calvo-Zaragoza, J.; Rizo, D. End-to-end neural optical music recognition of monophonic scores. *Appl. Sci.* **2018**, *8*, 606. [CrossRef]

4. Qiong, W.; Qiang, L.; Xin, G. Optical Music Recognition Method Combining Multi-Scale Residual Convolutional Neural Network and Bi-Directional Simple Recurrent Units. *Laser Optoelectron. Prog.* **2020**, *57*, 081006. [CrossRef]
5. Li, Y.; Liu, H.; Jin, Q.; Cai, M.; Li, P. TrOMR: Transformer-Based Polyphonic Optical Music Recognition. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
6. Ríos-Vila, A.; Rizo, D.; Iñesta, J.M.; Calvo-Zaragoza, J. End-to-end optical music recognition for pianoform sheet music. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2023**, *26*, 347–362. [CrossRef]
7. Hajič, J.; 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; Volume 1, pp. 39–46.
8. Hajic, J.; Pecina, P. Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression. *arXiv* **2017**, arXiv:1708.01806.
9. Calvo-Zaragoza, J.; Toselli, A.H.; Vidal, E. Handwritten music recognition for mensural notation with convolutional recurrent neural networks. *Pattern Recognit. Lett.* **2019**, *128*, 115–121. [CrossRef]
10. Baró, A.; Badal, C.; Fornés, A. Handwritten historical music recognition by sequence-to-sequence with attention mechanism. In Proceedings of the 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Dortmund, Germany, 7–10 September 2020; pp. 205–210.
11. Calvo-Zaragoza, J.; Rizo, D. Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores. In Proceedings of the ISMIR, Paris, France, 23–27 September 2018; pp. 248–255.
12. Liu, A.; Zhang, L.; Mei, Y.; Han, B.; Cai, Z.; Zhu, Z.; Xiao, J. Residual recurrent CRNN for end-to-end optical music recognition on monophonic scores. In Proceedings of the 2021 Workshop on Multi-Modal Pre-training for Multimedia Understanding, Taipei, Taiwan, 21 August 2021; pp. 23–27.
13. Shishido, T.; Fati, F.; Tokushige, D.; Ono, Y.; Kumazawa, I. Production of MusicXML from Locally Inclined Sheetmusic Photo Image by Using Measure-based Multimodal Deep-learning-driven Assembly Method. *Trans. Jpn. Soc. Artif. Intell.* **2023**, *38*, A-MA3_1–A-MA3_12. [CrossRef]
14. Alfaro-Contreras, M.; Valero-Mas, J.J. Exploiting the two-dimensional nature of agnostic music notation for neural optical music recognition. *Appl. Sci.* **2021**, *11*, 3621. [CrossRef]
15. Alfaro-Contreras, M.; Ríos-Vila, A.; Valero-Mas, J.J.; Iñesta, J.M.; Calvo-Zaragoza, J. Decoupling music notation to improve end-to-end Optical Music Recognition. *Pattern Recognit. Lett.* **2022**, *158*, 157–163. [CrossRef]
16. Rebelo, A.; Fujinaga, I.; Paszkiewicz, F.; Marcal, A.R.; 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]
17. Pinto, T.; Rebelo, A.; Giraldi, G.; Cardoso, J.S. Music score binarization based on domain knowledge. In Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Las Palmas de Gran Canaria, Spain, 8–10 June 2011; Springer: Berlin/Heidelberg, Germany; pp. 700–708.
18. Szwoch, M. Guido: A musical score recognition system. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, 23–26 September 2007; Volume 2, pp. 809–813.
19. Chen, G.; Zhang, L.; Zhang, W.; Wang, Q. Detecting the staff-lines of musical score with hough transform and mathematical morphology. In Proceedings of the 2010 International Conference on Multimedia Technology, Ningbo, China, 29–31 October 2010; pp. 1–4.
20. Miyao, H.; Nakano, Y. Note symbol extraction for printed piano scores using neural networks. *IEICE Trans. Inf. Syst.* **1996**, *79*, 548–554.
21. Li, C.; Zhao, J.; Cai, J.; Wang, H.; Du, H. Optical Music Notes Recognition for Printed Music Score. In Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 8–9 December 2018; Volume 1, pp. 285–288.
22. Pacha, A.; Hajič, J.; Calvo-Zaragoza, J. A baseline for general music object detection with deep learning. *Appl. Sci.* **2018**, *8*, 1488. [CrossRef]
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. Gao, C.; Tang, W.; Jin, L.; Jun, Y. Exploring Effective Methods to Improve the Performance of Tiny Object Detection. In Proceedings of the European Conference on Computer Vision, Virtual, 23–28 August 2020; Springer: Cham, Switzerland; pp. 331–336.
26. Feng, Y.; Wang, X.; Xin, Y.; Zhang, B.; Liu, J.; Mao, M.; Xu, S.; Zhang, B.; Han, S. Effective feature enhancement and model ensemble strategies in tiny object detection. In Proceedings of the European Conference on Computer Vision, Virtual, 23–28 August 2020; Springer: Cham, Switzerland; pp. 324–330.
27. Yu, X.; Han, Z.; Gong, Y.; Jan, N.; Zhao, J.; Ye, Q.; Chen, J.; Feng, Y.; Zhang, B.; Wang, X.; et al. The 1st tiny object detection challenge: Methods and results. In Proceedings of the European Conference on Computer Vision, Virtual, 23–28 August 2020; Springer: Cham, Switzerland; pp. 315–323.

28. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef]
29. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
30. Huang, J.; Zhu, P.; Geng, M.; Ran, J.; Zhou, X.; Xing, C.; Wan, P.; Ji, X. Range scaling global u-net for perceptual image enhancement on mobile devices. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018; pp. 230–242.
31. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
32. Van Der Wel, E.; Ullrich, K. Optical Music Recognition with Convolutional Sequence-to-Sequence Models. In Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, 23–27 October 2017; pp. 731–737.
33. Ríos-Vila, A.; Calvo-Zaragoza, J.; Inesta, J.M. Exploring the two-dimensional nature of music notation for score recognition with end-to-end approaches. In Proceedings of the 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Dortmund, Germany, 7–10 September 2020; pp. 193–198.
34. Edirisooriya, S.; Dong, H.W.; McAuley, J.; Berg-Kirkpatrick, T. An Empirical Evaluation of End-to-End Polyphonic Optical Music Recognition. *arXiv* **2021**, arXiv:2108.01769.
35. Shi, B.; Bai, X.; Yao, C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 2298–2304. [CrossRef]
36. Ríos-Vila, A.; Iñesta, J.M.; Calvo-Zaragoza, J. On the use of transformers for end-to-end optical music recognition. In Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Aveiro, Portugal, 4–6 May 2022; pp. 470–481.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
38. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
39. Tsai, T.J.; Yang, D.; Shan, M.; Tanprasert, T.; Jenrungrot, T. Using Cell Phone Pictures of Sheet Music To Retrieve MIDI Passages. *IEEE Trans. Multimed.* **2020**, *22*, 3115–3127. [CrossRef]
40. Fisher, R.; Perkins, S.; Walker, A.; Wolfart, E. *Hypermedia Image Processing Reference*; John Wiley & Sons Ltd.: Chichester, UK, 1996; pp. 118–130.
41. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
42. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
43. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
44. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
45. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703.
46. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
47. 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), Milan, Italy, 10–15 January 2021; pp. 9188–9195.