electronics

Article

Kernel Density Estimation and Convolutional Neural Networks for the Recognition of Multi-Font Numbered Musical Notation

Qi Wang [1], Li Zhou [2,*] and Xin Chen [1]

[1] School of Automation, China University of Geosciences, Wuhan 430074, China
[2] School of Arts and Communication, China University of Geosciences, Wuhan 430074, China
* Correspondence: zhouli@cug.edu.cn

Abstract: Optical music recognition (OMR) refers to converting musical scores into digitized information using electronics. In recent years, few types of OMR research have involved numbered musical notation (NMN). The existing NMN recognition algorithm is difficult to deal with because the numbered notation font is changing. In this paper, we made a multi-font NMN dataset. Using the presented dataset, we use kernel density estimation with proposed bar line criteria to measure the relative height of symbols, and an accurate separation of melody lines and lyrics lines in musical notation is achieved. Furthermore, we develop a structurally improved convolutional neural network (CNN) to classify the symbols in melody lines. The proposed neural network performs hierarchical processing of melody lines according to the symbol arrangement rules of NMN and contains three parallel small CNNs called Arcnet, Notenet and Linenet. Each of them adds a spatial pyramid pooling layer to adapt to the diversity of symbol sizes and styles. The experimental results show that our algorithm can accurately detect melody lines. Taking the average accuracy rate of identifying various symbols as the recognition rate, the improved neural networks reach a recognition rate of 95.5%, which is 8.5% higher than the traditional convolutional neural networks. Through audio comparison and evaluation experiments, we find that the generated audio maintains a high similarity to the original audio of the NMN.

Keywords: numbered musical notations; kernel density estimation; bar line criteria; convolutional neural networks

check for updates

Citation: Wang, Q.; Zhou, L.; Chen, X. Kernel Density Estimation and Convolutional Neural Networks for the Recognition of Multi-Font Numbered Musical Notation. *Electronics* **2022**, *11*, 3592. https://doi.org/10.3390/electronics11213592

Academic Editor: George A. Tsihrintzis

Received: 4 October 2022
Accepted: 1 November 2022
Published: 3 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Musical notation is a bridge between musicians and music, which can visualize music so that musicians can perform it. The staff and numbered musical notation (NMN) are the most widely used types of musical notation. Compared with the staff, NMN has many advantages, such as being easier to learn and easy to write, which makes it more widely used than staff, and it plays an essential role in promoting and popularizing mass music and cultural activities. When composing music, many musicians are accustomed to using easy-to-write NMN to record the initial creative thinking.

In the information age, the digitization of music information is critical. We need to use electronics to obtain the information in the score accurately and convert the music language in the score into the computer language in documents. In the music world, this task is called "optical music recognition (OMR)" [1], which includes comprehensive techniques such as image processing and artificial intelligence algorithms to convert the music notes into machine-readable symbolic format [2].

Regardless of the recognition NMN or staff, the basic framework [3] of optical score recognition must be followed. They both follow the same image preprocessing and representation construction strategy but differ in music symbol recognition [2]. A graph neural network (GNN) [4] that retrieves a particular representation of a graph has been developed to detect the staff (melody) lines. For music symbol recognition, a new data-generation technique [5,6] has been proposed to improve the layout analysis. The study [7,8] described this

paper's most important part and retrieved the staff region. Dalit, C. et al., 2008 [9] erased the staff lines through geometric taxonomy and skeletonization. The article [10–12] highlighted handwritten music recognition by using convolutional recurrent neural networks (CRNN). However, these articles are only dedicated to staff notation.

Most NMN consists of melody lines and lyrics lines. To correctly identify the valid music symbols in the NMN, it is necessary to separate the melody and lyrics lines and then make a classification of the symbols in the melody lines. At present, there are few types of research on the recognition of NMN. An automatic segmentation algorithm [13] has been proposed with a double-scale descent method to realize simplified image segmentation. However, the authors conducted little research on the recognition of musical notation symbols. Recognition of numbered notation is realized by using morphological and geometric features and template matching algorithms. The projection method [14–16] with a weighted classifier realizes the recognition of the NMN. However, both of studies use musical notation with one type of font. The work presented here aims to eliminate the vacancy of research in multi-font numbered musical note recognition.

For multi-font melody line detection, the separation of the melody lines and the lyric lines is a binary classification problem. Their symbol heights are arranged differently. Supervised learning-based methods may be impractical, due to the lack of large NMN line datasets. Thus, the unsupervised learning method is worth considering. Kernel density estimation (KDE) [17] is an unsupervised probability distribution learning method. In recent studies [18–20], it functions as a data analyst and classifier. The single KDE analyst fails to capture the features of data with many details and frequent variations such as symbols in NMN. Therefore, we propose bar line criteria, a set of statistical rules incorporating NMN image features that are combined with the KDE method to perform the separation task (Section 2.5).

The classification of melody line symbols is similar in nature to text recognition of scanned documents. Convolutional neural networks (CNN) are a common model for symbol classification. In recent years, many structurally improved CNN have been designed. In study [21], a faster-regional convolutional neural network was proposed for text line segmentation and detection. Yin, W. et al., 2016 [22] adopted multichannel variable-size convolution neural network (MVCNN) to classify English sentences. Improved CNN-like parallel CNN and deep CNN [23] enable much stronger semantic learning capacity. Although these networks are better at classification tasks, they cannot accommodate very diverse input sizes and styles. For the problem of variable symbol size in NMN, we improve the traditional network structure by adding a spatial pyramid pooling layer. For the problem of changing symbol types and styles in NMN, we noticed that the symbol arrangement of the melody line of the musical notation has a certain level and regularity, and the same level conforms to the style. Based on the above findings, we added a hierarchical analysis module specially designed for NMN to the network and perform parallel training on symbols at each level (Section 2.6).

Briefly, our study is dedicated to eliminating the lack of OMR research, which is scarcely involved in multi-font NMN. We find the difference in the arrangement of symbols between lyric lines and melody lines in NMN, and therefore proposed KDE with bar line criteria to separate the lyrics and melody. We carefully analyze the hierarchical features of symbols in the melody line and improve the traditional convolutional neural network to solve the identification problems associated with variable symbol size and style in NMN.

2. Materials and Methods

2.1. Background on Numbered Musical Notation

A composer or songwriter expects to be understood in the notation by converting a coherent composition of rhythm and pitches [24]. Notations attempt to reduce the practice of composing and playing with rules and ideas. In notations, notes are symbols that record the pitches and the durations. Pitches are referred to as numbers, dots, and so on. Durations

are note length, musical rests, bar lines, etc. Time signatures written as a fraction, e.g., 2/4, are a representation of tempo. Common music symbols are shown in Figure 1 [25].

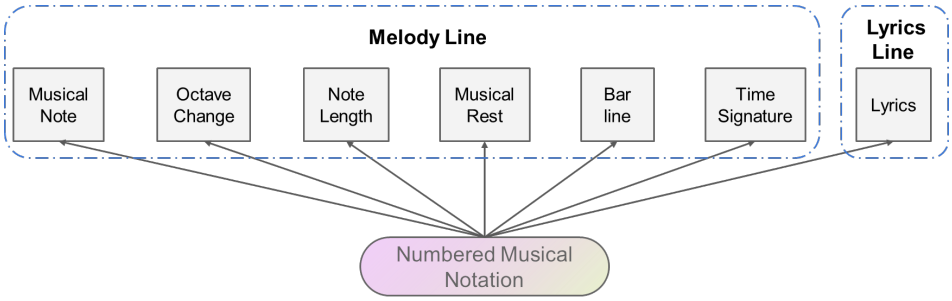


Figure 1. Common music symbols in NMN.

In Figure 1, musical notes and rests are Arabic numbers. Octave change symbols are mainly dots. Note length symbols are diverse lines. Time signatures are numbers and vertical lines whose styles and sizes are different from musical notes. These symbols form the melody line, which is shown in Figure 2. Lyrics are language words. As we can see in Figure 2, their positions are right below the melody lines. It can be concluded that the classification task is difficult due to the extensive types and diverse styles of different symbols.

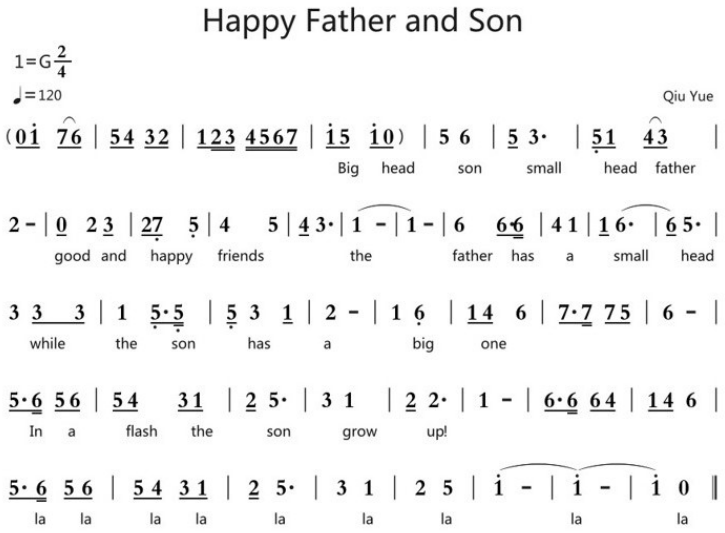


Figure 2. An NMN example in the dataset.

2.2. Block Diagram

The entire NMN recognition procedure with our proposed method is shown in Figure 3. An input NMN image example is presented in Figure 2. The NMN image undergoes processing, segmentation of melody and lyrics, and symbol classification.

2.3. Data Preparation

This article selected 200 complete NMNs of Chinese popular music. There are 10 fonts for the melody line symbols of the musical score, as shown in Figure 4. The musical scores are all monophonic, with lyric lines and melody lines. The symbols of the melody lines are composed of the melody symbols in Figure 1. The notation images are all scanned images, and the images are clear without tilt. An example is shown in Figure 2.

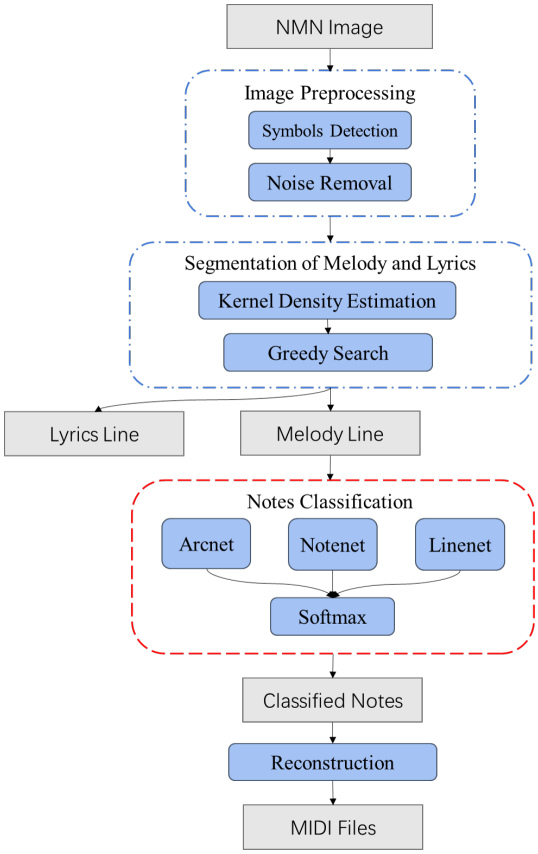


Figure 3. NMN recognition procedure.

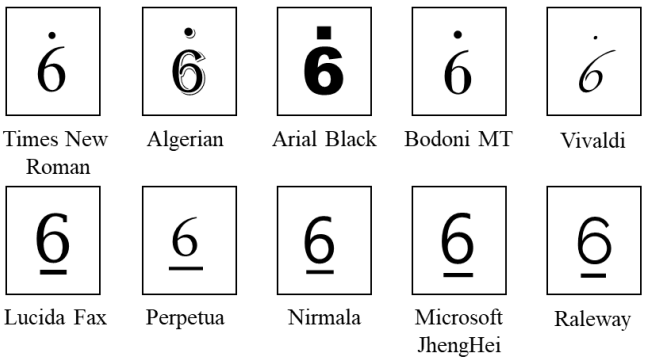


Figure 4. The 10 different NMN fonts.

The lengths of all 200 images with different fonts were all adjusted to 800, and the aspect ratio was kept unchanged during adjustment. All NMN images have been binarized. In other words, the pixel value of each pixel can only be either 0 or 255. Among the 200 different types of musical notations, we used 180 musical notations in the training set and 20 in the test set. All our model training and testing were completed on these 200 multi-font NMNs. In the training set, there are 18 NMNs of each font, and in the test set, there are 2 NMNs of each font.

Considering that different fonts have a great impact on the experimental results, we divided the NMNs tested into four batches for subsequent experiments:

Batch 1: One font (Microsoft JhengHei) and two NMNs.

Batch 2: Three fonts (Times New Roman, Algerian, Arial Black, and Bodoni MT) and six NMNs.

Batch 3: Six fonts (Vivaldi, Lucida Fax, Perpetua, Nirmala, and Microsoft JhengHei) and 12 NMNs.

Batch 4: 10 fonts (all the fonts mentioned in Figure 4) and 20 NMNs.

It can be seen that the fonts of the melody symbols of NMN are different, and the styles of the symbols are also different. Whether the symbols are numbers (musical notes), dots (octave change), or horizontal lines (note length), the style has changed.

In addition, we chose five specific NMN examples as the experimental data for generated audio evaluation. Their titles, durations, and note numbers are shown in Table 1.

Table 1. Specific information of generated audio evaluation samples.

Title	Duration	Number of Notes
Dreams	30 s	42
Painter	29 s	45
Secret	21 s	40
Ten Years	29 s	51
Edelweiss	24 s	42

2.4. Image Preprocessing

2.4.1. Symbols Detection

Projection [26] is an efficient method for the segmentation of NMN lines based on the alternate applications of the same technique along the x- and y-axes. Let the projection on the y-axis be Prj_i. The calculation method of the dividing line of each line of the musical notation is as follows:

$$\frac{1}{2\alpha}\Sigma_{i=-\alpha}^{\alpha} Prj_i \leq \epsilon \quad (1)$$

The position where the average projection value of a certain projection area is close to 0 is the dividing line.

The seed filling algorithm [27] extracts the position and size of each symbol. The basic process is as follows: when a seed point (x, y) is given, a section in the given area is first filled on the scan line where the seed point is located in the left and right directions [28], and the range [xLeft, xRight] is recorded. Then, the upper and lower two scan lines connected to this segment within the given area are determined and saved in sequence.

Let the parameter in inequality (1) α be 1 and the ϵ be 0.1; then, we obtain the line segmented NMN. By implementing the seed filling method, we detect the outline of each symbol; thus, we obtain its position and shape. An example of a symbol detection result is shown in Figure 5. The outlines of the individual symbols have been detected, and we have marked them with rectangular boxes.

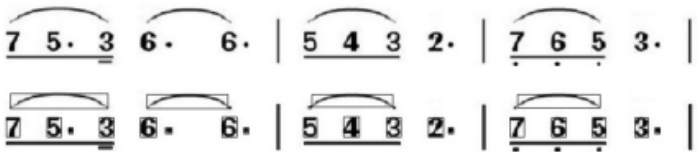


Figure 5. An example of symbol detection results.

2.4.2. Noise Removal

The recognition algorithm in this paper is mainly aimed at the symbols in Figure 1. For other symbols, we can treat them as "noise". We can divide this "noise" into three categories.

Obfuscated Symbols

In NMN, some music symbols are not difficult to recognize, but they interfere with the recognition of other music symbols. We need to remove them in the image preprocessing stage and assign special semantics to the positions of these "noise" symbols. The most common such noise symbols are repetitive symbols, whose left or right dots are easily identified as note dots.

The two points of the repeated symbol are very close to the vertical line, and the x-coordinates are the same. Therefore, in the process of seed filling, we record the *xleft* and *xright* of each scan. If i and j in fill space S are repeated, the two points on one side of the symbol must satisfy:

$$\begin{cases} X_i = X_j \\ yX_i = argmin_{i \in S}(xright_i - xleft_i) \end{cases} \quad (2)$$

Extraneous Symbols

There is also a "noise" symbol, although it does not affect the recognition of other symbols, its meaning has nothing to do with the pitch, length, and strength of the sound [29], such as a breath mark. Such symbols appear very infrequently and can be removed manually.

Special Symbols

In addition, the information contained in some symbols is very important, such as the key signature (1 = C, 2 = D, 3 = E, 4 = F, 5 = G, 6 = A, 7 = B), but it is difficult to incorporate into the symbol recognition algorithm system that follows. Therefore, we consider removing it as noise in the image preprocessing stage and recording the semantics to ensure that the symbol in the audio file information is included. The position of this symbol in the musical notation is fixed, and we use this position as the initial seed point, which can preferentially locate special symbols.

In Figure 6, the detected obfuscated symbols are colored orange. The extraneous symbols are green, and the special symbols are blue. Then, we record their positions and semantics and remove them.

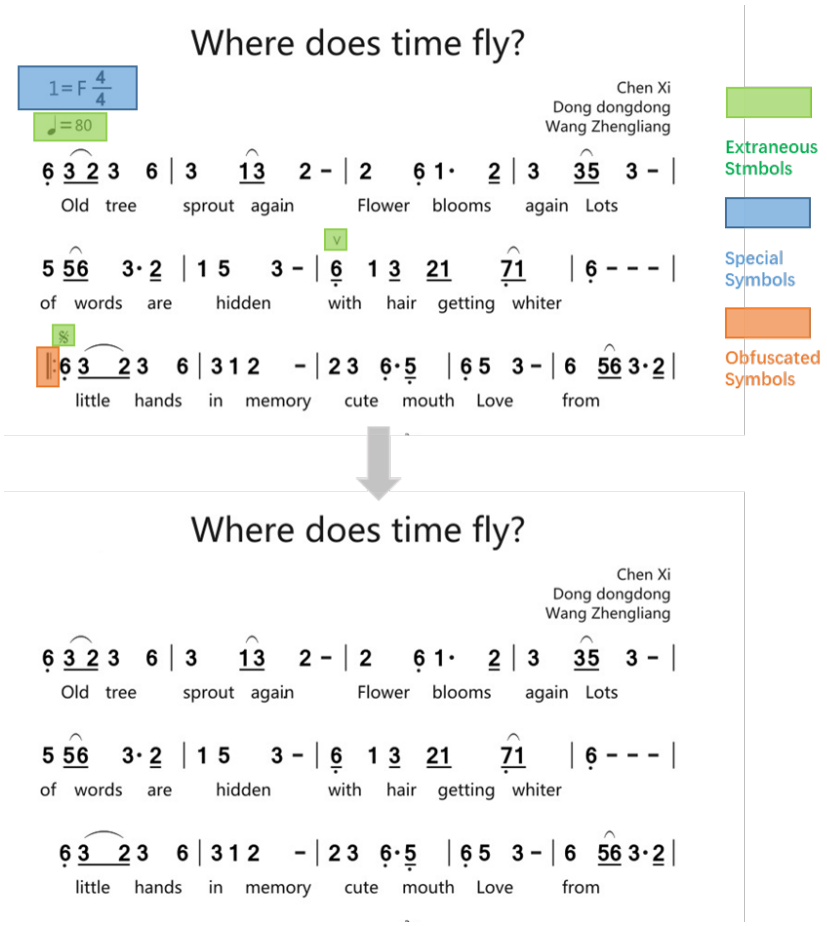


Figure 6. Noise removal diagram.

2.5. Line Separation

No matter how the form and style of the musical notation change, the symbols that make up the lyrics and melody in the musical notation must follow two different distribution rules. Therefore, we can use these two sets of rules as the basis for classification.

The difference in the symbol distribution between the melody line and the lyric line is mainly reflected in the height. The height data will show more details (recorded as height density) after being processed by KDE, which can be used as the basis for classification. No matter how the style of the musical notation changes, in the melody line, the height of the bar line is always greater than the height of the other symbols, and the height difference between a portion of the symbols and the bar line remains constant. However, in the lyric line, there is no symbol whose height is consistently the highest. This feature can be used as classification criteria in combination with height density data. We call the criteria the "bar line criteria". We detail these methods in this subsection.

2.5.1. Kernel Density Estimation

The kernel density estimation method is used to fit the density distribution of the symbol height. The estimated object is the symbol height h, and the independent and identically distributed samples extracted from H are set as $H_1, H_2, \ldots, H_n$. $f(h)$ is the density function that H conforms to, $x \in R$, the estimated value of the probability density function $f(\hat{h})$ at point h needs to be calculated, and the calculation equation of the probability density function is:

$$f(\hat{h}) = \frac{1}{nm}\Sigma_{i=1}^{n}\rho(\frac{h - H_i}{m}) \tag{3}$$

where n is the number of symbols in one line of the musical notation, m is the window width, and ρ is the kernel function. In this paper, a Gaussian kernel function is used, namely:

$$\rho = e^{-\frac{||h-h_i||^2}{2\sigma^2}} \tag{4}$$

where $||h - h_i||$ represents the difference between the current symbol height and the i-th sampling height.

We adopt the asymptotic mean integrated square error (AMISE) to estimate the optimal window width [17].

$$\text{AMISE}(\hat{f}) = T[\int_{-\infty}^{\infty}(bias f(\hat{h}))^2 dh + \int_{-\infty}^{\infty} var f(\hat{h}) dh] \tag{5}$$

where T represents the Taylor expansion, and we select the first four terms of the Taylor expansion, that is, the most important part.

The window width is certified according to Silverman's bandwidth [30]

$$h = 0.9 \times min(\hat{\sigma}, \frac{s}{1.34}) \times n^{-\frac{1}{5}} \tag{6}$$

where $\hat{\sigma}$ is the height sample's standard deviation, n is the sample size, and s is the sample's interquartile range.

2.5.2. Bar Line Criteria

The bar line has the largest relative height in the melody line. We add the minimum value of each line of symbols to the weighted estimate of the maximum and minimum symbols, which is μ, called the bar threshold.

$$\mu = min(H_i) + \text{peak}(-f(\hat{h})) \times \frac{max(H_i) - min(H_i)}{10k} \tag{7}$$

where k is the number of symbols in a line and the peak and $(-f(\hat{h}))$ is the negative peak value of $f(\hat{h})$.

Negative peaks describe the abruptness of the sign height. Because in the melody line the high degree of the mutation of bar lines and notes always exists, if there is no mutation, it must not be a melody line. At the same time, the mutation value describes the relative difference in height, and the threshold value can be obtained by adding the normalized value of the minimum value of the symbol and the maximum height difference of the symbol. We record the set of μ as S.

The melody line requires that S cannot be an empty set. The aspect ratio of the symbol with the highest relative height $W_{argmax_i H_i}$ conforms to the aspect ratio of the bar line. Furthermore, the number of bar lines $|S_{bar}|$ is reasonable, less than the number of notes.

$$\begin{cases} |S| \geq 1 \\ W_{argmax_i H_i} / H_i \leq 1/4 \\ |S_{bar}| \leq |S_{keys}| \end{cases} \tag{8}$$

The NMN line that satisfies the above conditions is the melody line, otherwise, it is the lyric line. In the subsequent numbered notation recognition process, our recognition object is the melody line.

2.6. Music Symbols Classification

Our proposed model is shown in Figure 7. The entire model is a structurally improved version of a CNN, which is specially prepared for the classification of musical symbols in melody lines. The characteristic of this model is that it adds a hierarchical analysis module and SPP layer, so we call it H-SPP-CNN.

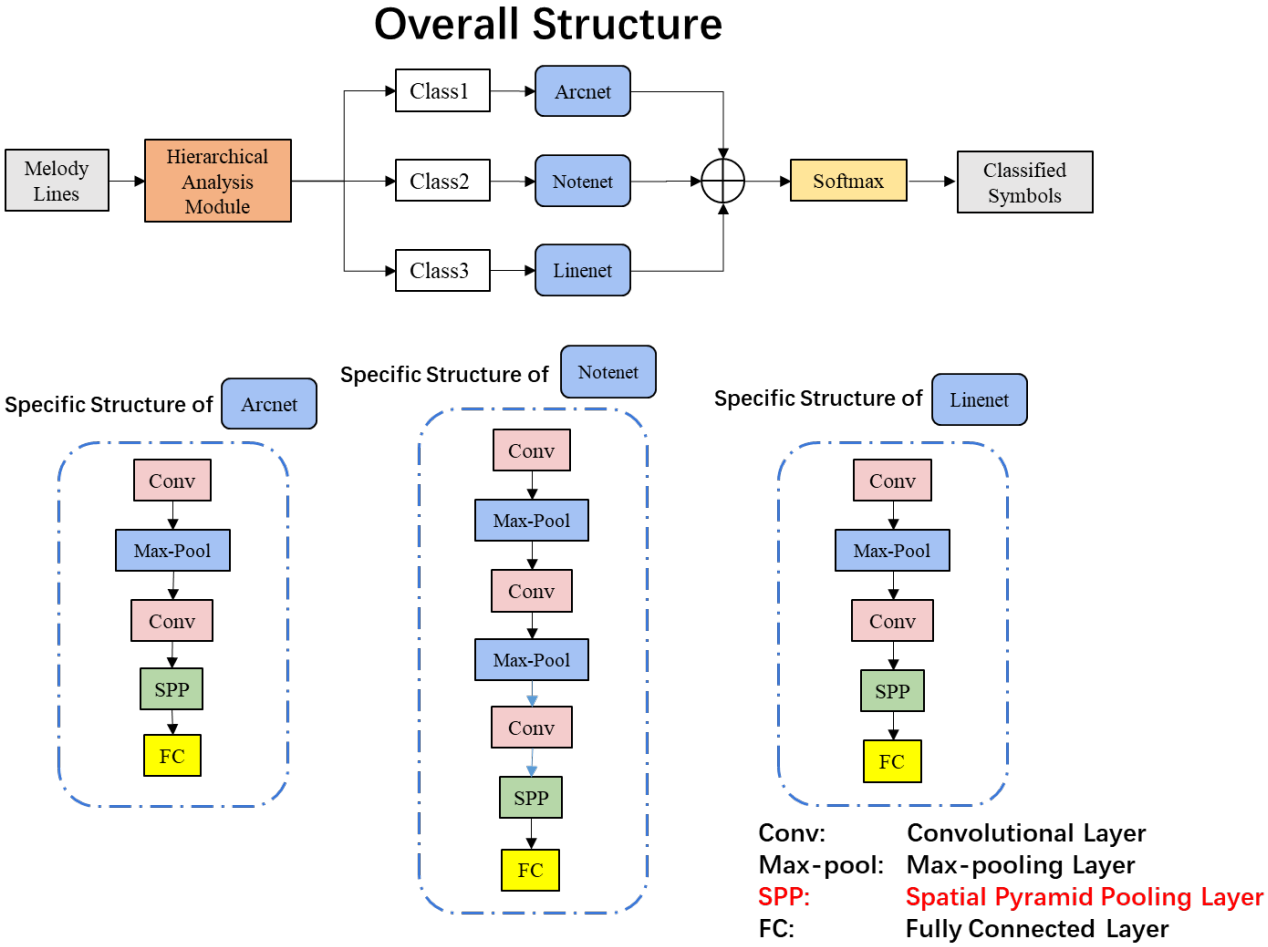


Figure 7. The structure of the proposed H-SPP-CNN.

To begin, the symbols in the melody lines are divided into three categories through the hierarchical analysis module (see Section 2.6.1). We name these symbols Class 1–3. Given that the majority of symbols in Class 1–3 are various kinds of arcs (a form of note length symbol), notes, and lines (see Figure 8 as an example), we design three parallel CNNs and name them Arcnet, Notenet, Linenet, respectively. The motivation of this design is that each symbol's size and style change too much to learn all the features with a simple improved CNN, but after the hierarchical analysis module, the internal differences of each type of symbol are smaller, and the training of "difference" in each type of symbol can ensure better model learning results. We summarize the recognition results of each CNN network with the softmax probability as the final recognition probability.

All melody lines of 10 fonts in the training set defined in Section 2.4 flow into the hierarchical module, which is the input entry of the improved CNN. The three parallel CNNs are trained independently, taking the hierarchically analyzed symbols as inputs. Every symbol is labeled in advance. The specific structure of each CNN is shown in Figure 7. The convolution kernel size of the convolution layer is five, the convolution kernel moving step size is one, and the pooling method selects max pooling. The number of layers for spatial pyramid pooling is set to three. The training epoch is set to 10, the batch size is set to 50, and the learning rate is set to 0.001. Using the Adam optimizer, the cross-entropy function is the loss function. The symbol recognition task in NMN is similar to a document recognition task. The symbols are small, and the color is black or white. Therefore, we refer to the optimal parameter setting method in the document recognition paper [23]. This parameter-setting method fits the documented scene and has a good learning effect. Notice that Notenet is deeper than Arcnet and Linenet, because Class 2 has slightly more symbolic features than the other two classes. All three CNNs have joined the SPP layer, which we will explain in Section 2.6.2.

2.6.1. Hierarchical Analysis Module

In contrast to staff notation, the hierarchical structure of the melody lines in NMN is more obvious. Jiang, Y. [14] realized the hierarchical definition and mathematical expression of symbols according to their position and size.

Note that the ordinate of the lower end of the symbol is y_a, the ordinate of the lower end of this line is y_b, and the height of the bar line is y_{sec}. The symbols can be divided into three classes.

Class 1

$$y_a - y_b \geq y_{sec} \times 2/3 \tag{9}$$

The first type of symbols include tuplets, repeated jump symbols, high-octave origin, and so on.

Class 2

$$y_a - y_b \leq y_{sec} \times 1/3 \tag{10}$$

The second category of symbols include symbols such as the lower octave origin, eighths, sixteenths, thirty-second note lines, etc.

Class 3

$$y_{sec} \times 1/3 + \epsilon \leq y_a - y_b \leq y_{sec} \times 2/3 - \epsilon \tag{11}$$

We set ϵ in Equation (11) to $imagewidth/1000$. Symbols that do not meet the conditions of the first and second types of symbols are the third type of symbols, such as ascending and descending signs, numbers, rests, brackets, and so on. Figure 8 is an example of the NMN melody line after hierarchical analysis.

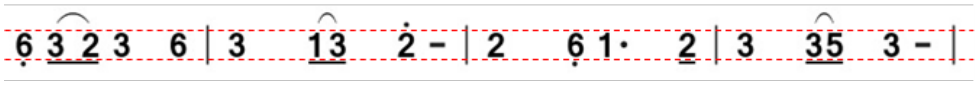


Figure 8. An example of NMN melody line after hierarchical analysis.

2.6.2. SPP Layer

Traditional CNN with a hierarchical analysis strategy requires fixed-size symbols and images. Artificial resizing may reduce recognition accuracy. Therefore, we add a spatial pyramid pooling (SPP) layer into the neural network, which can generate a fixed-length representation regardless of image size/scale [31].

The SPP layer is vivid, essentially maintaining a window, increasing it with a certain step size, pooling the convolutional features, and then increasing the side length. After adding the SPP layer, if the size of the input image changes, the vector size of the fully connected layer remains unchanged after mapping by the SPP layer so that the output result can be obtained smoothly. The SPP layer constructs the convolution kernel with the idea of multiplication. Let the multiplication object be Lv and the input image size be (h, w). Then, the size of the convolution kernel is:

$$\begin{cases} H_{kernel} = \lceil \frac{h}{Lv} \rceil \\ W_{kernel} = \lceil \frac{w}{Lv} \rceil \end{cases} \tag{12}$$

The padding area size is:

$$\begin{cases} H_{padding} = \lceil \frac{H_{kernel} \times n - h + 1}{2} \rceil \\ W_{padding} = \lceil \frac{W_{kernel} \times n - w + 1}{2} \rceil \end{cases} \tag{13}$$

where n is the number of pools. The parameters of the SPP layer are Lv in Equation (12) and the pooling type. Because the symbols are small, that is, the input size is not big, if the Lv is too high, the kernel will be too small. IF Lv is too low, the kernel will leave redundant information when it is too large to pool, so we set the value of Lv to 3. Notenet input symbols have no background characteristics, so pooling is better with max pooling [23].

In the CNN shown in Figure 7, the SPP layer is inserted after the convolutional layer. Although the input Notenet symbols vary in size, a fixed-size full-join vector (vector size set to 15) is output after the SPP layer, which solves the problem of different input sizes for NMNs based on mathematical ideas rather than manual operations.

The addition of the SPP layer means that the depth of the model increases and there is a possibility of overfitting. To solve the problem of overfitting, we use the dropout method [32]. In deep convolution networks, dropout can effectively alleviate the occurrence of overfitting and achieve a regularization effect.

2.7. Evaluation Metrics

The evaluation metric of the melody line and lyric line separation experiment is the recognition accuracy of the melody line. The number of melody lines of the NMN is denoted as N, and the accurately recognized melody behavior is denoted as M. Then, Melody recognition accuracy (MRA) is:

$$\text{MRA} = \frac{M}{N} \times 100\% \tag{14}$$

The evaluation metrics of the symbol classification experiment are classification accuracy. Let the total number of hierarchical $class_i$ (hierarchical classes are annotated in Figure 8) symbols be n_i, and the number of correct judgments be m_i. The accuracy is α_i, then:

$$\alpha_i = \frac{m_i}{n_i} \times 100\%. \tag{15}$$

After summarizing the internal recognition results of various symbols, we record the total recognition rate as α^*. Because the total number of symbols remains unchanged after hierarchical analysis, the total recognition rate is calculated as:

$$\alpha^* = \frac{\sum_{i=1}^{3} m_i}{\sum_{i=1}^{3} n_i} \times 100\% \tag{16}$$

All of the metrics above regard the probability of correct symbol recognition but do not analyze the probability of wrong symbol recognition. It is not objective to directly depict the wrongly identified symbols with a percentage of accuracy. For users, it is not objective enough to evaluate the recognition result of NMN only by the recognition rate of each symbol. The final recognition result of NMN is a reconstructed and encoded MIDI audio file of each melody symbol. We need to compare the similarity of the NMN's original audio file with the generated audio file from the user's point of view. The error value of the generated audio compared with the original audio is taken as the measurement standard of the number of false identifications.

We define several evaluation indicators of error [33], and the weight of each evaluation indicator is w_i, the overall number of notes is $|Notes|$:

1. High Impact Note

High impact notes (denoted as HN) are the most frequent notes in NMNs, such as the root note. Recognition error of high impact notes will affect the overall melody, so the weight of its evaluation is relatively large; we set it to be nine.

2. Low Impact Note

Low impact notes are less frequently occurring notes in NMN and notes with short duration. We set its weight to four and denote it as LN.

3. Duration of HNs

The duration of HNs is not too short, and the frequency is not too low. The duration of a quarter note can be used as a unit. We set its weight to five and record it as DHN.

4. Duration of LNs

Contrary to HNs, LNs should not appear too frequently and should not last as long. We take the time range of the 8th note as the unit, and set its weight to five, abbreviated as DLN.

The total error between the original audio and the generated audio is denoted as E, then:

$$E = (w_1 HN + w_2 LN + w_3 DHN + w_4 DLN)/|Notes| \tag{17}$$

We asked music experts in the project team and interviewed eight music major students. Expert scoring is conducted for our audio metrics to determine the upper bounds of weight and evaluation. The weight distribution has been given. For the design of the upper bounds, we can conclude that for audio with a speed of 80 bpm and a duration of more than 20 s and less than 30 s, the audio with an E value below 0.5 has a better recovery effect, and the audio with an E value below 0.3 has an excellent recovery effect and a very small error rate.

3. Results

The experiments were run on a computer with 11th Gen Intel Core i5-1135G7 @ 2.40 GHz quad-core CPU and 16 G memory, and the running platform was Anaconda Python 3.9.

3.1. NMN Line Segmentation Results

We randomly selected a line of melodies and lyrics from an NMN in the multi-font NMN dataset, and its height density function is shown in Figure 9.

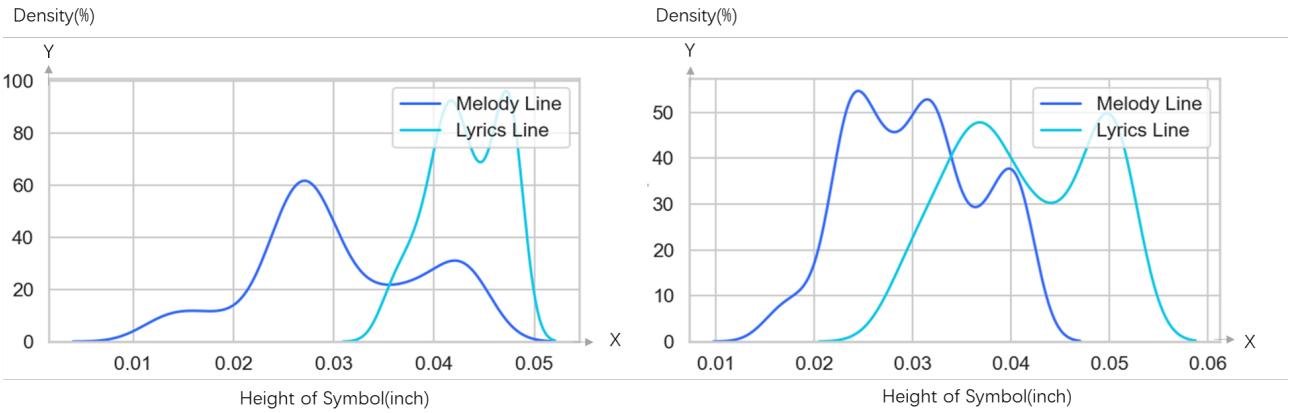


Figure 9. Height kernel density from a random NMN line in our dataset.

As we can see, symbols with high heights in melody lines have a greater density, and symbols with high heights in lyric lines have a greater density. This is because symbols with low heights, such as the midpoint and line of the melody line, occur frequently and there are very few such symbols in the lyric line. There are more symbols with relatively high heights in the lyric lines, but fewer symbols with high heights such as bars in the melody lines.

In this experiment, the baseline vertical projection method (baseline), the KDE with the bar line criteria method (KDE + BC) proposed in this paper, and the criteria without KDE (BC) were used to separate lyrics and melody.

The test data are all described in Section 2.3. The experimental subjects were 2 NMNs with the same font, 6 NMNs with 3 fonts, 12 NMNs with 6 fonts, and 20 NMNs with all the fonts in the test set (10 fonts). We refer to each subject by the number of fonts in the musical notation (1 font, 3 fonts, 6 fonts, 10 fonts). Table 2 reflects the MRA values for various cases.

Table 2. MRA of segmentation of melody lines and lyrics lines.

Methods	1 Font	3 Fonts	6 Fonts	10 Fonts
Baseline	97.9%	90.6%	88.5%	79.4%
KDE + BC	99.3%	92.7%	91.3%	89.5%
BC	97.3%	88.4%	87.5%	80.9%

Table 2 shows that when there was only one font, the MRAs of the three methods were all very large and remain above 97. When the number of NMN fonts increased, the MRA values of the three methods all decreased, but the MRA of the baseline method and BC method decreased significantly, showing a landslide decline. In the case with 10 fonts, they were 11.2 and 7.5 lower than that with 3 fonts, while the KDE+BC method dropped by 3.2, and the stability is much greater than the first two methods.

We performed deeper analysis in the 1 font experiment and tested all 10 types of fonts in the dataset. The MRA values are shown in Figure 10. The MRA of Microsoft JhengHei is the highest, while Vivaldi is the lowest due to its uncommon style. Therefore, the recommended font of printed NMN for OMR is Microsoft JhengHei.

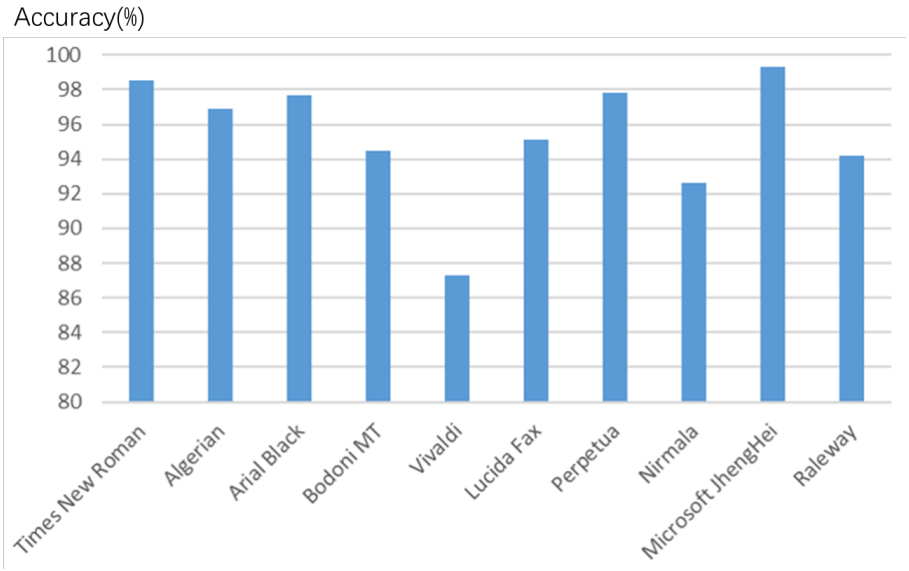


Figure 10. MRA in the condition of specific fonts.

3.2. Symbol Classification Results

We used 20 NMNs with all 10 fonts (defined as Batch 4 test set in Section 2.3) as symbol classification validation data. We compared the H-SPP-CNN proposed in this paper with the traditional CNN model. At the same time, it is compared with some improved CNNs, such as dilated CNN (DCNN) [34] and tiled CNN (TCNN) [35]. We adjusted the size of input symbols to be identical for training the TCNN, DCNN, and naive CNN. At the same time, we added the hierarchical analysis module (H-CNN, H-DCNN, H-TCNN) proposed in this paper in front of these classifications for comparative experiments. Models without a hierarchical analysis module are simply a single CNN classifier that uses all of the symbols in the melody lines without analyzing them in advance.

After adding the hierarchical analysis module, each classifier's learning characteristics were reduced, and thus the recognition rate was improved. After the traditional SPP-CNN model was added to the hierarchical analysis module, the recognition of bar lines increased by 15.1%, and the recognition rates of difficult-to-recognize numbers and lifting/reduction symbols increased by 10.2% and 5.5%, respectively.

We find that for symbols with few features and simple structures (such as dots), the recognition rate of the classifier is higher than for other symbols. Although the recognition rate of deep convolutional network methods such as DCNN and TCNN was improved to some degree, it only remained at around 80%. Although there were some improvements in many places such as the convolutional layers, the problem of feature extraction caused by different input image sizes has not been solved. We used the SPP-CNN model to solve this problem, making the recognition rate of dots and digital symbols greater than 90%.

The three parallel networks (Arcnet, Notenet and Linenet) were tested independently with the Batch 4 test set. First, we attempted to test the network without the hierarchical analysis module and used the CNN architecture and the SPP-CNN architecture. Then, we added the hierarchical analysis module to perform the comparative experiment. We combine the results of these three parallel networks to calculate the α^* in Equation (16).

As seen in Figure 11, after adding the SPP pooling layer, the recognition of each class and all symbols was increased, indicating that SPP-CNN is suitable for this recognition task with diverse fonts and different sizes. After adding the hierarchical analysis module proposed, the recognition rate of each class was further improved, indicating that the hierarchical idea in this paper is helpful for the accurate recognition of various symbols in NMN. Furthermore, the recognition rate of all symbols in Figure 10 is a reflection of the effect of the joined decision by concatenation of the results in each hierarchical class. The

decision helps compromise the result of each hierarchical class with a high recognition rate of over 95% in general.

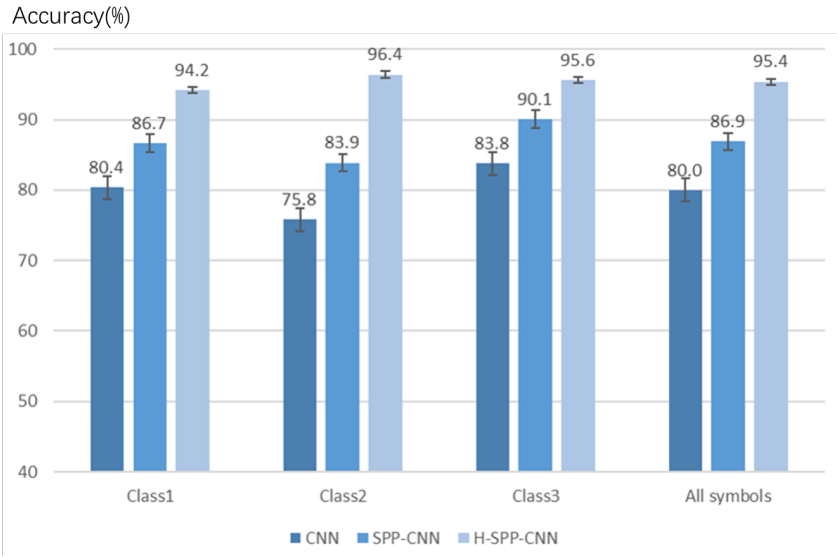


Figure 11. Recognition results of each class and all symbols.

3.3. Audio Evaluation

Five MIDI audio files (specifically explained in Section 2.3) generated by NMN were used as experimental objects. The experimental results are shown in Table 3.

Table 3. Audio evaluation results.

NMN Title	HN	LN	DHN	DLN	E
Dream	2.4%	0	4.8%	0	0.452
Secret	0	2.5%	2.5%	5.0%	0.475
Painter	0	4.4%	0	4.4%	0.4
Edelweiss	0	0	5.0%	0	0.25
Ten Years	0	2.0%	4.0%	0	0.275

It can be seen from the experimental results that although various types of errors rarely occur due to symbol recognition errors, there is no large-scale impact on the sense of hearing. According to the evaluation threshold specified by experts, the E value of all audio files is below 0.5, and the error recognition rate is up to standard. The E value of the two audio frequencies is less than 0.3, the error rate is very small, and the error caused by the error recognition has little impact on the overall hearing sense. This shows that the generated audio has high similarity to the original NMN audio, and the recognition result is apparently acceptable.

4. Discussions

In our present study, we focus on the recognition of multi-font NMN. A multi-font dataset was made for training and testing. We used kernel density estimation to perform unsupervised learning on the symbol height data according to the arrangement of symbols in melody lines and lyric lines. Based on the analysis, bar line criteria are proposed, which realize the classification of melody and lyric lines. In addition, we adopted the hierarchical spatial pyramid pooling convolutional neural network (H-SPP-CNN) to recognize the melody notes of diverse shapes and styles. The experimental results show that our line segmentation algorithm outperforms the baseline projection methods and single BC method. Our H-SPP-CNN exhibits higher recognition accuracy than other structurally improved CNNs. What is more, the recognition results are encoded into MIDI files. Results have proven a high similarity between the original music and generated MIDI files.

This shows that the method proposed in this paper can identify adequate music information in the musical notation and encode it into an audio file with a small error as the identification result.

The proposed KDE with BC method performs better because the baseline and BC methods do not use KDE to estimate the trough but indirectly or directly distinguish the melody line from the lyric line by counting and comparing the absolute value of the symbol height. This idea can obtain an accurate division basis when there is only one font, but in the case of diverse fonts, the relative value of the symbol height is no longer presented in a uniform data distribution. For example, Nirmala fonts and Lucida Fax fonts have different relative heights of numbers and bar lines, and discrete statistical methods are difficult to adapt to this situation. The KDE method can solve this problem. The estimated trough value can reflect the highly abrupt change. This feature always exists no matter how the font changes. After synthesizing this feature, the stability of the MRA value has been significantly improved.

The symbol classification results are optimized, because the H-SPP-CNN uses a combination of improvements at the micro and macro levels. At the macro level, multi-font NMN has many types of symbols and great changes in style, but the arrangement of symbols has certain rules and a clear sense of hierarchy. On the macroscopic basis of hierarchical analysis, employing three parallel CNNs for training and testing is theoretically superior to a single structurally improved CNN. The results in Table 4 and Figure 11 have proven our assumption. Therefore, at the macro level, we adopt the idea of "divide and conquer" to solve the identification problem. At the micro level, the main variation in the symbol images is in terms of style and size. Other improvement algorithms are more capable of capturing style features but cannot fit the extremely various input size. The SPP, which rarely occurs in document word classification, is right for solving this problem. So, there is an apparent increase in recognition accuracy Table 4.

Table 4. Symbol classification comparison test results.

	Symbols						
Method	**Bar**	**Lift/Restore**	**Tuplet**	**Length Marker**	**Cease**	**Dot**	**Number**
CNN	78.5%	80.1%	69.4%	75.3%	72.9%	85.1%	86.1%
TCNN	79.2%	80.4%	70.9%	76.1%	73.2%	86.4%	88.4%
DCNN	78.1%	83.7%	79.4%	76.8%	72.4%	88.2%	85.2%
SPP-CNN	80.2%	81.9%	86.7%	82.8%	79.6%	91.1%	92.1%
H-CNN	85.5%	89.9%	94.4%	90.3%	87.9%	90.6%	85.9%
H-TCNN	91.7%	89.3%	92.4%	88.5%	92.3%	93.9%	87.4%
H-DCNN	92.4%	90.9%	88.7.4%	94.5%	88.4%	92.6%	92.8%
H-SPP-CNN	95.3%	93.7%	96.9%	92.7%	94.1%	95.4%	97.6%

The macro and micro perspectives are integrated to make considerable progress in OMR types of research. However, our study had limitations, as our research focuses on printed NMN and the common symbols in them. In real life, NMN images are not so idealized. In the future, we will consider more objective changes in NMN images and go a step further.

Author Contributions: Conceptualization, Q.W. and X.C.; data curation, Q.W.; formal analysis, Q.W.; funding acquisition, L.Z. and X.C.; methodology, Q.W.; resources, L.Z.; writing—original draft, Q.W.; writing—review and editing, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Regular Projects of the Humanities and Social Sciences Fund of the Ministry of Education of Grant No.16YJAZH080.

Data Availability Statement: The generated audio files are available at https://github.com/Duoluoluos/Multi-font-NMN-Recognition-Results, accessed on 4 October 2022). Other data that support the finding of the study are available from the corresponding author upon reasonable request.

Acknowledgments: We extend our sincere appreciation to the editors and reviewers. Your suggestions contribute to the improvement of the paper's content.

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

References

1. Calvo-Zaragoza, J.; Hajič, J., Jr.; Pacha, A. Understanding optical music recognition. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–35. [CrossRef]
2. 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]
3. Novotný, J.; Pokorný, J. Introduction to Optical Music Recognition: Overview and Practical Challenges. In Proceedings of the DATESO, Grenoble, France, 9–13 March 2015; pp. 65–76.
4. Garrido-Munoz, C.; Rios-Vila, A.; Calvo-Zaragoza, J. A holistic approach for image-to-graph: Application to optical music recognition. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2022**, *2*, 1–11. [CrossRef]
5. Castellanos, F.J.; Garrido-Munoz, C.; Ríos-Vila, A.; Calvo-Zaragoza, J. Region-based Layout Analysis of Music Score Images. *arXiv* **2022**, arXiv:2201.04214.
6. Zheng, X.; Li, D.; Wang, L.; Zhu, Y.; Shen, L.; Gao, Y. Chinese folk music composition based on genetic algorithm. In Proceedings of the 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 9–10 February 2017; pp. 1–6.
7. Castellanos, F.J.; Gallego, A.J.; Calvo-Zaragoza, J. Unsupervised Domain Adaptation for Document Analysis of Music Score Images. Available online: https://archives.ismir.net/ismir2021/paper/000009.pdf (accessed on 4 October 2022).
8. Castellanos, F.J.; Gallego, A.J.; Calvo-Zaragoza, J.; Fujinaga, I. Domain adaptation for staff-region retrieval of music score images. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2022**, *5*, 1–12. [CrossRef]
9. Dalitz, C.; Droettboom, M.; Pranzas, B.; Fujinaga, I. A comparative study of staff removal algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 753–766. [CrossRef] [PubMed]
10. 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]
11. Baro, 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.
12. Mas-Candela, E.; Alfaro-Contreras, M.; Calvo-Zaragoza, J. Sequential Next-Symbol Prediction for Optical Music Recognition. In *International Conference on Document Analysis and Recognition*; Springer: Cham, Swizerland, 2021; pp. 708–722.
13. Deng, X.Y.; Yang, Y.H. Segmentation, Tilt Correction and Note Lyrics Extraction of Paper Numbered Musical Notation Images. *Acta Electonica Sin.* **2021**, *49*, 716.
14. Jiang, Y. Research on the Recognition Method of Numeral Notation. Master's Thesis, Zhejiang University, Hangzhou, China, 2006. Available online: https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CMFD0506&filename=2006033333.nh (accessed on 15 May 2006).
15. Min, D. Research on numbered musical notation recognition and performance in a intelligent system. In Proceedings of the 2011 International Conference on Business Management and Electronic Information, Guangzhou, China, 13–15 May 2011; Volume 1, pp. 340–343.
16. Wu, F.H.F. Applying Machine Learning in Optical Music Recognition of Numbered Music Notation. In *Cognitive Analytics: Concepts, Methodologies, Tools, and Applications*; IGI Global: Hsinchu, Taiwan, 2020; pp. 1915–1937.
17. Weglarczyk, S. Kernel density estimation and its application. In *ITM Web of Conferences*; EDP Sciences: Warszawska, Poland, 2018; Volume 23, p. 00037.
18. Lin, F.; Zhang, X.; Ma, Z.; Zhang, Y. Spatial Structure and Corridor Construction of Intangible Cultural Heritage: A Case Study of the Ming Great Wall. *Land* **2022**, *11*, 1478. [CrossRef]
19. Kisley, M.; Qin, Y.J.; Zabludoff, A.; Barnard, K.; Ko, C.L. Classifying Astronomical Transients Using Only Host Galaxy Photometry. *arXiv* **2022**, arXiv:2209.02784.
20. Kamalov, F.; Moussa, S.; Avante, R.J. KDE-Based Ensemble Learning for Imbalanced Data. *Electronics* **2022**, *11*, 2703. [CrossRef]
21. Jindal, A.; Ghosh, R. Text line segmentation in indian ancient handwritten documents using faster R-CNN. *Multimed. Tools Appl.* **2022**, 1–20. [CrossRef]
22. Yin, W.; Schütze, H. Multichannel variable-size convolution for sentence classification. *arXiv* **2016**, arXiv:1603.04513.
23. Chen, Y. Convolutional Neural Network for Sentence Classification. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2015.
24. Boretz, B. Meta-variations: Studies in the foundations of musical thought (I). *Perspect. New Music.* **1969**, *8*, 1–74. [CrossRef]

25. Suyanto, Y. Numbered Musical Notation and Latex Document Integration. In Proceedings of the 2018 4th International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, 7–8 August 2018; Volume 11, pp. 1–6.
26. Marinai, S.; Nesi, P. Projection based segmentation of musical sheets. In Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR'99 (Cat. No. PR00318), Bangalore, India, 22 September 1999; pp. 515–518.
27. Foley, J.D.; Van Dam, A.; Feiner, S.K.; Hughes, J.F.; Phillips, R.L. *Introduction to Computer Graphics*; Addison-Wesley: Amsterdam, The Netherlands, 2022; Volume 55.
28. Wang, D.; Fang, Y.; Huang, S. An algorithm for medical imaging identification based on edge detection and seed filling. In Proceedings of the 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, China, 22–24 October 2010; Volume 15, pp. V15-547–V15-548.
29. Rebelo, A.; Paszkiewicz, F.; Guedes, C.; Marcal, A.R.; Cardoso, J.S. A method for music symbols extraction based on musical rules. In Proceedings of the Bridges 2011: Mathematical Connections in Art, Music, and Science, Coimbra, Portugal, 27–31 July 2011; Volume 14, pp. 81–88.
30. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; Routledge: New York, NY, USA, 2018.
31. 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]
32. Park, S.; Kwak, N. Analysis on the dropout effect in convolutional neural networks. In *Asian Conference on Computer Vision*; Springer: Cham, Swizerland, 2016; pp. 189–204.
33. Velankar, M.R.; Sahasrabuddhe, H.V.; Kulkarni, P.A. Modeling melody similarity using music synthesis and perception. *Procedia Comput. Sci.* **2015**, *45*, 728–735. [CrossRef]
34. Li, Y.; Zhang, X.; Chen, D. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1091–1100.
35. Ngiam, J.; Chen, Z.; Chia, D.; Koh, P.; Le, Q.; Ng, A. Tiled convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2010**, *23*, 10–24.