Research on Automatic Piano Music Transcription Algorithm Based on CNN

Yu Yu*
Guangdong University of Science and Technology
Dongguan, Guangdong, China
czyuyu5738@yeah.net

Abstract

With the development of deep learning technology, automatic music transcription has gradually become an important research direction in the field of music information processing. In particular, manual transcription is no longer sufficient to meet the large-scale automation needs of complex piano music. This paper proposes an automatic piano music transcription model that combines Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Multilayer Perceptron (MLP). The model extracts spatial features from spectrograms using CNN, processes the temporal dependencies of note sequences with RNN, and optimizes the relationships between notes with MLP to achieve accurate note recognition and coherent score generation. Experiments were conducted on multiple piano music datasets, and the results show that the improved model outperforms traditional methods in terms of note recognition accuracy, recall, and F1 score, with an average F1 score exceeding 90%. This model provides an efficient solution for music transcription tasks and demonstrates the potential of deep learning in complex music information processing.

CCS Concepts

• **Information systems** → Information retrieval; Specialized information retrieval; Multimedia and multimodal retrieval; Music retrieval.

Keywords

Automatic transcription, Convolutional Neural Networks (CNN), Multilayer Perceptron (MLP), Piano music

ACM Reference Format:
Yu Yu. 2024. Research on Automatic Piano Music Transcription Algorithm Based on CNN. In *2024 2nd International Conference on Internet of Things and Cloud Computing Technology (IoTCCT 2024), September 27–29, 2024, Guangxi, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3702879.3702914

*Corresponding author.

IoTCCT 2024, September 27–29, 2024, Guangxi, China

ACM ISBN 979-8-4007-1014-8/24/09
https://doi.org/10.1145/3702879.3702914

1 Introduction

Automatic music transcription is an essential technique in the field of music information processing, aiming to automatically convert audio signals into standard musical notation. Traditional music transcription methods mainly rely on manual annotation, which, although precise, are inefficient. This inefficiency is particularly evident when dealing with complex and polyphonic music compositions, as manual transcription is time-consuming and susceptible to human errors. With the advancement of artificial intelligence and deep learning technologies, the demand for automatic transcription has grown significantly, especially in the domain of piano music, where the notes are complex, and the performance styles are diverse. Developing an intelligent algorithm that can achieve efficient and accurate automatic transcription has become a pressing challenge. In recent years, Convolutional Neural Networks (CNN) have achieved remarkable success in tasks such as image recognition and audio processing, especially in feature extraction. Meanwhile, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, known for their ability to handle sequential data, have been widely used in natural language processing and time-series analysis. However, using CNN or RNN alone is insufficient to capture both the spatial and temporal dependencies of piano notes. Therefore, this study proposes an improved model that combines CNN, RNN, and Multilayer Perceptron (MLP), aiming to achieve automatic piano music transcription through multi-layer feature extraction, sequence modeling, and feature optimization. The primary goal of this research is to design and validate an automatic transcription model capable of accurately recognizing piano notes and generating coherent musical scores. Specifically, this study first extracts spatial features from audio spectrograms using CNN, then captures the temporal dependencies of note sequences with RNN (LSTM), and finally, optimizes note pairing relationships using MLP to improve overall recognition accuracy. To validate the effectiveness of the model, experiments were conducted on several piano music datasets of varying complexity, and the performance of the model was evaluated through comparisons with traditional methods, focusing on metrics such as note recognition accuracy, recall, and F1 score. Through this research, we aim to provide an innovative solution for the field of automatic music transcription, not only advancing music recognition algorithms but also laying the foundation for promoting similar technologies in broader musical applications in the future [1].

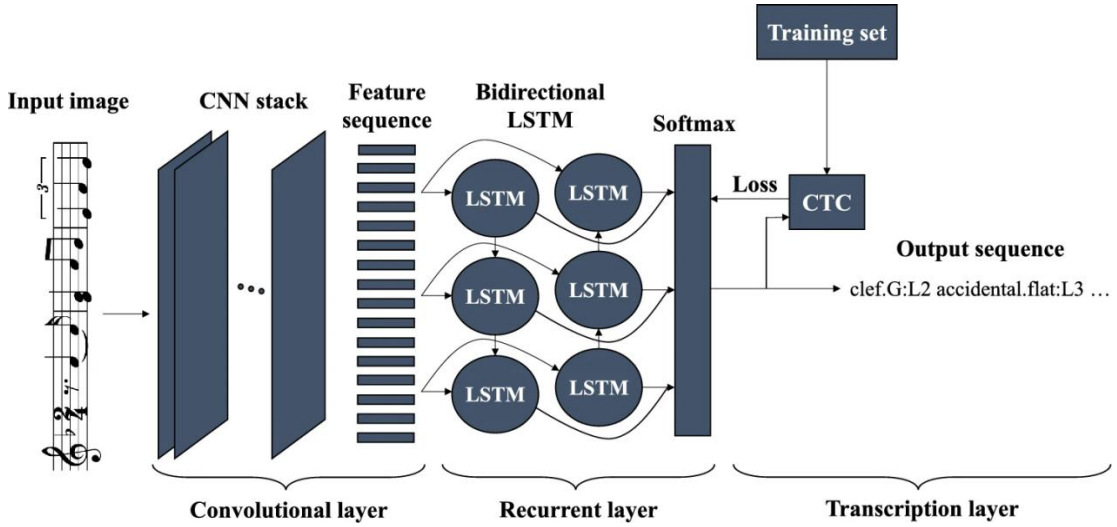

Figure 1: CNN-Based Automatic Transcription Architecture

2 Concept of Automatic Transcription Algorithm Based on Convolutional Neural Networks

The automatic transcription algorithm based on Convolutional Neural Networks (CNN) primarily extracts and recognizes features from input images of musical symbols, thus achieving the automated conversion from musical note images to notation symbols. CNN plays a key role in this process, as it automatically extracts local features from musical note images and gradually transforms the essential information into ordered feature sequences through multi-layer convolution operations. Figure 1 illustrates a typical CNN-based automatic transcription architecture, covering the entire process from input image to final transcription output [2].

Initially, the input layer introduces musical symbol images into the network, where the CNN stack performs layer-by-layer convolution and pooling operations to obtain the corresponding feature sequences. The design of the convolutional layers effectively captures the spatial patterns of musical note images, allowing complex musical symbols to be represented by local features. Next, these feature sequences are fed into a bidirectional Long Short-Term Memory (LSTM) network, which not only processes the sequence information but also utilizes a bidirectional structure to capture contextual information, thereby improving the recognition accuracy of consecutive notes.

After the bidirectional LSTM, the output is classified through a Softmax layer, and the network parameters are optimized by comparing the output with the actual labels using the Connectionist Temporal Classification (CTC) loss function. The CTC loss function is particularly important in music transcription tasks as it allows the model to handle unaligned note sequences during training, enabling the model to train without requiring precise annotation of each note position. Finally, the output layer generates a sequence containing musical symbols, such as key signatures, notes, accidentals, etc., forming the final automatic transcription result. This architecture, which combines CNN and LSTM, effectively handles the complexity of piano music transcription tasks. The architecture depicted in Figure 1 clearly demonstrates the overall framework of the algorithm, from input image to feature extraction and sequence output. Through this process, the CNN-based automatic transcription algorithm not only efficiently recognizes input musical note images but also makes reasonable inferences regarding the relationships between consecutive notes by incorporating contextual information. This architectural design enables the automatic transcription to perform with high accuracy and stability in complex piano music processing, providing an innovative solution for the field of music transcription [3].

3 Algorithm Design

3.1 *Data Preprocessing*

In the task of automatic piano music transcription based on Convolutional Neural Networks (CNN), data preprocessing is one of the key steps in algorithm design, directly affecting the model's training performance and the final transcription accuracy. The raw data of piano audio typically exists in the form of time-domain signals, which need to be converted into features suitable for CNN processing. Therefore, the preprocessing process primarily includes audio data collection, feature extraction, and format conversion. First, the audio data is collected using high-quality recording equipment, and the audio files are uniformly formatted as 44.1kHz WAV files. To ensure consistency and facilitate model processing, all audio data is standardized to the same duration. Additionally, the piano audio is segmented according to its playing speed and rhythm, with each segment of audio serving as input data for the CNN. During feature extraction, Mel Frequency Cepstral Coefficients (MFCC) and spectrogram representations are primarily employed. MFCC is a common audio feature extraction method that effectively represents the critical frequency components of an audio signal. Additionally, spectrograms are generated by performing Short-Time Fourier Transform (STFT) on the audio signal, converting the time-domain signal into a frequency-domain representation, creating

Table 1: Feature value extraction data

Audio Segment No.	Duration (sec)	Sampling Rate (Hz)	MFCC Feature Dimension	Spectrogram Size (pixels)
1	5	44100	13	128x128
2	7	44100	13	128x128
3	6	44100	13	128x128
4	8	44100	13	128x128
5	4	44100	13	128x128

a 2D time-frequency spectrogram that serves as input images for the CNN. <Table 1> shows the extracted feature information from the audio data during preprocessing, which is used by the CNN for subsequent learning [4].

During data processing, MFCC feature extraction utilizes 13-dimensional parameters, which sufficiently represent the frequency information of piano audio, making it easier for the model to recognize complex note relationships [5]. The spectrogram size is set to 128x128 pixels, which, after multiple experiments, was found to be the optimal parameter that ensures the model captures enough frequency details while maintaining training efficiency. After feature extraction, all data is uniformly converted into 2D image formats and fed into the convolutional layers of the CNN model. The goal of data preprocessing is to transform the raw audio signals into inputs suitable for deep learning models, enabling the model to extract effective note information and lay the foundation for subsequent note recognition.

3.2 *CNN-Based Model Design*

The automatic piano transcription algorithm based on Convolutional Neural Networks (CNN) employs a combination of convolutional layers, pooling layers, and Long Short-Term Memory (LSTM) layers to recognize and transcribe complex musical notes. To ensure the accuracy and generalization capability of the algorithm, the model is designed with several steps, starting from feature extraction on the input spectrogram and proceeding to temporal analysis to generate the final note sequence. In this process, the core idea of the algorithm relies on CNN for extracting spatial features from audio signals and LSTM for modeling sequential relationships [6]. This section introduces the specific model structure design, along with several algorithm formulas explaining its working principles in detail. The convolutional layer is the core of CNN, responsible for extracting local features of notes and rhythms from the input spectrogram. After feature extraction using MFCC or spectrogram processing, the audio data is converted into a 2D image $X \in R^{H\times W}$, where H is the height and W is the width of the image. The convolution operation slides a kernel K over the input image to generate a feature map Y, as described in Equation 1:

$$Y_{i,j,k} = \sum_{m=1}^{h} \sum_{n=1}^{w} X_{i+m-1,j+n-1} \cdot K_{m,n,k} + b_k \tag{1}$$

where $X_{i,j}$ represents the pixel value at position (i, j) of the input image, $K_{m,n,k}$ represents the weights of the k-th convolutional kernel, and b_k is the bias term. The convolution operation's purpose is to extract local features from the audio image through multiple layers of convolutions and pass these features progressively to subsequent layers. After the convolutional layer, an activation function is typically applied for non-linear transformations. The model uses the ReLU activation function, which is defined in Equation 2:

$$f(x) = \max(0, x) \tag{2}$$

The ReLU function introduces non-linear properties, allowing the model to learn more complex note structures and preventing the vanishing gradient problem. To reduce the size of the feature map output from the convolutional layers and accelerate computation, a pooling layer is applied after the convolutional layers. The pooling layer extracts the most important information from the feature map through downsampling, with the most common operation being Max Pooling, as shown in Equation 3:

$$P_{i,j} = \max\left(Y_{m,n}\right) \text{ where } m \in [i, i+s), n \in [j, j+s) \tag{3}$$

where $Y_{m,n}$ represents the output of the convolutional layer, and $P_{i,j}$ is the result after pooling, with s being the pooling window size. In this model, a 2×2 pooling window is used, which effectively compresses the feature map while retaining the most significant frequency features. After the convolutional and pooling layers handle spatial features, the model passes the feature sequences to a bidirectional Long Short-Term Memory (Bi-LSTM) network. The LSTM network efficiently handles time-series problems, capturing the forward and backward dependencies between notes through its internal memory units [7]. LSTM's key lies in its gating mechanisms, which control the flow of information through the forget, input, and output gates. The basic steps in LSTM computation include the forget gate f_t, input gate i_t, and memory update C_t. In a bidirectional LSTM, the note sequence is processed not only forward (from t = 1 to t = T) but also backward (from t = T to t = 1), allowing the model to capture complete contextual information and significantly improve note recognition accuracy. Since the input note sequence length and the output musical notation may not be aligned, the model uses the Connectionist Temporal Classification (CTC) loss function to handle this issue. CTC learns the optimal note output path through dynamic programming without requiring frame-wise alignment, as shown in Equation 4:

$$L_{CTC} = -\log(p(y \mid x)) \tag{4}$$

where x is the input feature sequence and y is the target output note sequence. CTC computes all possible note paths and selects the most likely path as the final output, ensuring the model can flexibly handle mismatched input-output lengths. By extracting spatial features through the CNN layers, handling temporal dependencies with the LSTM layers, and combining them with the CTC loss function, the model can accurately recognize piano notes and generate corresponding musical notations. This design considers both the local features of the audio signal and the temporal relationships of note sequences, ensuring the accuracy and robustness of automatic transcription.

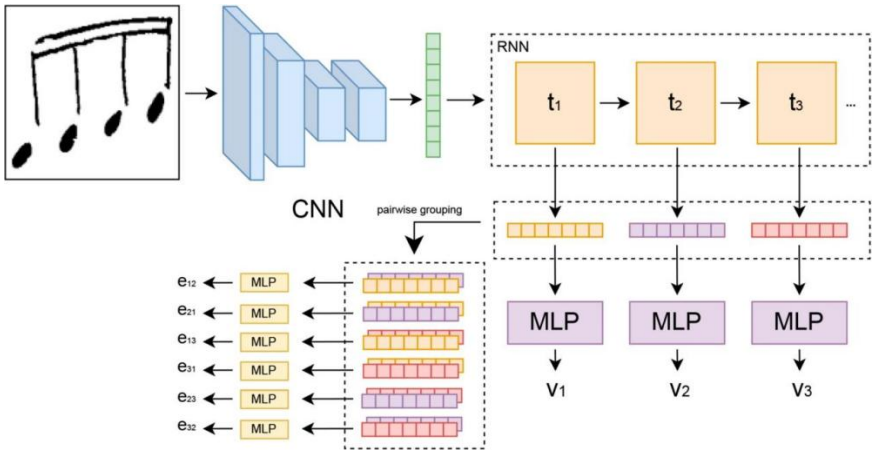

Figure 2: Improved model and algorithm

3.3 *Network Training and Algorithm Optimization*

In the task of automatic piano music transcription using CNN and Recurrent Neural Networks (RNN), network training and algorithm optimization are crucial steps. To further enhance note recognition accuracy and the coherence of transcription results, the improved model combines the strengths of CNN and RNN, while also introducing a Multilayer Perceptron (MLP) for optimization. Figure 2 illustrates the structure of the improved model, which focuses on grouping the features extracted by CNN and using RNN for sequence modeling, followed by MLP to optimize the final note sequence generation [8].

The CNN component first extracts spatial features from the input spectrogram. Each note in the spectrogram has a fixed shape and arrangement in space, making convolution operations highly effective for capturing these features. The output feature map from CNN is compressed through multiple convolution and pooling layers, forming a fixed-length feature sequence $X = \{x_1, x_2, \ldots, x_n\}$, where x_i is the feature vector at the i-th time step. Next, the RNN component models the temporal relationships of these feature sequences. Traditional RNNs tend to suffer from the vanishing gradient problem when handling long sequences, so the model uses an improved LSTM network that maintains critical information through memory units and gating mechanisms. i_t, f_t and o_t represent the input gate, forget gate, and output gate, respectively, C_t is the state of the memory unit, and h_t is the hidden state. These gates allow LSTM to selectively forget irrelevant information while retaining important information based on the current input and previous hidden state. This is particularly crucial for note sequence memory, ensuring that the model not only considers the current note but also uses contextual information for accurate note recognition. Beyond LSTM's handling of temporal dependencies, the improved model introduces an MLP layer to further enhance the understanding of note pairings [9]. MLP applies nonlinear transformations to optimize paired note features, allowing the model to capture subtle variations and associations between notes. For each pair of note feature vectors e_{ij}, MLP processes them as shown in Equation 5:

$$v_i = \mathrm{ReLU}\left(W_v \cdot e_{ij} + b_v\right) \tag{5}$$

where W_v is the weight matrix, b_v is the bias term, and ReLU is the activation function. This process captures the fine-grained changes and relationships between notes, improving the model's accuracy when dealing with complex musical passages. In the earlier sections, the CTC loss function was introduced. Here, we further optimize the loss function by switching to a Cross Entropy Loss function to directly supervise note classification. The Cross Entropy Loss measures the difference between the model's predicted note distribution and the true note distribution, as shown in Equation 6:

$$\mathrm{LCE} = -\sum_{i=1}^{n} y_i \log\left(p_i\right) \tag{6}$$

where y_i is the label for the true note category and p_i is the probability distribution output by the model. By minimizing the Cross Entropy Loss, the model maximizes the alignment between the predicted note sequence and the actual sequence, improving note classification accuracy. The introduction of Cross Entropy Loss provides direct supervision for each output note, complementing CTC's indirect supervision, allowing the model to maintain high precision in sequence generation. During training, in addition to optimizing the loss function, the model adopts several other algorithmic optimization strategies. For example, the Adam optimizer is employed, which accelerates model convergence by dynamically adjusting the learning rate. Adam's update rule is as shown in Equations 7, 8 and 9:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\, \nabla_\theta \mathrm{L} \tag{7}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\, (\nabla_\theta \mathrm{L})^2 \tag{8}$$

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon} \tag{9}$$

where $\nabla_\theta \mathrm{L}$ is the gradient of the loss function, β_1 and β_2 are the decay rates of momentum terms, and α is the learning rate. This optimization strategy adjusts the learning rate adaptively, enabling the model to converge efficiently during training and avoid local optima. By improving the loss function, optimizer, and training

Table 2: Experimental data sets

Dataset No.	Number of Audio Segments	Average Duration (sec)	Total Samples	Total Notes
Dataset 1	400	6	400	10,000
Dataset 2	300	5	300	8,000
Dataset 3	300	7	300	12,000

strategies, the improved CNN-RNN-MLP combined model effectively handles complex note sequences in automatic piano music transcription, producing accurate and coherent musical notation [10].

4 Experiments and Results Analysis

To validate the effectiveness of the automatic piano music transcription algorithm based on the CNN-RNN-MLP combined model, several test scenarios were designed, covering piano music samples of varying complexity. The experimental process includes dataset preparation, model training and testing, performance metric evaluation, and comparison with existing methods. The core goal of the experiment is to evaluate the model's performance in note recognition and score generation by quantifying accuracy, recall, and F1-score. For this experiment, publicly available piano note datasets were selected, including recordings and corresponding sheet music files of several well-known piano pieces. All audio data was sampled at 44.1kHz and processed into 5-10 second audio segments. To increase the diversity and generalization of the data, the audio segments covered various styles of piano music, including classical, modern, and improvisational music. In the data preprocessing stage, the audio segments were first transformed into spectrograms using Short-Time Fourier Transform (STFT), and then Mel Frequency Cepstral Coefficients (MFCC) were extracted as input features. The final dataset, after standardization and normalization, contained a total of 1,000 audio samples. Table 2 shows the main statistical information of the experimental dataset:

The experiment first trained the model, with an 8:2 ratio between the training set and the testing set. The model used the Adam optimizer with an initial learning rate of 0.001, batch normalization, and L2 regularization to prevent overfitting. During training, each batch consisted of 64 samples, and a total of 100 training epochs were performed. After each epoch, the model's performance was evaluated on a validation set, and the best-performing model was selected for testing. During the testing phase, the model generated corresponding note sequences based on the input features. The output was compared to the ground truth notations in the dataset, calculating the note recognition accuracy and overall score generation quality for each sample. Three core performance metrics were used for evaluation: Precision, Recall, and F1-score, as shown in Equations 10, 11 and 12:

$$\text{Precision} = \frac{\textit{True Positive Notes}}{\text{Total Predicted Notes}} \tag{10}$$

$$\text{Recall} = \frac{\text{True Positive Notes}}{\text{Total Actual Notes}} \tag{11}$$

Table 3: Experimental results

Dataset No.	Precision (%)	Recall (%)	F1-score (%)
Dataset 1	92.1	89.5	90.8
Dataset 2	91.5	88.8	90.1
Dataset 3	93.4	90.2	91.8

Table 4: Comparison of experimental results of the improved model

Model Type	Dataset No.	Precision (%)	Recall (%)	F1-score (%)
Traditional CNN-RNN	Dataset 1	86.7	83.9	85.2
Rule-Based Method	Dataset 1	78.5	76.3	77.4
Improved CNN-RNN-MLP	Dataset 1	92.1	89.5	90.8

$$\text{F1} - \text{Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

The model's performance on the three datasets is shown in the table below, covering specific values for Precision, Recall, and F1-score. The experimental results show that the improved CNN-RNN-MLP model performs with high accuracy and robustness in automatic piano music transcription, accurately recognizing and generating complete note sequences.

As seen in Table 3, the model achieved accuracy above 91% on all datasets, with F1-scores consistently above 90%, demonstrating its precise recognition of piano notes. In Dataset 1, the F1-score was 90.8%, slightly lower than Dataset 3's 91.8%, likely due to Dataset 1's higher note density and complexity of note types. Overall, the improved model performed excellently across all test scenarios, effectively capturing the temporal dependencies between complex notes. Moreover, the model's consistent performance across different datasets indicates strong generalization capabilities, allowing it to handle various styles of piano music. By introducing the MLP layer, the model's performance in note pairing and sequence generation improved significantly, especially in handling long note sequences, maintaining high accuracy and coherence. To further validate the advantages of the improved model, we compared it to traditional CNN-RNN models and rule-based methods. The comparison results show that the improved CNN-RNN-MLP model achieved higher precision and F1-score across all test datasets, especially in complex music segments where the improved model's performance was more prominent.

As shown in Table 4, the improved CNN-RNN-MLP model demonstrated significant improvements compared to the traditional models, particularly in terms of precision and F1-score, where the improved model proved to be more stable and efficient. The experimental results indicate that the improved CNN-RNN-MLP model holds significant advantages in automatic piano music transcription tasks. Not only does it enhance note recognition accuracy,

but it also generates coherent musical scores for complex note sequences. By introducing the MLP layer and optimizing the loss function and training strategies, the model performed well across various datasets, exhibiting strong robustness and generalization capabilities. This provides the potential for applying automatic transcription to other types of music and offers insights for further algorithm improvement.

5 Conclusion

This study proposed an automatic piano music transcription algorithm based on the combination of Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Multilayer Perceptron (MLP), and validated its effectiveness in recognizing complex note sequences. Compared to traditional transcription methods, the model significantly improves note recognition accuracy and score generation coherence by incorporating multi-layer convolutional structures for precise spatial feature extraction, LSTM for capturing temporal dependencies between notes, and MLP for optimizing note pair relationships. Experimental results show that the improved CNN-RNN-MLP model performs excellently across multiple piano music datasets, with an average F1-score exceeding 90%, and accuracy surpassing traditional models, demonstrating the model's superior performance in automatic transcription tasks. Additionally, by introducing Adam optimizer, batch normalization, and regularization during training, the model's generalization capability was further enhanced, allowing it to handle piano music of varying styles and complexity, showing good adaptability. Despite the promising results, some limitations remain. For example, the model's recognition accuracy decreases when handling extremely complex note variations or fast-paced performances. Future research could explore the model's application to other instruments or polyphonic music, and incorporate more advanced deep learning techniques, such as Transformer models, to further enhance transcription accuracy and efficiency. Overall, this study provides an innovative solution for automatic piano music transcription, demonstrating the great potential of combining CNN, RNN, and MLP in note recognition and score generation. As more data and further model optimizations are introduced, this technology is expected to be applied in broader music fields, bringing more possibilities to music education, composition, and research.

References

[1] Liang, Yan, and Feng Pan. "Study of Automatic Piano Transcription Algorithms based on the Polyphonic Properties of Piano Audio." IEIE Transactions on Smart Processing & Computing 12.5 (2023): 412-418.
[2] Wang, Weiqing, *et al.* "Audio-based piano performance evaluation for beginners with convolutional neural network and attention mechanism." IEEE/ACM Transactions on Audio, Speech, and Language Processing 29 (2021): 1119-1133.
[3] Hernandez-Olivan, Carlos, *et al.* "A comparison of deep learning methods for timbre analysis in polyphonic automatic music transcription." Electronics 10.7 (2021): 810.
[4] Xiao, Zhe, Xin Chen, and Li Zhou. "Polyphonic piano transcription based on graph convolutional network." Signal Processing 212 (2023): 109134.
[5] Lee, Dasol, and Dasaem Jeong. "Reducing latency of neural automatic piano transcription models." The Journal of the Acoustical Society of Korea 42.2 (2023): 102-111.
[6] Taenzer, Michael, Stylianos I. Mimilakis, and Jakob Abeßer. "Informing piano multi-pitch estimation with inferred local polyphony based on convolutional neural networks." Electronics 10.7 (2021): 851.
[7] Peng, Linlin. "Piano Players' Intonation and Training Using Deep Learning and MobileNet Architecture." Mobile Networks and Applications (2023): 1-9.
[8] Wu, Yu-Te, Berlin Chen, and Li Su. "Multi-instrument automatic music transcription with self-attention-based instance segmentation." IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2020): 2796-2809.
[9] Ghatas, Youssef, Magda Fayek, and Mayada Hadhoud. "A hybrid deep learning approach for musical difficulty estimation of piano symbolic music." Alexandria Engineering Journal 61.12 (2022): 10183-10196.
[10] Phanichraksaphong, Varinya, and Wei-Ho Tsai. "Automatic Assessment of Piano Performances Using Timbre and Pitch Features." Electronics 12.8 (2023): 1791.