applied sciences

MDPI

Article

Korean Pansori Vocal Note Transcription Using Attention-Based Segmentation and Viterbi Decoding

Bhuwan Bhattarai and Joonwhoan Lee *

Center for Advanced Image and Information Technology, Jeonbuk National University, Jeonju 54896, Republic of Korea; bhubon240@gmail.com
* Correspondence: chlee@jbnu.ac.kr

Abstract: In this paper, first, we delved into the experiment by comparing various attention mechanisms in the semantic pixel-wise segmentation framework to perform frame-level transcription tasks. Second, the Viterbi algorithm was utilized by transferring the knowledge of the frame-level transcription model to obtain the vocal notes of Korean Pansori. We considered a semantic pixel-wise segmentation framework for frame-level transcription as the source task and a Viterbi algorithm-based Korean Pansori note-level transcription as the target task. The primary goal of this paper was to transcribe the vocal notes of Pansori music, a traditional Korean art form. To achieve this goal, the initial step involved conducting the experiments with the source task, where a trained model was employed for vocal melody extraction. To achieve the desired vocal note transcription for the target task, the Viterbi algorithm was utilized with the frame-level transcription model. By leveraging this approach, we sought to accurately transcribe the vocal notes present in Pansori performances. The effectiveness of our attention-based segmentation methods for frame-level transcription in the source task has been compared with various algorithms using the vocal melody task of the MedleyDB dataset, enabling us to measure the voicing recall, voicing false alarm, raw pitch accuracy, raw chroma accuracy, and overall accuracy. The results of our experiments highlight the significance of attention mechanisms for enhancing the performance of frame-level music transcription models. We also conducted a visual and subjective comparison to evaluate the results of the target task for vocal note transcription. Since there was no ground truth vocal note for Pansori, this analysis provides valuable insights into the preservation and appreciation of this culturally rich art form.

Keywords: frame-level transcription; vocal transcription; Pansori music; attention mechanism; deep learning; Viterbi decoding

Citation: Bhattarai, B.; Lee, J. Korean Pansori Vocal Note Transcription Using Attention-Based Segmentation and Viterbi Decoding. *Appl. Sci.* **2024**, *14*, 492. https://doi.org/10.3390/app14020492

Academic Editor: Douglas O'Shaughnessy

Received: 28 November 2023
Revised: 29 December 2023
Accepted: 2 January 2024
Published: 5 January 2024

1. Introduction

Pansori music, originating in South Korea in the late seventeenth century, gained popularity among the privileged class by the mid-eighteenth century and has since been recognized as intangible heritage by UNESCO, highlighting the vital need for its preservation and transmission to future generations [1,2]. This traditional art form involves two key performers—the singer and the drummer. The drummer maintains the rhythm and provides essential cues for the singer, who, in turn, narrates melodic stories. The performance is structured into two distinct parts: storytelling, where the singer describes various characters and conveys the emotional depth of the story, followed by a segment where the singer projects their voice from the diaphragm, resonating powerfully with the audience [3]. Pansori music is characterized by three primary sources: the drum, the drummer's voice, and the singer's voice. The drummer and drum sounds are repeated throughout the song, typically within short timeframes. The drum sounds vary, comprising three distinct types or a combination thereof. Beyond its cultural significance, the need for vocal note transcription in Pansori music is crucial. Capturing the intricate details of the vocal notes and rhythms is essential for both preserving the authenticity of this

art form and facilitating its transmission to future generations. Accurate transcription allows for a deeper understanding of Pansori music's complex structure and provides a basis for analysis and study, offering insights into the interplay between the drummer, the singer, and the vocal and rhythmic patterns. This transcription process becomes essential for preserving the essence and integrity of Pansori music while facilitating its study in current contexts.

There are basically four levels of transcription; frame-level transcription, note-level transcription, stream-level transcription, and notation-level transcription. In this paper, we are concerned with frame-level transcription and note-level transcription. Frame-level transcription involves analyzing very short frames of audio to identify the pitch or fundamental frequency (F0), which has been already proposed for transcription of speech [4], singing voices [5], and musical instruments [6], while note-level transcription approaches estimate notes' onset and offset times including pitches [7,8]. The note-level transcription can be estimated in two ways. The first way is to estimate the onset time, offset time, and pitch of the note directly from the model [9], and the second way is to estimate the fundamental frequency first and apply post-processing like median filtering [10] or the hidden Markov model (HMM) [11] in the second step. In this work, we used the latter approach to estimate the vocal notes of Pansori. First, we used a pixel-wise semantic segmentation framework to calculate the fundamental frequency (F0) and applied Viterbi decoding obtained from the first step to estimate the vocal notes of Pansori music. We call this semantic segmentation framework for F0 estimation the source task and Viterbi decoding-based post-processing for Pansori vocal note estimation the target task.

The proposed attentional based encoder–decoder network model for F0 estimation presented in this research for the source task comprises a comparison of self-attention and channel attention with different loss functions like binary cross-entropy and focal loss [12]. Our selection for attention mechanisms is inspired and drawn from the foundational works of the field [13–16]. The proposed model is designed to operate on a combined frequency and periodicity (CFP) representation [10], which offers a detailed and comprehensive frequency–periodicity view of the audio data, essential for accurate F0 estimation. This architecture encompasses a series of convolutional layers, with subsequent batch normalization and SELU activation functions, aimed at extracting hierarchical features from the input CFP representation. The model integrates max pooling layers for downsampling and corresponding max unpooling layers for upsampling only in the frequency axis to pool only in the frequency dimension, facilitating an effective feature extraction and reconstruction process. Notably, the model incorporates a self-attention and channel attention mechanism specifically within the bottleneck. The self-attention layer added onto the final feature map of the encoder reduces the computational complexity because the frequency axis is reduced in final feature map of the encoder and the time axis in this stage is the same as the input because of the one-dimensional pooling layer.

The model acquired from the source task trained on the vocal melody dataset is an essential component for F0 estimation in Pansori vocal note transcription in the target task. This model provides crucial insights and features for estimating fundamental frequency values (F0) in the target Pansori audio data. To precisely estimate the vocal notes in the Pansori performance, the Viterbi decoding technique was employed, a core process within hidden Markov models (HMMs) [17]. Viterbi decoding involves the representation of different states (such as silence, onset, and sustain states) that correspond to various aspects of the vocal notes of Pansori within a defined range. The transition matrix is formulated to model the probabilities associated with transitioning between distinct vocal states within the HMM. This matrix reflects the likelihood of moving from one state (e.g., silence) to another (e.g., onset or sustain) during the progression of Pansori vocal performances. Similarly, the observation matrix is computed to assess the probability of observing specific vocal notes or events within Pansori music. It quantifies the likelihood of various vocal characteristics, such as note transitions, sustained notes, or silence, observed within the performance. By leveraging the Viterbi algorithm in conjunction with the constructed transition and

observation matrices, we aim to identify the most probable sequence of hidden states (representative of Pansori vocal notes) in the audio recordings. This sequence considers the observed vocal characteristics and the probable transitions between these vocal states. The application of these techniques seeks to offer a systematic and robust methodology for accurately transcribing and analyzing Pansori vocal performances, providing insights into the structured vocal elements and note transitions inherent in this traditional Korean music.

2. Related Work

Several research studies have explored frame-level transcription techniques in the context of music analysis. In one study [18], a method is introduced for estimating multiple concurrent piano notes, addressing overlapping overtones with smooth autoregressive models. This approach mitigates pitch estimation errors in the presence of background noise, providing a comprehensive model for piano note transcription, validated using real piano recordings from the MAPS dataset [18]. Another study [10] tackles the challenge of training data for estimating multiple pitches in music and proposed a novel approach called combined frequency and periodicity (CFP). This method combines different features to enhance accuracy in simultaneous pitch estimation, demonstrating effectiveness in western polyphonic music even under audio distortions like filtering and compression. While these two studies represent conventional approaches to frame-level transcription, recent attention from researchers has shifted toward neural network-based methods [19,20]. In [19], a supervised neural network model for polyphonic transcription on the MAPS dataset is introduced, incorporating an acoustic model and a music language model. The acoustic model, based on neural networks, estimates pitch probabilities in audio frames, while the recurrent neural network-based language model captures pitch correlations over time. In [20], the analysis of neural network-based frame-level piano transcription compares four different input representations, highlighting the importance of selecting the right input representation and fine-tuning hyperparameters, particularly the learning rate and its schedule, to improve frame-level transcription accuracy.

Vocal melody transcription represents a specific aspect of frame-level transcription. In [21], a high-resolution network (HRNet) is employed to separate vocals from polyphonic music, and an encoder–decoder network is used to estimate vocal F0 values. The experimental results indicate that this HRNet-based singing voice separation method effectively minimizes accompaniment interference, surpassing other state-of-the-art algorithms in most cases. Another study ([22]) focuses on single-instrument transcription, demonstrating the ability to estimate onset and offset times with arbitrary time resolution, outperforming accuracy in the MAESTRO dataset [23]. Additionally, Ref. [24] introduces Omnizart, a Python toolkit covering the entire deep learning-based automatic music transcription (AMT) life cycle. Featuring a compact command–line interface for user convenience, Omnizart provides models for a diverse range of instruments, including solo and ensemble instruments, percussion, and vocals. It also includes models for chord recognition and beat/downbeat tracking, addressing crucial tasks in music information retrieval (MIR) related to automatic music transcription. Similarly, Ref. [25] tackles the challenging task of automatic music transcription (AMT), highlighting the unique difficulties of transcribing multiple instruments simultaneously with fine-scale pitch and timing precision. By employing a general-purpose transformer model trained through sequence-to-sequence transfer learning, the research successfully demonstrates multi-task AMT across various datasets.

The note-based transcription approaches directly estimate notes, including pitches and onsets. This note-based transcription is one level higher than frame-level transcription. The work in [26,27] estimates the pitches and onsets in a single architecture. One example for this is [28], which jointly estimates the different attributes of notes like pitch, intensity, onset, and duration. It estimates the properties of notes concurrently using harmonic temporal structured clustering. Note-level transcription can also be achieved after the post-processing in frame-level transcription. First, the fundamental frequency is estimated concurrently and applied post-processing to estimate the musical notes in the second

step. The methods used during the post-processing steps are median filtering, hidden Markov models, and neural networks [10,11,29]. The work in [10] uses median filtering by comparing the estimated pitch in nearby frames for temporal smoothing. The moving median filter is used with 0.25 s with a hop size of 0.01 s. This post-processing method is reliable for connecting non-continuous pitch and can effectively delete the isolated one. Similarly, the work in [11] converts the output of the support vector machine to a posterior probability. The steps for pitch smoothing are performed for each note class by using the Viterbi search method on the transition matrix of 2×2. The note onset and offset is finally gathered from both the posterior probability of the support vector machine and the training data.

3. Method

3.1. Frame-Level Transcription

3.1.1. Input

The model input for frame-level transcription in the source task to extract vocal melody is the representation of combined frequency and periodicity (CFP) [10]. Generalized cepstrum and generalized cepstrum of spectrum are the periodicity representation, whereas the power scaled spectrogram is the frequency representation. The CFP can combine the spectral and temporal features of music which aggregates the complementary benefits of the two feature domains in different frequency ranges and improves the pitch detection algorithm. We use CFP representation as the input to our source task because the previous work related to music transcription [10,30,31] has already proved that CFP representation is beneficial for frame-level transcription because of the aggregation of the two feature domains.

3.1.2. Attention-Based Semantic Segmentation Framework

To conduct frame-level transcription for vocal melody extraction in the source task, we use encoder–decoder architecture which was originally developed for image segmentation [32]. The encoder takes an input CFP that represents both the frequency and periodicity information of the signal. The input has three channels that can represent the three aspects of music features. First, the input dimension of size $3 \times F \times T$ is passed into the encoder consisting of three convolution blocks. Each of the convolution blocks consists of a batch normalization layer, 2D convolution layer, scaled exponential linear unit (SELU) activation, and max pooling layer. The pooling layer has a size of 4×1 which pools only in the frequency dimension and consists of pooling indices between the pooling and unpooling layers. The final feature of the encoder is passed into the three consecutive self-attention blocks. Each self-attention block consists of a linear transformation of query, key, and value tensor applied to the output of the encoder. The attention weights are computed using the scaled dot-product attention mechanism between query and key. The softmax function is applied to obtain the normalized attention weights. The attention weights are multiplied with the value tensor to obtain the final attended values from the self-attention block. The output of the self-attention-based features has two paths, one for the decoder and another for the convolutional block. The path for the decoder consists of three blocks of convolution which are the same as the encoder and is used to upsample the features using transposed convolution. This path is constructed to detect the voicing frames of the audio. Similarly, another path for a single convolutional block is designed to estimate the non-voicing frames of the audio. As we apply the pooling operation only in frequency dimension, the size of this non-voicing detector is $1 \times T$. Finally, we concatenate the detection obtained from the two different paths to obtain the final output representation of vocal melody extraction. The architecture for the encoder–decoder network and self-attention-based vocal melody extraction is shown in Figures 1 and 2, respectively. The final output of the model architecture is obtained by concatenating the result of the bottom block and the upsampled result from the last convolutional block of the decoder along the frequency dimension ($dim = 2$).

The softmax function is then applied along this frequency dimension to obtain the final output, as shown in Equation (1).

$$output = softmax(torch.cat(bottom(pool3(conv3(pool2(conv2(pool1(conv1(x))))))), dim = 2)) \quad (1)$$

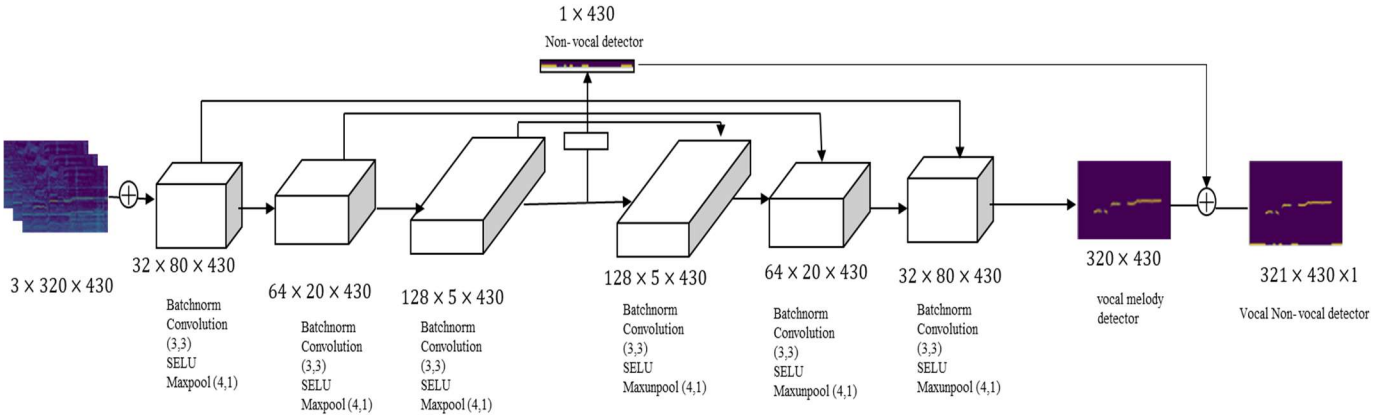

Figure 1. Encoder–decoder network for vocal melody extraction.

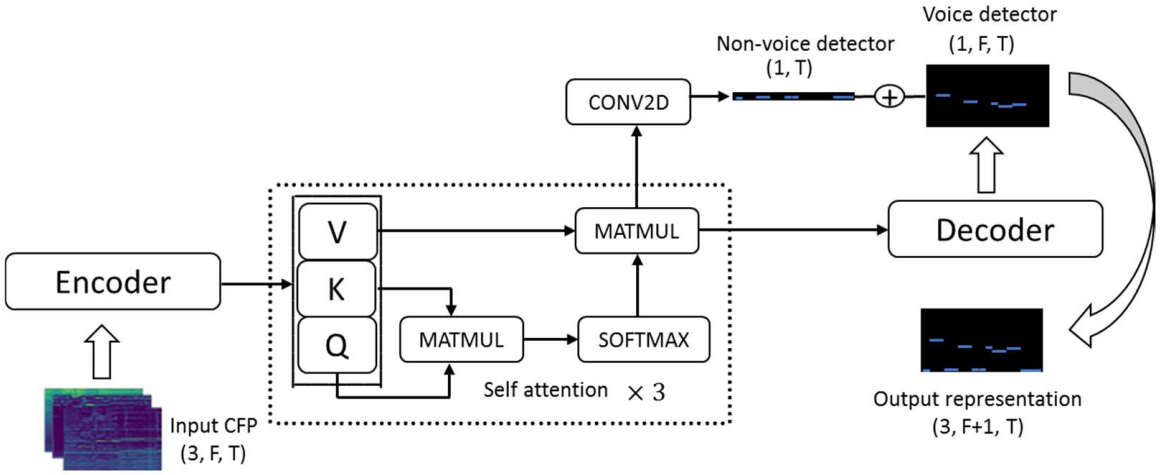

Figure 2. Self-attention-based vocal melody extraction.

We also integrate the channel attention mechanism after obtaining the final features of the encoder network and compare the performance with self-attention. The channel attention module introduces a channel-wise attention mechanism by selectively amplifying informative channels within the input tensor. This mechanism is particularly beneficial for tasks such as vocal melody extraction, where certain channels may carry more crucial pitch information. To apply the channel attention, first, an adaptive average pooling operation is applied to reduce the spatial dimensions of the tensor to 1×1, which can effectively summarize the global information across the channel dimension. Subsequently, the tensor passes through two linear fully connected layers. The first linear layer reduces the dimensionality of the channels to $input\_channels/reduction\ ratio$, followed by ReLU activation, where we keep the reduction ratio as 16. The tensor is then further processed by another linear layer, which restores the dimensionality of the original number of channels. The final attention weights are reshaped, which allows for element-wise multiplication with the original input obtained from the feature map of the encoder. The resulting feature map has again two paths, one for non-voice detection and another for voice detection, the same as for the self-attention described above.

3.1.3. Loss Function

We compare focal loss [12] and binary cross-entropy loss in our experiment for the source task of vocal melody transcription. The goal of these loss functions is to classify each time step in an audio signal as either belonging to the vocal melody (positive class) or not (negative class). The loss function computes the error by comparing the model's output logits, typically representing the likelihood of the presence of a melody, with the binary ground truth labels. By optimizing this loss during training, the model learns to distinguish between vocal melody and non-melody segments in the audio, which is essential for accurate melody extraction. In vocal melody extraction tasks, there may be a significant imbalance between voicing and non-voicing frames in audio data. The presence of voicing frames is often the minority class. Focal loss is designed to address such imbalances by assigning higher weights to hard-to-classify examples, which in this context would be the voicing frames. Following [12], we set the value of parameter for focal loss as $\alpha_t = 0.25$, and $\gamma = 2$.

3.2. Note-Level Transcription

The pre-trained model trained on the MedleyDB dataset [33] for vocal melody extraction in the source task provides the meaningful features for estimating F0 values in the target Pansori data. The vocal notes in the target task can be achieved after applying the post-processing in the Pansori estimated F0 of the source task in frame-level transcription. To conduct post-processing for the vocal notes in the Pansori performance, the Viterbi decoding technique is employed, a core process within hidden Markov models (HMMs). In the Viterbi decoding, the transition matrix T and observation matrix P play a specific role. More specifically, T represents the probabilities of transitioning between different hidden states and P represents the likelihood of observations given a particular state. By utilizing the Viterbi algorithm with T and P, the goal is to find the most likely sequence of hidden states (Pansori vocal notes) that best explains the observed musical events. Given the definition of states, S_0 for the silence state, S_i for the onset state for a particular note i, and S_i' for the sustain state for note i, the transition matrix T with dimensions $N \times N$ is constructed with specific probabilities for transitions between these states. The size of this transition matrix is determined by the range of musical notes, which we consider here from A2 to E6. Similarly, $P_{stay\_silence}$ represents the probability of staying in the silence state, P_{stay_note} represents the probability of the sustain state S_i' returning to itself, and $P\_$ represents the probability of transitioning from the silence state to an onset state. The structure of the transition matrix is given below.

$$T = \begin{bmatrix} P_{stay\_silence} & P\_ & 0 & 0 & \dots & 0 \\ 0 & 0 & P_{stay\_note} & 0 & \dots & 0 \\ 0 & 0 & 0 & P_{stay\_note} & 0 & \dots \\ . & . & . & . & .. & . \\ . & . & . & . & .. & . \\ 0 & 0 & 0 & 0 & 0 & P_{stay\_note} \\ P\_\_\_ & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Each value in the matrix T represents the probability of transitioning from a state (indicated by the row) to another state (indicated by the column) in the HMM. The specific values of $P_{stay\_silence}$, P_{stay_note}, and $P\_$ determine these transitions within the model's states.

Similarly, the observation matrix P is constructed based on several input parameters. To calculate the size of rows for matrix P, first, we defined the range of Pansori vocal notes from A2 to E6 and converted them into MIDI numbers. Hence, the number of notes n_{notes} can be obtained by subtracting from the maximum note $note_{max}$ to the minimum note $note_{min}$. In this context, the matrix P includes rows for the silence, onset, and sustain states for each vocal note. Hence, we multiply the number of notes by 2 for the onset and sustain states and add 1 for the silence state. So, the size of the rows for the observation matrix

P is calculated as $size_{rows} = 2(note_{max} - note_{min}) + 1$. Similarly, to calculate the size of the columns for the matrix P, the length of F0 values is calculated, which is obtained from our pre-trained model of the source task.

To convert the Pansori audio data into MIDI format, first, we passed the wave file of Pansori into our best vocal melody pre-trained model. The F0 for the Pansori vocal data can then be estimated using this pre-trained model. We also calculated the tuning frequency and adjusted the estimated F0. This corrected F0 was transformed to MIDI notes after applying the tuning correction. The tuning frequency aims to refine the accuracy of the estimated pitch by adjusting it based on any detected tuning discrepancies in the audio signal. The librosa library was employed for this purpose, contributing to the overall accuracy of the pitch estimation process [34]. To construct the observation matrix P, we defined the probability of $voice_{acc}$, $onset_{acc}$, and $pitch_{acc}$. If the predicted F0 value by the pre-trained model was non-zero, we assumed that the corresponding frame is considered the voiced frame. But, if the generated F0 value was zero, it was considered the non-voiced frame. We also detected the onset from the audio, and if the time frame was detected as an onset, it set the high probability for the onset state. But, if the time frame was detected as the non-onset state, it set the low probability for the non-onset state. To assign the probabilities to the sustain states of the observation matrix P, we looped over all possible MIDI note values from A2 to E6, and if it exactly matched with the transformed MIDI values, we set the high probability for the sustain state at that time frame. Similarly, if this did not match, we set the low probability for the sustain state.

The Viterbi algorithm works by considering the transition matrix T and observation probabilities P to identify the most probable sequence of the hidden state. For each time step in the observed sequence (along columns of P), the Viterbi algorithm assesses all possible state sequences at that time step. It computes the likelihood or probability of each state sequence occurring. It iterates through the observed sequence, updating the probability of arriving at each state based on the maximum likelihood computed from the previous time step. At each step, the algorithm computes the most probable path (sequence of states) by considering both the probability of the current observation given the state and the probability of transitioning to the current state from the previous states. By taking these probabilities into account and maximizing the likelihood, the algorithm infers the most likely sequence of hidden states that generated the observed sequence of events (here, Pansori vocal notes) based on the given models. We utilized the librosa library to obtain the most probable sequence of hidden states for Viterbi decoding (https://librosa.org/doc/main/generated/librosa.sequence.viterbi.html), accessed on 25 December 2023. After obtaining the sequence of states through Viterbi decoding, the subsequent processing involved converting these states into an intermediate piano-roll representation. The states, representing silence, onset, and sustain events, were iteratively analyzed to identify the onset and offset times, MIDI pitches, and note names of the vocal segments. The information obtained in the piano roll was finally converted into a MIDI file using the midiutil library (https://readthedocs.org/projects/midiutil/downloads/pdf/latest/), accessed on 25 December 2023.

4. Experimental Results

4.1. Frame-Level Transcription

We conducted the experiment of frame-level transcription by comparing self-attention and channel attention mechanisms, assuming the task as a pixel-wise semantic segmentation framework. The different loss functions like binary cross-entropy and focal loss were also compared. We evaluated the effectiveness of our method against state-of-the-art techniques on the MedleyDB [33] dataset, utilizing the same splitting ratio as described in [30,31,35]. The network hyper-parameters remained consistent with those outlined in [30]. The evaluation metrics used were voicing recall (VR), voicing false alarm (VFA), raw pitch accuracy (RPA), raw chroma accuracy (RCA), and overall accuracy (OA).

Our self-attention-based approach, integrated with focal loss, demonstrated notable results. The one-layer self-attention with focal loss achieved a voicing recall of 68.05%, with an overall accuracy of 78.74%. As we increased the self-attention layers to two and three, a slight decrease in performance was observed, suggesting potential saturation in model capacity. This observation prompted us to explore the reasons behind this trend and assess whether the increase in self-attention layers contributes significantly or not to model effectiveness.

Similarly, when employing binary cross-entropy as the loss function, the two-layer self-attention stands out with a high overall accuracy of 80.25%, showcasing the effectiveness of this combination. This notable performance of the two-layer self-attention brings a closer examination between attention mechanisms and loss functions. One potential avenue for this analysis is the impact of self-attention depth on the model's capacity to capture hierarchical dependencies within the data. This raises questions about the trade-offs involved in increasing or decreasing the depth of self-attention layers and how these choices influence the model's ability to generalize. Our channel attention-based methods, integrated with both focal loss and binary cross-entropy consistently exhibited strong performance. Notably, the focal loss variant stands out with an impressive overall accuracy of 80.33%, surpassing the performance of the binary cross-entropy counterpart, which achieved an accuracy of 79.87%. Focal loss is designed to address class imbalance by down-weighting well-classified examples, potentially enhancing the model's focus on challenging instances.

Comparing our proposed methods to established approaches, such as DSM [35], Lu and Su's [31], and SegNet [30], our techniques demonstrate competitive results. While DSM excels in voicing recall, our methods strike a balance, achieving strong performance across multiple metrics. Lu and Su's method performs admirably, particularly in voicing false alarm, showcasing its distinct strengths.

In conclusion, our proposed vocal melody extraction method, leveraging self-attention and channel attention mechanisms, coupled with carefully chosen loss functions, presents promising results on the MedleyDB dataset. The analysis of different configurations provides insights into the strengths and potential areas for improvement, paving the way for advancements in vocal melody extraction techniques. Table 1 below shows the comparison of various F0 estimation methods on the test data of MedleyDB.

Table 1. Comparison of various F0 estimation methods on the MedleyDB test data.

Method	VR	VFA	RPA	RCA	OA
One-layer SA + F	68.05	10.53	59.91	61.38	78.74
Two-layer SA + F	64.68	7.75	57.69	58.97	79.83
Three-layer SA + F	59.72	8.94	52.73	54.60	76.86
One-layer SA + CE	60.75	7.74	54.75	55.75	78.25
Two-layer SA + CE	63.68	**7.55**	58.30	59.12	80.25
Three-layer SA + CE	67.08	9.62	59.22	60.43	78.90
Channel attention + F	65.84	9.02	60.56	61.71	**80.33**
Channel attention + CE	65.73	8.77	59.40	60.63	79.87
DSM [35]	**88.4**	48.7	**72.0**	**74.8**	66.2
Lu and Su's [31]	77.9	22.4	68.3	70.0	70.0
SegNet [30]	73.7	13.3	65.5	68.9	79.7

We also visualize the results of Pansori vocal F0 using our best pre-trained model (channel attention + F) and other pitch estimation methods such pYin [36], PyWorld [37], and Segnet [30]. Figure 3 below shows the tracking of F0 visualization of our method along with other pitch estimation methods in one of the audio samples of Pansori. The

visualized result from our method shows similar tracking performance in comparison with other methods. To visualize the F0 tracking in the spectrogram, we first extract the F0 values of the audio sample from our method as well as other pre-trained models. The fundamental frequency values extracted from each model are then overlaid into the spectrogram. This comprehensive visualization aids in assessing the accuracy and consistency of our method in tracking F0 variations throughout the audio sample. The cyan lines in all four plots in Figure 3 represent the fundamental frequency values estimated by the model, providing insights into the pitch dynamics of the Pansori performance. This visual analysis is crucial for evaluating the efficacy of our approach in comparison to other state-of-the-art pitch estimation techniques. These visualized results contribute to the evaluation and validation of our approach in the broader context of pitch estimation for traditional Pansori vocal performances.

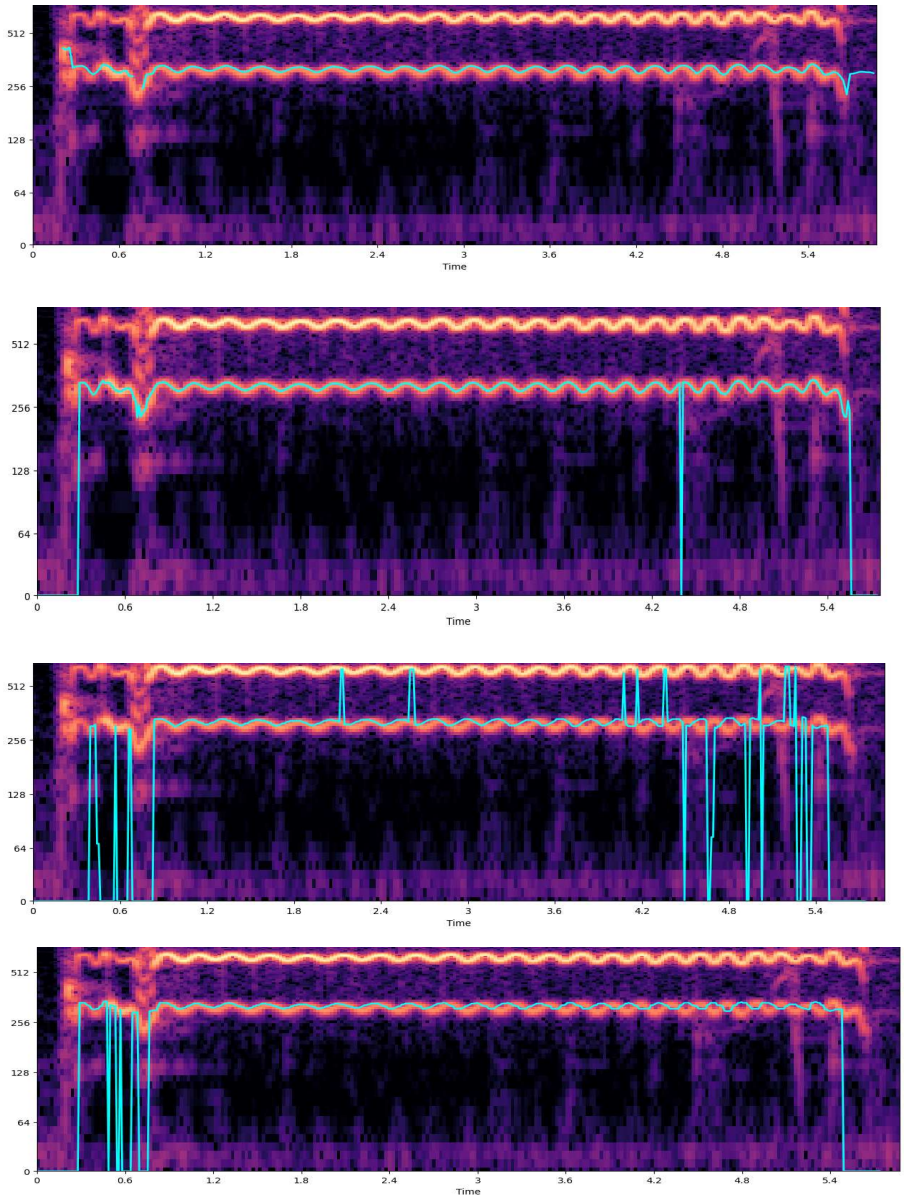

Figure 3. The tracking of F0 in one of the audio samples of Pansori. From top to bottom: pYin, PyWorld, Segnet, and channel attention + F (ours).

4.2. Note-Level Transcription

Note-level transcription can be achieved after the post-processing in frame-level transcription. First, the fundamental frequency of the Pansori raw wave file is estimated in the source task of vocal melody extraction and post-processing is applied using Viterbi decoding to estimate the vocal notes in the second step. The transition matrix of size 89×89 is constructed to represent the Pansori notes from A2 to E6. While constructing the transition matrix, we set the probability of the silence state, sustain state, and transitioning from the silence state to an onset state as $P_{stay\_silence} = 0.2$, $P_{stay\_note} = 0.9$, and $P\_ = \left(1 - P_{stay\_note}\right)/(n_{notes} + 1)$, where n_{notes} is the number of midi values between A2 and E6.

To construct the observation matrix P, first, we defined the probability of $voice_{acc} = 0.9$, $onset_{acc} = 0.8$, and $pitch_{acc} = 0.99$. If the F0 value generated by the pre-trained model was non-zero, we assumed that the corresponding frame is considered the voiced frame and set its probability as $voice_{acc} = 0.9$, but if the generated F0 value was zero, it was considered as the non-voiced frame, and we set its probability as $1 - voice_{acc}$. We also detected the onset from the audio and if the time frame was detected as an onset, it set the probability of onset state as $onset_{acc} = 0.8$. But if the time frame was detected as the non-onset state, it set the probability of non-onset state as $1 - onset_{acc}$. To assign the probabilities to the sustain states of the observation matrix P, we looped over all possible MIDI notes values from A2 to E6, and if it exactly matched with the transformed MIDI values, we set the probability of the sustain state as $pitch_{acc} = 0.99$ at that time frame. Similarly, if the absolute difference between these two MIDI values was 1, we set the probability of the sustain state as $pitch_{acc} \times 0.6 = 0.59$. If these two conditions did not match, we set the probability of the sustain state as $1 - pitch_{acc}$.

After obtaining the sequence of states through Viterbi decoding, we converted these states into an intermediate piano-roll representation. This transformation is designed to interpret the estimated states in the context of Pansori vocal performance. The piano roll consists of information such as onset and offset times, MIDI pitches, and note names of the vocal segments of Pansori.

First, we collected 15 chunks of Pansori audio samples. These samples were first used to estimate the F0 using our pre-trained vocal melody extraction methods. These F0 values were used in the Viterbi decoding algorithm to estimate the vocal note. We also extracted the F0 values from other state-of-the-art pitch estimation methods such as pYin [36], PyWorld [37], and Segnet [30] and visually compared the vocal notes with our best pre-trained methods. Figure 4 below shows the visual comparison of our method with pYin, PyWorld, and Segnet. It visualizes the audio, along with the MIDI, derived from one of the Pansori audio samples, employing both our method and other state-of-the-art techniques. We used the Audacity tool to visualize the audio signal along with its corresponding midi [38]. The X-axis in the top row in Figure 4 represents the time, whereas the Y-axis represents the amplitude of the raw audio Pansori. Similarly, the X-axis from second row to fifth row represents the vocal note of Pansori using our method and other state-of-the-art methods, where the Y-axis here represent the note names from C3 to C6. The original dataset of 15 chunks of Pansori along with their midi results are also available in the github (https://github.com/pratikshaya/music_note_transcription) accessed on 25 December 2023.

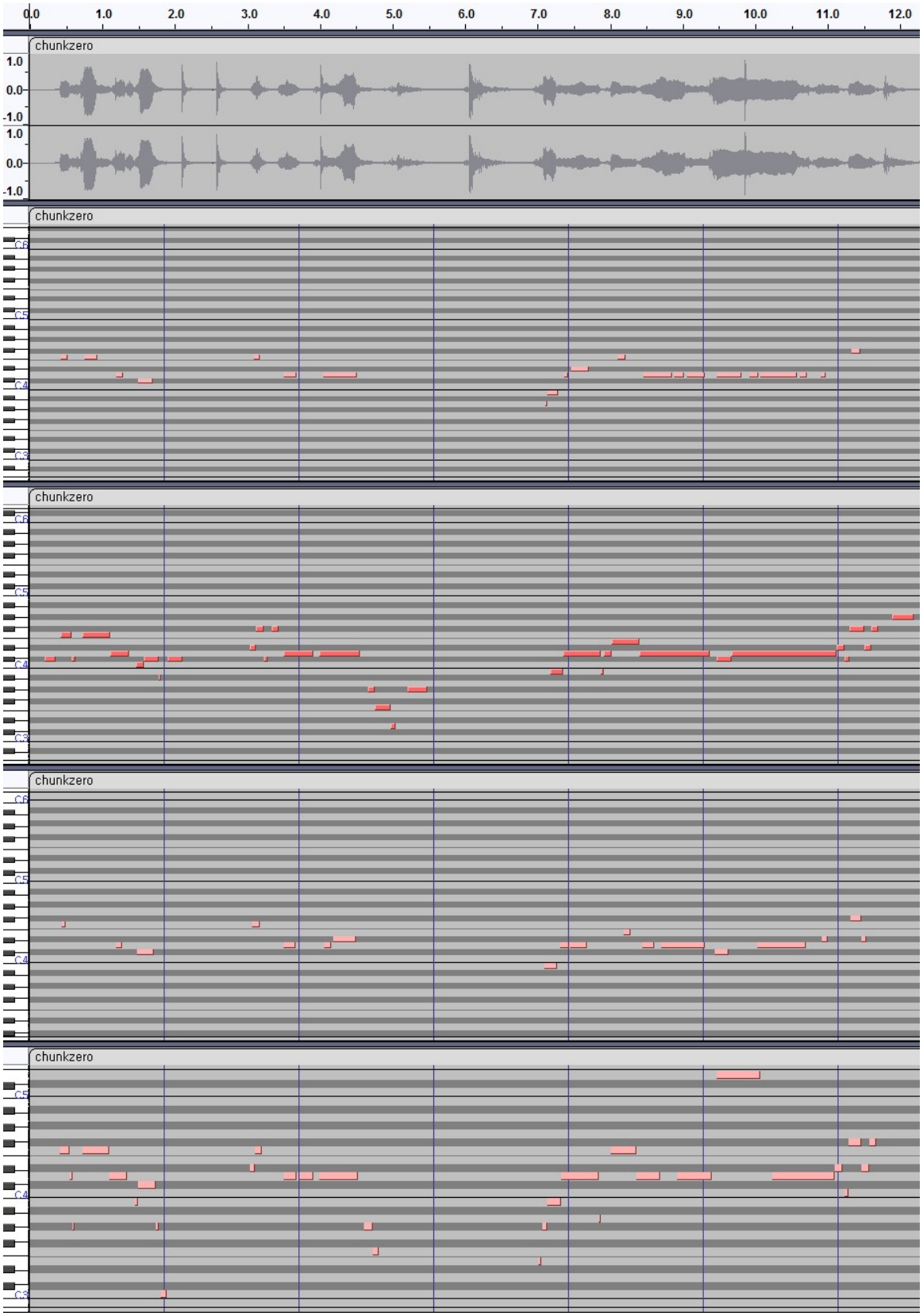

Figure 4. Visual comparison of our method with pYin, PyWorld, and Segnet. From top to bottom: original Pansori audio, channel attention + F (ours), pYin [36], PyWorld [37], and Segnet [30].

Due to the absence of note-level ground truth labels for Pansori data, a subjective evaluation was conducted. Our approach involved providing 15 note-level predicted MIDI files to Pansori experts for assessment. The MIDI files using Viterbi decoding obtained after F0 estimation from our best pre-trained model along with PyWorld and pYin were given to the expert. Five individuals were selected for this task, and each expert was assigned a rating between 0 and 5. A score of 0 represented the worst result, while a score of 5 indicated the best outcome. The average values from all five experts were then calculated for each of the 15 data samples. pYin obtained the average value of 2.56, PyWorld obtained 2.04, and our channel attention + focal loss obtained 2.32. This indicates that our Viterbi algorithm-based post-processing method is comparable with other popular state-of-the-art methods. The detailed rating values for each of the audio samples of Pansori using PyWorld, pYin, and channel attention + F are shown in Tables 2–4, respectively.

Table 2. Note-level subjective evaluation of Pansori vocal based on F0 estimation from PyWorld.

PyWorld						
Audio	**Expert1**	**Expert2**	**Expert3**	**Expert4**	**Expert5**	**Average**
chunk0	3	3	4	3	2	3
chunk1	2	3	2	3	2	2.4
chunk4	1	2	2	1	1	1.4
chunk5	2	1	2	2	1	1.6
chunk6	2	2	2	3	2	2.2
chunk7	3	2	3	3	2	2.6
chunk8	1	2	4	3	1	2.2
chunk9	3	2	3	2	2	2.4
chunk11	2	2	2	1	1	1.6
chunk12	2	2	2	1	2	1.8
chunk13	2	2	3	1	1	1.8
chunk14	3	2	1	2	1	1.8
chunk16	3	2	3	2	1	2.2
chunk17	3	2	2	2	2	2.2
chunk18	1	2	2	1	1	1.4
						2.04

Table 3. Note-level subjective evaluation of Pansori vocal based on F0 estimation from pYin.

pYin						
Audio	**Expert1**	**Expert2**	**Expert3**	**Expert4**	**Expert5**	**Average**
chunk0	3	3	4	3	2	3
chunk1	4	3	2	3	1	2.6
chunk4	5	4	3	3	2	3.4
chunk5	2	2	2	1	2	1.8
chunk6	3	3	2	2	3	2.6
chunk7	3	3	2	2	1	2.2
chunk8	2	2	4	2	3	2.6

Table 3. *Cont.*

pYin						
Audio	**Expert1**	**Expert2**	**Expert3**	**Expert4**	**Expert5**	**Average**
chunk9	3	3	2	3	2	2.6
chunk11	3	5	2	3	1	2.8
chunk12	4	4	2	3	1	2.8
chunk13	2	2	4	3	2	2.6
chunk14	3	2	1	2	1	1.8
chunk16	4	5	3	3	1	3.2
chunk17	3	2	3	3	1	2.4
chunk18	2	2	2	3	1	2
						2.56

Table 4. Note-level subjective evaluation of Pansori vocal based on F0 estimation from channel attention + F.

Channel Attention + F (Our)						
Audio	**Expert1**	**Expert2**	**Expert3**	**Expert4**	**Expert5**	**Average**
chunk0	4	5	3	2	3	3.4
chunk1	2	3	2	3	2	2.4
chunk4	3	4	2	2	2	2.6
chunk5	2	3	4	3	3	3
chunk6	3	4	2	2	3	2.8
chunk7	3	3	2	1	2	2.2
chunk8	2	2	3	1	2	2
chunk9	2	2	2	1	3	2
chunk11	2	2	2	2	2	2
chunk12	1	2	1	1	1	1.2
chunk13	3	4	3	3	2	3
chunk14	3	4	2	3	3	3
chunk16	1	1	2	2	1	1.4
chunk17	2	2	2	2	1	1.8
chunk18	1	3	2	2	2	2
						2.32

5. Conclusions

In conclusion, this paper presents a comprehensive exploration of frame-level transcription for vocal melody extraction using attention mechanisms in the source task, followed by the application of the Viterbi algorithm for note-level transcription in the context of Korean Pansori. Leveraging knowledge from the source task, a pre-trained model is employed for fundamental frequency (F0) estimation, crucial for the subsequent vocal note transcription in Pansori. The Viterbi decoding technique is employed to decode the vocal notes in Pansori, addressing the challenge of the absence of note-level ground truth labels for this traditional art form. The transition matrix, capturing the probabilities of transitioning between different vocal states, and the observation matrix, assessing the likelihood of specific vocal events, are constructed. By employing the Viterbi algorithm

with these matrices, this paper achieves the goal of identifying the most likely sequence of hidden states representative of Pansori vocal notes. The presented results, including the comparison of attention mechanisms and the evaluation of our methodology against other state-of-the-art methods, highlight the effectiveness and comparability of our approach in the intricate task of Pansori vocal note transcription.

In summary, this research not only contributes to the advancement of frame-level transcription models but also addresses the unique challenges posed by the transcription of traditional music forms like Pansori. The proposed methodology lays a foundation for preserving and appreciating the rich cultural heritage embedded in Pansori music, offering a valuable tool for future research and exploration of this traditional Korean art form.

Author Contributions: The first author, B.B., contributed to the whole project, which includes conceptualization, methodology, validation, analysis, and writing the original draft. The corresponding author, J.L., contributed to funding acquisition, conceptualization, and supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Research Foundation of Korea (NRF) under the Development of AI for Analysis and Synthesis of Korean Pansori Project (NRF-2021R1A2C2006895).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

References

1. Kang, B. *UNLV Theses, Dissertations, Professional Papers, and Capstones*; UNLV Theses: Las Vegas, NV, USA, 2016; p. 2789. [CrossRef]
2. Um, H. Performing Pansori music drama: Stage, story and sound. *Rediscovering Tradit. Korean Perform. Arts* **2012**, 72.
3. Bhattarai, B.; Pandeya, Y.R.; Lee, J. Parallel stacked hourglass network for music source separation. *IEEE Access* **2020**, *8*, 206016–206027. [CrossRef]
4. Jouvet, D.; Laprie, Y. Performance analysis of several pitch detection algorithms on simulated and real noisy speech data. In Proceedings of the 25th European Signal Processing Conference (EUSIPCO), Kos, Greece, 28 August–2 September 2017; IEEE: Piscataway, NJ, USA; pp. 1614–1618.
5. Babacan, O.; Drugman, T.; d'Alessandro, N.; Henrich, N.; Dutoit, T. A comparative study of pitch extraction algorithms on a large variety of singing sounds. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; IEEE: Piscataway, NJ, USA; pp. 7815–7819.
6. Von Dem Knesebeck, A.; Zölzer, U. Comparison of pitch trackers for real-time guitar effects. In Proceedings of the 13th International Conference on Digital Audio Effects, Graz, Austria, 6–10 September 2010.
7. Duan, Z.; Temperley, D. Note-level Music Transcription by Maximum Likelihood Sampling. In Proceedings of the 15th International Society for Music Information Retrieval Conference, Taipei, Taiwan, 27–31 October; 2014; pp. 181–186.
8. Kim, S.; Hayashi, T.; Toda, T. Note-level automatic guitar transcription using attention mechanism. In Proceedings of the 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, 29 August–2 September 2022; IEEE: Piscataway, NJ, USA; pp. 229–233.
9. Hsu, J.Y.; Su, L. VOCANO: A note transcription framework for singing voice in polyphonic music. In Proceedings of the 22nd International Society for Music Information Retrieval Conference, Online, 7–12 November 2021.
10. Su, L.; Yang, Y.H. Combining spectral and temporal representations for multipitch estimation of polyphonic music. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 1600–1612. [CrossRef]
11. Nam, J.; Ngiam, J.; Lee, H.; Slaney, M. A Classification-Based Polyphonic Piano Transcription Approach Using Learned Feature Representations. In Proceedings of the 12th International Society for Music Information Retrieval Conference, Miami, FL, USA, 24–28 October 2011; pp. 175–180.
12. 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.
13. 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*.
14. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.

15. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7151–7160.
16. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
17. Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* **1967**, *13*, 260–269. [CrossRef]
18. Emiya, V.; Badeau, R.; David, B. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Trans. Audio Speech Lang. Process.* **2009**, *18*, 1643–1654. [CrossRef]
19. Sigtia, S.; Benetos, E.; Dixon, S. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 927–939. [CrossRef]
20. Kelz, R.; Dorfer, M.; Korzeniowski, F.; Böck, S.; Arzt, A.; Widmer, G. On the potential of simple framewise approaches to piano transcription. *arXiv* **2016**, arXiv:1612.05153.
21. Gao, Y.; Zhang, X.; Li, W. Vocal melody extraction via hrnet-based singing voice separation and encoder-decoder-based f0 estimation. *Electronics* **2021**, *10*, 298. [CrossRef]
22. Kong, Q.; Li, B.; Song, X.; Wan, Y.; Wang, Y. High-resolution piano transcription with pedals by regressing onset and offset times. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 3707–3717. [CrossRef]
23. Hawthorne, C.; Stasyuk, A.; Roberts, A.; Simon, I.; Huang, C.Z.; Dieleman, S.; Elsen, E.; Engel, J.; Eck, D. Enabling factorized piano music modeling and generation with the MAESTRO dataset. ICLR. *arXiv* **2019**, arXiv:1810.12247. Available online: https://arxiv.org/abs/1810.12247 (accessed on 25 December 2023).
24. Wu, Y.T.; Luo, Y.J.; Chen, T.P.; Wei, I.; Hsu, J.Y.; Chuang, Y.C.; Su, L. Omnizart: A general toolbox for automatic music transcription. *arXiv* **2021**, arXiv:2106.00497. [CrossRef]
25. Gardner, J.; Simon, I.; Manilow, E.; Hawthorne, C.; Engel, J. MT3: Multi-task multitrack music transcription. In international conference in learning representation. *arXiv* **2022**, arXiv:2111.03017.
26. Berg-Kirkpatrick, T.; Andreas, J.; Klein, D. Unsupervised transcription of piano music. *Adv. Neural Inf. Process. Syst.* **2014**, *27*.
27. Ewert, S.; Plumbley, M.D.; Sandler, M. A dynamic programming variant of non-negative matrix deconvolution for the transcription of struck string instruments. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; IEEE: Piscataway, NJ, USA; pp. 569–573.
28. Kameoka, H.; Nishimoto, T.; Sagayama, S. A multipitch analyzer based on harmonic temporal structured clustering. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 982–994. [CrossRef]
29. Boulanger-Lewandowski, N.; Bengio, Y.; Vincent, P. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv* **2012**, arXiv:1206.6392.
30. Hsieh, T.H.; Su, L.; Yang, Y.H. A streamlined encoder/decoder architecture for melody extraction. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; IEEE: Piscataway, NJ, USA; pp. 156–160.
31. Lu, W.T.; Su, L. Vocal Melody Extraction with Semantic Segmentation and Audio-symbolic Domain Transfer Learning. In Proceedings of the 19th International Society for Music Information Retrieval, Paris, France, 23–27 September 2018; pp. 521–528.
32. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
33. Bittner, R.M.; Salamon, J.; Tierney, M.; Mauch, M.; Cannam, C.; Bello, J.P. Medleydb: A multitrack dataset for annotation-intensive mir research. In Proceedings of the 15th International Society for Music Information Retrieval Conference, Taipei, Taiwan, 27–31 October 2014; pp. 155–160.
34. McFee, B.; Raffel, C.; Liang, D.; Ellis, D.P.; McVicar, M.; Battenberg, E.; Nieto, O. librosa: Audio and music signal analysis in python. In Proceedings of the 14th Python in Science Conference, Austin, TX, USA, 6–12 July 2015; Volume 8, pp. 18–25.
35. Bittner, R.M.; McFee, B.; Salamon, J.; Li, P.; Bello, J.P. Deep Salience Representations for F0 Estimation in Polyphonic Music. In Proceedings of the International Society for Music Information Retrieval Conference, Suzhou, China, 23–28 October 2017; pp. 63–70.
36. Mauch, M.; Dixon, S. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In Proceedings of the 2014 IEEE International Conference On Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; IEEE: Piscataway, NJ, USA; pp. 659–663.
37. Morise, M.; Yokomori, F.; Ozawa, K. WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Trans. Inf. Syst.* **2016**, *99*, 1877–1884. [CrossRef]
38. Audacity Software. 1999. Available online: http://thurs3.pbworks.com/f/audacity.pdf (accessed on 25 December 2023).