arXiv:2304.05917v1 [cs.SD] 12 Apr 2023

用于音符级歌声转录的音素信息神经网络模型

[1]*Sangeon Yong*, [2]*Li Su*, [1]*Juhan Nam*

[1]韩国大田 KAIST 文化技术研究生院[2]台湾台北中央研究院信息科学研究所

摘要

音符级自动音乐转录是最具代表性的音乐信息检索（MIR）任务之一，长期以来，人们一直借助它研究各种乐器，以理解音乐。然而，由于缺乏高质量的标注数据，许多乐器的转录仍是一项具有挑战性的任务。尤其是歌唱，由于其音高、音色和力度富于表现力，因此难以准确识别音符。本文提出一种方法，利用歌唱所具有、其他乐器所不具备的语言特征，更准确地确定歌声的音符起音。所提出的模型将梅尔频谱图和音素后验图（PPG）作为起始检测网络的输入；音素后验图（PPG）是逐帧的音素似然度，由使用歌唱和语音数据预训练的网络生成。为验证语言特征对起始检测的影响，我们比较了不同语言数据集上的评估结果，并对起始类型进行分类，以开展详细分析。我们的方法显著提升了歌声转录的性能，从而凸显了语言特征在歌唱分析中的重要性。

索引词— 歌声转录、起始检测、音素分类、音乐信息检索

1. 引言

音符级歌唱转录是一项音乐信息检索（MIR）任务，旨在从人声演唱的音频录音中预测音符事件的属性（即起始时间、止音时间和音高值）。尽管这项任务已被研究多年，但歌声转录的表现通常不如其他乐器，例如复调钢琴音乐 [1, 2]。缺乏大规模标注数据集是主要技术障碍之一。此外，歌声在音高、音色、力度以及歌词的发声方式等方面表现出高度多样性。例如，颤音、滑音和连滑音等演唱技巧使得确定音符边界和音符级音高变得困难。这种多变性甚至使人类专家进行手动音符转录也颇具难度 [3]。这反过来又导致高质量标注数据集的缺乏。

与其他乐器明显不同的歌声另一项重要特征是，它通过歌词传达语言信息，而这会影响音符分割。鉴于大多数歌唱音符都是音节式的（即一个文本音节对应一个音乐音符）且花唱式的（即一个音节用多个音符演唱），音节变化与音符变化之间的关系十分复杂。这使得歌声中出现了其他乐器所没有的某些音符模式。因此，在自动歌声转录模型中，我们需要考虑这种语言特征。

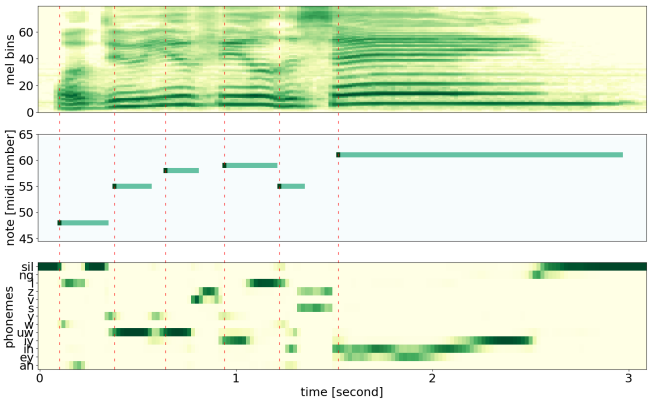

图 1。一个歌声示例：梅尔频谱图（上）、标注音符起始点和音高的钢琴卷帘（中），以及歌声的音素后验概率图（PPG）（下）（此示例中概率低于 0.5 的音素已省略）。

本文提出了一种神经网络模型，将语言信息纳入输入，以改进歌声的音符级歌唱转录。与早期研究类似，我们使用对数尺度梅尔频谱图作为主要输入。此外，我们还将预训练音素分类器生成的音素后验图（PPG）作为第二路输入。如图1所示，PPG呈现出有别于梅尔频谱图的模式；可以看出，在1.2秒和2.8秒处，PPG的过渡模式能够更好地描述起始事件。我们提出了一种以卷积循环神经网络（CRNN）为骨干网络的双分支神经网络模型，以有效表征这两种输入特征。在实验中，我们开展了消融研究，以考察模型设计、梅尔频谱图和PPG的有效性。此外，我们还比较了梅尔频谱图和PPG对过渡与再起音这两类歌声转录中具有挑战性的起始事件的影响。最后，我们证明，所提出的模型优于若干最先进的音符级歌唱转录模型，尤其是在起始检测方面。

2. 相关研究

传统研究主要使用各种类型的频谱差异来检测音频信号的起始点 [4]。频谱差异在寻找打击乐器的起始点方面尤为有效，但对于起音柔和的表现力乐器，效果不佳。深度神经网络也已广泛应用于歌声领域。Nishikimi 等 [5] 提出了一种采用长短期记忆（LSTM）模块的基于注意力的编码器-解码器网络。Fu 等。

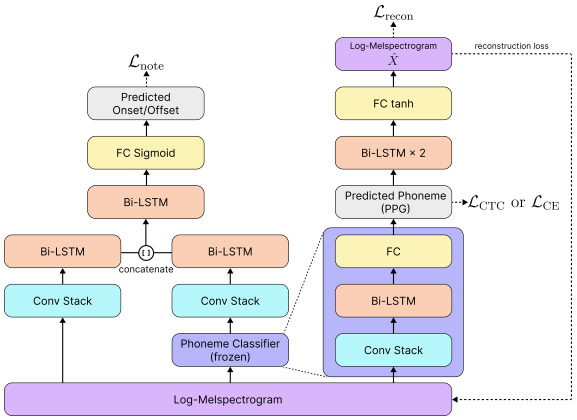

图 2。所提出的模型架构

[6] 提出了一种音符变化状态的层次结构，用于分割歌声中的音符，并采用多通道特征来提升性能。Hsu 等人 [7] 提出了一种半监督 AST 框架。近期， [8] 提出了一种基于目标检测的方法，显著提升了歌声起始/终止检测的性能。尽管其中大多数方法依赖旋律标签中的音符起音和止音信息，但最近有一项研究尝试将音素信息作为输入特征的一部分用于音符分割 [9]。然而，其性能并不理想。在本研究中，我们提出了一种神经网络架构，以有效利用音素信息。

3. 所提出的方法

3.1. 模型架构

如图 2 所示，我们提出的模型架构由两个分支网络和一个带有全连接层的单个循环神经网络组成。其中一个分支网络接收对数尺度梅尔频谱图 X，另一个分支网络接收来自预训练音素分类器的音素后验图（PPG） P。两个分支均为卷积循环神经网络，其中卷积神经网络架构是对 [10], 中提出的卷积网络 *ConvNet* 进行修改后的版本，该卷积网络通常用于钢琴转录任务 [1, 11]。为获得更宽的时间尺度感受野，我们将第一个卷积层替换为在时间帧轴上膨胀率为 2 的空洞卷积。为了预测音符事件，我们通过拼接两个分支网络的输出，并将其连接到额外的循环神经网络层和全连接层，从而融合这两个分支网络。输出层表示为一个三维 sigmoid 向量，其中每个元素分别以二进制状态检测起始点、止音和激活状态。激活状态表示每一帧中的音符处于开启还是关闭状态。

3.2. 帧级音素分类器

我们使用音素分类器提取了语音信息，该分类器将输出以PPG的形式返回。我们使用带有全连接层的单个卷积循环神经网络实现了它。对于卷积神经网络部分，我们使用了原始的卷积网络架-构。我们尝试了两种损失函数来训练音素分类器网络。一种是逐帧交叉熵损失，当我们拥有时间对齐的音素标签时即可使用该损失。由于很难获得帧级别的时间对齐音素标签，尤-其是对于歌声，我们还使用了连接主义时间分类（CTC）损失函数 [12] ，该函数可以处理预测的音素序列（$\hat{}p$）与地面真实音素序列（S）的长度并不相等。CTC 算法会沿着可能的预测路径 $\mathcal{B}$ 预测插入了空白标签的音素序列。由于 CTC 损失函数是针对预测整个序列进行优化的，预测模式往往呈现尖峰且稀疏，因此难以准确找到音素的边界 [12, 13]。为解决这个问题，我们使用了两层双向长短期记忆层和一个全连接层来重构输入的对数尺度梅尔频谱图（$\hat{}X$）。这是一种在使用 CTC 损失时增强时间对齐效果的方法 [14]。对于重构损失（$\mathcal{L}_{\text{recon}}$），我们将对数尺度梅尔-$\tilde{}$频谱图从 -1 归一化到 1（X），并使用 tanh 函数作为激活函数，同时使用 L_2 损失函数。损失函数定义如下:

$$\begin{aligned}\mathcal{L}_{\text{CTC}} &= -\log \sum_{\hat{p},\mathcal{B}(\hat{p})=p} \prod_{t=0}^{T-1} \mathbb{P}(\hat{p}_t|X)\,,\\ \mathcal{L}_{\text{recon}} &= \|\hat{X}-\tilde{X}\|^2\,,\\ \mathcal{L}_{\text{PPG}} &= \mathcal{L}_{\text{CTC}}+\mathcal{L}_{\text{recon}}\,,\end{aligned} \tag{1}$$

其中， T是时间步总数， p 是真实音素序列， $\mathbb{P}(\hat{}p_t|X)$是时间 t 处的 PPG。

3.3. 标签平滑

与其他乐器不同，在歌唱数据集 [15]中，合成的或自动对齐的起始点/止音标签几乎无法获得。此外，由于歌唱起始点在时间上较为柔和，具有柔和的起始点，要通过波形或梅尔频谱图定位歌唱的精确起始点位置绝非易事。起始点的这种柔和性是使歌声起始点更难训练的因素之一。以往逐帧起始检测研究 [6, 7] 延长了起始点标签的持续时间，以解决这一问题。

根据上述研究，我们还使用了一种平滑方法来增加起始点和止音标签的长度。具体而言，我们对一维独热起始点标签序列$y_{\text{on}} := y_{\text{on}}[n]$ （n表示时间索引）和止音标签序列 y_{off} : $= y_{\text{off}}[n]$ 与经过缩放的三角窗函数 $w_{\text{tri}}[n]$ 进行线性卷积，从而同时提高精度。三角函数的缩放因子 N 表示非零值的帧数。为了使平滑后的标签中心值为 1，我们仅将奇数用作缩放因子 N。卷积过程表示为

$$\begin{aligned}w[n] &= \begin{cases}1-\left|\frac{n}{(N+1)/2}\right| & \text{if } |n| \le \frac{(N+1)}{2}\\ 0 & \text{otherwise.}\end{cases}\\ y_{\text{on_s}}[n] &= y_{\text{on}}[n] * w_{\text{tri}}[n]\\ y_{\text{off_s}}[n] &= y_{\text{off}}[n] * w_{\text{tri}}[n]\end{aligned} \tag{2}$$

其中，运算 $*$ 表示线性卷积， n 表示帧索引。

3.4. 音符解码

为了从预测输出中找出起始点的位置，我们设定一个固定阈值，并将高于该阈值的最大值所在帧设为起始点的位置。在确定音符的止音时，我们首先在当前起始时间与下一个起始时间之间寻找止音候选点。止音候选点可以是止音预测中最高的峰值，也可以是激活值

训练数据集 评估数据集		SSVD v2.0 ISMIR2014		SSVD v2.0		经 CSD 优化的		CSD-精炼版 ISMIR2014		SSVD v2.0	
	特征	COn	COff	COn	COff	COn	COff	COn	COff	COn	COff
(a) 单个卷积循环神经网络	X	0.8244	0.7751	0.8956	0.8983	0.9797	0.9719	0.8812	0.7524	0.8866	0.8007
(b) 双卷积循环神经网络 +一个循环神经网络	X, X	0.9133	0.8513	0.9486	0.9566	0.9888	0.9838	0.9004	0.7636	0.8988	0.8089
(c) 单个卷积循环神经网络	$\hat{P}$	0.8655	0.7776	0.9223	0.9105	0.9890	0.9660	0.9048	0.7685	0.9063	0.8296
(d) 双卷积循环神经网络 +一个循环神经网络	$\hat{P}, \hat{P}$	0.9094	0.8310	0.9342	0.9470	0.9907	0.9638	0.9090	0.7733	0.9142	0.8336
(e) 双卷积神经网络 + 一个循环神经网络	$X, \hat{P}$	0.9024	0.8349	0.9439	0.9420	0.9877	0.9791	0.9016	0.7852	0.9098	**0.8340**
(f) 双卷积神经网络 + 两个循环神经网络	$X, \hat{P}$	0.9230	0.8538	0.9496	0.9531	0.9914	0.9839	**0.9150**	**0.7804**	**0.9199**	0.8328
(g) 双卷积循环神经网络 +一个循环神经网络	$X, \hat{P}$	**0.9305**	**0.8576**	**0.9569**	**0.9692**	**0.9923**	**0.9864**	0.9145	0.7723	0.9166	0.8257

表 1。采用两种输入特征的各种神经网络架构的起音/止音检测结果。X 和 $\hat{}P$ 分别表示梅尔频谱图和 PPG。(g) 对应于图 2 中的神经网络架构。

预测值低于 0.5 的时间帧。如果存在多个止音候选点，我们将最晚的候选点设为止音点。如果没有找到止音候选点，则将下一个起始点所在的时间帧设为该音符的止音点。起始点和止音的阈值均设为 0.2。为了确定阈值，我们使用 0.1 至 0.9 之间、间隔为 0.1 的阈值评估验证集，以找出最优阈值。

对于音符级歌唱转录，我们按照 [6]，根据音符片段逐帧的基频（F0）估计音符级音高，以确定音符的音高。我们使用 PYIN 算法 [16], 提取基频，该算法是最准确的音高跟踪器之一。为了将基频轮廓压缩为音符级音高，我们采用加权中位数算法，该算法会在按给定权重排列的元素中找到第 50 百分位数。在本实验中，我们使用归一化的汉宁窗函数作为加权中位数的权重，其长度与音符片段的帧数相同，以减小边界附近基频的影响，因为边界处是表现力最丰富的部分。由于所有权重值之和应为 1，因此通过将汉宁窗函数除以窗元素之和来进行归一化。

4. 实验

4.1. 数据集

我们使用 SSVD v2.0 作为主要数据集 [8]。其中包含多段视唱录音，训练集和验证集共有 67 个歌唱音频文件，测试集有 127 个音频文件。人工标注内容包括起始点、止音和平均音符音高。为了在给定音频的同时使用音素标签和音符标签，我们还使用了 CSD 数据集 [17], 中的 50 首韩语歌曲，这些歌曲均包含一位女性专业歌手的音符和音素标签。由于 CSD 的原始音符标注旨在用于歌声合成，我们发现这些标注需要进一步完善，才能用于音符转录任务。因此，我们按照 [3]提出的规则，重新标注了 CSD 中的 50 首歌曲，用于本实验。CSD 的重新标注标签可在我们的 GitHub 页面上找到 [1]。完善后的 CSD 数据集按 35、5 和 10 首歌曲分别划分为训练集、验证集和测试集。

为训练音素分类器，我们使用了 TIMIT [18]，其中包含带有时间对齐音素标签的英语语音，用于训练采用 SSVD v2.0的模型。TIMIT 包含5.4 小时的英语语音音频。在训练音素分类器网络时，我们根据卡内基梅隆大学发音词典 [19] 将音素类型减少至39 种。对于采用 CSD 的模型，我们使用 CSD 中未对齐的音素标签进行训练。

[1]https://github.com/seyong92/CSD_重新标注

为了将所提模型的转录性能与以往研究进行比较，我们还使用了 ISMIR2014 [3] 数据集作为测试集，其中包含由成人和儿童演唱的 38 首歌曲。

4.2. 评估指标

我们使用mir_eval 库 [20] 进行起始点/止音检测和音符级转录，以评估这些模型。我们采用了 [3] 中提出的指标：COn（正确起始）的 F1 值、COff（正确结束）的F1值、COnOff（正确起始和结束）的 F1 值、COnP（正确起始和音高）的 F1 值，以及 COnPOff（正确起始、结束和音高）的F1值。我们使用了 mir_eval 的默认参数，即将起始容差设为 50ms，将结束容差设为50 ms与音符时长的0.2倍中的较大值，并将音高容差设为50音分。此外，考虑到歌唱起始点的柔和性，我们还报告了起始点/止音阈值为 100 ms 时的结果。

4.3. 训练细节

将音频文件重采样至 16 kHz 后，我们计算了 80 个频带的梅尔频谱图 X，跳跃长度为 320 个采样点（20 毫秒），快速傅里叶变换的窗口长度为 1024 个采样点。对于修改后的卷积网络 模块，我们为卷积层分别设置了 48/48/96 个节点，并为全连接层设置了 768 个节点。所有双向长短期记忆层均使用 768 个节点，并将音符起始/激活检测器中的最后一个全连接层设置为两个独立节点，分别用于起始点检测和激活检测。对于标签平滑，我们使用了 5 的缩放因子，将标签长度扩展至 100 毫秒；在我们的实验中，这一设置效果最佳。

为训练音符起音/止音检测网络，我们使用 AdamW 优化器 [21]，批量大小为 8，学习率为 1e-6。每 1000 步，我们按 0.98 的衰减因子降低学习率。训练期间，我们随机截取 5 秒的音频片段。每 500 步在验证集上评估一次；如果连续 10 次验证后模型没有进步，就停止训练。为训练音素分类器，我们使用 Adam 优化器，批量大小为 16，学习率为 2e-4。每 900 步，我们按 0.98 的衰减因子降低学习率。对于音素分类器，我们每 500 步验证一次；如果连续 5 次验证后模型没有进步，就停止训练。

5. 结果与讨论

5.1. 消融研究

我们进行了消融研究，以考察输入特征和模型架构的影响。图 2 所示的所提出模型

模型	正确起始 (50毫秒)			正确起始 (100毫秒)			正确结束（50毫秒）			正确结束（100毫秒）		
	P	R	F	P	R	F	P	R	F	P	R	F
TONY [22]	0.7068	0.6326	0.6645	0.8402	0.7486	0.7877	0.7862	0.6981	0.7358	0.8405	0.7471	0.7870
Omnizart [7, 23]	0.7797	0.8229	0.7951	0.8667	0.9153	0.8843	0.7698	0.8132	0.7852	0.8394	0.8842	0.8554
MusicYOLO（重新训练）[8]	0.9427	0.8970	0.9176	**0.9711**	0.9247	0.9456	**0.8924**	**0.8504**	**0.8693**	**0.9476**	0.9024	0.9227
所提出的方法	**0.9448**	**0.9188**	**0.9305**	0.9652	**0.9387**	**0.9506**	0.8701	0.84730.8576		0.9429	**0.9176**	**0.9290**

表2。ISMIR2014上的起音/止音检测结果。Musicyolo和所提出的模型均使用SSVDv2.0进行训练。Omnizart是一个预训练的音符转录模型包（未使用SSVDv2.0）。TONY是一款用于音高和音符转录的免费开源应用程序。

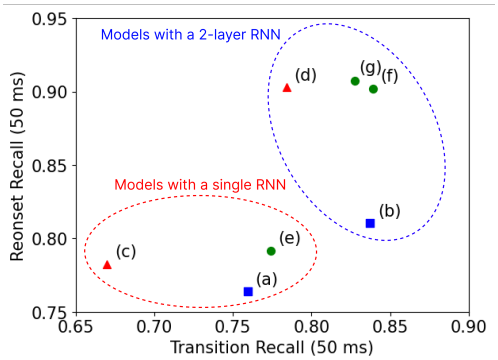

图 3. ISMIR2014消融研究中各模型的过渡和再起音召回率。红色三角形表示使用梅尔频谱图的模型，蓝色方块表示使用PPG的模型，绿色圆形表示同时使用两种特征的模型。

对应于 (g) 中的“Dual CRNNs + one RNN”。我们首先将其与仅使用一种特征类型（(a) 中的梅尔频谱图或 (c) 中的 PPG）的单个 CRNN 模型进行比较。考虑到模型架构会影响性能，我们还将所提出的模型与相同的“Dual CRNNs + one RNN”进行比较，但后者的两个输入均使用同一种输入特征（(b) 中的梅尔频谱图或 (d) 中的 PPG）。以所提出的模型为基础，我们还移除了 (e) 中每个 CRNN 分支内的 RNN 模块，然后在 (f) 中的 (e) 上方再堆叠一个 RNN 模块。

表 1 展示了所有对比模型的起音/止音检测结果。仅使用一种输入特征的单个卷积循环神经网络模型（a）和（c），其准确率明显低于所提出的模型（g）。使用 CSD 训练时，性能差距相对较小。有趣的是，使用 PPG 的单个卷积循环神经网络模型始终优于使用梅尔频谱图的模型。对比（b）、（d）和（g）中采用相同模型架构、但输入特征不同的结果可知，同时使用梅尔频谱图和 PPG 比单独使用其中一种更有效。不过，这种差距小于与（a）和（c）中的单个卷积循环神经网络模型对比时的差距。这表明模型架构对于提升性能也很重要。同样，（e）、（f）和（g）的结果表明，神经网络的设计选择会影响性能。由于 CSD 是一个小型数据集，所提出的模型容易对其过拟合。总体而言，（g）中的所提出模型表现最佳。

我们进一步通过考察两种特殊起始点的召回准确率，研究输入特征的影响：再起音和过渡。它们都是与前一个音符的止音相隔 20 毫秒或更短的音符起音。两者的区别在于音高是否发生变化（过渡）或保持不变（再起音）。当歌词中的音节或能量发生变化，但音高保持不变时，通常会出现再起音。请注意，由于我们的模型并不

	ISMIR2014		SSVD v2.0	
模型	COnP	COnPOff	COnP	COnPOff
Tony [22]	0.6009	0.4621	0.7311	0.6794
Omnizart [7, 23]	0.6174	0.4992	0.6047	0.5151
提出的模型	**0.8975**	**0.7728**	**0.8558**	**0.8303**

表3。在 ISMIR2014 和 SSVD v2.0 上的音符转录结果。所提出的模型使用 SSVD v2.0 进行训练。

预测起始类型时，只能计算召回准确率。如图3所示，使用梅尔频谱图的模型（红色）倾向于检测出更多过渡，表明其对音高变化更加敏感。另一方面，使用PPG的模型（蓝色）倾向于检测出更多重新起始，说明其能够很好地捕捉语音变化。最后，同时使用两种特征的模型在过渡和重新起始上的准确率更加均衡。演示示例、更多分析以及预训练模型可在配套网站上获取。 [2]

5.2. 与先前工作的比较

表2展示了在ISMIR2014数据集上与先前工作的比较，该数据集已被广泛用于歌声起始/终止检测（或音符分割）。为进行公平比较，我们使用与所提出模型相同的数据集重新训练了一个最新的最先进模型 [8]。我们提出的模型在两种容差下的起始F值均优于最先进模型，但在50ms容差下的终止F值略低。公开可用的音符转录软件（TONY）和模型包（Omnizart）的准确率明显低于这两个模型。最后，为了解包含音高信息的歌唱音符转录性能，我们在表3中测量了ISMIR2014和SSVDv2.0上的COnP和COnPOff。结果表明，所提出的模型始终取得了优于TONY和Omnizart的性能。

6. 结论

我们提出了一种用于音符级歌声转录的神经网络架构，该架构利用了梅尔频谱图之上的PPG。通过消融研究，我们结合两种输入特征考察了多种架构，结果表明，额外的音素信息有助于起音/止音检测。此外，我们还表明，所提出的模型在ISMIR2014和 SSVDv2.0 上优于对比模型。未来，我们计划探索能够在大规模数据集上有效处理来自噪声旋律和歌词标签的弱监督的模型 [24]。

[2]https://seyong92.github.io/phoneme-informed-transcriptio n-blog/

7. 参考文献

[1] Curtis Hawthorne、Erich Elsen、Jialin Song、AdamRoberts、IanSimon、Colin Raffel、Jesse Engel、SageevOore和 Douglas Eck，《起始点与帧：双目标钢琴转录》，载于第*19*届国际音乐信息检索学会会议论文集，法国巴黎，2018年，第50–57页。[2] Qiuqiang Kong、Bochen Li、XuchenSong、Yuan Wan和YuxuanWang，《通过回归起始点和止音时间实现带踏板的高分辨率钢琴转录》，*IEEE/ACM*音频、语音与语言处理汇刊，第29卷，第3707–3717页，2021年10月。[3] Emilio Molina、Ana M. Barbancho、Lorenzo J.Tardón 和Isabel Barbancho，《自动歌声转录的评估框架》，载于第*15*届国际音乐信息检索学会会议论文集，台湾台北，2014年，第567–572页。[4] Simon Dixon，《重新审视起始检测》，载于第*9*届数字音频效果国际会议论文集，加拿大蒙特利尔，2006年，第133–137页。[5] Ryo Nishikimi、Eita Nakamura、Satoru Fukayama、Masataka Goto 和 KazuyoshiYoshii，《基于带有弱监督注意力机制的编码器-解码器循环神经网络的自动歌声转录》，载于*IEEE* 声学、语音与信号处理国际会议，英国布莱顿，2019年，第161–165页。[6] Zih-SingFu 和 Li Su，《用于歌声分割与转录的层次分类网络》，载于第*20*届国际音乐信息检索学会会议论文集，荷兰代尔夫特，2019年，第900–907页。[7] Jui-YangHsu和 LiSu，《VOCANO：用于多声部音乐中歌声的音符转录框架》，载于第*22*届国际音乐信息检索学会会议论文集，线上，2021年，第293–300页。[8] Xianke Wang、Wei Xu、WeimingYang 和 WenqingCheng，《Musicyolo：一种基于目标检测而非频谱帧的视唱起音/止音检测框架》，载于*IEEE*声学、语音与信号处理国际会议，新加坡，2022年，第396–400页。[9] Yukun Li、Emir Edmirel、PolinaProutskova 和Simon Dixon，《单声部声乐的音素引导的音符分割》，载于第*2*届音乐与口语音频自然语言处理研讨会（*NLP4MusA*）论文集，线上，2021年，第17–21页。[10] Rainer Kelz、Matthias Dorfer、Filip Korzeniowski、Sebastian Böck、AndreasArzt 和Gerhard Widmer，《逐帧方法在钢琴转录中的潜力》，载于第*17*届国际音乐信息检索学会会议论文集，奥地利林茨，2016年，第475–481页。[11] Taegyun Kwon、DasaemJeong和 Juhan Nam，《使用自回归多状态音符模型的多声部钢琴转录》，载于第*21*届国际音乐信息检索学会会议论文集，加拿大蒙特利尔，2020年，第454–461页。

[12] Alex Graves、Santiago Fernández、Faustino Gomez 和 Jür- gen Schmidhuber，《连接时序分类：使用循环神经网络对未分段序列数据进行标注》，载于第*23*届国际机器学习会议论文集，美国宾夕法尼亚州匹兹堡，2006年，第369–376页。

[13] Ha.sim Sak、Andrew Senior、Kanishka Rao、Ozan Irsoy、Alex Graves、Françoise Beaufays 和 Johan Schalkwyk，《学习声学 ic frame labeling f或语音 h 识别 ition with 循环神经网络》，载于*IEEE*国际声学、语音与信号处理会议，澳大利亚昆士兰州南布里斯班，2015年，第4280–4284页。[14] Yann Teytaut 和 Axel Roebel，《利用循环神经网络实现说话与歌唱声音的音素与音频对齐》，载于*Interspeech 2021*，捷克布尔诺，2021年，第61–65页。[15] Curtis Hawthorne、Andriy Stasyuk、Adam Roberts、Ian Simon、Cheng-Zhi Anna Huang、Sander Dieleman、Erich Elsen、Jesse Engel 和 Douglas Eck，《实现因子化钢琴音乐建模与生成：MAESTRO数据集》，载于国际学习表征会议，美国路易斯安那州新奥尔良，2019年。[16] Matthias Mauch 和 Simon Dixon，《PYIN：一种利用概率阈值分布的基频估计器》，载于*IEEE*国际声学、语音与信号处理会议，意大利佛罗伦萨，2014年，第659–663页。[17] Soonbeom Choi、Wonil Ki m、Saebyul Park、Sangeon Yong 和 Juhan Nam，《用于歌唱声音研究的儿童歌曲数据集》，载于*ISMIR*最新研究与演示论文，加拿大蒙特利尔，2020年。[18] John S. Garofolo、Lori F. Lamel、William M. Fisher、Jonathan G. Fiscus、David S. Pallett、Nancy L. Dahlgren 和 Victor Zue，《TIMIT声学-语音连续语音语料库》，*LDC93S1*，费城：语言数据联盟，1993年。[19]《卡内基梅隆大学发音词典》，http://www.speech. cs.cmu.edu/cgi-bin/cmudict，访问日期：2022年10月22日。[20] Colin Raffel、Brian McFee、Eric J. Humphrey、Justin Sala- mon、Oriol Nieto、Dawen Liang 和 Daniel P. W. Ellis，《mireval：常用音乐信息检索指标的透明实现》，载于第*15*届国际音乐信息检索学会会议论文集，台湾台北，2014年，第367–372页。[21] Ilya Loshchilov 和 Frank Hutter，《解耦权重衰减正则化》，载于国际学习表征会议，美国路易斯安那州新奥尔良，2019年。[22]Matthias Mauch、Chris Cannam、Rachel Bittner、Geroge Fazekas、Justin Salamon、Jiajie Dai、Juan Bello 和 Simon Dixon，《使用Tony软件进行计算机辅助旋律音符转录：准确性与效率》，载于声音与音乐计算会议论文集，爱尔兰梅努斯，2015年。[23] Yu-Te Wu、Yin-Jyun Luo、Tsung-Ping Chen、I-Chieh Wei、Jui- Yang Hsu、Yi-Chin Chuang 和 Li Su，《Omnizart：用于自动音乐转录的通用工具箱》，《开放源代码软件期刊》，第6卷，第68期，第3391页，2021年12月。[24] Gabriel Meseguer-Brocal、Alice Cohen-Hadria 和 Geoffroy Peeters，《DALI：一个利用教师-学生机器学习范式自动创建的同步音频、歌词与音符大型数据集》，载于第*19*届国际音乐信息检索学会会议论文集，法国巴黎，2018年，第431–437页。