一种用于复音音符转录和多音高估计的轻量级乐器无关模型

Rachel M. Bittner[,*Juan José Bosch*[,*David Rubinstein*[,*Gabriel Meseguer-Brocal*],*Sebastian Ewert*[

[Spotify,]IRCAM

摘要

自动音乐转录(AMT)已被公认为具有广泛应用的关键使能技术。考虑到任务的复杂性，最好的结果通常是专注于特定设置的系统，例如，特定于乐器的系统往往比与乐器无关的方法产生更好的结果。同样，当只估计帧级f_0值并忽略较难的音符事件检测时，可以获得更高的精度。尽管准确率很高，但这种专门的系统往往无法在现实世界中部署。存储和网络的限制禁止使用多个专用模型，而内存和运行时的限制限制了它们的复杂性。本文中，我们提出了一种用于乐器转录的轻量级神经网络，支持复音输出，并泛化到各种各样的乐器(包括人声)。我们的模型被训练为联合预测帧级起音、多音高和音符激活，我们实验表明，这种多输出结构提高了由此产生的帧级音符精度。尽管它很简单，但基准测试结果表明，我们的系统的笔记估计比可比的基线要好得多，其帧级精度仅略低于专业的最先进的AMT系统。通过这项工作，我们希望鼓励社区进一步研究低资源，仪器不可知的AMT系统。

索引术语-自动音乐转录，音符估计，多音高估计，复调，低资源

1. 介绍

音乐的自动抄写已经研究了四十多年[1]。在此期间，特别是深度学习兴起后，系统得到了相当大的改进。然而，这项任务仍然没有解决，部分原因是各种内在挑战[1]，但也由于缺乏一个客观的基本事实，人类一致同意[2]。由于任务本身的难度，AMT系统通常被设计在一个有限的范围内，并专注于一个子任务。在AMT中有许多常见的子任务，它们沿着三个维度分支:(1)输出复调的程度(单音，复调)(2)要估计的输出类型(音符，f_0))和(3)输入音频的类型(流行歌曲，钢琴独奏，吉他独奏，爵士合奏等)。例如，专门针对特定的乐器类允许模型利用特定于乐器的特征来提高转录的准确性，例如钢琴[3-5]、吉他[6,7]或歌声[8,9]。同样，为估计特定输出类型而建立的模型，或仅限于单音设置[10]的模型，可以进一步提高这些场景中的准确性。在许多现实世界的应用中，部署大量的专用系统变得棘手，例如由于存储、网络和维护的限制。此外，对于许多仪器来说，创建一个足够大的数据集来训练现代方法是具有挑战性的。应用程序还可以对模型的大小、(峰值)内存消耗和运行时间添加额外的限制。因此，在最新发布的技术水平和实际可以在一系列设置中部署的模型之间往往存在差距。

在这项工作中，我们考虑了一个广泛的场景:一个与乐器无关的[1]复调AMT模型，它可以估计音符和多音高输出。所提出的模型是一个轻量级的神经网络，由于其低内存和处理时间要求，可以在低端设备上高效运行。除非另有说明，我们处理单一乐器类的复调录音(例如钢琴独奏、小提琴合奏、声乐独奏、合唱团等)，但不限制我们考虑哪些类。它被联合训练以预测帧级起始、多音高和音符后验图。在推理过程中，我们对帧级后验图进行后处理，以获得音符事件和多基音信息。我们研究了所提出的模型在不重新训练的情况下转录各种乐器和人声的能力，并与最近的一个与乐器无关的复音音符估计的基线模型进行了比较。此外，我们通过消融研究评估了所提出模型的组件的贡献。本文讨论的所有代码和训练过的模型都公开了[2]。此外，我们仅使用公共数据集进行训练和评估，以促进可重复性。

2. 背景及相关工作

在AMT方面有大量的工作。由于篇幅限制，我们参考[1,11]以获得更全面的概述。如前所述，AMT系统有三个维度:(1)考虑输出复音的程度，(2)估计输出的类型和(3)输入音频的类型。在这项工作中，我们考虑了复音设置，其中一个以上的音符/音高可能同时出现在输出中;请注意，单音AMT是复音AMT的严格子集，因此我们也支持单音源。AMT输出通常是帧级多音高估计(MPE)或音符级估计，它们转录不同粒度级别的复调音乐[1]。两者都是有用的，这取决于应用程序:MPE提供较低层次的表达性能信息(如颤音、滑音)，而音符级估计提供更接近乐谱的信息。MPE方法预测在给定时间范围内活跃的基频(f_0s)(请注意，即使不是严格等效，我们也可以根据该领域的文献[1]互换使用基音和f_0)。它们通常首先估计一个基音后验图[12,13]，其中每个时频箱被分配一个在给定时间基频活跃的可能性的估计。这样的矩阵

1 By “instrument agnostic” we mean “not specific to an instrument class”.

2 https://github.com/spotify/basic-pitch

通常每个半音包含多个箱子，这允许估计音高的小(“连续”)变化。各种方法旨在估计并随后将来自复调录音的MPE输出分组为音符事件[14-18]，或者尝试将音高分组为轮廓[11,12,19]。音符估计(或音符跟踪)方法旨在估计音符事件(定义为:音高、起始时间、偏移时间)。音符不能从MPE系统的输出中轻易地估计出来，因为MPE信息不编码起始/偏移，并且保留了音调的波动，不应该总是量化到最接近的半音。特别是，音符估计对于歌唱声音来说是困难的，与钢琴等乐器相比，歌唱声音在中心音高[9]周围可能有高度的波动。已有多种方法被提出用于从音高后验图估计音符，例如使用中值滤波[11]，隐马尔可夫模型[16]或神经网络[20,21]。虽然大多数方法都独立考虑每个半音，但有些方法试图使用谱似然模型[1,18]或音乐语言模型[3,17]对音符之间的相互作用进行建模。变形金刚最近被应用于AMT，从钢琴音乐的谱图中直接预测类似midi的音符事件[5]。一些AMT模型同时执行音符和音高估计[14,18,22]，并且大多数使用单音数据。关于输入音频特性，传统的基于信号处理的AMT方法比最近的方法更适用于多种乐器，并且更简单、更快[1,12]。然而，性能最好的系统往往是以更高的计算要求和对仪器特定系统[4]的关注为代价的。

3. 模型

我们的目标是创建一个AMT模型，该模型可以泛化一组复调(或单音)乐器，而无需重新训练，同时足够轻量，可以在低资源环境中运行。我们在运行推理时同时考虑了速度和峰值内存使用，并故意将自己限制在一个浅层架构中，以保持低内存需求。注意，模型的参数数量并不一定与它的内存使用量相关;例如，卷积层需要很少的参数，但由于特征图的大小，仍然可以有较高的内存使用率。

谐波叠加。给定输入音频，该模型首先计算一个常数q变换(Constant-Q Transform, CQT)，每个半音有3个箱子，跳大小≈11 ms。我们不是使用，例如梅尔语谱图，并最终使用密集或LSTM层(这要求模型具有全频率感受野)[4]学习到输出对数间隔频率尺度上的投影，而是从具有所需频率尺度的表示开始。谐波CQT (HCQT)[13]是CQT的一种变换，它沿着第三维对齐谐波相关的频率，允许小卷积核捕获谐波相关的信息。作为HCQT的有效近似，在[23]之后，我们复制CQT并通过每个谐波对应的频率箱数垂直移动它。在这项工作中，我们使用了7次谐波和1次谐波。

体系结构。图1所示的架构是一个以音频为输入，并产生三个后验图输出的全卷积模型，总共只有16782个参数。该模型的三个输出后验图是时频矩阵编码，如果(1)一个音符的开始发生(Y_o)(2)一个音符是活跃的(Y_n)和(3)一个音高是活跃的(Y_p)。所有输出都具有与输入CQT相同的时间帧数量，并且在频率上，Y_o和Y_n的分辨率都为每个半音1个bin，而Y_p的分辨率为每个半音3个bin。除了具有不同的频率分辨率外，Y_n和Y_p被训练来捕获不同的概念:Y_n捕获帧级音符事件

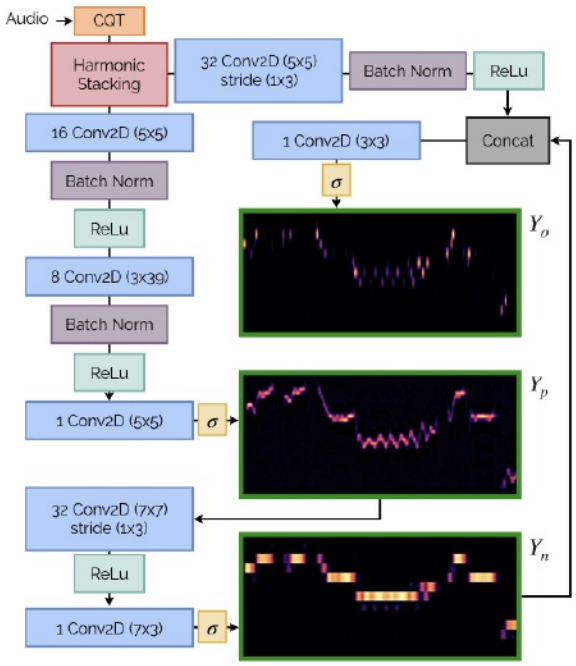

图1所示。NMP架构。矩阵后验图输出Y_o，Y_p，和Y_n用绿色概述。σ表示sigmoid激活。

在时间和频率上“音乐量化”的信息，而Y_p编码帧级的多基音信息，捕捉颤音等细节。在训练过程中，这些输出的每一个目标都是由音符和音高注释生成的二进制矩阵。

该架构是结构化的，以便利用三个输出的不同属性。我们假设Y_p是与输入音频“最接近”的输出。估计Y_p的架构类似于[13]，但具有更少的卷积层以减少内存使用。值得注意的是，我们在频率上采用了相同的octave +一个半音阶大小的内核，我们发现这有助于避免octave错误。这一堆卷积执行了一种形式的“去噪”，以强调多基音后验输出，并去强调瞬态、谐波和其他非基音内容。在频率上使用有限的感受野的一个额外好处是，它消除了对基音变换数据增强的需要。Y_p随后的两个小卷积层被用来估计Y_n。这些卷积可以看作是“音乐量化”层，学习如何将多基音后验图进行非平凡的分组，转化为音符事件后验图。最后，与[24]一样，Y_o是使用Y_n和从音频中计算出的卷积特征作为输入来估计的，这些特征对于识别瞬态是必要的。

培训。使用二进制交叉熵作为每个输出的损失函数，总损失是三个损失的总和。然而，对于Y_o来说，存在严重的类不平衡，驱动模型到处输出Y_o= 0。作为对策，我们使用类平衡的交叉熵损失，其中负类的权重为0.05，正类为0.95(通过观察结果Y_o)的属性来经验设置，这有助于模型在保持稀疏的同时捕获发病。在训练过程中，模型输入是22050 Hz采样率下的2秒音频。我们用16的批量大小训练模型，并使用学习率为0.001的Adam优化器。在训练过程中，对音频应用随机标签保持增强，包括添加噪声、均衡化滤波器和混响。

Posteriorgram后处理。类似于许多音符或轮廓创建后处理方法，我们创建音符事件，由开始时间t^0，结束时间t^1和音高f定义，通过运行一个后处理步骤，使用Y_o和Y_n作为输入[1]，遵循类似于Onsets和Frames[4]中描述的过程。一组起始候选$\{(t^0_i, f_i)\}$通过跨时间的峰值选择Y_o来填充，并丢弃可能性< 0.5的峰值。按t^0_i降序为每个i创建Note事件，通过Y_n及时向前跟踪，直到可能性低于阈值τ_n的时间超过允许的容忍度(11帧)，然后结束Note。当创建音符时，Y_n的所有对应帧的似然更新为0。在使用了所有起始点之后，通过按降序迭代具有似然>τ_n的Y_n箱子来创建额外的笔记事件，遵循相同的笔记创建过程，但在时间上同时向前和向后跟踪。最后，删除短于≈120 ms的note事件。多音高估计是通过简单地在频率上选取Y_p峰值并保留所有大于τ_n的峰值来创建的。

4. 实验

在本节中，我们研究了所提出的方法“音符和多音高”(NMP)的性能，重点关注音符估计任务，但也简要评论了MPE任务。AMT方法通常使用为MIREX[3]评估任务提出的一组指标进行评估。在这项工作中，我们报告了音符级F-measure (F)，如果音高在四分之一音内，开始时间在50毫秒内，偏移量在音符持续时间的20%内，音符被认为是正确的，音符级F-measure-no-offset (Fno)具有与F-measure相同的标准，但忽略偏移，以及帧级音符精度(Acc)，这是为跳跃大小为10毫秒的帧计算的。我们使用Fno作为整体音符估计精度的主要度量，因为偏移的定义不如起始(例如，由于混响，维持踏板，注释过程)客观[25]。我们使用mir_eval[26]来计算这些指标。对于NMP和每个消融研究，我们在验证数据集上微调笔记创建参数τ_n，使其最大化Fno。

为了评估NMP和基线在不同仪器类别中的表现，我们使用了跨越多种仪器类型的各种各样的训练和测试数据，总结在表1中(参见被引用的论文以获得更具体的细节)，使用了mirdata[27]库。从训练集中随机抽取5%的曲目用于验证。我们注意到一些数据集的一些额外细节:我们使用重复数据删除的“redux”版本的Slakh，并在120个非冲击测试集的仪器平衡子集上进行测试，其中沉默最少;MedleyDB和iKala中的注释是使用pyin-notes自动生成的[22];MedleyDB的音频文件来自音高跟踪子集[4]，而对于iKala，我们使用孤立的人声;对于Phenicx，我们使用了42种乐器分组的茎(例如小提琴，巴松管)和注释。

4.1. 音符转录基线比较

我们将我们的模型与最近的强基线模型MI-AMT[34]进行了比较，后者是一种复调、乐器不可知的音符估计方法。它使用带有注意力机制的U-Net架构，输出一个总共超过20M参数的音符激活后图，在MAESTRO和MusicNet上进行训练。音符后验图经过后处理，以便创建音符事件。

Dataset	*Polyphony*	*Instrument*	*Labels*	*Train*	*Test*
Molina [28]	Mono	Vocals	N	-	38
GuitarSet [29]	Mono / Poly	Ac. Guitar	N + P	648	72
MAESTRO [4]	Poly	Piano	N	1154	128
Slakh [30]	Poly	Synthesizers	N	1590	120
Phenicx [31]	Poly	Orchestral	N	-	42
iKala [32]	Mono	Vocals	N + P	252	-
MedleyDB [33]	Mono	Multiple	N + P	103	-

表1。使用的数据集总结。*Train*和*Test*列表示轨道的数量。标签列显示可用的注释类型:(N)音符，(P)多音调。

我们提出的方法和MI-AMT的结果如表2所示。我们首先注意到，NMP在所有测试数据集和指标上都大大优于基线MI-AMT，除了MAESTRO(钢琴)和Slakh(合成器)上的可比较Acc。NMP对复音乐器(MAESTRO, Slakh, Phenicx, GuitarSet的$^1\boxtimes_2$)和单音乐器(Molina和GuitarSet的$^1\boxtimes_2$)的数据集表现强劲，尽管没有对输出音符估计施加单音限制。此外，我们在不同仪器类型的数据集上看到了一致的性能，验证了NMP在不需要特定于仪器的情况下表现良好。

	Molina			GuitarSet			Maestro			Slakh			Phenicx		
	Acc	Fno	F	Acc	Fno	F	Acc	Fno	F	Acc	Fno	F	Acc	Fno	F
MI-AMT	.48	.31	.11	.43	.59	.27	.39	.30	.07	.40	.23	.07	.13	.12	.05
NMP	**.63**	.52	.35	**.70**	**.79**	**.56**	.38	.71	.11	**.44**	.42	.21	**.53**	.49	.35
NMP - P	.60	.55	.38	.67	.78	.55	.36	.65	.12	.40	**.43**	**.23**	.50	**.51**	**.36**
NMP - H	.45	.36	.20	.50	.65	.40	.27	.48	.10	.33	.36	.17	.37	.39	.23

表2。基准算法、提出的方法和消融实验在所有测试数据集上的平均音符事件指标。每一列的最佳得分以粗体显示。绿色阴影表示该分数与最佳分数的距离，白色表示最差分数。通过配对t检验，与NMP(每指标/数据集)相比，所有非下划线结果均有统计学显著差异，$p < 0.05$。

4.2. 消融实验

谐波叠加。为了检验谐波叠加作为输入表示的使用，我们训练了一个模型，它省略了谐波叠加层，但在其他方面是等效的，在表2中表示为NMP - H。不出意外的是，考虑到较小的感受野，谐波堆叠的省略大大降低了所有指标和数据集的性能，这与[13, 23]中进行的类似实验的结果一致。这表明谐波叠加有效地允许模型使用更小的卷积核，同时仍然捕获相关信息。这种比较的一个局限性是，当省略谐波堆叠时，通道数量减少，这反过来降低了模型的容量。

Y_p的效果。我们通过训练一个等效模型来测量Y_n上的监督瓶颈层Y_p对note估计的影响，其中Y_p是没有监督的，其中Y_n是在它之前的卷积堆栈的输出，图1中的*批Norm*→ReLu→1 Conv2D (5x5)层省略了。这种情况下的结果在表2中用NMP - P表示。我们首先看到Y_p引入的约束在所有数据集中一致地提高了Acc，但是对Fno和F的影响是混合的;GuitarSet、Slakh和Phenicx之间无显著差异，$Y_p$im-

[3] http://www.music-ir.org/mirex/

[4] https://zenodo.org/record/2620624

MAESTRO的性能略有提高，Molina的性能略有下降。这表明，即使额外的监督对起始/偏移检测是中性的，它对识别音符音高是有帮助的，我们得到了包含一些装饰和表现力信息的额外输出的好处。

4.3. 与特定乐器方法的比较

	Molina (Vocano)			GS-solo (TENT)			Maestro (OF)		
	Acc	Fno	F	Acc	Fno	F	Acc	Fno	F
Baseline	61.6	**64.2**	**51.3**	63.2	76.3	54.6	**43.8**	**95.2**	**36.4**
NMP	**62.6**	52.3	34.6	**71.7**	**84.0**	**65.0**	37.5	70.9	10.5

表3。NMP上的平均音符事件指标与人声、吉他和钢琴的特定乐器模型。每一列的最佳乐谱以粗体显示。比较的特定于工具的模型名称在括号中的列标题中指出。通过配对t检验，所有非下划线结果与NMP比较，$p < 0.05$，差异有统计学意义。

我们已经看到，所提出的模型在各种数据集上的表现优于可比的工具不可知基线。为了进一步了解我们模型的上限，我们提供了与最近的开源特定工具模型的比较。起跳和帧(OF)[4]是一种在MAESTRO数据集上训练的复调钢琴转录方法，它使用由大约18M个参数组成的CNN和RNN联合预测起跳和音符后图，然后是一个音符创建的后处理阶段。Vocano[9]是一种单音人声转录方法，它首先进行人声源分离，然后应用预训练的音高提取器，然后使用单音人声数据训练的音符分割神经网络。TENT[6]是一种单音吉他独奏转录方法，它首先进行旋律轮廓提取，然后利用CNN架构对弦弯、滑、振音等常见吉他元素进行演奏技术检测，后处理阶段根据旋律轮廓得到最终音符，并识别每个时间框架不同的演奏技术。因此，我们只报告独奏的结果，GuitarSet的单音部分。

对于吉他，NMP在所有指标上都优于TENT，更重要的是，据我们所知，这些都是GuitarSet上最先进的结果。对于人声(Molina)， Vocano在Fno和F上优于NMP，但帧级音高精度(Acc)与NMP相当，这表明Fno可以随着起音检测的改进而提高。NMP和特定乐器方法之间最大的性能差异是在MAESTRO数据集中，与专门为钢琴转录训练的OF相比，实现了95.2%的Fno，而我们的方法为70.9%(值得注意的是，这仍然是这个任务的一个相当高的分数)。性能差异的主要原因似乎是由于OF的起始检测精度更高，因为两种方法的Acc更相似(OF为42.8%，NMP为37.5%)。有趣的是，根据[24]中获得的结果，NMP将在钢琴数据上与Melodyne[5]进行竞争，即使不可能进行直接比较，因为他们报告了另一个类似钢琴数据集的结果。

4.4. MPE基线

在这里，我们简要验证了NMP在MPE下的表现，将NMP的MPE输出与深度显著性模型的输出进行了比较[13]。我们报告了Bach10[12]和Su[2]数据集的结果，每个数据集包含10个西方复调古典室内乐合奏的录音。NMP的MPE输出优于Bach10数据集的深度显著性，帧级精度为72.5±3.8，深度显著性为55.7±2.9。然而，深度显著性在Su为43.6±7.9时获得更好的结果，而NMP为37.7±15.4。虽然这是一个小规模的验证，但这些结果表明，Y_p捕获的信息是有意义的，并可能与强大的基线模型竞争。虽然对于这项任务来说，3-bin / semi -声调分辨率的后验图可能看起来分辨率相对较低，但它们可以通过使用估计的$f_0$bin的振幅值以及其相邻bin在频率上的振幅值来估计连续的多基音估计。请注意，尽管没有在多乐器混合上进行训练，但它似乎取得了令人信服的结果。

4.5. 效率

为了说明NMP的计算效率，我们将峰值内存使用和总运行时间与MI-AMT进行比较。基准测试是在2017年的Macbook Pro上进行的，配备3.1GHz四核Intel Core i7 CPU和16GB 2133MHz LPDDR3内存。所有的基准测试都是首先使用一个“短”(。35秒的白噪声文件以近似系统的开销，以及来自Slakh数据集的“长”(7分钟45秒)文件，以便为每个方法显示一个更真实的输入。在测量之前，音频文件被重采样到该方法的期望采样率。我们发现这两种方法在估计开销方面是相当的，NMP使用490 MB峰值内存，耗时7秒，MI-AMT使用561 MB峰值内存，耗时10秒;然而，在长文件上，NMP大大优于MI-AMT，仅使用951 MB峰值内存，耗时24秒，而MI-AMT使用3.3 GB，耗时96秒。有趣的是，特定于仪器的型号的峰值内存甚至更高，of使用5.4 GB, Vocano使用8.5 GB。

5. 结论

我们证明了所提出的基于低资源神经网络的模型(NMP)可以成功地应用于与乐器无关的复调音符转录和MPE。NMP在五个不同的数据集上优于最近的强基线笔记估计模型，并且与MPE的深度显著性相似。进一步，我们看到谐波堆叠的使用允许我们的模型在保持其性能的同时保持低资源。当与特定于乐器的模型进行比较时，我们看到NMP在GuitarSet上实现了最先进的结果。然而，在钢琴和人声方面，它的表现并没有超过特定乐器模型。然而，NMP具有“一刀切”解决方案的优点，并且具有更低的计算需求。我们希望鼓励进一步研究低资源、多用途的AMT系统，并相信所提出的解决方案可以成为一个有价值的基线。

未来的工作可以探索包含许多乐器的音频混合的低资源转录，以及在这种低资源环境下使用偏移预测。所提出的音符事件创建方法是基于启发式的，更仔细设计的类似[16,17]的模型可能会导致音符事件创建的改进。虽然这项工作旨在从一开始就创建一个轻量级的模型，但我们没有探索经典的模型修剪或压缩技术，这些技术将进一步提高效率。最后，

[5] version 4.1.1.011, http://www.celemony.com/en/melodyne

音符和多音高输出之间的相互作用可以被探索，例如，估计音符级别的音高弯曲。

6. 引用

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Process. Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[2] L. Su and Y.-H. Yang, "Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription," in *Proc. CMMR*, 2015.

[3] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 24, no. 5, pp. 927–939, 2016.

[4] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAE-STRO dataset," in *Proc. ICLR*, 2019.

[5] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with transformers," *arXiv preprint arXiv:2107.09142*, 2021.

[6] T.-W. Su, Y.-P. Chen, L. Su, and Y.-H. Yang, "TENT: Technique-embedded note tracking for real-world guitar solo recordings," *TISMIR*, vol. 2, no. 1, pp. 15–28, 2019.

[7] A. Wiggins and Y. Kim, "Guitar tablature estimation with a convolutional neural network," in *Proc. ISMIR*, 2019, pp. 284– 291.

[8] A. McLeod, R. Schramm, M. Steedman, and E. Benetos, "Automatic transcription of polyphonic vocal music," *Applied Sci-ences*, vol. 7, no. 12, p. 1285, 2017.

[9] J.-Y. Hsu and L. Su, "VOCANO: A note transcription framework for singing voice in polyphonic music," in *Proc. ISMIR*, 2021.

[10] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convo-lutional representation for pitch estimation," in *Proc. ICASSP*, 2018, pp. 161–165.

[11] A. Klapuri and M. Davy, *Signal Processing Methods for Music Transcription*. Springer US2007.

[12] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 8, pp. 2121–2133, 2010.

[13] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 estimation in polyphonic music," in *Proc. ISMIR*, 2017, pp. 63–70.

[14] M. P. Ryynanen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *Proc. WASPAA*, 2005.

[15] Z. Duan, J. Han, and B. Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 1, pp. 138–150, 2013.

[16] E. Benetos, "Polyphonic note and instrument tracking using linear dynamical systems," in *Proc. AES Int. Conf. Semantic Audio*, 2017.

[17] A. Ycart and E. Benetos, "Polyphonic music sequence transduc-tion with meter-constrained LSTM networks," in *Proc. ICASSP*, 2018, pp. 386–390.

[18] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii, "Bayesian singing transcription based on a hierarchical gen-erative model of keys, musical notes, and F0 trajectories," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 1678–1691, 2020.

[19] R. M. Bittner, J. Salamon, J. J. Bosch, and J. P. Bello, "Pitch contours as a mid-level representation for music informatics," in *Proc. AES Int. Conf. Semantic Audio*, 2017.

[20] S. Ewert and M. B. Sandler, "An augmented Lagrangian method for piano transcription using equal loudness thresholding and LSTM-based decoding," in *Proc. WASPAA*, 2017, pp. 146–150.

[21] R. Nishikimi, E. Nakamura, M. Goto, and K. Yoshii, "Audio-to-score singing transcription based on a CRNN-HSMM hybrid model," *APSIPA Trans. Signal Inf. Process.*, vol. 10, 2021.

[22] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon, "Computer-aided melody note transcription using the Tony software: Accuracy and efficiency," in *Proc. Int. Conf. Tech. Music Notation Representation*, 2015.

[23] J. Balhar, "Melody extraction using a harmonic convolutional neural network," in *Proc. ISMIR*, 2018, p. 4.

[24] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and Frames: Dual-objective piano transcription," in *Proc. ISMIR*, 2018, pp. 50–57.

[25] C.-Y. Liang, L. Su, Y.-H. Yang, and H.-M. Lin, "Musical offset detection of pitched instruments: The case of violin," in *Proc. ISMIR*, 2015, pp. 281–287.

[26] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "mir_eval: A transparent implementation of common MIR metrics," in *Proc. ISMIR*, 2014, pp. 367–372.

[27] M. Fuentes, R. Bittner, M. Miron, G. Plaja, P. Ramoneda, V. Lostanlen, D. Rubinstein, A. Jansson, T. Kell, K. Choi, and et al., "mirdata v.0.3.0," Jan 2021.

[28] E. Molina, A. M. Barbancho-Perez, L. J. Tardon-Garcia, I. Barbancho-Perez *et al.*, "Evaluation framework for automatic singing transcription," in *Proc. ISMIR*, 2014, pp. 567–572.

[29] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "Guitarset: A dataset for guitar transcription." in *Proc. ISMIR*, 2018, pp. 453–460.

[30] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cut-ting music source separation some Slakh: A dataset to study the impact of training data quality and quantity," in *Proc. WASPAA*, 2019, pp. 45–49.

[31] M. Miron, J. J. Carabias-Orti, J. J. Bosch, E. Gómez, and J. Janer, "Score-informed source separation for multichannel orchestral recordings," *Jour. of Electrical Computer Engineering*, vol. 2016, 2016.

[32] T.-S. Chan, T.-C. Yeh, Z.-C. Fan, H.-W. Chen, L. Su, Y.-H. Yang, and R. Jang, "Vocal activity informed singing voice separation with the ikala dataset," in *Proc. ICASSP*, 2015, pp. 718–722.

[33] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research." in *Proc. ISMIR*, 2014, pp. 155–160.

[34] Y.-T. Wu, B. Chen, and L. Su, "Multi-instrument automatic music transcription with self-attention-based instance segmenta-tion," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2796–2809, 2020.