AUTOMATIC CONVERSION OF POP MUSIC INTO CHIPTUNES FOR 8-BIT PIXEL ART

Shih-Yang Su[1,2], *Cheng-Kai Chiu*[1,2], *Li Su*[1], *Yi-Hsuan Yang*[1]

[1]Research Center for Information Technology Innovation, Academia Sinica, Taiwan,
[2]Department of Computer Science, National Tsing Hua University, Taiwan

ABSTRACT

In this paper, we propose an audio mosaicing method that converts Pop songs into a specific music style called "chiptune," or "8-bit music." The goal is to reproduce Pop songs by using the sound of the chips on the old game consoles in 1980s/1990s. The proposed method goes through a procedure that first analyzes the pitches of an incoming Pop song in the frequency domain, and then synthesizes the song with template waveforms in the time domain to make it sound like 8-bit music. Because a Pop song is usually composed of the vocal melody and the instrumental accompaniment, in the analysis stage we use a singing voice separation algorithm to separate the vocals from the instruments, and then apply different pitch detection algorithms to transcribe the two separated sources. We validate through a subjective listening test that the proposed method creates much better 8-bit music than existing nonnegative matrix factorization based methods can do. Moreover, we find that synthesis in the time domain is important for this task.

Index Terms— Audio mosaicing, chiptune, synthesis

1. INTRODUCTION

Chiptune music, or the so-called 8-bit music, is an old style music that were widely used in the 1980s/1990s game consoles, with the theme song of the classic game *Super Mario Bros.* being one good example.[1] The music consists of simple waveforms such as square wave, triangular wave and saw wave. Although the game consoles in the old days have faded away, the chiptune music style does not disappear [1, 2]. Actually, recent years have witnessed a revival of interests in old pixel art [3, 4], both in the visual and audio domain.[2] Chiptune style has begun to reclaim its fames in the entertainment and game industry, and many people have been publishing hand-crafted 8-bit version of Pop songs online.[3]

Being motivated by the above observations, we are interested in developing an automatic process that converts existing Pop music to chiptunes by signal processing and machine learning techniques. This task can be considered related to an *audio antiquing* problem [5, 6], which aim to simulate the degradation in audio signals like those in the old days, and also, an instance of the so called *audio mosaicing* problem [7–12]. However, no attemps have been made to tackle this task thus far, to the best of out knowledge.

In general, the goal of audio mosaicing is to transfer a given audio signal (i.e. the target) with sound of another audio signal (i.e. the source). An example is to convert a human speaking voice into the barking sound of a dog. In this example, the human sound is the target, while the sound of dog is the source. Our task is also an audio mosaicing problem, but in our case the aesthetic quality of the converted sound is important. On the one hand, we require that the converted song is composed of only the sounds of simple waveforms that appear in the old game consoles. On the other hand, from the converted song the main melody of the target song needs to be recognizable, the converted song needs to sound like a 8-bit music, and it should be acoustically pleasing.

To meet these requirements, we propose a novel analysis/synthesis pipeline that combines state-of-the-art algorithms developed in the music information retrieval (MIR) community for this task. In the analysis stage, we firstly use a singing voice separation algorithm to highlight the vocal melody, and then use different pitch detection algorithms to transcribe the vocal melody and the instrumental accompaniments. In the synthesis stage, we firstly perform a few post-processing steps on the transcribed pitches to reduce the complexity and unwanted fluctuations due to errors in pitch estimation. We then use templates of simple waveforms to synthesize an 8-bit music clip based on given the pitch estimates. The whole pipeline is illustrated in Fig. 1 and details of each step will be described in Section 2.

We validate the effectiveness of the proposed method over a few existing general audio mosaicing methods through a subjective listening test. The human subjects were given the original version and the automatically generated 8-bit versions of a few Pop songs and were asked to rate the quality of the 8-bit music using three criteria corresponding to pitch accuracy, 8-bit resemblance, and overall quality. Experimental result presented in Section 3 shows that automatic 8-bit music conversion from Pop music is viable.

[1]Audio file online: `https://en.wikipedia.org/wiki/File:Super_Mario_Bros._theme.ogg` (last accessed: 2016-12-23).

[2]For example, pixel art is used in SIGGRAPH 2017 as their visual design: `http://s2017.siggraph.org/` (last accessed: 2016-12-23).

[3]Audio files online: `https://soundcloud.com/search?q=8bit` (last accessed: 2016-12-23).

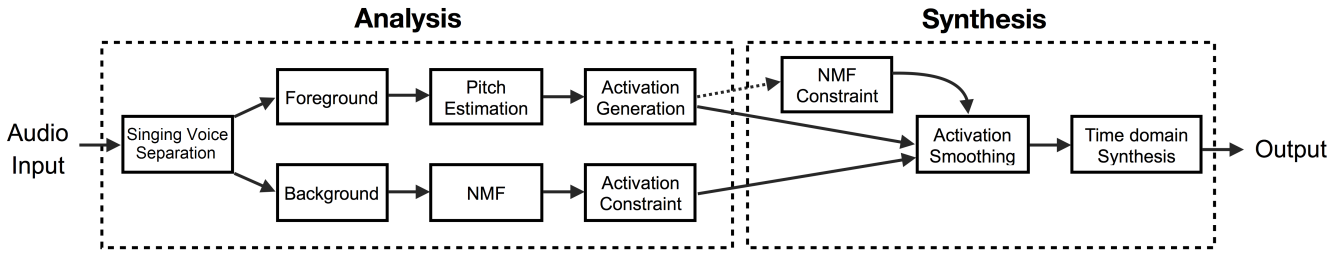


Fig. 1. System diagram of the proposed method for 8-bit music conversion.

1.1. Related Work on Audio Mosaicing

Many methods have been proposed for audio mosaicing. The *feature-driven synthesis* method [7–9] splits the source sound into short segments, analyzes the feature descriptors of the target sound such as temporal, chroma, mel-spectrogram characteristics, and then concatenates those segments by matching those feature descriptors. On the other hand, the *corpus-based concatenative synthesis* method [10–12], selects sound snippets from a database according to a target specification given by example sounds, and then use the concatenative approach [13, 14] to synthesize the new clip.

More recently, methods based on *non-negative matrix factorization* (NMF) [15] become popular. NMF decomposes a non-negative matrix $\mathbf{V} \in \mathbb{R}_{\geq 0}^{m \times n}$ into a *template* matrix $\mathbf{W} \in \mathbb{R}_{\geq 0}^{m \times k}$ and an *activation* matrix $\mathbf{H} \in \mathbb{R}_{\geq 0}^{k \times n}$, such that $D(\mathbf{V}|\mathbf{WH})$ is small, where $D(\cdot|\cdot)$ is a distortion measure such as the β-divergence [16] and ≥ 0 denotes non-negativity. For audio, we can use the magnitude part of the short-time Fourier transform (STFT), i.e. the spectrogram, as the input $\mathbf{V}$; in this case, m, n and k denote respectively the number of frequency bins, time frames, and templates. Assuming that there is a *one-to-one pitch correspondence* between the pre-built template matrices $\mathbf{W}^{(s)}$ and $\mathbf{W}^{(t)}$ for the source and target sounds, given the spectrogram of a target clip $\mathbf{V}^{(t)}$ we can compute the activation by $\mathbf{H}^{(t)} = \arg\min_{\mathbf{H}} D(\mathbf{V}^{(t)}|\mathbf{W}^{(t)}\mathbf{H})$, and then obtain the mosaicked version $\widehat{\mathbf{V}}^{(s)}$ by $\widehat{\mathbf{V}}^{(s)} = \mathbf{W}^{(s)}\mathbf{H}^{(t)}$. We can then reconstruct the time-domain signal by inverse STFT, using the phase counterpart of $\mathbf{V}^{(t)}$.

The one-to-one pitch correspondence condition requires that the two templates $\mathbf{W}^{(t)}$ and $\mathbf{W}^{(s)}$ are of the same size and each column of them corresponds to the same pitch. This condition is however hard to meet if the target is a Pop song, for it involves sounds from vocals and multiple instruments. To circumvent this issue, Driedger *et al.* [17] recently proposed to use the source template $\mathbf{W}^{(s)}$ directly to approximate the input to get $\mathbf{H}^{(t)} = \arg\min_{\mathbf{H}} D(\mathbf{V}^{(t)}|\mathbf{W}^{(s)}\mathbf{H})$, and treat $\mathbf{W}^{(s)}\mathbf{H}^{(t)}$ as the synthesis result. Because our target is also Pop music, we consider this as a baseline NMF method in our evaluation. For better result, Driedger *et al.* [17] further extended this method by imposing a few constraints on the learning process of NMF to reduce repeated or simultaneous activation of notes and to enhance temporal smoothness. The resulting "Let-it-bee" method can nicely convert Pop songs into sounds of bees, whales, winds, racecars, etc [17].

Our experiments will show that neither NMF nor the more advanced Let-it-bee method provides perceptually satisfactory 8-bit conversion. This is mainly due to the specific aesthetic quality required for 8-bit music, which is less an issue for atonal or noise-like targets such as bee sounds.

2. PROPOSED METHOD

This section presents the details of each component of the proposed method, whose diagram is depicted in Fig. 1.

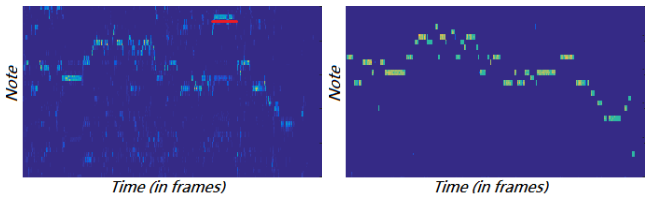
2.1. Singing Voice Separation

The vocal part and the instrumental part of a Pop song is usually mixed in the audio file available to us. As the singing voice usually carries information about the melody of a song, we propose to use singing voice separation (SVS) algorithms to separate the two sources apart.[4] This is achieved by an unsupervised algorithm called the *robust principle component analysis* (RPCA) [18, 19] in this paper. RPCA approximates a matrix (i.e. the spectrogram) by the sum of a low-rank matrix and a sparse one. In musical signals, the accompaniment is polyphonic and usually repetitive, behaving like a low-rank matrix in the time-frequency representation. In contrast, the vocal melody is monophonic and changes over time, behaving more like a sparse signal [20]. Therefore, by reconstructing the time-domain signals of the low-rank and the sparse parts, we can recover the two sources. Although there are many other algorithms for SVS, we adopt RPCA for its simplicity and well-demonstrated effectiveness in the literature [21]. If the vocal part in musical signal is centered, we instead subtract the left channel from the right one to cancel the vocal part, and thus obtain a better accompaniment signal.

2.2. Pitch Analysis of the Accompaniments (Background)

As the instrumental accompaniment is polyphonic, we can transcribe it by using any *multi-pitch estimation* (MPE) al-

[4] In the literature of auditory source separation, a 'source' refers to one of the audio signals that compose the mixture. Hence, the term 'source' here should not be confused with the term 'source' used in audio mosaicing.



(a) NMF-based estimate (b) pYIN-based estimate

Fig. 2. The pitch estimation result for the singing voice.

gorithms. In this paper, we simply use the baseline NMF method [17] for MPE. This is done by computing $\mathbf{H}^{(t)}$ from the separated instrument part of $\mathbf{V}^{(t)}$ using the template of chiptune notes $\mathbf{W}^{(s)}$. As different columns of $\mathbf{W}^{(s)}$ are built to correspond to different pitches, the resulting $\mathbf{H}^{(t)}$ provides pitch estimates. The method is adopted for its simplicity, but in our pilot study we found that false positives in the estimate would make the synthesized sound too busy and sometimes even noisy. To counter this, we impose a simple constraint on $\mathbf{H}$, assuming that the instrumental accompaniment can have at most three active notes at any given time. Specifically, for each time frame we consider only the top three pitch candidates with the strongest activations, and discard all the others by setting their activation to zero. In this way, we trade recall rate for better precision rate by having fewer false positives. As the main character in the 8-bit music should be the singing melody, it seems to be fine to downplay the instrumental part by presenting only at most three pitches at a time.

2.3. Pitch Analysis of the Singing Voice (Foreground)

The singing voice is usually monophonic (assuming only one singer per song) and features continuous pitch changes such as vibrato and glissando. As a result, NMF cannot perform well for transcribing the singing voice, as shown in Fig. 2(a). In light of this, we instead use a *monophonic pitch detection* algorithm called pYIN [22] for the separated vocal part. Assuming that any two detected pitches in consecutive frames cannot differ from each other by more than one octave, we postprocess the result of pYIN by moving the higher note in such cases one octave down. As illustrated in Fig. 2, pYIN can better capture the singing melody than NMF.

2.4. Activation Smoothing and NMF Constraint

We implement two additional post-processing steps for the aesthetic quality of the conversion result. First, we apply a median filter of width 9 frames to temporally smooth the result of pitch estimation for the vocal and instrumental parts separately. Although this smoothing process may remove frequency modulations such as vibratos in the singing voice, perceptually it seems better to suppress the effect of vibratos in the 8-bit music. Figure 3 illustrates the effect of smoothing.

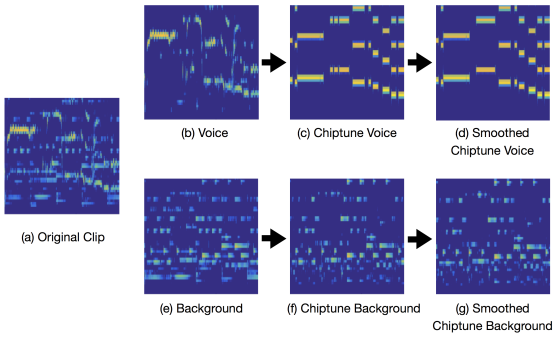


Fig. 3. The spectrograms of a song after each major step.

Second, we hypothesize that the pitch estimate of the vocal and instrumental parts might be related and it is possible to use one of them to help the other. Therefore, we try to use the pitch range determined by the pitch estimate of the instrumental part (i.e, the pitch range is set by the maximal and minimal values of the detected pitches) to set constraint on the pitch estimate of the vocal part. We refer to this as the 'NMF constraint' and will test its effectiveness in our experiments.

2.5. Time-domain Synthesis

The final synthesis makes use of a pre-recorded collection of simple narrow pulse waves and spike waves of different pitches serving as the template chiptune tones. From the result of the preceding stages, we examine every note in any given time frame to find consecutive time frames with the same notes, which are then considered as note segments. If a note segment contains only one frame, the segment will be discard. Each note segment determines a set of pitches, their amplitudes (i.e. energy), their common starting time and duration. From this information, we concatenate the template chiptune tones using overlap-and-add techniques directly in the time-domain [13, 14], with proper duration and amplitude scaling of the chiptune tones. The major benefit of synthesis in the time domain is to avoid the influence of phase errors for we are not given phase information in the synthesis stage.

3. EXPERIMENT

To evaluate the performance of 8-bit music conversion, we invited human subjects to take part in a subjective listening test. This is deemed better than any objective evaluation for the purpose of 8-bit music conversion is for the human listeners to enjoy them. We are able to recruit 33 participants who are acknowledgeable of how a usual 8-bit music (not necessarily a 8-bit version of a Pop song but tunes that have been used in video games) for the listening test. 23 participants are 18–24 years old, while the others are 24–40 years old. 30 of them are male. The participants were asked to listen to four set of clips. Each set contains a clip of Pop music (each 10–30 sec-

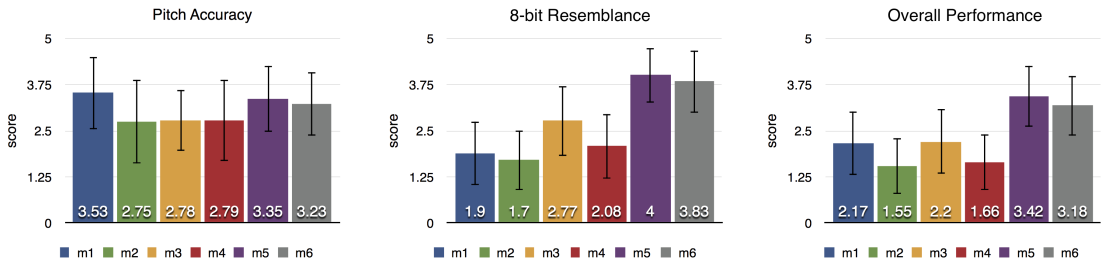


Fig. 4. Result of subjective evaluation: the mean ratings on the six methods described in Section 3 in terms of (left) pitch accuracy, (middle) 8-bit resemblance, and (right) overall quality. The error bars indicate the standard deviation of ratings.

onds in length), and 6 different 8-bit versions of the same clip generated respectively by the following methods:

(m1) Baseline NMF for audio mosaicing of Pop songs [17].
(m2) SVS + baseline NMF: we apply NMF to the separated vocals and instrumental background separately.
(m3) SVS + proposed pitch analysis (Sections 2.2–2.4) but synthesize in the frequency domain using NMF.
(m4) SVS + Let-it-bee [17] for the two separated sources respectively.
(m5) SVS + proposed pitch analysis (Sections 2.2–2.4) + time-domain synthesis, excluding the NMF constraint.
(m6) SVS + proposed pitch analysis (Sections 2.2–2.4) + time-domain synthesis.

In our implementation, we set the window size to 1 024 samples, hop size to 256 samples, and $\lambda = 1$ (a regularization parameter) for RPCA; window size to 2 048 samples, hop size to 256 samples, and beta threshold to 0.15 for pYIN; window size to 2 048 samples and hop size to 1 024 samples and the KL divergence as the cost function for NMF.

The four audio clips employed in the listening test are: *Someone Like You* by Adele *All of Me* by John Legend, *Jar of Hearts* by Christina Perri, and a song entitled *Gospel* by MusicDelta from the MedleyDB dataset [23]. The main criteria in selecting these songs are: 1) at most two accompanying instruments at any given time, 2) the main instrument is piano, 3) only one singer per song. We found in our pilot study that RPCA can better separate the singing voice from the accompaniments for such songs.

After listening to the clips (presented in random order and without names), the participants were asked to evaluate the 8-bit versions in the following 3 aspects, from one (very poor) to five (excellent) in a five-point Likert scale:

- **Pitch accuracy**: the perceived pitch accuracy of the converted 8-bit music.
- **8-bit resemblance**: the degree to which the converted clip captures the characteristics of 8-bit music.
- **Overall performance**: whether the clip sounds good or bad, from a pure listener point of view.

The mean ratings are depicted in Fig. 4 along with the error bars, where the following observations are made. First, in terms of pitch accuracy, the performance of the six considered methods is similar, with no significant difference according to the Students t-test. The mean pitch accuracy appears to be moderate, suggesting future work for further improvement.

Second, in terms of 8-bit resemblance, we see that the proposed method (m6) and its variant (m5) perform significantly better than the other four (p-value$<$0.05). The method (m3) is a variant of the proposed method which performs the final synthesis operation in the frequency domain instead of the time domain. We see that this method still performs better than the existing NMF or Let-it-bee methods, confirming the adequacy of the use of SVS and the proposed pitch analysis procedure for this task. However, the major performance gap between (m3) and (m6) indicates that time-domain synthesis is critical. Moreover, we see that while the proposed method attains an average 8-bit resemblance near to 4 (good), baseline NMF or Let-it-bee methods have average 8-bit resemblance only about 2 (poor). We find that NMF-based methods cannot perform well because the resulting conversion still sounds like the original song. Moreover, as the result of (m5) and (m6) are close, the NMF constraint seems not needed.

Finally, the result in overall performance seems to be correlated with 8-bit resemblance, but the average values are in general lower, suggesting room for improvement.

Audio examples of the original clips and the converted ones can be found in an accompanying website.[5] We will also release part of the source codes for reproducibility.

4. CONCLUSION

In this paper, we have proposed a novel task of converting Pop music into 8-bit music and an analysis/synthesis pipeline to achieve it, bringing together state-of-the-art singing voice separation and pitch detection algorithms. A listening test validates the advantages of the proposed method overall two existing NMF-based audio mosaicing methods. As a first attempt, we consider the result promising. From the feedbacks of the participants, future work can be directed to improve the onset clarity of the notes. It is also interesting to extend our work to Pop music accompanied by other instruments.

[5] https://lemonatsu.github.io/

5. REFERENCES

[1] Kevin Driscoll and Joshua Diaz, "Endless loop: A brief history of chiptunes," *Transformative Works and Cultures*, vol. 2, 2009.

[2] Alex Yabsley, "The sound of playing: A study into the music and culture of chiptunes," *Unpublished dissertation. Griffith University*, 2007.

[3] Johannes Kopf and Dani Lischinski, "Depixelizing pixel art," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)*, vol. 30, no. 4, pp. 99:1 – 99:8, 2011.

[4] Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita, "Pixel art with refracted light by rearrangeable sticks," *Computer Graphics Forum*, vol. 31, pp. 575–582, 2012.

[5] Vesa Välimäki, Sira González, Ossi Kimmelma, and Jukka Parviainen, "Digital audio antiquing—signal processing methods for imitating the sound quality of historical recordings," *Journal of the Audio Engineering Society*, vol. 56, no. 3, pp. 115–139.

[6] David T. Yeh, John Nolting, and Julius O. Smith, "Physical and behavioral circuit modeling of the SP-12 sampler," in *Proc. International Computer Music Conference*, 2007.

[7] Graham Coleman, Esteban Maestre, and Jordi Bonada, "Augmenting sound mosaicing with descriptor-driven transformation," in *Proc. Digital Audio Effects*, 2010.

[8] Ari Lazier and Perry Cook, "MoSievius: Feature driven interactive audio mosaicing," in *Proc. Digital Audio Effects*, 2003.

[9] Jordi Janer and Maarten De Boer, "Extending voice-driven synthesis to audio mosaicing," in *Proc. Sound and Music Computing Conference, Berlin*, 2008, vol. 4.

[10] Diemo Schwarz, "Corpus-based concatenative synthesis," *IEEE signal processing magazine*, vol. 24, no. 2, pp. 92–104, 2007.

[11] Gilberto Bernardes, *Composing music by selection content-based algorithmic-assisted audio composition*, Ph.D. thesis, University of Porto, 2014.

[12] Pierre Alexandre Tremblay and Diemo Schwarz, "Surfing the waves: Live audio mosaicing of an electric bass performance as a corpus browsing interface," in *Proc. New Interfaces for Musical Expression*, 2010, pp. 447–450.

[13] Diemo Schwarz, "Current research in concatenative sound synthesis," in *Proc. International Computer Music Conference*, 2005, pp. 1–1.

[14] Diemo Schwarz, "Concatenative sound synthesis: The early years," *Journal of New Music Research*, vol. 35, no. 1, pp. 3–22, 2006.

[15] Daniel D Lee and H Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Proc. Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[16] Emmanuel Vincent, Nancy Bertin, and Roland Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010.

[17] Jonathan Driedger, Thomas Prätzlich, and Meinard Müller, "Let It Bee–towards NMF-inspired audio mosaicing," in *Proc. the International Society for Music Information Retrieval Conference*, pp. 350–356, [online] `https://www.audiolabs-erlangen.de/resources/MIR/2015-ISMIR-LetItBee`.

[18] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright, "Robust principal component analysis?," *Journal of the ACM*, vol. 58, no. 3, pp. 11, 2011.

[19] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Advances in neural information processing systems*, 2009, pp. 2080–2088.

[20] Po-Sen Huang, Scott Deeann Chen, Paris Smaragdis, and Mark Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 57–60.

[21] Tak-Shing Chan, Tzu-Chun Yeh, Zhe-Cheng Fan, Hung-Wei Chen, Li Su, Yi-Hsuan Yang, and Roger Jang, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 718–722.

[22] Matthias Mauch and Simon Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 659–663, [online] `https://code.soundsoftware.ac.uk/projects/pyin`.

[23] Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Bello, "MedleyDB: A multitrack dataset for annotation-intensive mir research," in *Proc. International Society for Music Information Retrieval Conference*, 2014, pp. 155–160.