GCT634/AI613: Musical Applications of Machine Learning

Automatic Music Transcription: Monophonic



Juhan Nam

Overview of Automatic Music Transcription (AMT)

- Predict score information from acoustic music signals
 - Pitch contour: frame-level continuous pitch curves
 - MIDI: note-level events or piano rolls
 - Sheet music: symbolic music notation



- Pitch estimation (frame-level)
 - Monophonic pitch estimation from a single sound source
 - Polyphonic pitch estimation from multiple sound sources
 - Polyphonic single instrument: piano, guitar
 - Polyphonic multiple instrument: violin + cello + piano (stream-level)
 - Melody estimation: single melodic pitch estimation from multiple sound sources



Automatic Music Transcription: An Overview, Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert, IEEE SPM, 2018

- Note Transcription (note-level)
 - Based on the frame-level pitch estimation
 - Identify a note by detecting the onset and offset \rightarrow MIDI note on/off events
 - The task is challenging when the pitch is expressive (e.g. singing, violin)

MIDI Messages

"Note on: note=61, vel=80"



Polyphonic piano transcription <u>https://ai-midi.com/</u>



Singing Note Transcription (Tony) https://www.sonicvisualiser.org/tony/

AMT Tasks

- The sheet music needs a lot more information (complete AMT)
 - Based on the note transcription
 - Metric analysis: tempo estimation, beat/downbeat detection
 → quantize onset and offsets to beat-based time units
 - Key detection: 12 pitch classes and major/minor (e.g. C major)
 - Notes: clef, stem, beam
 - Expressions: dynamics (e.g. piano/forte), articulation (e.g. staccato), phrasing, tempo, and more
 key



Monophonic Pitch Estimation

- When a tone is generated with a pitch, the waveform is periodic (or nearly periodic) and the spectrum is harmonic (or nearly harmonic)
- Pitch is often called fundamental frequency or "F0" (F0 =1/period)
- Traditional approaches (digital signal processing)
 - Time-domain approach: estimate the period of the waveform
 - Frequency-domain approach: exploit the harmonic pattern



Waveforms of Flute A4 note



Spectrum of Flute A4 note

Time-Domain Approach

- Measure the period: x(t) = x(t + T)
- Calculate the distance between a segment in a fixed window and another segment in a sliding window
 - Auto-correlation function (ACF): distance by the inner product
 - Average magnitude difference function (AMDF) : Euclidean distance
 - AMDF is more robust to the amplitude changes compared to ACF
- Find the time difference (lag) that makes the best match



Time-Domain Approach: YIN

- Based on the normalized AMDF
 - Choose the minimum notch below a threshold

$$\hat{d}(l) = \begin{cases} d(l) / [\frac{1}{l} \sum_{u=1}^{l} d(u)] & \text{otherwise} \\ 1 & l = 0 \end{cases} \qquad d(l): \text{AMDF}$$



YIN, a fundamental frequency estimator for speech and music, Alain de Cheveigne, Hideki Kawahara, JASA, 2002

Frequency-domain Approach

- Pattern matching: cross-correlation between log-frequency spectrum with a pre-defined harmonic template
 - SWIPE: use the spectrum of sawtooth waveform
- Harmonic product sum (HPS): successive product with harmonically down-sampled spectra
 - Use as a pitch salience function



A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music, Arturo Camacho and John G Harris, JASA, 2008.

- Decompose spectrum into harmonic partials (periodic) and frequency envelope (slowly-varying)
 - Real Cepstrum: $c_x(l) = real{FFT^{-1}(log|FFT(x)|))}$
 - "Liftering" to remove the frequency envelope



Cepstrum Pitch Determination, A. Michael Noll, JASA, 1967.

Post processing: Smoothing

- Median filtering
 - Handy and useful to remove outliers (e.g. octave jump)
- Viterbi decoding (based on hidden Markov model)
 - Find the best path with the maximum likelihood considering pitch transition
 - PYIN: use a probabilistic threshold and find the best sequence (Tony)



PYIN: A fundamental frequency estimator using probabilistic threshold distributions, Matthias Mauch, Simon Dixon, ICASSP, 2014

CREPE: Monophonic Pitch Estimation Using Supervised CNN

- Classification-based approach using CNN
 - Input: a single frame of waveforms (1024 samples, resampled to 16kHz)
 - Output: quantized pitch with a resolution of 20 cents --> 360 classes
 - The output labels are smoothed using a Gaussian function: softening the penalty for near-correct predictions



CREPE: A Convolutional Representation for Pitch Estimation, Jong Wook Kim, Justin Salamon, Peter Li, Juan P. Bello, ICASSP, 2018

CREPE: Monophonic Pitch Estimation Using Supervised CNN

- Higher performance than traditional DSP methods
 - Higher raw pitch accuracy (RPA) and raw chroma accuracy (RCA)
 - Robust when a certain noise is added
 - But, low accuracy when the training set does not cover a wider pitch range or the pitch annotation is not consistent



CREPE: A Convolutional Representation for Pitch Estimation, Jong Wook Kim, Justin Salamon, Peter Li, Juan P. Bello, ICASSP, 2018

CREPE: Monophonic Pitch Estimation Using Supervised CNN

- Dataset issue in the supervised-learning approach
 - It is very tedious to annotate frame-level pitch labels on audio files. How can we obtain large-scale pitch annotations?
- Solution: Analysis-Resynthesis approach
 - Use a monophonic pitch estimator and obtain **pseudo pitch labels** (F0)
 - Re-synthesize the input source using the F0 values
 - Originally used for melody extraction in mixed audio



An Analysis/Synthesis Framework for Automatic f0 Annotation of Multitrack Datasets, Justin Salamon, et al., ISMIR, 2017

SPICE: Monophonic Pitch Estimation Using SSL

- Train a pitch estimation network without pitch labels
 - Siamese network to estimate a pitch difference when a pair of constant-Q transform has the same relative pitch difference



$$\begin{split} & \underset{0 \leq y_{t} \leq 1}{\text{Relative pitch difference estimation error}} 0 \leq y_{t} \leq 1 \quad e_{t} = |(y_{t,1} - y_{t,2}) - \sigma(k_{t,1} - k_{t,2})| \\ & \mathcal{L}_{\text{pitch}} = \frac{1}{T} \sum_{t} h(e_{t}) \quad h(x) = \begin{cases} \frac{\mathbf{x}^{2}}{2}, & |x| \leq \tau \\ \frac{\tau^{2}}{2} + \tau(|x| - \tau) \\ \text{Huber norm} \end{cases} \\ & \underset{\text{Huber norm}}{\text{Reconstruction error}} \\ & \mathcal{L}_{\text{recon}} = \frac{1}{T} \sum_{t} \|\mathbf{x}_{t,1} - \hat{\mathbf{x}}_{t,1}\|_{2}^{2} + \|\mathbf{x}_{t,2} - \hat{\mathbf{x}}_{t,2}\|_{2}^{2}, \\ & \underset{\text{Confidence level estimation}}{\text{L}_{\text{conf}} = \frac{1}{T} \sum_{t} |(1 - c_{t,1}) - e_{t}/\sigma|^{2} + |(1 - c_{t,2}) - e_{t}/\sigma|^{2}} \end{split}$$

Final pitch estimation: the scale and offset are learned using a small labeled dataset $\hat{p}_{0,t} = b + s \cdot y_t = b + s \cdot Enc(\mathbf{x}_t)$

SPICE: Self-Supervised Pitch Estimation, B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirovic, IEEE TASLP, 2020

PESTO: Monophonic Pitch Estimation Using Equivariant SSL

- Minimize the equivariance and invariance
 - The output of the network y is a multinomial distribution (pitch classification)
 - Equivariance: shifting + cross-entropy for k transposition

$$\phi : \mathbb{R}^{d} \to \mathbb{R} \\ \mathbf{y} \mapsto (\alpha, \alpha^{2}, \dots, \alpha^{d}) \mathbf{y} \quad \mathcal{L}_{\text{equiv}}(\mathbf{y}, \mathbf{y}^{(k)}, k) = h_{\tau} \left(\frac{\phi(\mathbf{y}^{(k)})}{\phi(\mathbf{y})} - \alpha^{k} \right) \qquad \mathcal{L}_{\text{SCE}}(\mathbf{y}, \mathbf{y}^{(k)}, k) = \sum_{i=0}^{d-1} y_{i} \log \left(y_{i+k}^{(k)} \right)$$

• Invariance: add pitch-invariant transform (e.g. add noise) and cross-entropy



PESTO: Pitch Estimation with Self-supervised Transposition-equivariant Objective, Alain Riou, Stefan Lattner, Gaëtan Hadjeres, Geoffroy Peeters, ISMIR, 2023

PESTO: Monophonic Pitch Estimation Using Equivariant SSL

- Transpostion-preserving architecture
 - 1D-Conv: the frequency resolution remains unchanged
 - A softmax layer for the output
 - Toeplitz fully-connected layer: preserve the transposition (m + n 1)elements instead of mn): equivalent to convolution



Figure 3. Architecture of our network f_{θ} . The number of channels varies between the intermediate layers, however the frequency resolution remains unchanged until the final Toeplitz fully-connected layer.

$$A = \begin{pmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-n+2} & a_{-n+1} \\ a_1 & a_0 & a_{-1} & \ddots & \ddots & a_{-n+2} \\ a_2 & a_1 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_{m-1} & \cdots & \cdots & \cdots & \cdots & a_{m-n} \end{pmatrix}$$

Toeplitz Matrix

PESTO: Pitch Estimation with Self-supervised Transposition-equivariant Objective, Alain Riou, Stefan Lattner, Gaëtan Hadjeres, Geoffroy Peeters, ISMIR, 2023

PESTO: Monophonic Pitch Estimation Using Equivariant SSL

• A lightweight model

			Raw Pitch Accuracy	
Model	# params	Trained on	MIR-1K	MDB-stem-synth
SPICE [19]	2.38M	private data	90.6%	89.1%
DDSP-inv [45]	-	MIR-1K / MDB-stem-synth	91.8%	88.5%
PESTO (ours)	28.9k	MIR-1K	96.1%	94.6%
PESTO (ours)	28.9k	MDB-stem-synth	93.5%	95.5%
CREPE [16]	22.2M	many (supervised)	97.8 %	96.7%

Robustness test

	Raw Pitch Accuracy (MIR-1K)				
Model	clean	20 dB	10 dB	0 dB	
SPICE [19]	91.4%	91.2%	90.0%	81.6%	
PESTO					
eta=0	94.8%	90.7%	79.2%	50.0%	
$\beta = 1$	94.5%	94.2%	92.9%	83.1%	
$\beta \sim \mathcal{U}(0,1)$	94.7%	94.4%	92.9%	81.7%	
$\beta \sim \mathcal{N}(0,1)$	94.8%	94.5%	93.0%	82.6%	
$\beta \sim \mathcal{N}(0, \frac{1}{2})$	94.8%	94.5%	92.9%	81.0%	
CREPE [16]	97.8%	97.3%	95.3%	84.8%	

Table 2. Robustness of PESTO and other baselines to background music with various Signal-to-Noise ratios. Adding background music to training samples significantly improves the robustness of PESTO (see section 4.4.2).

• Ablation Study

	MIR-1K		MDB	
	RPA	RCA	RPA	RCA
PESTO baseline	96.1%	96.4%	94.6%	95.0%
Loss ablations				
w/o \mathcal{L}_{equiv}	5.8%	8.6%	1.3%	6.1%
w/o \mathcal{L}_{inv}	96.1%	96.4%	92.5%	94.5%
w/o $\mathcal{L}_{ ext{SCE}}$	96.1%	96.5%	86.9%	93.8%
Miscellaneous				
no augmentations	94.8%	95.4%	94.8%	95.2%
non-Toeplitz fc	5.7%	8.7%	1.2%	6.1%

 Table 3. Respective contribution of various design choices of PESTO for a model trained on *MIR-1K*.

PESTO: Pitch Estimation with Self-supervised Transposition-equivariant Objective, Alain Riou, Stefan Lattner, Gaëtan Hadjeres, Geoffroy Peeters, ISMIR, 2023

Source Code / Inference Models

- PYIN: <u>https://librosa.org/doc/main/generated/librosa.pyin.html</u>
- CREPE: <u>https://github.com/marl/crepe</u>
- SPICE: <u>https://www.tensorflow.org/hub/tutorials/spice</u>
- PESTO: <u>https://github.com/SonyCSLParis/pesto</u>

• The neural pitch estimators use food names...

- Extract melodic pitch contours from polyphonic music
 - Pre-dominant pitch estimation in the presence of multiple sound sources
- Methods
 - Salience-based approach: use a saliency function (e.g. HPS)
 - Source separation approach: separate the melodic source and use the monophonic pitch estimation
 - Classification-based approach: use CNN or CRNN



Melody Extraction from Polyphonic Music Signals: Approaches, Applications and Challenges, J. Salamon, E. Gómez, D. P. W. Ellis and G. Richard, IEEE SPM, 2014

Singing Melody Extraction

- Joint learning of singing voice detection and vocal pitch estimation
 - Combining the loss functions from the two tasks
 - Vocal pitch classification (CRNN)
 - ResNet stacks: "no pooling over time"
 - Bi-directional LSTM-RNN to learn temporal dependency
 - Use the Gaussian blurring in the output layer
 - Singing voice detector (CRNN)
 - Use the shared features from the three layers of the pitch classifier: "hierarchical" audio features (e.g., vocal formant, vibrato, portamento)
 - Bi-directional LSTM-RNN
 - o <u>https://github.com/keums/melodyExtraction_JDC</u>



		-	-		
	Main	Main + AUX	Main	Main + AUX	
Input	-		31×513		
Conv block	[3 >	× 3, 64] × 2	31	× 513, 64	
ResNet Block 1	[3 ×	3,128] × 2	31	× 128, 128	
ResNet Block 2	[3 ×	3, 192] × 2	31	× 32, 192	
ResNet Block 3	[3 ×	3,256] × 2	31	× 8, 256	
Pool block		-	31	× 2, 256	
Bi-LSTM	256	256 + 32	31×512	$31 \times (512 + 64)$	
FC	722	722 + 2	31×722	31 × (722 + 2)	

Singing Melody Extraction Using Teacher-Student Models

- Semi-supervised learning
 - Labeled data: used to train a teacher network
 - Unlabeled data: used to train a student network to predict the output of the teacher network

	Dataset	Number of Tracks	Total Length
T	RWC	100	6h 47m
(Labeled)	MedleyDB	61	2h 39m
(Labeled)	iKala	262	2h 6m
Training (Unlabeled)	In-house	535	6h 21m
	FMA_small	3,521 / 8,000	25h / 60h
	FMA_medium	10,639 / 25,000	89h / 208h
	FMA_large	40,505 / 106,574	337h / 888h
Test	ADC04	12	4m
	MIREX05	9	4m
	MedleyDB	12	43m
	AST218	218	14h 53m

• Random data augmentation makes the student network perform betters



Semi-Supervised Learning Using Teacher-Student Models for Vocal Melody Extraction, Sangeun Kum and Juhan Nam, 2020