Freely available online
M. Schwabe, S. Murgul and M. Heizmann,
"Dual Task Monophonic Singing Transcription"
J. Audio Eng. Soc., vol. 70, no. 12, pp. 1038–1047, (2022 December).
DOI: https://doi.org/10.17743/jaes.2022.0040

Dual Task Monophonic Singing Transcription

MARKUS SCHWABE, SEBASTIAN MURGUL, AND MICHAEL HEIZMANN
(markus.schwabe@kit.edu) (sebastian.murgul@klangio.com) (michael.heizmann@kit.edu)

Institute of Industrial Information Technology (IIIT), Karlsruhe Institute of Technology, Karlsruhe, Germany

Automatic music transcription with note level output is a current task in the field of music information retrieval. In contrast to the piano case with very good results using available large datasets, transcription of non-professional singing has been rarely investigated with deep learning approaches because of the lack of note level annotated datasets. In this work, two datasets are created concerning amateur singing recordings, one for training (synthetic singing dataset) and one for the evaluation task (SingReal dataset). The synthetic training dataset is generated by synthesizing a large scale of vocal melodies from artificial songs. Because the evaluation should represent a realistic scenario, the SingReal dataset is created from real recordings of non-professional singers. To transcribe singing notes, a new method called Dual Task Monophonic Singing Transcription is proposed, which divides the problem of singing transcription into the two subtasks onset detection and pitch estimation, realized by two small independent neural networks. This approach achieves a note level F1 score of 74.19% on the SingReal dataset, outperforming all state of the art transcription systems investigated with at least 3.5% improvement. Furthermore, Dual Task Monophonic Singing Transcription can be adapted very easily to the real-time transcription case.

0 INTRODUCTION

One of the most important parts of the music information retrieval (MIR) task is the automatic music or melody transcription. It tries to create a symbolic representation of an input audio signal in order to generate a human-readable note sheet. Several transcription approaches using simple spectrogram thresholding, autocorrelation, probabilistic methods, or neural networks and deep learning have been proposed [1]. A simple but effective method for frame-wise monophonic fundamental frequency (F0) estimation is the YIN algorithm [2] based on the autocorrelation of the audio signal. It is suited for music and speech, even with high pitches. Mauch et al. [3] improved this algorithm by a probabilistic thresholding for the pitch candidates, which reduces short-time errors of the original YIN. This improved algorithm is called probabilistic YIN (pYIN). In 2015, the pYIN method was extended for the note transcription task by adding a second, independent hidden Markov model (HMM) that models the attack, stable, and silent states for each note [4]. This implementation is called Tony.

Recent approaches like CREPE [5] showed that data-driven pitch tracking outperforms previous methods for signals from various instruments. CREPE is based on a deep convolutional neural network and operates directly on the waveform. Preprocessed input data, like Mel-scaled spectrograms with logarithmic magnitude in the onsets and frames (OaF) algorithm [6], enable smaller models with less computational resource demands. The OaF algorithm is a dual objective transcription approach that jointly predicts note onsets and pitches of piano signals.

Among the transcription of different instruments, one of the most challenging tasks is the transcription of sung melodies because the pitch evolution within one note is often unstable. Ryynänen and Klapuri [7] further processed F0 estimation results by an HMM based acoustic and musicological model to improve automatic singing transcription (AST). A similar approach based on two models has been developed by McLeod et al. [8] for the more challenging polyphonic vocal transcription of a cappella music. But both model-based transcription systems have problems in the case of unstable pitches like vibrato. Thus, they are inappropriate for untrained singers, for which the varying pitch represents a main AST problem.

Therefore, Molina et al. [9] classified stable voice and unvoiced signal regions and performed a monophonic singing transcription based on the YIN algorithm with a hysteresis defined on the pitch-time curve. Yang et al. [10] improved the pYIN F0 results by means of a hierarchical HMM consisting of an upper ergodic HMM for note transitions and a pitch dynamic model for pitch fluctuations. Although some YIN or pYIN errors can be reduced, the performance of those approaches is mainly dependent on the vanilla YIN or pYIN algorithm, which was outperformed by, e.g., OaF in the case of piano signals. A new probabilistic approach is the Bayesian singing transcription of Nishikimi et al. [11] that combines an F0 trajectory model with the information of local keys and musical note rhythms in a Bayesian hierarchical hidden semi-Markov model.

Because additional information has to be given in advance or can be estimated erroneously, most approaches are data-driven and use neural networks. Rigaud and Radenen [12] proposed the separate F0 estimation and segmentation of the vocal signal by a deep neural network for both of the tasks. The respective results are combined by a Viterbi tracking as postprocessing. For an isolated singing voice in popular music, Nishikimi et al. [13] developed an encoder-decoder model with an attention mechanism. This model represents an end-to-end approach for joint estimation of note pitch and time values. Because it is no frame-based approach, one time error has an effect on the whole estimated note sequence; therefore, F1 score results are low, and note insertions, substitutions, or deletions have to be considered in suitable evaluation metrics.

Convolutional neural networks with preprocessing are used by Cuesta et al. [14] for the multiple F0 estimation of vocal ensembles. Magnitude and phase representations of the harmonic CQT are calculated and fed into the network that can estimate more than one sung note of choir singers per time step. Choir singers are assumed to sing stable pitches, thus unstable pitches lead to worse performance in that approach.

One main challenge in singing transcription is the detection of onsets and offsets; therefore, Fu and Su [15] improved the note segmentation of monophonic singing signals by a hierarchical classification. They defined the states silence, activation, onset, or offset and estimated the sequence of states based on several input representations and a residual network. During postprocessing, the state sequence and a separate pitch estimation are combined for a note level transcription. This approach was further improved by Hsu and Su [16], in which the residual network was replaced by a PyramidNet architecture. Moreover, virtual adversarial training was investigated to incorporate unlabeled data in the training process, but the performance depends on the model and the data properties. An implementation of that approach is integrated in the MIR project Omnizart [17].

This work presents a dual task algorithm for monophonic amateur singing transcription. Because the combination of time and F0 information is essential in music transcription, a dual model with separated subtasks for onsets and pitches inspired by [6] is used, aiming at a note level AST. Because the evaluation of a data-driven approach needs realistic testing data, SingReal, a new dataset with annotated amateur recordings including non-professional singing characteristics like pitch instabilities, was created. This is supplemented by a sufficiently large training dataset that consists of artificially created singing recordings.

After the definition of all relevant feature extraction algorithms in SEC. 1, the used singing transcription datasets are presented in SEC. 2, including the new SingReal dataset. The Dual Task Monophonic Singing Transcription (DTMST) model with its components is explained in detail in SEC. 3, and the transcription results are evaluated in SEC. 4.

1 FEATURE EXTRACTION

The recorded discrete monophonic singing signal $x[n]$, separated from other music source signals, represents the data input of this approach. If the vocal part is not recorded separately, music source separation algorithms like Demucs [18] or Spleeter [19] can extract this track in an additional preprocessing step. But the impact of such an additional step is not investigated in this work.

Firstly, the separated discrete signal $x[n]$ is converted into a time-frequency representation $X[m, k]$ with time index m and frequency index k in order to highlight characteristic time-dependent and frequency-dependent features. A common representation for MIR algorithms is the short-time Fourier transform (STFT) [20] X_{STFT}, which has a linear frequency scale. Alternatively, X_{STFT} can be further processed to calculate the Mel spectrogram

$$S_{\mathrm{Mel}}[m,r] = \sum_{k=0}^{K-1} F_r[k] \cdot |X_{\mathrm{STFT}}[m,k]|^2 . \tag{1}$$

This representation uses the psychoacoustic Mel scale based on human perception [21] in the frequency dimension that is calculated by the filters F_r of a Mel filter bank. In case of instrument transcription, Mel spectrograms lead to comparable results as STFT spectrograms despite a massively reduced number of frequency bins [22].

Another suitable time-frequency representation for music signals is the constant-Q transform (CQT) [23]

$$X_{\mathrm{CQT}}(m,k) = \sum_{n=m-\lfloor N_k/2 \rfloor}^{m+\lfloor N_k/2 \rfloor} x[n]\, a_k^* \left[n - m + \frac{N_k}{2} \right] \tag{2}$$

with floor operator $\lfloor \cdot \rfloor$ and the window function

$$a_k[n] = \frac{1}{N_k}\, w\left[\frac{n}{N_k} \right] e^{-\mathrm{j}2\pi \frac{f_k}{f_\mathrm{s}}} \tag{3}$$

including the sampling rate f_s and window $w[n]$ of frequency-dependent length N_k. With its logarithmic frequency scale, the CQT ensures a constant resolution for all octaves and therefore fits to the discrete semitones. A modification of the CQT is the harmonic CQT (HCQT) [24], which calculates several CQTs for different octaves and concatenates them in a multi-dimensional representation. Consequently, the fundamental frequency in the first channel is concatenated with its harmonics in the higher channels in one frequency bin of the HCQT.

At a note onset, the spectrogram intensity is increasing in the respective frequencies. In order to detect such intensity variations in successive frames, the spectral flux [25]

$$\Phi[m] = \sum_{r=0}^{R-1} \max\left(0,\ S[m,r] - S_{\mathrm{ref}}[m-1,r]\right) \tag{4}$$

can be calculated based on an arbitrary spectrogram S with R frequency bins and reference spectrogram S_{ref}. In the case of the classical spectral flux, S_{ref} equals S. Consequently, Eq. (4) represents the sum of the positive temporal derivatives of all frequency bins and therefore correlates with the strength of a note onset at time index m. A revised version, which is more sensitive to soft onsets, uses a maximum filtered spectrogram [25],

$$S_{\max}[m, r] = \max\left(S[m, r-1 : r+1]\right), \quad (5)$$

as reference S_{ref}. It is called super flux.

2 SINGING TRANSCRIPTION DATASETS

Deep learning depends on the availability of large datasets, which allow for the training of complex neural networks. For music transcription tasks in general, there are many datasets available. An overview over lots of common datasets available for MIR is given on the ISMIR website.[1] For singing voice, pitch tracking datasets like MedleyDB [26] are available. Unfortunately, pitch tracking datasets aim at the frame-based fundamental frequencies but do not contain discrete note pitches and onset or offset annotations and are therefore not suited for note level transcription. In [27], a dataset of karaoke recordings of the mobile app "Smule" is presented but does not contain note level annotations either. The Choral Singing Dataset [28] is a collection of three pieces performed by 16 singers of the Anton Bruckner Choir from Barcelona (Spain). This dataset even comes with aligned MIDI files, but, still, it is not large enough to use it for deep learning.

Two large datasets for singing transcription are CMedia[2] and MIR-ST500 [29], which represent collections of 100 or 500 Chinese pop songs from YouTube with human-labeled annotations. Unfortunately, only the mixed songs and not the separate singing signals are available. Another large dataset is one part of the RWC dataset [30], which is widely used and consists of popular music recordings with English and Japanese lyrics. But those songs are performed by professional musicians in a professional recording studio and hence do not meet the current needs of an amateur recording dataset. Moreover, isolated singing signals are also not available.

Because of the lack of annotated datasets of amateur recordings, a new dataset with recordings of untrained singers has been created in this work. In order to collect realistic examples, a mobile app has been developed and distributed to voluntary participants worldwide.[3] After the user has recorded their voice, it is uploaded to a server on which a state of the art algorithm processes the file and creates a musical score that can be viewed in an integrated score viewer. This workflow leads to genuine recordings because the participant uses the app for the purpose of a transcription app.

In order to create a universally usable dataset, three kinds of annotations are made in a postprocessing step after the user's recording. Firstly, a musical score is transcribed manually by ear and stored as a MusicXML file. Secondly, the bars of the musical score are synchronized to the audio signal using local tempo annotations. Lastly, the MusicXML file is translated to MIDI and corrected manually on a note level in order to finely align each note's timing and synchronize it with the audio. The resulting dataset is called "SingReal," and it is used for the evaluation of amateur singing transcription. It consists of a total of 35 singing recordings, 18 of female and 17 of male amateur singers, each with a duration between 5 and 60 s. The pitch distribution of the SingReal dataset is visualized in Fig. 1.

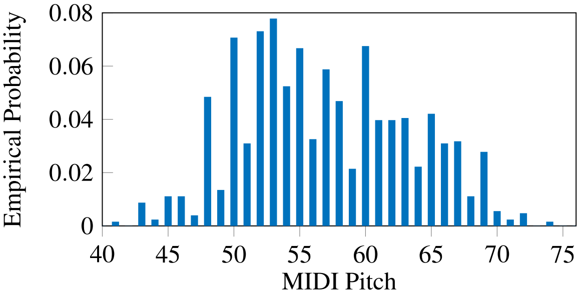

Fig. 1. MIDI pitch distribution of the SingReal evaluation dataset of amateur singing recordings.

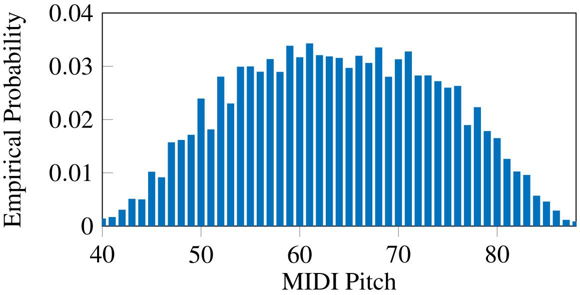

Fig. 2. Pitch distribution of the synthetic singing dataset.

For the training process, a large synthesized dataset has been generated in this work. This approach avoids the very time-consuming manual annotation of real recordings, similar to the idea of Emiya et al. in [31] for a piano dataset. Moreover, different timbres and sound characteristics can be utilized to create well-balanced data and several styles. In Fig. 3, the dataset generation process is visualized.

The developed synthetic singing dataset is based on a collection of monophonic folk tunes [32] containing over 4,000 concatenated ABC notation files of melodies without lyrics. Lyrics are included by a poetry generation dataset [33] containing song lyrics from different artists like Adele, Bob Marley, or Eminem. So, a variety of genres, which all have their own vocabulary, is used to create realistic singing data. All words of the lyrics are split by Pyphen [34], a python library for hyphenation, into syllables that can be assigned to the individual notes. The notes of the folk tunes are quantized into duration steps of 1/16. To generate an evenly distributed pitch probability in the training dataset, the merged songs are transposed randomly with respect to a general pitch range of 40 to 88, which represents a note

[1] https://ismir.net/resources/datasets/.

[2] https://www.music-ir.org/mirex/wiki/2020:Singing_Transcription_from_Polyphonic_Music.

[3] https://klangio.com/sing2notes.

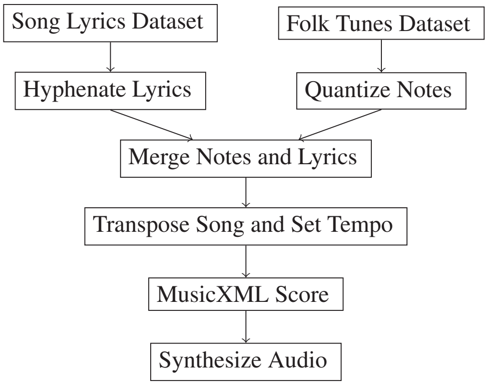

Fig. 3. Schematic for the generation of a large synthesized singing recording dataset.

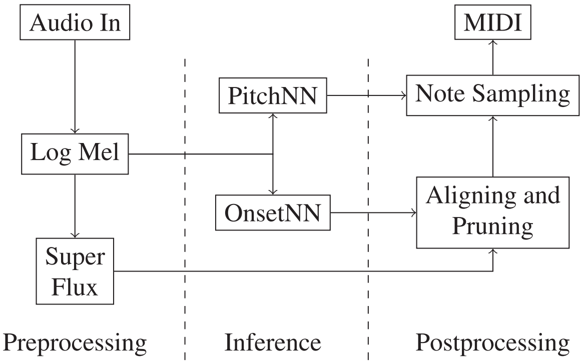

Fig. 4. Schematic of the Dual Task Monophonic Singing Transcription (DTMST) system structure. OnsetNN = Onset Detection Neural Network; PitchNN = Pitch Estimation Neural Network.

range from E2 to E6. This pitch range comprises amateur bass to amateur soprano singing ranges and the pitch range of the SingReal dataset. The resulting pitch distribution is visualized in Fig. 2.

In order to create diversity in MIDI note duration, a tempo in the range of 50 to 120 bpm is sampled from a uniform distribution. The resulting modified synthetic song is converted to a MusicXML score, which is then used to synthesize a realistic singing voice audio by SinSy [35], an HMM-based singing voice synthesis system. For a balanced dataset distribution, an English male or female speaker is used randomly. The heuristic weight for the Sinsy expression parameter "vibrato" is sampled from a uniform random variable with a range from 0.5 to 1.5. For this work, 1,100 synthesized audio files with a sampling rate of 44.1 kHz and total duration of 14.5 h have been generated. In contrast to the real singing recordings, the synthetic audios have a bit more stable pitches with less fluctuations or vibrato and include less background noise. The two datasets, including all audio files and annotations, and the code of the DTMST model are freely available online.[4]

3 MODEL DESCRIPTION

The DTMST approach of this work divides the problem of singing transcription into the two subtasks onset detection and pitch estimation, realized by two small independent neural networks. Its architecture is visualized in Fig. 4 and can be grouped into three main sections: preprocessing, inference, and postprocessing. In the preprocessing step, the required features are extracted from the input audio signal. Then, the neural networks predict the current onsets and pitches based on the logarithmic Mel spectrogram in the inference phase. The estimated onsets are improved using heuristics and the spectral flux. Finally, the improved onsets are used to sample MIDI notes from the pitch estimation output in the postprocessing step.

This approach has low resource requirements, e.g., less than 5.2 MB of checkpoint and code size, and a flexible input length because no fixed audio duration is necessary. Furthermore, the implementation is very fast: 60 s of audio take about 0.75 s computation time for preprocessing, inference, and postprocessing. Consequently, the DTMST approach is real-time capable.

3.1 Preprocessing

During preprocessing the required time-frequency representations are computed from the input audio signal. Because both neural networks take the Mel spectrogram as input representation in the inference phase, it is calculated as introduced in SEC. 1 and employed with logarithmic magnitude values. This is called the logarithmic Mel spectrogram here. The input audio signals are resampled to a sampling rate of 16 kHz, which is common in music transcription [6]. A window length of 2,048 and hop size of 512 are used for the spectrogram calculation. A total number of 229 frequency bins is used, starting at a minimal frequency of 55 Hz. The time-dependent super flux vector is calculated as defined in Eqs. (4) and (5) from the logarithmic Mel spectrogram for the onset postprocessing step (see SEC. 3.4).

3.2 Onset Detection Neural Network

Onsets of note candidates are estimated by the first neural network of DTMST, the Onset Detection Neural Network (OnsetNN). Its architecture is visualized in Fig. 5. The logarithmic Mel spectrogram of the full audio signal is used as input representation. OnsetNN consists of three convolutional layers, followed by a dropout layer and

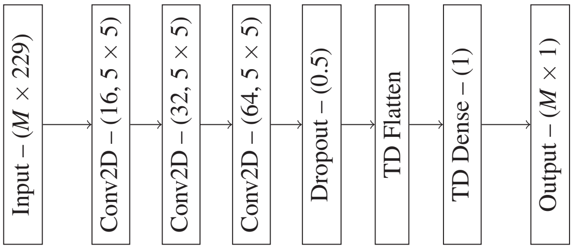

Fig. 5. Onset detection network using logarithmic Mel spectrogram as input feature.

[4] https://github.com/klangio/dtmst.

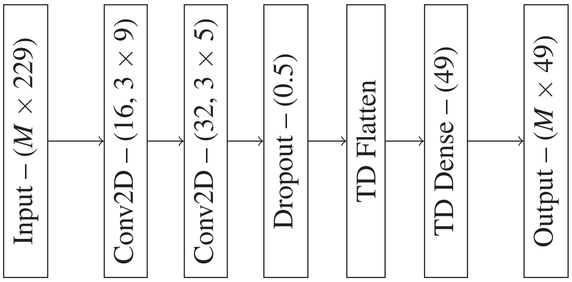

Fig. 6. Network architecture of the pitch estimation neural network with logarithmic Mel spectrogram as input feature.

time-distributed dense layer. All convolution kernels have a square shape, and the activation function used for the convolutional layers is Rectified Linear Unit. For the dropout layer, a dropout rate of 0.5 is selected. The time-distributed flatten layer is needed to reshape the dropout output into a 1D array for the dense layer, which uses a sigmoid activation function.

Binary vectors corresponding to the M spectrogram time frames are used as labels. If the current, previous, or next frame is an onset, the respective coordinate is “1,” and otherwise, it is “0.” Several variations of the architecture, input representations, and labels have been investigated. But in most cases, they had only a small effect on the accuracy. Detailed evaluation results can be found in SEC. 4.

3.3 Pitch Estimation Neural Network

Pitches of the sung notes are estimated by the second neural network of DTMST, Pitch Estimation Neural Network (PitchNN). In contrast to frame-based pitch tracking, the note pitches with discrete semitones and not the fundamental frequencies are estimated to get a musical score quantization. Consequently, the approach aims to reliably estimate the most accurate note pitches, regardless of a poor singing performance with shaky pitch, which is typical for amateur singers.

The architecture of PitchNN is shown in Fig. 6. Similar to OnsetNN, the input is the logarithmic Mel spectrogram of the full audio signal. PitchNN uses two convolutional layers with Rectified Linear Unit as activation function for feature extraction. Because spectral relations are more important than temporal ones, both convolution kernels are oblong with shapes of 3 × 9 and 3 × 5, respectively. The convolutional layers are followed by a dropout layer with a dropout rate of 0.5 and time-distributed dense layer with softmax activation function. This layer consists of 49 neurons because of the considered pitch range from MIDI numbers 40 to 88, which represents a note range from E2 to E6. The pitch labels are given in a 2D binary matrix of 49 pitch bins and M time frames. Tones are often not held stable by (untrained) singers and tend toward lower pitches in the release phase. Therefore, the pitch labels of sung notes are only set to “1” during the attack phase, which is the most robust pitch detection interval.

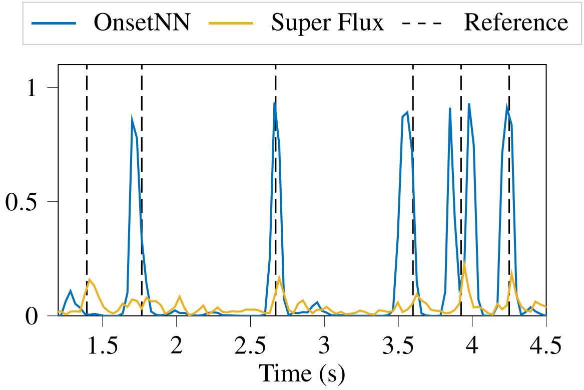

Fig. 7. Onset Detection Neural Network (OnsetNN) estimates, super flux values, and onset references for a real singing recording example.

3.4 Onset Postprocessing

A deep neural network is often handled as a black box that tries to learn a mapping between input and output from given examples. As a result, the network can sometimes lead to uncertain behavior, which can be overcome by postprocessing its predictions using additional sanity checks.

Including the super flux in the onset postprocessing represents such a sanity check, because the local maxima of OnsetNN are not as precise as those of the super flux in most cases. Thereby, the super flux is calculated from the preprocessed logarithmic Mel spectrogram and has the same time discretization as OnsetNN. In Fig. 7, the differences between the OnsetNN output and generally oversensitive super flux are visualized for a real singing recording. The exact onset time mostly corresponds to a local maximum in the super flux curve of the audio signal, but there are more local maxima of the super flux than onsets. Therefore, the estimated onset times, which are determined from the OnsetNN output by comparing it with a threshold of 0.5, are corrected to the time points with the nearest local maximum in the super flux.

Besides the super flux, there also exist other onset strength functions, as given in the overview in [36]. Different onset strength functions have been compared, as presented in SEC. 4.1, but super flux led to the best results.

After the super flux–based time aligning, a pruning step is performed. All onsets detected up to 93.75 ms after the previous detected onset (which corresponds to 75% of the duration of a 16th note at 120 bpm) are interpreted as fluctuations in the tone’s envelope and are therefore dropped.

3.5 Note Sampling

The note sampling step segments the outputs of the pitch estimation and onset detection into unquantized MIDI notes. An estimated MIDI note consists of the three parameters start time, end time, and pitch.

To retrieve the pitch of a note detected by onset estimation, the output of PitchNN is analyzed three frames after the detected onset frame. The pitch bin with the highest value at this specific sampling point during the attack phase is taken as the note pitch. This exact sampling point of three time frames after the onset is chosen empirically and corresponds to a delay of 96 ms after the onset. Because this time delay of about 75% of the duration of a 16th note at 120 bpm represents an unrealistic fast singing, the pitch in that frame is considered to belong to the detected onset.

The start time of the note is the estimated onset time, and the end time of it is chosen as the start time of the following note. Rests are not considered within a transcribed melody because of the connected characteristic of sung melodies. For the last note of the music piece or a separated melody, the duration is set to a constant value of 2 s, which corresponds to a whole note at 120 bpm. If the tempo is estimated in future work, this constant duration can be adapted to a whole note at the detected tempo very easily.

3.6 Training

For the purpose of training, the synthetic singing dataset presented in SEC. 2 is used. Both networks are trained together by means of a combined loss function of independent entropies; therefore, both networks are optimized independently during only one training procedure. OnsetNN is trained using the binary cross-entropy l_b, whereas PitchNN is trained using the categorical cross-entropy l_c. The combined loss of both tasks is the weighted sum of both losses

$$l = \sum_{t=0}^{T-1} \alpha \cdot l_\mathrm{b}\left(\hat{o}(t), o_\mathrm{ref}(t)\right) + \beta \cdot l_\mathrm{c}\left(\hat{p}(t), p_\mathrm{ref}(t)\right), \qquad (6)$$

with estimated onsets $\hat{o}$ and pitches $\hat{p}$ and their references o_ref and p_ref, respectively. The weighting factors α and β can be used to adjust the focus for the optimization. Because both parts of the dual task loss converge in the same range, both factors are set to 1. The note loss l is minimized using the Adam optimizer [37]. To fit the data on a standard graphic board with 6 GB of memory, a batch size of 32 is used. Because of the large size of the training dataset and shallowness of the chosen models, the neural networks converge within a few epochs.

4 EXPERIMENTS AND RESULTS

In this section, DTMST is evaluated on the SingReal dataset and compared with state of the art methods. Precision, recall, and F1 score are calculated using the default settings of mir_eval [38] on the following three levels:

Onsets: An onset is considered correct if it is within a 50-ms interval to a reference onset.

Notes: A note is considered correct if its onset is correct and the pitch is within 50 cents.

Notes with Offset: A note with correct onset and pitch is only considered correct if also the note's offset is within a 50-ms interval to the reference offset.

4.1 Ablation Study

Table 1 shows the impact of different input feature spectrograms on the onset detection without postprocessing. Mel spectrogram with linear magnitude scale performs worst; the others achieve F1 scores of about 72% and are suitable inputs. Using logarithmic magnitude leads to improvements of at least 1% compared to linear scale for all representations. Therefore, a logarithmic magnitude scale generally improves onset detection based on time-frequency representations. Because the HCQT with logarithmic magnitude performs best in this evaluation with an F1 score of 73.37%, additional information of concatenated octaves seems to help onset detection. Unfortunately, the calculation of HCQT is very time-consuming and takes much longer than the calculation of the logarithmic Mel spectrogram. Therefore, the logarithmic Mel spectrogram is used as input in this approach, as shown in Fig. 4. That enables the real-time calculation discussed in SEC. 4.3.

Table 1. Onset level results for various OnsetNN input representations with different magnitude scale.

Input	Scale	Precision (%)	Recall (%)	F1 Score (%)
Mel	Lin	56.99	51.70	61.97
Mel	Log	72.83	71.89	72.17
CQT	Lin	**74.76**	68.14	70.60
CQT	Log	69.11	**75.64**	72.09
HCQT	Lin	73.62	71.93	72.39
HCQT	Log	74.16	73.62	**73.37**

CQT = constant-Q transform; HCQT = harmonic CQT;
Log = logarithmic; OnsetNN = Onset Detection Neural Network.

Table 2. Note level results for various PitchNN input representations with different magnitude scale.

Input	Scale	Precision (%)	Recall (%)	F1 Score (%)
Mel	Lin	71.66	70.17	70.72
Mel	Log	**74.25**	**72.80**	**73.32**
CQT	Lin	73.58	72.18	72.68
CQT	Log	73.83	72.37	72.90
HCQT	Lin	72.82	71.40	71.91
HCQT	Log	71.08	69.35	70.04

CQT = constant-Q transform; HCQT = harmonic CQT;
Log = logarithmic; PitchNN = Pitch Estimation Neural Network.

For the pitch estimation, Table 2 presents the impact of different input feature spectrograms on the estimation results. Although the F1 scores do not differ much for the investigated representations, the Mel spectrogram with logarithmic magnitude leads to the best pitch estimation results. The logarithmic magnitude does not have such a distinct effect for pitch estimation as for onset detection. In the case of the HCQT, the logarithmic scale even leads to a decreased F1 score. Because the HCQT has lower pitch estimation results than the CQT, additional information about the harmonic frequencies is not required for pitch estimation.

The impact of different onset strength functions for use in postprocessing is given in Table 3. Functions calculated from the logarithmic Mel spectrogram show better results than the ones calculated from the STFT, which confirm the better onset results with logarithmic scale in Table 1. In the case of both input spectrograms, super flux shows an improvement of about 1% over the classical spectral flux;

Table 3. Onset level results for different onset strength functions for onset correction.

Onset Function	Spectrogram	F1 Score (%)
No Correction	...	72.17
Spectral Flux	Log Mel	76.85
Super Flux	Log Mel	**77.53**
Spectral Flux	STFT	71.67
Super Flux	STFT	72.59
Complex Domain [39]	STFT	69.55

Log = logarithmic; STFT = short-time Fourier transform.

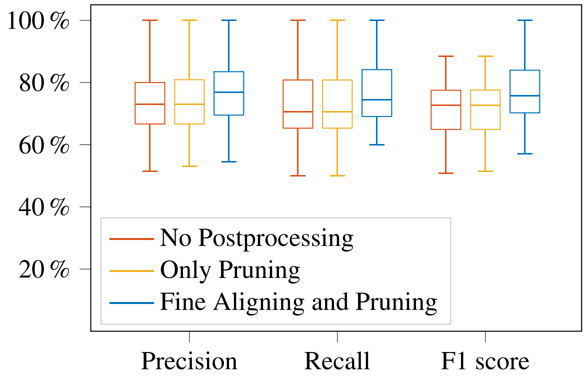

Fig. 8. Evaluation of pruning and fine aligning with super flux as postprocessing for the onset detection.

therefore, it is better suited for onset detection of singing signals. In contrast, the use of the complex domain onset strength function [39] yields to a decreased F1 score and is not suitable for the considered task. Because super flux achieves the highest onset level F1 score of 77.53%, it is used in the DTMST implementation.

In Fig. 8, the effects of different postprocessing steps are compared. Best results are achieved when pruning and super flux alignment are applied. Pruning by itself does not lead to an improvement because OnsetNN does not predict a significant amount of too-short notes on the SingReal dataset. But pruning is generally still useful as a sanity check to prevent invalid note durations. A clear improvement in the onset F1 score of about 4.5% is achieved by fine-aligning the onsets by means of the nearest peak in the super flux curve.

4.2 Transcription Evaluation

In order to evaluate the performance of the DTMST approach, the transcription results for the SingReal dataset of real amateur singing recordings are analyzed on the three levels Onsets, Notes, and Notes with Offset. Table 4 compares those results of DTMST with state of the art methods containing algorithmic (YIN), probabilistic (Tony), and neural network–based approaches (CREPE, Onsets and Frames, and Omnizart). Additionally, the monophonic mode of the commercial software Melodyne (v5) [40] is evaluated. The input audio signals are resampled to 16 kHz for all methods except for Melodyne and Tony, which receive the original 44.1 kHz signals, because they yield better results with the original sampling rate.

Because YIN and CREPE estimate F0 pitch contours in their basic versions, both were extended by a note segmentation stage. Firstly, the pitch contours are rounded to the nearest halftone step. Then pitch changes and silences are used as note boundaries. Notes shorter than 100 ms are discarded.

DTMST shows the best results for nearly all metrics and outperforms the Tony algorithm, representing the best-performing state-of-the-art method, by more than 3.5% concerning the note level. For Notes with Offset, DTMST is the best approach by a large margin of at least 18 dB. The dual-objective approach OaF, from which the dual task approach is inspired, has been trained for about 10,000 epochs on the synthetic singing dataset. It has originally been designed for polyphonic transcription of piano pieces and might have a too-specific architecture that is not suited for singing data. For example, it still predicts a polyphonic output that is not reasonable for sung melodies of one singer. YIN and CREPE results are comparable and depend on their F0 estimation and the used note segmentation, which can be further improved. Nevertheless, the authors expect that the results with improved note segmentation would stay below Tony, which is an improved YIN algorithm.

The best neural network-based state of the art results are achieved by Omnizart, but the F1 scores are still much lower than DTMST. According to the different datasets during training, these differences could result from the lack of generalizability, which often occurs in data-driven approaches. Melodyne performs worst compared to the other state of the art methods, although it offers a special monophonic note detection mode. Because it is primarily used for professional music postprocessing and editing, the note segmentation criteria are probably focused on comprising the whole note with attack and reverberation to enable a meaningful pitch correction. Therefore, the note onsets might be estimated too early.

4.3 Real-Time Transcription

DTMST can easily be modified to transcribe audio in a real-time scenario. Lots of pitch tracking approaches are not real-time capable, but several implementations like real-time F0 estimation of human voice [41] or real-time pitch tracking of instruments with an extended complex Kalman filter [42] have been developed. The main advantage of such a real-time capable transcription system is a better user experience because of the direct feedback that can be given to the musician while recording.

The structure of the real-time DTMST is visualized in Fig. 9. The main difference between an online (real-time) and offline scenario is that the incoming frames of an audio stream are gradually accessed in the online case. Consequently, there is only a limited time span available to preprocess each frame. For a sampling rate of 16 kHz and hop size of 512 like in this work, the available time for each preprocessing step is 32 ms.

Table 4. Comparison of DTMST results with state of the art methods using SingReal dataset concerning precision P, recall R, and F1 score $F1$.

	Onsets			Notes			Notes with Offset		
Approach	P (%)	R (%)	$F1$ (%)	P (%)	R (%)	$F1$ (%)	P (%)	R (%)	$F1$ (%)
YIN [2]	51.61	47.86	48.46	51.05	47.33	47.92	10.90	10.67	10.56
Tony [4]	78.13	68.01	72.30	**76.22**	66.39	70.56	30.00	27.12	28.37
CREPE [5]	38.95	53.07	44.31	38.75	52.71	44.05	10.64	13.73	11.85
OaF [6]	58.58	58.43	58.15	57.03	57.02	56.68	34.03	33.58	33.71
Omnizart [17]	60.87	59.97	59.82	59.41	58.60	58.41	22.55	22.57	22.34
Melodyne [40]	26.68	26.40	26.17	20.08	19.65	19.54	10.93	11.33	11.06
DTMST	**78.41**	**77.06**	**77.53**	75.07	**73.71**	**74.19**	**52.26**	**51.50**	**51.76**

DTMST = Dual Task Monophonic Singing Transcription; OaF = Onsets and Frames.

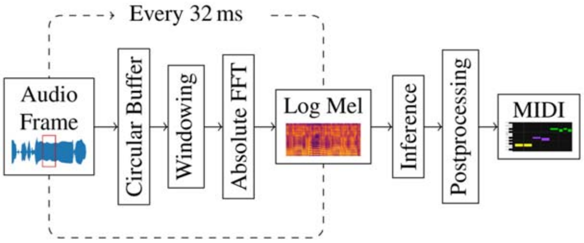

Fig. 9. Structure of Dual Task Monophonic Singing Transcription (DTMST) for real-time processing.

Because Mel spectrograms can be calculated frame by frame, the calculation is real-time capable when a circular buffer is used. Each column of the Mel spectrogram is calculated by means of the magnitude of the fast Fourier transform and a precalculated Mel filter bank. In order to follow the real-time constraint, only preprocessing is done in the audio computing thread. The preprocessing of each audio frame takes less than 8 ms in average using a Dell XPS 13 2018 (9370) with an Intel Core i7-8550U.

Inference and postprocessing are done in a separate computing thread and are based on the complete previous singing recording. Therefore, they do not have to comply with the real-time constraint of the audio thread because the transcription can be done with an independent repetition rate. Every update cycle, the whole spectrogram is inferred, and a MIDI representation is drawn on the screen. In these experiments, the refresh rate of the second computing thread was about 5 Hz in average, but it decreases with increasing recording time because of the higher data amount. Furthermore, more temporal context is available to the model if the recording time is higher. Hence, the model improves its estimation with each new frame. After a complete recording has been input, the corresponding results match the ones of the offline system.

5 CONCLUSION

DTMST, a dual task approach with two independent neural networks, for onset detection and pitch estimation, has been proposed for monophonic singing transcription. In order to transcribe non-professional vocals, the networks have been trained on a synthetic dataset of artificial song melodies. The performance is evaluated on the new SingReal dataset, which consists of real recordings of non-professional singers. DTMST outperforms all state of the art algorithms by at least 3.5% in F1 score. Furthermore, a real-time transcription is realizable with DTMST.

In future works, the impact of synthesized training data instead of real recordings should be analyzed. Therefore, a large corpus of annotated real recordings is necessary, for example, by active learning for semi-automatic annotation. Furthermore, notes' offset estimation could be integrated by a third neural network similar to OnsetNN or as an additional label in the onset detection.

6 REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic Music Transcription: An Overview," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 20–30 (2019 Jan.). https://doi.org/10.1109/MSP.2018.2869928.

[2] A. De Cheveigné and H. Kawahara, "YIN, A Fundamental Frequency Estimator for Speech and Music," *J. Acoust. Soc. Am.*, vol. 111, no. 4, pp. 1917–1930 (2002 Apr.). https://doi.org/10.1121/1.1458024.

[3] M. Mauch and S. Dixon, "PYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 659–663 (Florence, Italy) (2014 May). https://doi.org/10.1109/ICASSP.2014.6853678.

[4] 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 *Proceedings of the 1st International Conference on Technologies for Music Notation and Representation (TENOR)*, pp. 23–30 (Paris, France) (2015 May).

[5] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A Convolutional Representation for Pitch Estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 161–165 (Calgary, Canada) (2018 Apr.). https://doi.org/10.1109/ICASSP.2018.8461329.

[6] C. Hawthorne, E. Elsen, J. Song, et al., "Onsets and Frames: Dual-Objective Piano Transcrip-

tion," *arXiv preprint arXiv:1710.11153* (2017 Oct.). https://doi.org/10.48550/arXiv.1710.11153.

[7] M. Ryynänen and A. Klapuri, "Transcription of the Singing Melody in Polyphonic Music," in *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pp. 222–227 (Victoria, Canada) (2006 Oct.).

[8] A. McLeod, R. Schramm, M. Steedman, and E. Benetos, "Automatic Transcription of Polyphonic Vocal Music," *Appl. Sci.*, vol. 7, no. 12, paper 1285 (2017 Dec.). https://doi.org/10.3390/app7121285.

[9] E. Molina, L. J. Tardón, A. M. Barbancho, and I. Barbancho, "SiPTH: Singing Transcription Based on Hysteresis Defined on the Pitch-Time Curve," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 2, pp. 252–263 (2015 Feb.). https://doi.org/10.1109/TASLP.2014.2331102.

[10] L. Yang, A. Maezawa, J. B. L. Smith, and E. Chew, "Probabilistic Transcription of Sung Melody Using a Pitch Dynamic Model," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 301–305 (New Orleans, LA) (2017 Mar.). https://doi.org/10.1109/ICASSP.2017.7952166.

[11] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii, "Bayesian Singing Transcription Based on a Hierarchical Generative Model of Keys, Musical Notes, and F0 Trajectories," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 1678–1691 (2020 May). https://doi.org/10.1109/TASLP.2020.2996095.

[12] F. Rigaud and M. Radenen, "Singing Voice Melody Transcription Using Deep Neural Networks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 737–743 (New York, NY) (2016 Aug.).

[13] R. Nishikimi, E. Nakamura, S. Fukayama, M. Goto, and K. Yoshii, "Automatic Singing Transcription Based on Encoder-Decoder Recurrent Neural Networks With a Weakly-Supervised Attention Mechanism," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 161–165 (Brighton, UK) (2019 May). https://doi.org/10.1109/ICASSP.2019.8683024.

[14] H. Cuesta, B. McFee, and E. Gómez, "Multiple F0 Estimation in Vocal Ensembles Using Convolutional Neural Networks," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, pp. 302–309 (Montréal, Canada) (2020 Oct.).

[15] Z.-S. Fu and L. Su, "Hierarchical Classification Networks for Singing Voice Segmentation and Transcription," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 900–907 (Delft, The Netherlands) (2019 Nov.).

[16] J.-Y. Hsu and L. Su, "VOCANO: A Note Transcription Framework for Singing Voice in Polyphonic Music," in *Proceedings of the 22nd International Society of Music Information Retrieval Conference (ISMIR)*, pp. 293–300 (Online) (2021 Nov.).

[17] Y.-T. Wu, Y.-J. Luo, T.-P. Chen, et al., "Omnizart: A General Toolbox for Automatic Music Transcription," *arXiv preprint arXiv:2106.00497* (2021 Jun.). https://doi.org/10.48550/arXiv.2106.00497.

[18] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music Source Separation in the Waveform Domain," *arXiv preprint arXiv:1911.13254* (2019 Nov.). https://doi.org/10.48550/arXiv.1911.13254.

[19] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: A Fast and Efficient Music Source Separation Tool With Pre-Trained Models," *J. Open Source Softw.*, vol. 5, no. 50, paper 2154 (2020 Jun.). https://doi.org/10.21105/joss.02154.

[20] J. B. Allen and L. R. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," *Proc. IEEE*, vol. 65, no. 11, pp. 1558–1564 (1977 Nov.). https://doi.org/10.1109/PROC.1977.10770.

[21] S. Stevens, J. Volkmann, and E. B. Newman, "A Scale for the Measurement of the Psychological Magnitude Pitch," *J. Acoust. Soc. Am.*, vol. 8, no. 3, pp. 185–190 (1937 Jan.). https://doi.org/10.1121/1.1915893.

[22] K. W. Cheuk, K. Agres, and D. Herremans, "The Impact of Audio Input Representations on Neural Network Based Music Transcription," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6 (Glasgow, UK) (2020 Sep.). https://doi.org/10.1109/IJCNN48605.2020.9207605.

[23] C. Schörkhuber and A. Klapuri, "Constant-Q Transform Toolbox for Music Processing," in *Proceedings of the 7th Sound and Music Computing Conference*, pp. 3–64 (Barcelona, Spain) (2010 Jul.).

[24] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep Salience Representations for F_0 Estimation in Polyphonic Music," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 63–70 (Suzhou, China) (2017 Oct.).

[25] S. Böck and G. Widmer, "Maximum Filter Vibrato Suppression for Onset Detection," in *Proceedings of the 16th International Conference on Digital Audio Effects*, vol. 7, pp. 55–61 (Maynooth, Ireland) (2013 Sep.).

[26] R. M. Bittner, J. Salamon, M. Tierney, et al., "MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 155–160 (Taipei, Taiwan) (2014 Oct.).

[27] Smule, Inc., "DAMP-VPB: Digital Archive of Mobile Performances - Smule Vocal Performances Balanced," *Zenodo* (2017 Nov.). https://doi.org/10.5281/zenodo.2616690.

[28] H. Cuesta, E. Gómez Gutiérrez, A. Martorell Domínguez, and F. Loáiciga, "Analysis of Intonation in Unison Choir Singing," in *Proceedings of the 15th International Conference on Music Perception and Cognition*, paper 561 (Graz, Austria) (2018 Jul.).

[29] J.-Y. Wang and J.-S. R. Jang, "On the Preparation and Validation of a Large-Scale Dataset of Singing Transcription," in *Proceedings of the IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 276–280 (Toronto, Canada) (2021 Jun.). https://doi.org/10.1109/ICASSP39728.2021.9414601.

[30] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, Classical and Jazz Music Databases," in *Proceedings of the 3rd International Conference on Music Information Retrieval*, pp. 287–288 (Paris, France) (2002 Oct.).

[31] V. Emiya, R. Badeau, and B. David, "Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 6, pp. 1643–1654 (2010 Aug.). https://doi.org/10.1109/TASL.2009.2038819.

[32] S. Raj, "ABC Notation of Tunes," https://kaggle.com/raj5287/abc-notation-of-tunes (accessed Jul. 6, 2022).

[33] P. Mooney, "Song Lyrics: Poetry and Lyrics (TXT Files)," https://kaggle.com/paultimothymooney/poetry (accessed Jul. 6, 2022).

[34] " Pyphen," https://pyphen.org (accessed Jul. 6, 2022).

[35] K. Oura, A. Mase, T. Yamada, et al., "Recent Development of the HMM-Based Singing Voice Synthesis System — Sinsy," in *Proceedings of the 7th ISCA Workshop on Speech Synthesis*, pp. 211–216 (Kyoto, Japan) (2010 Sep.).

[36] S. Dixon, "Onset Detection Revisited," in *Proceedings of the 9th International Conference on Digital Audio Effects*, vol. 120, pp. 133–137 (Montréal, Canada) (2006 Sep.).

[37] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980* (2014 Dec.). https://doi.org/10.48550/arXiv.1412.6980.

[38] C. Raffel, B. McFee, E. J. Humphrey, et al., "mir_eval: A Transparent Implementation of Common MIR Metrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 367–372 (Taipei, Taiwan) (2014 Oct.).

[39] J. P. Bello, C. Duxbury, M. Davies, and M. Sandler, "On the Use of Phase and Energy for Musical Onset Detection in the Complex Domain," *IEEE Signal Process. Lett.*, vol. 11, no. 6, pp. 553–556 (2004 Jun.). https://doi.org/10.1109/LSP.2004.827951.

[40] Celemony Software GmbH, "Melodyne," https://www.celemony.com (accessed Jul. 6, 2022).

[41] W. B. Kuhn, "A Real-Time Pitch Recognition Algorithm for Music Applications," *Comput. Music J.*, vol. 14, no. 3, pp. 60–71 (1990 Fall). https://doi.org/10.2307/3679960.

[42] O. Das, J.O. Smith, and C. Chafe, "Real-Time Pitch Tracking in Audio Signals With the Extended Complex Kalman Filter," in *Proceedings of the 20th International Conference on Digital Audio Effects*, pp. 118–124 (Edinburgh, UK) (2017 Sep.).

THE AUTHORS

Markus Schwabe

Sebastian Murgul

Michael Heizmann

Markus Schwabe studied electrical engineering and information technology at the Karlsruhe Institute of Technology (KIT) and received his Master of Science in 2016. He is currently working at the Institute of Industrial Information Technology (IIIT) at the KIT as a research associate. His research interests include signal and audio processing, machine learning, and music signal separation.

•

Sebastian Murgul studied electrical engineering and information technology at the Karlsruhe Institute of Technology (KIT) and received his Master of Science in 2020. During this time, he founded the start-up company Klangio GmbH, where he is now working as the chief executive officer. His research interests include signal and audio processing, machine learning, and music transcription.

•

Michael Heizmann is professor of Mechatronic Measurement Systems and director at the Institute of Industrial Information Technology (IIIT) at the Karlsruhe Institute of Technology (KIT). His research areas include machine vision, image processing, image and information fusion, measurement technology, machine learning, artificial intelligence, and their applications.