Pattern Recognition Letters 128 (2019) 340–347

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

# Automatic music transcription for traditional woodwind instruments *sopele*

Arian Skoki, Sandi Ljubic, Jonatan Lerga, Ivan Štajduhar*

*University of Rijeka, Faculty of Engineering, Department of Computer Engineering, Vukovarska 58, Rijeka 51000, Croatia*

ARTICLE INFO

*Article history:*
Received 4 June 2019
Revised 23 September 2019
Accepted 25 September 2019
Available online 25 September 2019

*Keywords:*
Automatic music transcription
Traditional woodwind instrument
Sopele
Discrete Fourier transform
Machine learning

ABSTRACT

*Sopela* is a traditional hand-made woodwind instrument, commonly played in pair, characteristic to the Istrian peninsula in western Croatia. Its piercing sound, accompanied by two-part singing in the hexatonic Istrian scale, is registered in the UNESCO Representative List of the Intangible Cultural Heritage of Humanity. This paper presents an insight study of automatic music transcription (AMT) for *sopele* tunes. The process of converting audio inputs into human-readable musical scores involves multi-pitch detection and note tracking. The proposed solution supports this process by utilising frequency-feature extraction, supervised machine learning (ML) algorithms, and postprocessing heuristics. We determined the most favourable tone-predicting model by applying grid search for two state-of-the-art ML techniques, optionally coupled with frequency-feature extraction. The model achieved promising transcription accuracy for both monophonic and polyphonic music sources encompassed in the originally developed dataset. In addition, we developed a proof-of-concept AMT system, comprised of a client mobile application and a server-side API. While the mobile application records, tags and uploads audio sources, the back-end server applies the presented procedure for converting recorded music into a common notation to be delivered as a transcription result. We thus demonstrate how collecting and preserving traditional *sopele* music, performed in real-life occasions, can be effortlessly accomplished on-the-go.



## 1. Introduction

Music transcription is a process of converting audio signals into musical scores. A complete transcription would require determining the pitch, timing, and instrument of all sound events comprised in a given audio source. Due to a high-level complexity of such a task, the goal is usually redefined to notate a particular aspect of the music signal, for example the dominant melody or the sound of the most prominent instrument [12]. In general, Automatic Music Transcription (AMT) is a challenging problem that can be divided into several sub-domains, with pitch detection and rhythm quantisation being the ones most investigated.

Pitch is determined by the combination of different frequencies. Namely, a musical tone consists of a fundamental frequency, which is accompanied by other, higher frequency harmonics (the concept illustrated in the Supplement).

The sound of an instrument gets its characteristics from the fundamental frequency and variation of accompanied harmonics. Pitch detection quality usually depends on the type of instrument that is being played. The instrument most commonly used for pitch detection studies is a piano - however, this paper focuses on a traditional woodwind instrument, whose pitch is additionally influenced by the artist's performance. Specifically, deviations in tone frequency can be introduced as a result of the artist's (in)ability to blow air constantly and with enough power.

While the problem of pitch detection for monophonic signals might be considered solved, the estimation of concurrent pitches in a time window, also called multiple-F0 or multi-pitch detection, still remains open [5]. In such context, polyphonic transcription represents a more complex issue because multiple tones interfere with each other and create a mixture, which is sometimes hard to recognise. The problem gets more difficult as the number of sound sources (i.e. instruments) increases. Therefore, a vast majority of AMT systems, which perform multi-pitch detection and note tracking, enforce certain limitations to the degree of polyphony or the instrument type.

The traditional woodwind instrument *sopela*, described in detail in Section 2, is always played in pair (plural form - *sopele*), hence only two tones are played at the same time. This alleviates the polyphonic transcription complexity, as the frequency domain is not so congested. This paper builds upon the mentioned fact, and aims for an AMT solution for the *sopele*, an internationally recognised cultural heritage.

* Corresponding author.
*E-mail address:* istajduh@riteh.hr (I. Štajduhar).

https://doi.org/10.1016/j.patrec.2019.09.024
0167-8655/

### 1.1. Related work

Contemporary AMT systems are still evidently inferior to expert human musicians in both accuracy and flexibility, but successful achievements have been reported for polyphonic music of limited complexity. Many existing solutions can be inspected within Music Information Retrieval Evaluation eXchange (MIREX) ecosystem,[1] which provides a set of formal assessments through which various systems, algorithms and techniques, are evaluated under controlled conditions.

Most multi-pitch estimation and note tracking solutions utilise methods derived from signal processing, wherein notes are detected according to audio features from time-frequency domain. Spectrogram factorisation methods consider the input spectrogram as a matrix to be decomposed into a series of pitch templates and activations. Different strategies based on that principle can be used for pitch estimation, such as non-negative matrix factorisation (NMF) [22,26], probabilistic latent component analysis (PLCA) [2,4], and sparse coding [7,17]. One of the most recent attempts to provide a holistic approach to AMT successfully integrates improved PLCA-based multi-pitch detection with rhythm quantisation method, based on a metrical hidden Markov model [15].

Apart from spectrogram factorisation techniques, a significant interest has been given recently to heuristics and machine learning (ML) techniques able to perform AMT-based tasks. Corresponding solutions aim to directly classify meaningful descriptors extracted from audio windows to the output pitches. This concept assumes deriving appropriate classifiers that can be trained by making use of available learning datasets. As shown in [25], supervised classification can be used for the note tracking part of the AMT process alone, thus replacing typical strategies that rely on custom hand-crafted rules. Random forests (RF) classifier has been successfully used for automatic identification of instruments in audio records [14], as well as for AMT of *a cappella* performances having multiple singers [19]. In recent years, neural networks have also been applied to musical data. Specifically, in [16], a deep belief network is used for the classification-based polyphonic piano transcription, with multiple-note training being suggested for improving transcription accuracy. In yet another AMT approach for piano music, a specific combination of an acoustic model - based on a convolutional neural network (CNN), and a music language model - based on a recurrent neural network (RNN) - has been effectively implemented [21]. Additionally, CNNs were shown to be applicable for classifying music genres [20], and the most prominent drum set instruments [10].

As noted before, piano is evidently the most investigated instrument in the AMT context. Significantly less AMT studies can be found for other instruments, such as drums [10,27] or guitar [1,3]. Woodwind instruments are not specifically tackled whatsoever, possibly because of pitch deviations that are inherent to the performer's playing style. Nevertheless, MIREX dataset, used to evaluate general AMT solutions, involves an audio source from a woodwind quintet, comprised of a flute, an oboe, a clarinet, a horn, and a bassoon.

While analysing a large number of available methods and techniques suitable for AMT tasks, we made our choice in accordance with specific features of the *sopela* woodwind instrument, its typical playing setup, and the underlying musical scale. In order to deal with the unwanted variations in the *sopele* polyphonic-sound pitch, we utilised two state-of-the-art supervised ML approaches, optionally coupled with a frequency-spectrum-extraction technique. We juxtaposed an RF model, a "traditional" ML representative, and a CNN model, a deep-learning alternative. Hence, in our case, multi-pitch detection is based on classifying either audio waveforms, or audio features, which are previously extracted from the waveforms using discrete Fourier Transform (DFT). Best model hyperparameters were estimated using grid search. Because there were no existing datasets containing representative *sopele* audio clips, we had to generate training data from scratch.

## 2. *Sopela* woodwind instrument and the Istrian scale

Back in 2009, two-part singing and playing in the hexatonic Istrian scale (shown in the Supplement), characteristic to the Istrian peninsula in western Croatia, has been registered in the UNESCO Representative list of the Intangible Cultural Heritage of Humanity.[2] Typical musical instruments used in the corresponding style are *sopele* - traditional hand-made wooden aerophones of piercing sound. *Sopela* can be considered as a descendant of the shawn, which was very popular during medieval and renaissance periods. Today, only remaining descendants of the old shawn are found in the Swiss Alps, the Italian Apennines and in western Croatia - namely in the Kvarner Bay and Istria. Body of a *sopela* is made from a single piece of wood having six finger holes. The mouthpiece consists of a double-reed and a pirouette, similar to oboe. *Sopele* are always played in pair, consisting of a *small sopela* having a higher pitch, and a *great sopela* having a lower pitch. Playing *sopele* often allows for improvisation and decorations in musical sentences, however, the core of every music piece provides a certain structure with constant rhythm periods. Being the important part of a cultural heritage, the sound of *sopele* is still a part of everyday life and festive occasions, traditionally involved in wedding ceremonies, community and family gatherings and religious services.

From the AMT perspective, the *sopela* instrument imposes both advantageous and unfavourable features. Each *sopela* can produce only six distinct tones (in line with the hexatonic Istrian scale), which is not a lot when compared to contemporary music instruments. Being traditionally played in pair, without any extra instruments around, *small* and *great sopela* will generate two concurrent tones at most. Hence, related pitch recognition process should be able to detect merely 12 distinct tones in monophonic modality, and their 36 non-permutative combinations in polyphonic modality. Seeing that two concurrently played *sopele* will not make the frequency domain considerably congested, the multi-pitch detection problem in the respective context becomes mitigated to a certain extent. Nevertheless, despite the relative simplicity of the *sopele* playing setting, a generalised transcription model for associated sound is still hard to achieve. The main constraint lies in the very core of the *sopele* production - due to its rareness and specificity, this instrument is exclusively made by hand of very few craftspeople, who had acquired their skills and knowledge from their elders. Consequently, every instrument is literally unique, producing intrinsically different tone instances. Another issue is inherent to all wind instruments - sound characteristics depend on the performers' ability to blow air. Namely, if air is not blown under constant pressure, the tone pitch will descend with falling pressure. Although *sopela* seems a simple instrument at first glance, it requires a lot of knowledge, practice, and the correct technique in order to be played properly. A significant amount of sample data has to be collected in order to alleviate the mentioned problems.

This paper can be considered as a first attempt to formally propose an AMT solution for the *sopele* sound. We developed a proof-of-concept full-stack system in order to provide the possibility of

[1] https://www.music-ir.org/mirex/wiki/MIREX_HOME.

[2] https://ich.unesco.org/en/RL/two-part-singing-and-playing-in-the-istrian-scale-00231.

preserving this internationally-recognised (intangible) cultural heritage *in-situ*. This system is comprised of a prototype client mobile application and a server-side API. While the mobile application records, tags and uploads audio sources, the back-end server utilises ML techniques for converting recorded music into a common notation to be delivered as a transcription result. We, thus, demonstrate how non-sophisticated apparatus can be used for obtaining a note transcript on the smartphone screen immediately after recording the related *sopele* sound in real-life locations.

## 3. Material and methods

### 3.1. Data acquisition

As mentioned before, to the extent of our knowledge, publicly available datasets containing representative *sopele* music sources do not exist. In that respect, the first initiative of this insight study was to generate one for further analysis and eventual dissemination. For data acquisition, we used one set of *sopele*, which consisted of one *small* and one *great sopela*.

In order to learn the characteristics of specific tones, we gathered audio recordings of every tone. Recordings of the same tone had to be different in their signal strength and ambience noise. That way, the model would be robust to signal strength invariance and background noise. *Sopele* were played from several different positions to achieve this. One recording was recorded in front of the microphone, whereas others had been recorded from a set distance, ranging from 1 to 10 m away from the microphone. The third way of recording was made by walking away from the microphone, and then going back. One person was playing the *sopele* in all the recordings. Polyphonic music transcription required all combinations of tones coupled with one another, therefore we merged single tones of the *small* and *great sopela* into stereo files. In the end, we recorded silence to enable detection of ambient noise in the music piece recordings. Apart from single tone instances, we also recorded some traditional songs and choruses, performed by the same player. We collected all the recordings using the same recording device.

### 3.2. Data preprocessing

The first data-preprocessing step was splitting the audio recordings into smaller time windows of 10 ms. A time window of 10 ms is small enough to capture every single tone from a set of tones played in a faster tempo, yet is large enough to preserve the spectral quality of the tone pitch. Polyphonic model generates merged-tones data by creating stereo files. Channels in a stereo file may not be of the same strength, because of different recording positions, as described in Section 3.1. That could result in a misclassification of a merged tone as a single tone. To prevent that, we linearly amplified the channel that was weaker to the same level as the stronger channel. After that, all audio files were monophonic, balanced, and 10 ms long, additionally attributed with the corresponding class label. In the following text, we denote this instance of the curated dataset with the WAV prefix, indicating its usage for learning directly from single-channel waveform data.

#### 3.2.1. Frequency-feature extraction

Because ML models normally benefit from using some sort of feature extraction to deal with the hard-to-model nonlinearities and variations in the data, we introduced an additional (optional) preprocessing step, namely frequency-spectrum extraction. This was done in order to explicitly analyse the benefits of using this kind of feature extraction together with the ML techniques, as opposed to learning from raw audio-waveform data. We extracted the frequencies needed for determining the pitch of the tone using the one-dimensional DFT, providing an insight into frequency content of the analysed audio data [9,24]. Because it is expected that separate time windows contained tone occurrences irrelative to their duration, we utilised the one-dimensional DFT rather than two-dimensional time-frequency distributions, due to its computational efficiency. All recordings had to be brought to the same scale, therefore we used maximum amplitude division in order to unit-scale the values (range from 0 to 1). We cut all amplitudes below a set threshold of 50 dB (which is comparable to a quiet restaurant) to 20% of their size. This way, we significantly lowered the influence of inessential harmonics, thus enabling easier recognition of the class *rest* (ambient noise). We stored the resulting array of frequency amplitudes in a Hierarchical Data Format 5 (HDF5) file, as well as the class labels accompanying them. In the following text, we use the DFT prefix to represent the models utilising the described frequency-feature extraction.

### 3.3. Tone classification models

AMT for woodwind instruments has to deal with the problem of ambiguous intra-class and inter-class variations in the pitch. In the case of *sopele*, these problems are further accentuated by the fact that each instrument has unique properties (because it is hand-crafted), and its sound is heavily influenced by the artist's performance (Section 2). Furthermore, *sopele* are always played in pair, normally on festive occasions, in public (presence of ambient noise), which requires a robust polyphonic AMT solution. Consequently, we considered the use of supervised ML techniques as a solution to these problems. Due to the evident lack of research efforts targeting AMT for woodwind instruments, there is no representative baseline approach for our target domain. Hence, we opted for using one "traditional" state-of-the-art ML technique (RF), and one state-of-the-art deep learning technique (CNN), in order to determine which one is more suitable for AMT in this domain.

Because the performance of both ML techniques depends on the choice of several hyperparameters, we explored and evaluated the most promising hyperparameter combinations using grid search over a validation subset. We selected the most influential hyperparameters for varying, and determined the boundaries and step sizes of their value ranges using common sense, inspired by similar reported work. Furthermore, we either coupled both ML techniques with DFT-extracted features, or applied them directly to audio waveforms (WAV prefix). This was done in order to inspect whether such feature extraction is beneficial for predictive accuracy. Therefore, we inspected a total of 4 modelling approaches (models) in more detail: (1) WAV+RF, (2) DFT+RF, (3) WAV+CNN, and (4) DFT+CNN. Because CNNs are capable of learning feature extraction, using DFT for input transformation might seem unnecessary for a CNN. We have shown, however, that CNNs are incapable of dealing with this kind of representation, probably due to the insufficient amount of available training data.

#### 3.3.1. RF models

RF is an ensemble of classification trees, where each tree is learned independently from a randomly sampled subset of the training dataset [6]. Each tree is grown greedily by using a randomly sampled subset of features at each recursive-partitioning step, preventing overly-correlated specialisations of trees, which in turn reduces the variance of the ensemble.

We considered the following hyperparameter values for the RF models: the number of trees, $T \in \{800, 900, 1000\}$; the minimum number of data instances falling into a specific node, required for considering a split, $n_{split} = \{2, 4, 6\}$; the minimum number of data instances required to fall into each leaf node, to allow the split,

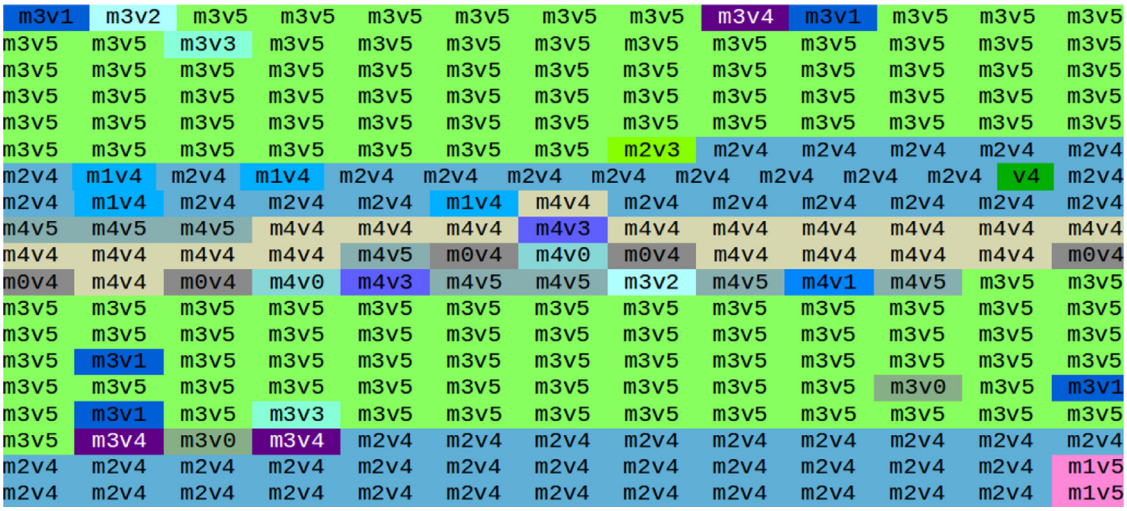


**Fig. 1.** A depiction of the results of raw prediction from individual time windows of 10 ms. Every class is represented with a class name and a distinct colour.

$n_{leaf} = \{1, 2, 4\}$; the number of sampled features for performing the next split, $f = \sqrt{n}$, where $n$ is the total number of data instances; feature-splitting criterion, $\gamma = \{Gini$ impurity, information gain$\}$; the maximum tree depth, $d = \{60, 80, 100\}$; and whether bootstrap is used or not, $\mathcal{B} = \{$true, false$\}$.

#### 3.3.2. CNN models

A standard CNN model stacks a trainable feature extractor (convolutional layers) on top of a classifier (fully-connected layers), enabling learning the parameters of both, simultaneously [13]. Convolutional layers offer a degree of invariance to translation, rotation and scale in the data, while down-scaling the input representation.

In this work, we considered a standard convolutional feed-forward neural network only [11]. We chose a relatively simple and shallow network architecture, in order to cover smaller deviations from tonal-harmonics original positions, aiming at increasing model expressiveness and flexibility. Distinct convolutional layers in our setup consist of the following building blocks: a convolutional filter-bank sub-layer (consisting of several one-dimensional convolutional filters), following a max-pooling sub-layer (stride and size 2), and a nonlinear transform sub-layer (activation function). The last convolutional layer is followed by a fully-connected layer, following a *softmax* classification layer. Both the convolutional and the fully-connected layers utilised the rectified linear unit (*ReLU*) as the activation function. For preventing overfitting, each convolutional layer was followed by a dropout layer, using a 25% dropout rate, whereas the fully-connected layer used a 50% dropout rate. CNN models were trained using *Adadelta* search, minimising categorical cross-entropy (loss function), for 12 epochs, using mini-batch size of 1000. These, as well as other choices in regard to the model architectures, were inspired mostly by [21].

We considered the following hyperparameter values for the CNN models: the number of stacked convolutional layers, $L_c = \{1, 2, 3, 4, 5\}$; the number of filters in the first convolutional layer, $f_1 = \{16, 32\}$; the number of filters in each subsequent convolutional layer, $f_2 = \{16, 32, 64, 128\}$; convolutional layer filter size, $c = \{3, 5, 7\}$; and the number of neurons in the fully connected layer, $h = \{64, 128, 256\}$. If multiple convolutional layers were stacked, the same values of $f_2$ and $c$ were applied to all of them.

### 3.4. Postprocessing for the full-stack AMT system

Each model predicts classes for individual, 10 ms long, windows. Consecutive predicted classes represent a tone of uninterrupted duration on the observed interval. Fig. 1 renders raw prediction. Every class is represented with a class name and a distinct colour. Class name consists of a letter *m* or *v*, which determines the type of *sopela* (*small* or *great*, respectively), and a number ranging from 0 to 5 depending on tone pitch (0 being the highest pitch). As can be seen, misclassification may occur and interrupt the duration of a correctly classified tone.

Our postprocessing method first strips silence at the beginning and at the end of the music file. Next, it applies an ignore-threshold of 3 windows in order to reduce the influence of smaller groups of misclassified instances (similar to minimum duration pruning [8]). It groups consecutive predicted tones, and eliminates the groups whose length is below the ignore-threshold by appending them to the next largest group. Looking back to Fig. 1, windows of class 'm3v5' (green colour) were grouped together, and misclassifications that occurred in-between were ignored and added to that group count.

Following the auto-correction described above, generating musical scores takes place. In order to infer music sheets from raw predictions, we had to set some constraints. One can determine specific tone duration by the number of time windows compared to the beat. We calculate the beat by transforming beats per minute to beats per second, and then dividing that number by the time-window length. For example, beat value of 33 means that 33 consecutive equivalent tones should be predicted for a quarter note to be recognised. We should note that wrong rhythm estimation may occur depending on the number of predicted time windows, compared to the predefined note duration (shown in the Supplement).

For clarity purposes, Fig. 2 depicts a complete pipeline of the proposed transcription process. The libraries used for AMT system implementation were: *NumPy* (frequency features extraction in the preprocessing stage), *scikit-learn* (RF modelling), *Keras* (CNN modelling), *Lilypond* and *Abjad* (creating music sheets in the postprocessing stage). If not stated otherwise in the text, default values of built-in functions were used.

## 4. Results

Following preprocessing (Section 3.2), we randomly split the dataset into three stratified disjoint subsets: training (64%), validation (16%) and test (20%). In absolute numbers, this split corresponded to 18104/4526/5657 data instances for monophonic models, and 68240/17060/21323 for polyphonic models. We used the

**Table 1**
Hyperparameter values of models giving the best $F_1$ performance over the validation set. We selected these as the standard for subsequent tests, reported in Table 2 and in Fig. 3.

| Model architecture | Hyperparameter values |
|---|---|
| Monophonic setup | |
| WAV+RF | $T = 900$; $\gamma = Gini$ impurity; $n_{split} = 2$; $n_{leaf} = 1$; $d = 80$; $\mathcal{B} =$ false |
| DFT+RF | $T = 1000$; $\gamma =$ information gain; $n_{split} = 2$; $n_{leaf} = 1$; $d = 60$; $\mathcal{B} =$ false |
| WAV+CNN | $L_c = 2$; $f_1 = 32$; $f_2 = 64$; $c = 3$; $h = 128$ |
| DFT+CNN | $L_c = 2$; $f_1 = 32$; $f_2 = 32$; $c = 5$; $h = 128$ |
| Polyphonic setup | |
| WAV+RF | $T = 1000$; $\gamma = Gini$ impurity; $n_{split} = 6$; $n_{leaf} = 1$; $d = 60$; $\mathcal{B} =$ false |
| DFT+RF | $T = 900$; $\gamma = Gini$ impurity; $n_{split} = 2$; $n_{leaf} = 1$; $d = 80$; $\mathcal{B} =$ false |
| WAV+CNN | $L_c = 2$; $f_1 = 32$; $f_2 = 32$; $c = 3$; $h = 128$ |
| DFT+CNN | $L_c = 3$; $f_1 = 32$; $f_2 = 32$; $c = 3$; $h = 256$ |

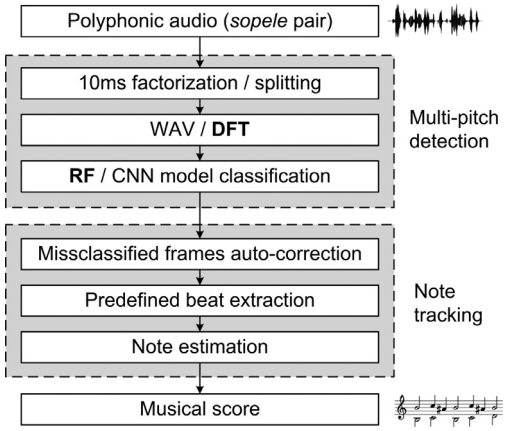


**Fig. 2.** The proposed AMT system for *sopele* music.

**Table 2**
The results on several standard metrics obtained using the most promising hyperparameter values (Table 1). Best results in the monophonic/polyphonic setup are emphasised for each metric.

| Model architecture | Precision | Recall | $F_1$ | Classif. accuracy |
|---|---|---|---|---|
| Monophonic setup | | | | |
| WAV+RF | 0.9878 | 0.9882 | 0.9880 | 0.9878 |
| DFT+RF | 0.9986 | 0.9985 | 0.9985 | 0.9985 |
| WAV+CNN | 0.8629 | 0.7227 | 0.7866 | 0.7189 |
| DFT+CNN | **0.9996** | **0.9996** | **0.9996** | **0.9996** |
| Polyphonic setup | | | | |
| WAV+RF | 0.5644 | 0.5336 | 0.5486 | 0.5337 |
| DFT+RF | **0.9294** | **0.9269** | **0.9281** | **0.9270** |
| WAV+CNN | 0.4730 | 0.6215 | 0.5372 | 0.6217 |
| DFT+CNN | 0.9230 | 0.9138 | 0.9184 | 0.9141 |

validation subset for estimating the best-performing model architectures, whereas the test subset was used to quantitatively evaluate the best-performing models. Finally, we tested our full-stack AMT system for inferring music sheets by using a recording of a traditional *sopele* music piece.

We evaluated model performance quantitatively using standard classifier-evaluation metrics for AMT, namely: precision, recall, $F_1$ score, and classification accuracy [18]. Although precision, recall and $F_1$ score are inherently used for binary classifiers, in this multi-class setup they were estimated in a one-vs-all manner by calculating the mean values obtained over each distinct class [23].

We put both RF and CNN modelling techniques to the test, coupled with the optional feature extraction using DFT, giving a total of four types of models under inspection. We tested these models under a monophonic, and under a polyphonic setup, which differ in the number of classes. Monophonic setup uses 13 classes (six for every type of *sopela*, and additional one for detecting ambient noise), whereas the polyphonic setup uses 49 classes (monophonic setup is expanded with 36 non-permutative combinations of tones from both types of *sopela*). The polyphonic model is more suitable for real-world application, because *sopele* are always played in pair.

### 4.1. Estimating model architectures

We performed grid search for all 8 modelling instances independently (mono/poly: WAV/DFT+RF/CNN), by varying hyperparameter values as described in Section 3.3. We selected the models achieving the highest $F_1$ score over the validation subset as target model architectures to be additionally evaluated by subsequent tests. The best-performing-model hyperparameters are shown in Table 1. When comparing the top five performing model architectures within a specific RF setup (e.g., DFT+RF), we noticed that they do not differ significantly in the $F_1$ score from the top-performing one, $\Delta F_1 < 0.002$. This was expected, as bagging is not that susceptible to the choice of (reasonable) hyperparameter values, because vote averaging over hundreds of trees constrains the variance. On the other hand, CNN model setups were noticeably more sensitive to the choice of hyperparameters, having $\Delta F_1 < 0.05$. This was also expected, because CNN performance is highly susceptible to the choice of hyperparameters [11]. It should be noted, though, that because both the RF and the CNN estimation techniques are innately stochastic (depending on the outcome of a random event, e.g. feature subset selection, or minibatch dataset split), repeated experiments can, and probably would, lead to slightly different top-performing model architectures. The differences ($\Delta F_1$) were notably larger for the remaining contestants, on average, which was expected. The performance of the best model architectures, and their selected hyperparameters, is analysed next.

### 4.2. Model evaluation

Model-evaluation results over the test subset are presented in Table 2, for both setups. In the monophonic setup, all DFT-coupled models are highly accurate ($F_1 \approx 0.999\%$). This is in line with related work, stating that pitch detection for monophonic signals might be considered solved. On the other hand, with raw data being fed into the classifier, RF performs only slightly worse ($F_1 = 0.988$), whereas the CNN classifier under-performs significantly ($F_1 = 0.787$). Unlike monophonic models, dealing with only one line of melody (13 classes), polyphonic models try to predict individual or paired tones for both types of *sopela* (49 classes), which is more demanding. Using DFT feature extraction proves to be highly beneficial for polyphonic data ($F_1 \approx 0.92\%$), because feeding raw audio data into models results in their poor performance

($F_1 < 0.55$). In the polyphonic setup, DFT+RF performs slightly better than DFT+CNN ($\Delta F_1 \approx 0.01$).

Outstanding performance of the RF classifier when using raw waveform inputs of monophonic data, although unexpected, can be explained by its inherent variance-suppression ability, which successfully deals with the time-domain representation of the data. Selected model hyperparameters (Table 1) in all setups suggest that weakly regularised RF models perform better (shallower trees are preferred, whereas the splitting criteria, $n_{split}$ and $n_{leaf}$, are almost completely ignored). In the case of polyphonic WAV+RF, a slightly more regularised model is preferred ($n_{split} = 6$), probably due to larger variations in the data. Also, for all RF models, larger ensembles are preferred, which was expected. Surprisingly, all RF models benefited from using the entire training dataset for growing trees, instead of using sub-sampling (bootstrap).

Both top-performing CNN architectures in the monophonic setup have two convolutional layers and average-sized filter banks, suggesting that shallower and simpler models are preferred. This is understandable, as the complexity of these models was already large ($\approx 90 \cdot 10^4$ parameters for WAV+CNN; $\approx 45 \cdot 10^4$ parameters for all other CNN models), compared to the training subset size ($\approx 1.8 \cdot 10^4$ / $\approx 6.8 \cdot 10^4$). This relative disproportion of model complexity, when compared against the amount of available data, could also explain the inability of the WAV model to converge to a better local optimum. This holds for both the monophonic and the polyphonic setup. Furthermore, because convolutional filter sizes varied widely across the top-five-performing CNN models (grid-search), we can assume that their choice is less impactful on model performance, when compared to the choice of other hyperparameters. Also, the best performing DFT+CNN model uses one additional convolutional layer, which reduces the dimensionality of the input by half (compared to other CNN models), yet is compensated in complexity by the doubled number of neurons in the fully-connected layer. The evidence suggests that the convolutional filters are incapable of learning how to filter raw audio waveforms properly in this setup, probably due to the relatively small size of the available training dataset.

The discussion concerning the chosen hyperparameter values of top-performing models is concordant with the values of the top five performing models for each respective model type. The performance of the described top performers is additionally explained by the confusion matrices, which can be found in the Supplement.

### 4.3. Full-stack AMT system performance

Because the polyphonic context is of particular interest for real-world application, we put four best performing model architectures for multi-pitch detection (Table 1) under concluding test. We incorporated each model in the full-stack AMT system, with the main objective of producing the music sheet for a recording of a traditional *sopele* music piece called “Sadila je Mare” (roughly translated to “Mare has been planting”). We recorded the song in the process of data acquisition, by making use of the original music sheet, which is considered a ground truth (Fig. 3(a)). We composed a corresponding stereo source from two audio files recorded by both types of *sopele*. We limited the tempo of this piece to 80 beats per minute.

Visual comparison of the predicted sheets (Fig. 3(b)–(e)) against ground truth data gives a good insight into the AMT system efficiency and models’ performance. Musical scores generated by both CNN and RF models show to be reasonably accurate. No significant difference can be observed between the results of DFT-enabled models and models that use raw audio data only. Although model-evaluation results suggest that this difference should be more evident, performance deficiency of polyphonic WAV+RF and WAV+CNN models is barely visible (and in the rhythm, mostly), which is likely due to the postprocessing actions that rectified the misclassifications.

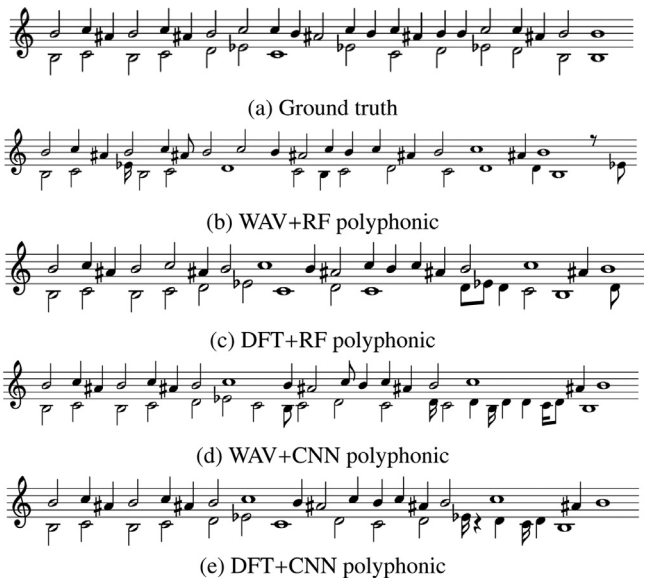


**Fig. 3.** A comparison of original and predicted sheets for polyphonic models. Two parallel note sequences represent *small* and *great sopela* on the same bar.

It can be seen that predicted tones of *great sopela* (bottom sequences) are, in all cases, more erroneous when compared to the *small sopela* predictions (top sequences). The reason behind this outcome can be found in the fact that the tone of the *small sopela* does not oscillate as much as the tone of its counterpart instrument. Furthermore, as already stated before, the pitch of the tone greatly depends on the player’s breathing capability and embouchure, which is more obvious for the *great sopela*. This implies that using a larger dataset for model estimation would be a reasonable step towards improving model accuracy and, consequently, the AMT system efficiency in general.

For the purpose of completeness, we also tested the monophonic models for transcribing a real-world music piece. In this case, the models are predicting music sheets for each *sopela* individually. The corresponding results are in the Supplement.

## 5. Client mobile application and back-end server

Traditional dances accompanied by *sopele* music are usually found on folklore gatherings. These events provide an opportunity for people to get together, compete and preserve cultural heritage. If a *sopele* player enjoyed a song played by another musician, he/she could have recorded it and then tried to mimic it at home. By using the transcription model provided in this paper, he/she could get the transcription sheet almost immediately after recording the song. That was the main motivation behind the development of an end-to-end AMT system, including an Android application (Fig. 4) and a back-end server that implements ML logic and associated pre/postprocessing actions. Using the provided mobile application, one can record an audio file and listen to it later, before deciding to export it into a music sheet via the available API.

Obtaining a music sheet consists of uploading the audio file to the server, and retrieving the transcription result. Audio splitting process does not involve creating multiple smaller audio files on the server filesystem; instead, everything is done using server memory only. The audio file is fragmented, and as soon as one piece is cut, that same piece undergoes identical preprocessing steps as described in Section 3.2. When preprocessing is finished, the model immediately predicts class labels for specific fragments.

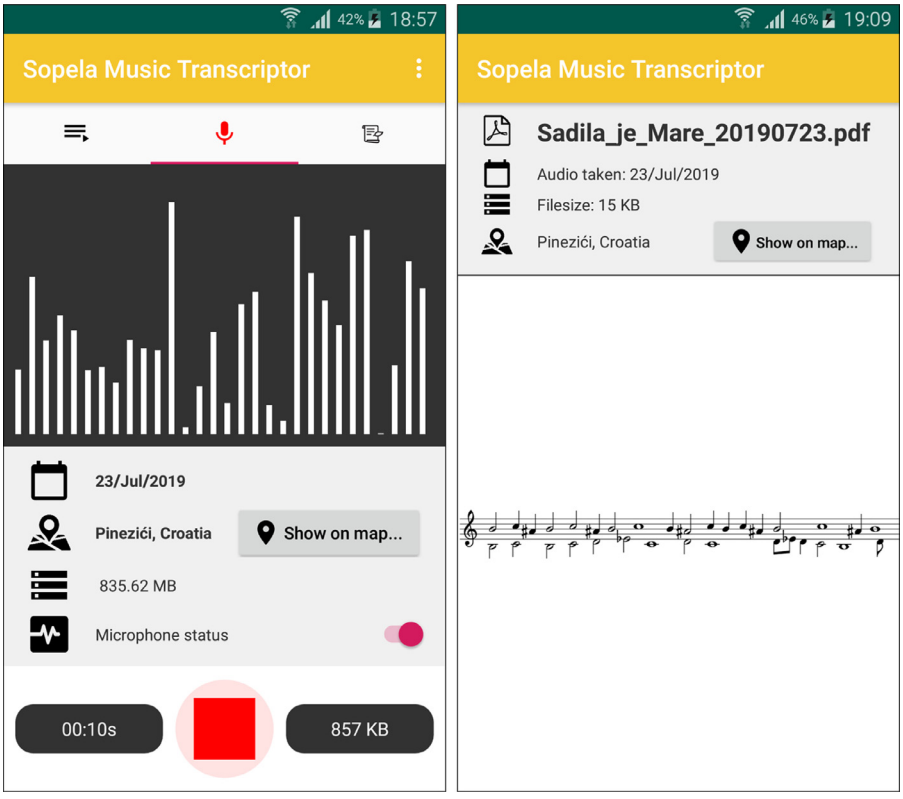


**Fig. 4.** Android application activities: audio recording with automatic geo-tagging (left), generated music score (right).

After all fragments have been classified, the results are stored in a HDF5 file which is used in conjunction with the postprocessing actions (Section 3.4) in order to create music sheets. The resulting sheet is sent to the client mobile application and is stored in the local application folder. Server response time is almost instantaneous. However, it should be noted that the recordings used here were no more than 40 s long. Following the obtained evaluation outcomes, we incorporated the DFT+RF polyphonic model in the final version of the back-end server.

## 6. Conclusion and future work

This paper provides a complete proof-of-concept AMT solution which targets a specific traditional woodwind instrument. The underlying motivation of the presented work is to provide support for collecting and cataloguing traditional *sopele* music, performed in real-life occasions in an effortless way, thus enabling the preservation of this globally acknowledged cultural heritage. One of the main issues in AMT is multi-pitch detection from a polyphonic input source. A number of instrument-specific AMT solutions have been provided, typically with piano being in the primary focus. On the other hand, there exists an evident lack of research efforts targeting AMT for woodwind instruments.

As *sopele* are always played in pair, we had to support polyphonic transcription. Multi-pitch classification for these woodwind instruments is highly susceptible to the ambiguous within-class and between-class variations in the sound they produce, caused by the instruments' unique build, as well as the performer's play style and air-blowing aptitude. To cope with this, we examined the feasibility of an AMT solution based on ML, using RF or CNN. In order to alleviate modelling complexity, we also considered the use of frequency-feature extractors for learning tone specifics from audio time-windows (DFT), as an optional preprocessing step. We evaluated the AMT solution under the monophonic setup (for detecting distinct *sopele* tones), and under the polyphonic setup (for detecting all possible tone combinations of a *sopele* pair), giving a total of eight models under consideration, i.e. RF/CNN with or without DFT under monophonic or polyphonic setup. Both RF and CNN models took very little time to train on a Xeon dual processor (32 logic cores) setup, having 3 GPUs (training time per model measured in minutes, on average).

Experimental evaluation over the given test dataset reveals a high-level pitch-detection performance of both RF and CNN under both setups. We have shown that the use of DFT is highly beneficial, especially in the polyphonic setup, where both RF and CNN under-perform significantly if it is not used. Both RF and CNN seem to be performing equally well under both setups when using DFT, although RF is negligibly better performing in the polyphonic setup. When it comes to AMT for a *sopele*-based real music piece, distinct model performance is concordant with the presented quantitative results, albeit, the use of DFT as a preprocessing step is to some extent cancelled-out by the postprocessing step, where smaller windows of (probably) misclassified tones are substituted with their more numerous neighbours. Furthermore, the predicted music score for *small sopela* track showed to be generally more accurate, with less misclassified tones, when compared against the *great sopela*.

This paper reports an insight study, focused only on the problem of AMT over one pair of physical instruments played by one performer. However, by gathering sufficient quantities of data, we plan to develop a more generally applicable model in the future. Furthermore, with considerably larger datasets available, additional ML techniques could come into consideration (e.g. RNN). Alongside the efforts concerning *sopele*-based music data collection, a rhythm estimation model should be built in order to further enhance audio-to-score transcription accuracy.

## Funding

This work has been supported in part by the University of Rijeka under the project number *uniri-tehnic-18-15* and project number *uniri-tehnic-18-17*.

## Declaration of Competing Interest

None

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2019.09.024.

## References

[1] L. Alcabasa, N. Marcos, Automatic guitar music transcription, in: 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), IEEE Computer Society, Los Alamitos, CA, USA, 2012, pp. 197–202.
[2] V. Arora, L. Behera, Multiple f0 estimation and source clustering of polyphonic music audio using plca and hmrfs, IEEE/ACM Trans. Audio Speech Lang. Proc. 23 (2) (2015) 278–287.
[3] A.M. Barbancho, A. Klapuri, L.J. Tardon, I. Barbancho, Automatic transcription of guitar chords and fingering from audio, Trans. Audio Speech Lang. Proc. 20 (3) (2012) 915–921.
[4] E. Benetos, S. Dixon, A shift-invariant latent variable model for automatic music transcription, Comput. Music J. 36 (4) (2012) 81–94.
[5] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, A. Klapuri, Automatic music transcription: challenges and future directions, J. Intell. Inf. Syst. 41 (3) (2013) 407–434.
[6] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[7] A. Cogliati, Z. Duan, B. Wohlberg, Context-dependent piano music transcription with convolutional sparse coding, IEEE/ACM Trans. Audio Speech Lang. Proc. 24 (12) (2016) 2218–2230.
[8] A. Dessein, A. Cont, G. Lemaitre, Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence, in: Proceedings of the 11th International Society for Music Information Retrieval Conference, Utrecht, The Netherlands, 2010, pp. 489–494.
[9] W. van Drongelen, Continuous, Discrete, and Fast Fourier Transform, in: Signal Processing for Neuroscientists, Elsevier, 2018, pp. 103–118.
[10] N. Gajhede, O. Beck, H. Purwins, Convolutional neural networks with batch normalization for classifying hi-hat, snare, and bass percussion sound samples, in: Proceedings of the Audio Mostly 2016, in: AM '16, ACM, New York, NY, USA, 2016, pp. 111–115.
[11] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT press, 2016.
[12] A. Klapuri, M. Davy, Signal Processing Methods for Music Transcription, 1st, Springer Publishing Company, Incorporated, 2010.
[13] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Advances In Neural Information Processing Systems 25 (2012) 1097–1105.
[14] E. Kubera, M.B. Kursa, W.R. Rudnicki, R. Rudnicki, A.A. Wieczorkowska, All that jazz in the random forest, in: M. Kryszkiewicz, H. Rybinski, A. Skowron, Z.W. Raś (Eds.), Foundations of Intelligent Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 543–553.
[15] E. Nakamura, E. Benetos, K. Yoshii, S. Dixon, Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 101–105.
[16] J. Nam, J. Ngiam, H. Lee, M. Slaney, A classification-based polyphonic piano transcription approach using learned feature representations, in: Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24–28, 2011, 2011, pp. 175–180.
[17] K. O'Hanlon, H. Nagano, N. Keriven, M.D. Plumbley, Non-negative group sparsity with subspace note modelling for polyphonic transcription, IEEE/ACM Trans. Audio Speech Lang. Proc. 24 (3) (2016) 530–542.
[18] D.M. Powers, Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation, J. Mach. Learn. Technol. 2 (1) (2011) 37–63.
[19] R. Schramm, E. Benetos, Automatic transcription of a cappella recordings from multiple singers, in: Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio, 2017, pp. 1–8.
[20] C. Senac, T. Pellegrini, F. Mouret, J. Pinquier, Music feature maps with convolutional neural networks for music genre classification, in: Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, in: CBMI '17, ACM, New York, NY, USA, 2017, pp. 19:1–19:5.
[21] S. Sigtia, E. Benetos, S. Dixon, An end-to-end neural network for polyphonic piano music transcription, IEEE/ACM Trans. Audio Speech Lang. Proc. 24 (5) (2016) 927–939.
[22] P. Smaragdis, J. Brown, Non-negative matrix factorization for polyphonic music transcription, in: Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2003, pp. 177–180.
[23] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manage. 45 (4) (2009) 427–437.
[24] L. Tan, J. Jiang, Discrete Fourier Transform and Signal Spectrum, in: Digital Signal Processing, Elsevier, 2019, pp. 91–142.
[25] J.J. Valero-Mas, E. Benetos, J.M. Iñesta, A supervised classification approach for note tracking in polyphonic piano transcription, J. New Music Res. 47 (3) (2018) 249–263.
[26] E. Vincent, N. Bertin, R. Badeau, Adaptive harmonic spectral decomposition for multiple pitch estimation, Trans. Audio Speech Lang. Proc. 18 (3) (2010) 528–537.
[27] C.-W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Muller, A. Lerch, A review of automatic drum transcription, IEEE/ACM Trans. Audio Speech Lang. Proc. 26 (9) (2018) 1457–1483.