Markus Schwabe* and Michael Heizmann

# Influence of input data representations for time-dependent instrument recognition

## Einfluss von Eingangsdaten-Darstellungen für die zeitabhängige Instrumentenerkennung

**Abstract:** An important preprocessing step for several music signal processing algorithms is the estimation of playing instruments in music recordings. To this aim, time-dependent instrument recognition is realized by a neural network with residual blocks in this approach. Since music signal processing tasks use diverse time-frequency representations as input matrices, the influence of different input representations for instrument recognition is analyzed in this work. Three-dimensional inputs of short-time Fourier transform (STFT) magnitudes and an additional time-frequency representation based on phase information are investigated as well as two-dimensional STFT or constant-Q transform (CQT) magnitudes. As additional phase representations, the product spectrum (PS), based on the modified group delay, and the frequency error (FE) matrix, related to the instantaneous frequency, are used. Training and evaluation processes are executed based on the MusicNet dataset, which enables the estimation of seven instruments. With a higher number of frequency bins in the input representations, an improved instrument recognition of about 2 % in F1-score can be achieved. Compared to the literature, frame-level instrument recognition can be improved for different input representations.

**Keywords:** Instrument recognition, polyphonic music signals, time-frequency representations, neural networks.

**Zusammenfassung:** Ein wichtiger Vorverarbeitungsschritt für verschiedene Musiksignalverarbeitungsalgorithmen ist die Schätzung der spielenden Instrumente in Musikaufnahmen. Zu diesem Zweck wird die zeitabhängige Instrumentenerkennung in diesem Ansatz durch ein neuronales Netz mit Residual-Blöcken realisiert. Da Musiksignalverarbeitungsaufgaben unterschiedliche Zeit-Frequenz-Darstellungen als Eingabematrizen verwenden, wird in dieser Arbeit der Einfluss verschiedener Eingangsdarstellungen für die Instrumentenerkennung analysiert. Dabei werden sowohl dreidimensionale Eingänge von Kurzzeit-Fourier-Transformation (STFT) mit einer zusätzlichen auf Phaseninformation basierenden Zeit-Frequenz-Darstellung als auch die Magnituden der zweidimensionalen STFT oder der Constant-Q-Transformation (CQT) untersucht. Als zusätzliche Phasendarstellungen werden das Produktspektrum (PS), das auf der modifizierten Gruppenlaufzeit basiert, und die Frequenzfehlermatrix (FE-Matrix), welche von der Momentanfrequenz abgeleitet ist, verwendet. Die Trainings- und Evaluierungsprozesse werden auf Basis des MusicNet-Datensatzes durchgeführt, der die Schätzung von sieben Instrumenten ermöglicht. Durch eine höhere Anzahl an Frequenzbins in den Eingangsdarstellungen kann eine um etwa 2 % im F1-Score verbesserte Instrumentenerkennung erreicht werden. Im Vergleich zur Literatur kann die Instrumentenerkennung auf Frame-Ebene für verschiedene Eingangsdarstellungen verbessert werden.

**Schlagwörter:** Instrumentenerkennung, polyphone Musiksignale, Zeit-Frequenz-Darstellungen, neuronale Netze.

## 1 Introduction

Instrument recognition is highly relevant for different music information retrieval (MIR) tasks. The playing instruments in polyphonic music signals are important features that are used for example as a part of the audio tags in automatic tagging [4]. Additionally, the information about playing instruments can facilitate the detection of other instrument-dependent features and tags like genre or mood. Furthermore, audio source separation and automatic music transcription, as for example in [13], can be improved by instrument recognition in preprocessing, because the separation or transcription algorithm shrinks to a tailored estimation for a much smaller amount of known instruments.

Most of instrument recognition algorithms have focused on clip-wise recognition, which means that the playing instruments were estimated for the whole music ex-

***Corresponding author: Markus Schwabe,** Institute of Industrial Information Technology (IIIT), Karlsruhe Institute of Technology (KIT), Hertzstraße 16, 76187 Karlsruhe, Germany, e-mail: markus.schwabe@kit.edu
**Michael Heizmann,** Institute of Industrial Information Technology (IIIT), Karlsruhe Institute of Technology (KIT), Hertzstraße 16, 76187 Karlsruhe, Germany

cerpt fed to the algorithm. Han et al. [6] developed a deep convolutional neural network (CNN) for instrument recognition based on mel-spectrogram inputs and aggregation of multiple outputs from sliding windows over the audio data. Pons et al. [12] analyzed the architecture of CNNs in order to formulate an efficient design strategy to capture the relevant information about timbre. Both approaches were trained and validated by the IRMAS dataset [2], which contains polyphonic music excerpts.

Beside the exclusive consideration of absolute values of the input audio data, as it is done through the transformation of mel-spectrograms, there are several possibilities to incorporate phase information. Diment et al. [5] used for example the modified group delay (MODGD) feature, which includes phase information calculated from the Fourier transform, and trained Gaussian mixture models with MFCCs and MODGD features for instrument recognition. Sebastian and Murthy [15] trained a recurrent neural network for music source separation with a phase representation of music signals derived from the MODGD features. Furthermore, phase information can be incorporated implicitly by using raw music signals as input data. Li et al. [10] built such an end-to-end learning approach based on CNN. This network, like the CNN for automatic tagging of Dieleman and Schrauwen [4], needs only very little domain knowledge, but performs slightly lower compared to approaches with preprocessed input data such as spectrograms [6].

Especially for improving audio source separation by preprocessed instrument recognition, the clip-wise recognition is not sufficient. Thus, frame-level instrument recognition was developed by Hung and Yang [8]. They used the absolute values of the constant-Q transform (CQT) and separately estimated pitch information of the music signal as input for their deep neural network. Another approach based on the short-time Fourier transform (STFT) of the input music signal was presented in [14], where additional time-dependent phase information was included in the input representation by the product spectrum (PS), a combination of STFT absolute values and group delay function results. A combined estimation of the playing instruments and notes for each frame was presented by Hung et al. [9], in which the proposed model is forced to estimate the interaction between timbre and pitch.

As there are different time-frequency representations used in literature approaches, the influence of those input data representations on the instrument recognition performance is investigated in this work. Representations with and without additional phase information are analyzed, because timbre details are included in the phase information of music signals. Instrument recognition for each time frame is too fine for the temporal resolution of the human ear, therefore time-dependent recognition with a resolution of about 100 ms is sufficient. First, the relevant time-frequency representations are defined in Section 2 and the proposed model structure for instrument recognition is explained in Section 3. The experiments for the evaluation of the analyzed model configurations and the different input data representations are described in Section 4. In Section 5, the results are summarized.

## 2 Time-frequency representations

A very common preprocessing step in MIR algorithms is the calculation of the short-time Fourier transform (STFT)

$$X[m,k] = \sum_{n=0}^{N-1} x[n]\, \gamma^*_{mk}[n] = |X[m,k]|\, \mathrm{e}^{j\theta[m,k]} \tag{1}$$

of the discrete input music signal $x[n]$ and the time and frequency shifted window $\gamma^*_{mk}[n]$ of length $N$. Many algorithms only use the absolute values $|X[m,k]|$ for their task and therefore neglect the phase $\theta[m,k]$. Since the values of $\theta[m,k]$ aren't limited to $[0, 2\pi]$, phase values have to be unwrapped for a meaningful interpretation.

The resulting phase ambiguity caused by unwrapping can be avoided by calculating the discrete realization of the continuous group delay function [1]

$$\tau_\mathrm{g}(\omega) = -\frac{d}{d\omega}\,\theta(\omega) = -\Im\left(\frac{d}{d\omega}\log\left(X(\omega)\right)\right), \tag{2}$$

which is defined as [5]

$$\tau_\mathrm{g}[m,k] = \frac{X_\mathrm{R}[m,k]\, Y_\mathrm{R}[m,k] + X_\mathrm{I}[m,k]\, Y_\mathrm{I}[m,k]}{|X[m,k]|^2}. \tag{3}$$

Thereby, $Y[m,k]$ is the STFT of the signal $y[n] = n \cdot x[n]$ and the indices R and I stand for the real and the imaginary part, respectively. A combination of magnitude and phase information is realized by the product of squared absolute and group delay function values [17]

$$P[m,k] = |X[m,k]|^2 \cdot \tau_\mathrm{g}[m,k] \tag{4}$$

$$= X_\mathrm{R}[m,k]\, Y_\mathrm{R}[m,k] + X_\mathrm{I}[m,k]\, Y_\mathrm{I}[m,k]. \tag{5}$$

The resulting phase-dependent time-frequency representation $P[m,k]$ is called product spectrum (PS) here.

Another possibility for the incorporation of phase information is based on the estimation of instantaneous frequency (IF) [11], which uses phase differences in order to refine the STFT frequency bin values. Since a time-frequency representation with the IF values as frequency

bins is not realizable, the additional phase information is incorporated by a time-frequency representation of the frequency errors (FE) for each bin. This error matrix

$$E[m,k] = \frac{\theta[m,k] - (\theta[m-1,k] + \omega[k] \cdot \Delta t)}{2\pi \cdot \Delta t} \tag{6}$$

is calculated based on the angular frequency $\omega[k]$ of the STFT frequency bin $k$ and the STFT time resolution $\Delta t$. Thereby, the numerator values are considered to be in the interval $[-\pi, \pi]$ although there are nonsolvable ambiguities for high frequencies and large $\Delta t$.

Beside the STFT, a common time-frequency representation for music signals is the constant-Q transform (CQT) [3]. Its discrete representation is defined as

$$X_{\mathrm{CQT}}[m,k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} x[n]\, w_{mk}[n]\, \mathrm{e}^{-j2\pi \frac{Qn}{N[k]}} \tag{7}$$

with a window function $w_{mk}[n]$ and a frequency-dependent window length $N[k]$. This frequency dependency of $N[k]$ ensures a constant resolution factor

$$Q = \frac{f[k]}{f[k+1] - f[k]} = \frac{1}{2^{\frac{1}{b}} - 1}, \tag{8}$$

where $f[k]$ is the frequency value at step k and $b$ defines the number of frequency bins per octave. Consequently, the time-frequency representation calculated by CQT has a logarithmic frequency axis.

# 3 Proposed model

Active instruments in the input music signal are estimated by convolutional neural networks with residual blocks in this work. The model and the residual block structure is described in detail in Section 3.2. In order to analyze input representations with and without phase information, the preprocessing step, which is explained in Section 3.1, calculates all relevant time-frequency representations. For the training of the network, the utilized dataset and the label generation are described in Section 3.3.

## 3.1 Preprocessing

A time-frequency representation of magnitudes is a common input format for time-dependent instrument detection. The magnitude representations investigated in this work are the absolute values of the STFT (Equation (1)) and the CQT (Equation (7)) of the analyzed music signal. In order to improve identification of instruments in polyphonic music at frame level, time-frequency representations based on phase information are calculated and concatenated with the magnitude representation in the channel dimension to cover the correlation between them in time and frequency. This additional information has led to better results for some representations [14]. As phase information representations, the product spectrum of Equation (4), or the frequency error matrix calculated by Equation (6) are utilized in this work.

First, magnitude and phase representations are calculated for the raw audio input signal. Then the magnitude and PS representations are normalized to their maximum amplitude and the values are converted into a logarithmic scale according to

$$X[m,k]_{\mathrm{dB}} = 20 \cdot \log_{10}\left(|X[m,k]| + \epsilon\right) \tag{9}$$

with $\epsilon = 10^{-10}$. This allows the consideration of high dynamics and provides a differentiated representation of the harmonics. In addition, the logarithmic representation corresponds better to human perception.

Because of memory restrictions during training and operation of the neural network, all representations are divided into segments. In this work, segments of 92.88 ms have been chosen, which represent 4096 time samples at the sampling rate of 44.1 kHz. For the STFT calculation, window lengths of 1024 and 4096 samples are investigated, which represent 23.22 ms and 92.88 ms. During that time period, the input music signal is assumed to be stationary, respectively. The parameter $b$ of the CQT calculation is chosen as 12 and 48, which means 1 or 4 bins per note and results in 88 or 400 CQT bins. Thereby, the minimum frequency for the CQT calculation is chosen as 27.5 Hz, which represents the frequency of note A0. Furthermore, different overlaps of 512 and 2048 samples between successive windows are analyzed in the STFT and the CQT calculation. Consequently, the considered input representations are 3-dimensional data of shape $\mathbb{R}^{i_b \times i_t \times i_p}$ with $i_b \in \{88, 400, 513, 2049\}$ frequency bins, $i_t \in \{60, 240\}$ time bins and $i_p \in \{1, 2\}$ channels.

## 3.2 Model architecture

All considered instruments are estimated as active or silent in the respective input segment by a neural network. Its architecture is presented in Figure 1. After one 2D convolution in the first layer, the network consists of four structure blocks, which consist of one 1D convolution, an increasing number of residual blocks, and a max pooling layer

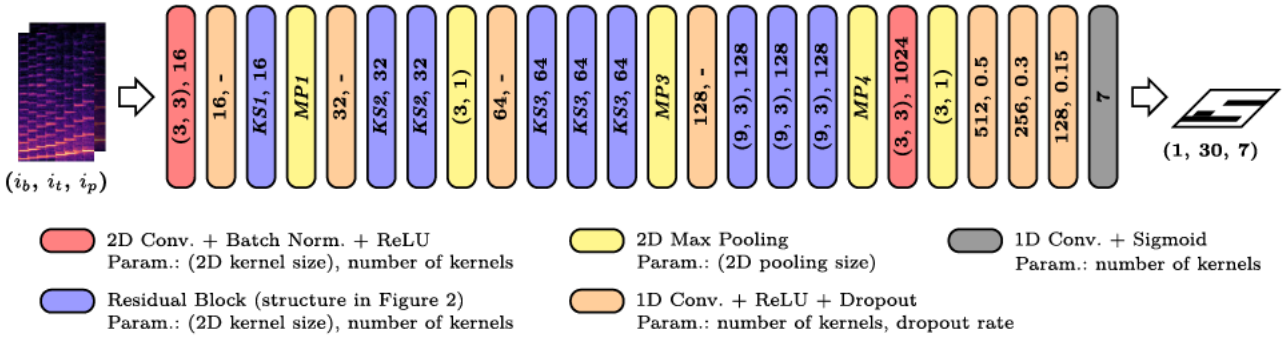


**Figure 1:** Schematic model structure with layer parameters. Variable parameters (in *italic*) are listed in Table 1.

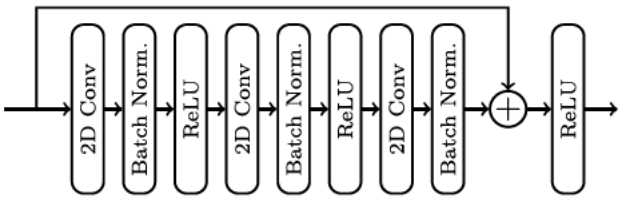


**Figure 2:** Schematic model structure of a residual block.

at the end. Thereby, the blue residual blocks in Figure 1 are based on the idea of residual building blocks for deep neural networks [7] and their structure is presented in Figure 2. The increasing number of residual blocks with increasing kernel size in frequency dimension is realized in order to enable complex features and comprise a wider frequency interval in deeper layers. Depending on the basic time-frequency transform (STFT or CQT), the parameters for the kernel sizes are adjusted with that intention. Additionally, the parameters for several max pooling layers have to be adapted according to the different input shapes to ensure that the output shape $(1, 30)$ for frequency and time is met. This leads to the variable parameters in Table 1.

**Table 1:** Parameter configurations for the investigated model of Figure 1.

| $I_b$ | $I_t$ | KS1 | KS2 | KS3 | MP1 | MP3 | MP4 |
|---|---|---|---|---|---|---|---|
| 88 | 60 | (3,3) | (5,3) | (7,3) | (2,1) | (2,1) | (2,2) |
| 88 | 240 | (3,3) | (5,3) | (7,3) | (2,2) | (2,2) | (2,2) |
| 400 | 60 | (3,3) | (5,3) | (7,3) | (2,1) | (4,1) | (4,2) |
| 400 | 240 | (3,3) | (5,3) | (7,3) | (2,2) | (4,2) | (4,2) |
| 513 | 60 | (5,1) | (5,1) | (7,1) | (3,1) | (3,1) | (5,2) |
| 513 | 240 | (5,1) | (5,1) | (7,1) | (3,2) | (3,2) | (5,2) |
| 2049 | 60 | (5,1) | (5,1) | (7,1) | (6,1) | (5,1) | (5,2) |
| 2049 | 240 | (5,1) | (5,1) | (7,1) | (6,2) | (5,2) | (5,2) |

Each 2D convolutional layer in the network and also in the residual block is followed by a batch normalization for regularization and a rectified linear unit (ReLU) as activation function. For each structure block, the number of kernels is increasing from 16 to 128 in order to extract a large number of features with high degree of abstraction. The last 2D convolution layer, colored in red in Figure 1, extracts with 1024 the largest number of features and represents the transition between the residual and the fully connected layer (FCL) part of the network. After the fifth max pooling layer, there are 1024 feature maps of shape $(1, 30)$, which realizes the desired time resolution for the instrument estimation output frames.

The last four layers of the neural network can be interpreted as FCLs that are implemented by $1 \times 1$ convolutions. Consequently, the number of kernels is interpreted as the number of nodes in an FCL. Dropout with the given percentage is used for those FCLs to improve regularization. The 7 output nodes, representing the 7 output instruments considered in the dataset (Section 3.3), are achieved by decreasing number of kernels from 512 to 7 across the FCLs. All 1D convolutional layers, colored in orange in Figure 1, are followed by a ReLU activation function, except for the output layer, colored in gray, whose activation function is the sigmoid function.

## 3.3 Dataset and label generation

For training and evaluating the developed model, the MusicNet dataset [16] with 34 hours of 330 freely-licensed chamber music recordings is utilised. Thereby, the predefined training partition is split in a validation set of 34 and a training set of 286 recordings. Although there are 11 instruments in the entire dataset, only 7 instruments are active in the 10 test recordings. Consequently, only these 7 instruments piano, violin, viola, cello, clarinet, bassoon, and horn are estimated and evaluated in this work. Sounds

of the other 4 instruments oboe, flute, harpsichord, and string bass are not removed from training and validation dataset, but have not been labeled. They are assumed as unwanted additional signals during training.

In order to achieve a time-dependent instrument recognition, the resolution of output time frames is chosen to be 4096 time samples, which is about 92.88 ms at a sampling rate of 44.1 kHz. The model estimates the presence of instruments for 30 time frames, therefore each input segment represents a signal duration of about 2.79 s. The corresponding label matrix of an input segment, which represents the ground truth for the instrument recognition, is generated as a Boolean matrix of shape $(7, 30)$. If an instrument has been played at any time during the particular 4096 time samples of a time frame, it is assumed as active and labelled with '1' in the respective row for that instrument and the column for this time frame.

# 4 Experiments

The model described in Section 3 is trained and evaluated with the MusicNet dataset, whose labels are built according to Section 3.3. Keras with Tensorflow has been used for the model's implementation and its application during the experiments. Further details about the implementation, especially for estimation and evaluation, are described in Section 4.1. In Section 4.2, the results of the experiments are presented and discussed.

## 4.1 Implementation details

In order to analyze the impact of different input data representations, a total of 13 models are trained for 50 epochs each in this work. Their structure parameters are adapted to the different input representations, as described in Section 3.2. The input time-frequency representations are based on either STFT or CQT and include the magnitude values of the respective basis transform. Overall, the four cases STFT magnitudes, STFT magnitudes with PS representation, STFT magnitudes with FE matrix, and CQT magnitudes are investigated with different shapes, like explained in Section 3.1. All models are trained using stochastic gradient descend (SGD) with momentum 0.9 as the optimization algorithm. Binary cross entropy (BCE) is used as the cost function, because it is suited for binary instrument activation. For the first 5 epochs, an initial learning rate of $l = 0.1$ is defined. After that, a scheduler decreases the learning rate with

$$l = 0.1 \cdot 0.5^{\left\lfloor \frac{n_{\text{epoch}}+1}{5} \right\rfloor} \tag{10}$$

according to the epoch number $n_{\text{epoch}}$.

Due to the sigmoid activation function in the output layer, all estimations for active instruments are continuous values in the range $[0, 1]$, which represent probabilities for their presence at the respective time frames. Since we consider an instrument either active or not in a defined time frame, the output is binarized with the threshold $b = 0.5$. This threshold avoids extreme binarization sensitivities for specific instruments, which is more desired than a small performance increase by choosing best thresholds [14].

Instrument recognition results are evaluated based on the MusicNet test dataset and the F1-score. This metric is the harmonic mean of the metrics precision and recall, which are ratios of the number of true positive estimations to all positive estimations (precision) or all positive labels (recall). The F1-scores are calculated independently for each instrument, but combined for all considered test recordings. An average F1-score is calculated over all instrument results to get a simple performance metric.

## 4.2 Results

After a successful training of 50 epochs, the performance of the different models is compared based on the F1-scores for the MusicNet test dataset. They are calculated as described in Section 4.1 for each instrument and an average value over all considered instruments. The resulting values for all 13 models are given in Table 2.

As presented in Table 2, the incorporation of additional phase information representations like PS or FE doesn't lead to an improved instrument recognition in this case, but to a comparable result to the recognition with only the absolute STFT values. Consequently, those additional informations are unnecessary for the instrument recognition with the investigated model structure, because they lead to a higher amount of input data and calculations with no performance enhancement.

In general, instrument recognition is performed best for piano, violin, and cello. One reason is that the MusicNet test data contain solo recordings of those three instruments, but only recordings of trios for the rest of the considered instruments. Since instrument recognition is much easier for solo recordings, the three solo instruments show the best F1-scores here.

Furthermore, an improved instrument recognition can be determined for a larger time-frequency input representation, regardless of the representation type. The results

**Table 2:** F1-scores for all investigated models with different input representations based on the MusicNet test dataset. The best average result for each method is highlighted.

| Method | Input Shape | Piano | Violin | Viola | Cello | Clarinet | Bassoon | Horn | Average |
|---|---|---|---|---|---|---|---|---|---|
| STFT | (513, 240, 1) | 0.9804 | 0.9482 | 0.8111 | 0.9039 | 0.8775 | 0.8278 | 0.7553 | 0.8720 |
| | (2049, 60, 1) | 0.9793 | 0.9455 | 0.7992 | 0.9062 | 0.9097 | 0.8389 | 0.7926 | 0.8816 |
| | (2049, 240, 1) | 0.9778 | 0.9490 | 0.8192 | 0.9136 | 0.8926 | 0.8349 | 0.8203 | **0.8868** |
| STFT + PS | (513, 240, 2) | 0.9753 | 0.9513 | 0.8250 | 0.9115 | 0.8740 | 0.8304 | 0.7624 | 0.8757 |
| | (2049, 60, 2) | 0.9771 | 0.9464 | 0.8082 | 0.9121 | 0.8934 | 0.8400 | 0.7982 | 0.8822 |
| | (2049, 240, 2) | 0.9806 | 0.9479 | 0.8144 | 0.9114 | 0.8938 | 0.8328 | 0.8206 | **0.8859** |
| STFT + FE | (513, 240, 2) | 0.9823 | 0.9464 | 0.8138 | 0.9062 | 0.8900 | 0.8265 | 0.7759 | 0.8773 |
| | (2049, 60, 2) | 0.9779 | 0.9427 | 0.8019 | 0.9130 | 0.8965 | 0.8316 | 0.7897 | 0.8790 |
| | (2049, 240, 2) | 0.9797 | 0.9470 | 0.8096 | 0.9112 | 0.9090 | 0.8442 | 0.8014 | **0.8860** |
| CQT | (88, 60, 1) | 0.9773 | 0.9370 | 0.7938 | 0.9051 | 0.8824 | 0.8252 | 0.7771 | 0.8711 |
| | (88, 240, 1) | 0.9762 | 0.9353 | 0.8109 | 0.9054 | 0.8837 | 0.8249 | 0.7581 | 0.8706 |
| | (400, 60, 1) | 0.9801 | 0.9471 | 0.8157 | 0.9159 | 0.8828 | 0.8474 | 0.7812 | **0.8815** |
| | (400, 240, 1) | 0.9809 | 0.9440 | 0.8099 | 0.9162 | 0.8862 | 0.8426 | 0.7845 | 0.8806 |

for 400 CQT frequency bins are about 1 % better than those with 88 frequency bins. In case of the STFT representations, the improvement is about 1 %–1.5 % between 2049 and 513 frequency bins. That can be explained by the finer frequency resolution, which is automatically achieved by more frequency bins and leads to finer instrument-specific spectra that can be learned by the neural network. Beside the average values, this effect can also be recognized at the improved results for cello, clarinet, bassoon, and horn.

An additional improvement of about 0.5 % in the average F1-score for STFT-based input representation can be achieved by a smaller hop size, which leads to a higher number of time frames in the time-frequency representation. The enlargement from 60 to 240 time bins is realized by a hop size decrease from 2048 to 512 samples, which represents $\frac{1}{8}$ of the window length. Especially horn detection results can be increased about 1 %–2.5 % by that enlargement. In contrast, CQT results can't be improved by more input time frames. As horn notes are predominantly in a low frequency range, where the STFT resolution is not very fine, additional frames could help the neural network to recognize a playing horn in the low frequency range.

In order to evaluate the frame-level instrument recognition of this work, results for the approach of Hung and Yang [8], which is the best approach for frame-level instrument recognition in literature so far, and an approach of previous work [14] are compared to our best residual models without additional phase information in Table 3. F1-score, precision, and recall values are taken from the comparison in [14] for both literature approaches. Hung and Yang use the CQT of the analyzed music (shape $(88, 258)$) and additionally harmonic series features (HSF) for pitch estimation as input information, whereas all other approaches do not need any pitch estimation. The shape of the input representation of the convolutional network from previous work is $(513, 259, 2)$.

According to the F1-scores in Table 3, our model with residual block structure outperforms the literature approaches for both input representations STFT and CQT, although the thresholds of the literature models have been chosen instrument-dependently. The different thresholds of HSF-5 in the range $[0.01, 0.99]$ are one reason for the highest precisions of that approach, but also for the lower recalls compared to the other models. Since higher recalls ensure a larger coverage of positive labels, our residual approach realizes an increased detection of active instruments. Besides the best frame-level instrument recognition results due to F1-score, that is a further advantage for subsequent signal processing, because the instrument detection should include most of the occurring instruments in the analyzed music recording. In general, instrument recognition results are influenced by the choice of an appropriate model structure as well as by input data representations.

# 5 Conclusion

Based on a neural network with residual blocks, the influence of different time-frequency representations of magnitude and phase information has been investigated. Larger representations with a higher number of time and frequency bins led to an increased performance for both STFT and CQT based calculations. Additional phase representations like the product spectrum or the frequency error ma-

**Table 3:** Evaluation metrics for instrument recognition models of the literature compared to the best approaches of this work. The best average metric values are highlighted.

| Method | Metric | Piano | Violin | Viola | Cello | Clarinet | Bassoon | Horn | Average |
|---|---|---|---|---|---|---|---|---|---|
| Res. network [8] | Precision | 0.9777 | 0.9383 | 0.7678 | 0.9175 | 0.8801 | 0.7931 | 0.7061 | **0.8544** |
| CQT + HSF-5 | Recall | 0.9904 | 0.9679 | 0.8953 | 0.9069 | 0.9237 | 0.8544 | 0.8188 | 0.9082 |
| | F1-score | 0.9840 | 0.9529 | 0.8267 | 0.9122 | 0.9014 | 0.8226 | 0.7583 | 0.8797 |
| Conv. network [14] | Precision | 0.9700 | 0.9298 | 0.7395 | 0.8628 | 0.7961 | 0.6798 | 0.6616 | 0.8057 |
| STFT + PS | Recall | 0.9949 | 0.9788 | 0.9583 | 0.9643 | 0.9711 | 0.9332 | 0.6977 | 0.9283 |
| best thresholds | F1-score | 0.9823 | 0.9537 | 0.8348 | 0.9108 | 0.8750 | 0.7866 | 0.6792 | 0.8603 |
| STFT | Precision | 0.9750 | 0.9428 | 0.7284 | 0.9032 | 0.8365 | 0.7372 | 0.7558 | 0.8398 |
| (2049, 240, 1) | Recall | 0.9805 | 0.9553 | 0.9359 | 0.9243 | 0.9568 | 0.9626 | 0.8969 | **0.9446** |
| | F1-score | 0.9778 | 0.9490 | 0.8192 | 0.9136 | 0.8926 | 0.8349 | 0.8203 | **0.8868** |
| CQT | Precision | 0.9743 | 0.9341 | 0.7389 | 0.8888 | 0.8403 | 0.7824 | 0.7041 | 0.8376 |
| (400, 60, 1) | Recall | 0.9859 | 0.9605 | 0.9104 | 0.9447 | 0.9298 | 0.9242 | 0.8773 | 0.9333 |
| | F1-score | 0.9801 | 0.9471 | 0.8157 | 0.9159 | 0.8828 | 0.8474 | 0.7812 | 0.8815 |

trix could not further improve the instrument recognition results of STFT magnitude inputs. The described residual model outperforms other frame-level approaches in the literature for several input representations.

In future works, the algorithm has to be tested with larger datasets and more instruments to improve applicability in music signal processing and specific MIR tasks.

# References


1. H. Banno, J. Lu, S. Nakamura, K. Shikano, and H. Kawahara. Efficient representation of short-time phase based on group delay. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 2*, pages 841–864, 1998.
2. J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In *13th International Society for Music Information Retrieval (ISMIR) Conf.*, pages 559–564, 2012.
3. J. C. Brown. Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–434, Jan. 1991.
4. S. Dieleman and B. Schrauwen. End-to-end learning for music audio. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6964–6968, 2014.
5. A. Diment, P. Rajan, T. Heittola, and T. Virtanen. Modified group delay feature for musical instrument recognition. In *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research*, pages 431–438, 2013.
6. Y. Han, J. Kim, and K. Lee. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):208–221, 2017.
7. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
8. Y.-N. Hung and Y.-H. Yang. Frame-level instrument recognition by timbre and pitch. In *19th International Society for Music Information Retrieval (ISMIR) Conference*, pages 135–142, 2018.
9. Y.-N. Hung, Y.-A. Chen, and Y.-H. Yang. Multitask learning for frame-level instrument recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 381–385, 2019.
10. P. Li, J. Qian, and T. Wang. Automatic instrument recognition in polyphonic music using convolutional neural networks. *arXiv preprint arXiv:1511.05520*, 2015.
11. M. Müller. *Fundamentals of Music Processing – Audio, Analysis, Algorithms, Applications*. Springer International Publishing Switzerland, 2015. ISBN 978-3-319-21944-8.
12. J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra. Timbre analysis of music audio signals with convolutional neural networks. In *25th European Signal Processing Conf.*, pages 2744–2748, 2017.
13. M. Schwabe, M. Weber, and F. Puente León. Notenseparation in polyphonen Musiksignalen durch einen Matching-Pursuit-Algorithmus. *tm - Technisches Messen*, 85(s1):s103–s109, 2018.
14. M. Schwabe, O. Elaiashy, and F. Puente León. Incorporation of phase information for improved time-dependent instrument recognition. *tm - Technisches Messen*, 87(s1):s62–s67, 2020.
15. J. Sebastian and H. A. Murthy. Group delay based music source separation using deep recurrent neural networks. In *International Conference on Signal Processing and Communications*, pages 1–5, 2016.
16. J. Thickstun, Z. Harchaoui, and S. M. Kakade. Learning features of music from scratch. In *International Conference on Learning Representations*, 2017.
17. D. Zhu and K. K. Paliwal. Product of power spectrum and group delay function for speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 1*, pages I125–I128, 2004.