Circuits, Systems, and Signal Processing (2024) 43:4239–4271
https://doi.org/10.1007/s00034-024-02641-1

Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music Using Discrete Wavelet Transform

Sukanta Kumar Dash[1] · **S. S. Solanki**[1] · **Soubhik Chakraborty**[2]

Received: 4 June 2023 / Accepted: 19 January 2024 / Published online: 19 March 2024

Abstract

In this article, a new multi-input deep convolutional neural networks (deep-CNNs) model architecture is addressed for the recognition of predominant instruments in polyphonic music using discrete wavelet transform (DWT). The proposed deep-CNNs model employs a fusion of Mel-spectrogram and Mel-frequency cepstral coefficient (MFCC) features as its first input and a concatenation of statistical features extracted from decomposed signals obtained through DWT as its second input. Particle swarm optimization (PSO), a feature selection algorithm, is employed to minimize the feature dimensionality by excluding the irrelevant features. The proposed model is experimentally tested on the IRMAS dataset using fixed-length single-labeled train data for model training and variable-length multi-labeled test data for model evaluation. The proposed model is evaluated using several DWT feature dimensions, and a feature dimension of 250 yields the best outcomes. The model performance is assessed by averaging the precision, recall, and $F1$ measures on a micro- and macro-level. For a set of optimal model hyperparameter values, our proposed model can reach micro and macro $F1$ measures of 0.695 and 0.631, which are 12.28% and 23.0% greater as compared to the benchmark Han et al. (IEEE/ACM Trans Audio Speech Lang Process 25(1):208–221, 2016. https://doi.org/10.1109/taslp.2016.2632307) CNN model, respectively.

Keywords Predominant instrument recognition · Deep convolutional neural networks · Mel-spectrogram · MFCC · DWT · PSO

✉ Sukanta Kumar Dash
skdash@bitmesra.ac.in

S. S. Solanki
sssolanki@bitmesra.ac.in

Soubhik Chakraborty
soubhikc@yahoo.co.in

[1] Department of Electronics and Communication Engineering, Birla Institute of Technology, Mesra, Ranchi, Jharkhand 835215, India

[2] Department of Mathematics, Birla Institute of Technology, Mesra, Ranchi, Jharkhand 835215, India

1 Introduction

Music information retrieval (MIR), a rapidly growing field of study with numerous practical applications, is useful for classifying, modifying, and synthesizing music. The most important MIR subtask, the predominant instrument recognition in polyphonic music is addressed in this article. Identifying the predominating instruments among several instruments being played simultaneously is the task of predominant instrument recognition [17]. Musical instrument recognition has recently drawn a lot of research interest because of its distinctiveness and significant commercial potential. Real-life music is polyphonic, which is characterized by the interference caused by the simultaneous occurrence of different musical sounds with significant variance in playing style, audio quality, and timbre, which makes instrument recognition even harder for computers and poses a major obstacle to the domain of MIR [16]. Due to the current focus on deep learning and artificial intelligence (AI), these technologies are being used widely in the MIR domain, which has aided in breakthroughs in several sub-fields that have been encountering bottlenecks.

Due to the growing amount of music files that are available in digital format, there is a significant need for music search. Searching for music is challenging because input queries are typically in text format, unlike text search. Automatic predominant instrument identification is crucial because source distinction performance can be greatly enhanced by understanding the instrument type [17]. If the instrument information is tagged, people can use their preferred instrument to search for their desired music. Additionally, a variety of audio/music applications can make use of the obtained instrument information, like creating music playlists [21], classifying sports audio [20], classifying acoustic scenes [44], browsing news videos [52], automatic music transcription [4], sound source separation [41], etc. Despite their diversity, these applications mainly rely on developing classification algorithms for musical instrument recognition, which necessitate the extraction of features from available music data.

1.1 Related Works

Present-day research now deals with polyphonic music, which is more representative of real-life music as compared to monophonic music. Different machine learning algorithms have been addressed as a preliminary effort for musical instrument recognition that dealt with polyphonic music audio synthesized from studio-recorded single tones. Kitahara et al. [22] used principal component analysis (PCA) and linear discriminant analysis (LDA) algorithms along with a range of spectral, temporal, and modulation features to categorize five different instruments using a music database produced by merging audio samples from solo musical instruments. The reported instrument recognition accuracy for a duo, trio, and quartet was 84.1%, 77.6%, and 72.3%, respectively. For sound separation in polyphonic audio, Heittola et al. [19] used a unique source-filter model with Gaussian mixture model (GMM) and Mel-frequency cepstral coefficient (MFCC) features and achieved a recognition level of roughly 59% for a set of six polyphonic notes selected at random from 19 distinct instruments on a single music database [15]. Fuhrmann et al. [11] aimed to identify the most recognizable

instruments in audio snippets to create a semantic relationship between them. With the help of a set of unique timbral features obtained from mean and variance statistics, they trained the support vector machine (SVM) model on a dataset of 11 modeled instruments acquired from diverse sources [3]. Their performance evaluation resulted in an F-measure greater than 0.65 and a precision score of around 0.86. Wu et al. [51] reported a joint modeling combining sustained sound and attack sound for the recognition of instruments on isolated notes spanning nine different instruments acquired from a collection of three music databases [15, 32, 33]. They used logarithmic transformation, which increased the correlation between individual timbre perception and obtained a distribution closer to the Gaussian. The authors used PCA to normalize the training set of data after transforming features into a low-dimensional vector. Utilizing the SVM model and the proposed set of features, they were able to improve instrument recognition performance by 20% and 6%, respectively, over the MFCC and source-filter features. Bosch et al. [5] proposed a novel source separation technique to train the SVM model employing typical hand-crafted audio timbral features with mean and variance statistics generated frame-by-frame to identify the predominant instruments on the IRMAS dataset of 11 instruments. They obtained the $F1$ measures for micro and macro as 0.50 and 0.43, respectively, for their proposed model. Giannoulis et al. [13] suggested a mask estimation method that made use of the probabilistic reliability of multiple feature vectors with missing features to recognize multi-pitch musical instruments without a sound source separation. Several masks were tested with the proposed method and obtained a maximum recognition average accuracy of about 68% for 10 instruments from a combination of two music databases [15, 33]. Duan et al. [8] introduced a cepstrum-based novel approach known as unified discrete cepstrum (UDC) for instrument classification. As a result, without the need for source separation, the individual sources could be estimated from a mixed spectrum directly. Authors employed SVM with UDC and the Mel-scale analogue of UDC to classify 13 various Western instruments on a single music database [32]. The recognition accuracy reported was about 37% and 25% for two and six polyphonic notes of randomly mixed chords, respectively. Thus, all these reported algorithms for identifying musical instruments need precise mining of hand-crafted features as input, which necessitates in-depth knowledge of the pertinent field.

However, with the advent of deep learning [23], the need for handcrafted features as input has been reduced drastically. Deep learning is a technique for system design that stacks numerous nonlinear modules to produce a higher-level representation automatically from the raw audio data. It trains its parameters using backpropagation algorithms, which can convert the raw inputs into useful task-specific representations. Deep learning algorithms have recently been extensively used across several research areas due to their improved performance [26, 27, 31, 40, 49]. Convolutional neural networks (CNNs), a well-liked deep learning technique, build a feature hierarchy for classification by iteratively convoluting the input source image with trained filters. The hierarchical technique thus enables the higher layers to achieve more complex features. CNNs have emerged as the most widely used technique for musical instrument recognition in recent years. Li et al. [25] demonstrated that CNNs trained on raw audio signals can outperform the conventional techniques of information retrieval that employ hand-crafted features. Han et al. [17] addressed a deep-CNN model

architecture for the recognition of predominant instruments in polyphonic music by aggregating various outcomes from sliding windows spanning the audio data on the IRMAS dataset [5] with Mel-spectrogram as input to the CNN model. The deep-CNN model was evaluated using multi-labeled testing data after being trained using single-labeled training data and obtained $F1$ measures of 0.619 and 0.513 for the micro and macro, respectively. Han's CNN model architecture was enhanced by Pons et al. [34] with the addition of single-layer and multi-layer techniques. Authors aggregated the predictions from SoftMax outputs on the IRMAS dataset [5] and applied a threshold of 0.2 for the identification of pertinent timbre information of different instrument classes and obtained $F1$ measures of 0.589 and 0.516 for the micro and macro, respectively, for their model. Yu et al. [53] extended Han et al. [17] work with a CNN model architecture based on the auxiliary classification to classify multiple instrument classes through multitask learning technique and achieved $F1$ measures of 0.685 and 0.597 for the micro and macro, respectively, for their proposed model. Raghunath et al. [39] proposed a transformer-based, multi-visual instrument recognition system on an ensemble of tempogram, modgd-gram, and Mel-spectrogram functions and were successful in achieving $F1$ measures of 0.66 and 0.62 for the micro and macro, respectively. Lekshmi et al. [24] addressed the predominant instrument recognition task using CNN model architecture and managed to achieve the $F1$ measures for micro and macro as 0.69 and 0.62, respectively, through a feature fusion of modgd-gram and Mel-spectrograms with late fusion. Each of these reported CNN-based classification algorithms used single-labeled training data for training the model and multi-labeled testing data to evaluate the model on the IRMAS dataset [5] to identify predominant instruments in polyphonic music.

1.2 Motivation

Recent research on music information retrieval (MIR), which includes works on musical instrument recognition, has focused on developing complex classification models based on feature mining. However, the type and class of the features employed as inputs for these models have not been given more importance. However, in the field of healthcare [42, 43], medical image processing [48, 54], biomedical-signal processing [46, 47], etc., one can find a substantial change in the feature diversity and feature mining modalities. In the current literature [1, 2], researchers used discrete wavelet transform (DWT) feature modality and were able to obtain notable performance enhancement. It inspired us to use a fusion of statistical features concatenated from decomposed signals obtained through DWT with perceptual features like Mel-spectrogram and MFCC, taken as input to the proposed deep-CNNs model architecture for the recognition of predominant instruments in polyphonic music.

1.3 Our Contributions

Our proposed work makes the following key contributions.

- We propose a multi-input deep-CNNs model architecture for the recognition of predominant instruments in polyphonic music on the IRMAS dataset.

- The proposed deep-CNNs model is fed with inputs that combine perceptual features like Mel-spectrogram and MFCC with seven statistical features taken from DWT.
- Feature selection algorithm, particle swarm optimization (PSO), is used to reduce feature dimensionality by removing the irrelevant features.
- Our proposed model successfully achieves the $F1$ measures for micro and macro as 0.695 and 0.631, respectively for an optimal set of model hyperparameters obtained through experiments.

The remaining section of this article is organized as follows. In Sect. 2, the system description is offered, which gives a thorough explanation of the audio pre-processing, feature selection, and feature extraction methods that were employed to identify the predominant instrument. Also covered in detail are the proposed network architecture design and training configuration. Section 3 discusses the system evaluation, explaining the IRMAS dataset, the testing configuration, and the metrics for evaluating system performance. Section 4 covers the experimental results and discussion. This section outlines the proposed model performance analysis, followed by instrument-wise performance analysis and a comparison to existing model algorithms already in use for the recognition of the predominant instrument in polyphonic music. Finally, in Sect. 5, we conclude our research work.

2 System Description

This section outlines the system description by first describing the audio data preprocessing method, then extraction of features, and finally feature selection. After that, the proposed deep-CNNs network architecture and training configuration are discussed.

2.1 Audio Pre-processing

Before the extraction of audio features, the training and testing audio samples from the IRMAS dataset were preprocessed identically as described in [17]. To normalize the audio data, it is converted from stereo to mono, sampled at 22.05 kHz, and then divided by the highest value employing the integrated library of Librosa (https://librosa.org/doc/latest/index.html). The built-in modules of Librosa [29] are used to compute the perceptual features: Mel-spectrogram and MFCC from the preprocessed audio data, which are utilized as the first line of input to the proposed deep-CNNs model. The number of the Mel-frequency bins and MFCCs is chosen to be 128 and 20, respectively. Using the pywavelets library (https://pywavelets.readthedocs.io/en/latest/), we compute DWT coefficients up to level five, resulting in six coefficients as cD1, cD2, cD3, cD4, cD5, and cA5. Seven distinct statistical features are derived from the preprocessed, decomposed signals obtained through DWT and used as the second line of input to the proposed deep-CNNs model. In our experimental work, we employ the same ideal window size of 1024 samples (about 46 ms) and hop size of 512 samples (about 23 ms) as those described in [17].

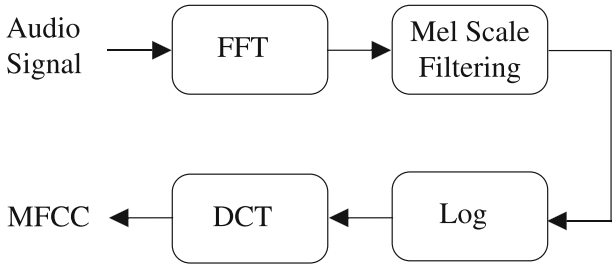


Fig. 1 Block diagram for MFCC computation

2.2 Feature Extraction

Deep-CNNs are capable of learning hierarchical feature representations automatically from raw data, but employing hand-crafted and domain-specific features as inputs to CNNs greatly enhances the performance. This work is largely focused on employing perceptually informed features like Mel-spectrogram, MFCC, and DWT-based statistical features.

2.2.1 Mel-Spectrogram

Mel-spectrograms [38] frequently employed in audio and speech processing applications [12, 45] are computed through STFT-extracted coefficients with relation to compositional frequencies. To extract Mel-spectrograms, which simulate the nonlinear perception of sound by the human ear, which is better at distinguishing between lower frequencies than higher ones, each frame of the frequency-domain representation is processed by a Mel filter bank. The formula for converting frequency (f) in Hertz to Mel-frequency (f_m) is described in [7] as:

$$(f_m) = 2595\, log_{10}\left(1 + \frac{f}{700}\right) \tag{1}$$

2.2.2 MFCCs

For a long time, audio processing applications have relied on Mel-frequency cepstral coefficients (MFCCs) as a standard representation of acoustic features [36]. According to Wikipedia [50], the short-term power spectrum of a sound is represented by the Mel-frequency cepstrum (MFC), which is based on a linear cosine transform of a log power spectrum on the nonlinear Mel-scale of frequency. Mel is a numerical value that relates to a pitch, much like a pitch specifies the frequency. Figure 1 depicts the fundamental approach for computing MFCCs.

2.2.3 Discrete Wavelet Transform (DWT)

Wavelet is a type of mathematical function, which is localized in the time and frequency domains. Wavelet transform (WT) employs wavelets as its basis functions in contrast to

Table 1 Seven most prevalent wavelet types with 52 mother wavelets

Sl. no.	Wavelet type	Mother wavelets
1	Coiflets (coif)	coif1, coif2, coif3, coif4, coif5
2	Daubechies (db)	db1, db2, db3, db4, db5, db6, db7, db8, db9 db10
3	Discrete Meyer (dmey)	dmey
4	Haar (haar)	haar
5	Symlets (sym)	sym2, sym3, sym4, sym5, sym6, sym7, sym8
6	Reverse biorthogonal (rbio)	rbio1.1, rbio1.3, rbio1.5, rbio2.2, rbio2.4 rbio2.6, rbio2.8, rbio3.1, rbio3.3, rbio3.7 rbio3.9, rbio4.4, rbio5.5, rbio6.8
7	Biorthogonal (bior)	bior1.1, bior1.3, bior1.5, bior2.2, bior2.4 bior2.6, bior2.8, bior3.1, bior3.3, bior3.7 bior3.9, bior4.4, bior5.5, bior6.8

the conventional Fourier transform, which uses sines and cosines of a fixed frequency [1]. Using WT, a multi-resolution analysis (MRA)-based method, a signal is split into different frequency bins, notably high and low-frequency bins [6]. The WT can be applied in continuous form (CWT) or in discrete form (DWT), which is employed in our experimental work. In CWT, the signal is represented by a group of basis functions referred to as mother wavelets. These mother wavelets are interrelated to one another by simple scaling and translation. In DWT, digital filtering techniques are used to represent the digital signals in their time-scale equivalents. Redundancy is one of the drawbacks of CWT, but DWT is more effective because it uses a frequency filter bank to remove undesirable frequencies and decompose the signal into different levels. There are different types of wavelets based on the frequency components they are associated with. As a result, the choice of specific wavelet type(s) determines the first step of wavelet-based digital signal processing (DSP). Multiple mother wavelets result in distinct levels of DWT for the same audio segment, which eventually leads to multiple class detection outcomes. Table 1 depicts the seven most prevalent wavelet types which were considered in this article [1]. Each wavelet type consists of individual members (mother wavelets) with various filter lengths and the resultant mother wavelet can be more smoothly characterized by a higher filter length. We use a total number of 52 mother wavelets out of these seven distinct types of wavelets. For a given audio segment, each mother wavelet generates its distinct coefficients, which might lead to varying recognition performances for the same piece of audio signal.

Computationally, DWT is calculated using a multi-level decomposition algorithm [18], a similar process of sub-band filtering. This involves processing a signal through a sequence of low-pass and high-pass filters. This procedure yields two outputs at each level: an approximation coefficient (cA) and a detail coefficient (cD) by convoluting the input signal with the coefficients of a pair of low-pass (Lp) and high-pass (Hp) half-band filters as depicted in Fig. 2. The convolution operation can be defined mathematically as:

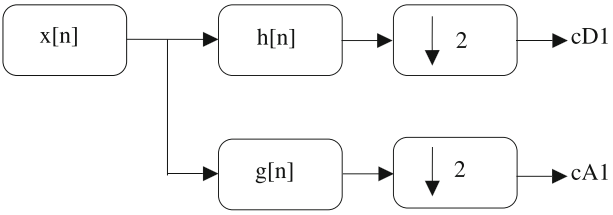


Fig. 2 Block diagram for the filter analysis

$$x[n] * f[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot f[n-k] \tag{2}$$

where $x[n]$ is the input signal and $f[n]$ is the filter impulse response. Considering, $g[n]$ and $h[n]$ are the low-pass and high-pass half-band filters, the output of the new decomposed level can be obtained as:

$$y_{\text{low}}[n] = \sum_{k} x[k] \cdot g[2k-n] \tag{3}$$

$$y_{\text{high}}[n] = \sum_{k} x[k] \cdot h[2k-n] \tag{4}$$

where $y_{\text{low}}[n]$ and $y_{\text{high}}[n]$ are the set of output signals. The resulting DWT coefficients [28] can be obtained as follows:

$$D[a,b] = \frac{1}{\sqrt{a}} \sum_{m=0}^{p-1} x\,(t_m) \cdot \Psi\left(\frac{t_m - b}{a}\right) \tag{5}$$

where a represents the scale parameter ($a = 2^j$), b represents the translation parameter ($b = k2^j$) with $j, k \in Z$, j represents scale index, k represents wavelet transform signal index, m represents discrete-time stamp that needs to be summed up varying from $0\ to\ p = 2^j$ and D represents DWT coefficients. These coefficients form the basis for the feature extraction step. In this article, we compute the DWT coefficients up to level five, yielding six coefficients: cD1, cD2, cD3, cD4, cD5, and cA5 using the pywavelets library (https://pywavelets.readthedocs.io/en/latest/). From these six DWT coefficients, we then extract seven significant statistical features including mean absolute value (MAV), average power (AVP), variance (VAR), standard deviation (SD), mean (MEAN), skewness (SKW), and Shannon entropy (SE), as described in [1]. Therefore, a total of 2184 DWT features (52 mother wavelets $\times$ 6 wavelet coefficients $\times$ 7 statistical indicators) are employed for each audio segment.

2.2.4 Statistical Features

We employ seven statistical functions to extract meaningful numerical representations from the output coefficients of DWT. These functions are listed below.

$$\text{MAV} = \frac{1}{n}\sum_{i=1}^{n}|x_i| \tag{6}$$

$$\text{AVP} = \frac{1}{n}\sum_{i=1}^{n}|x_i|^2 \tag{7}$$

$$\text{VAR} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu)^2 \tag{8}$$

$$\text{SD} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu)^2} \tag{9}$$

$$\text{MEAN} = \frac{1}{n}\sum_{i=1}^{n}x_i \tag{10}$$

$$\text{SKW} = \frac{\sum_{i=1}^{N}(x_i-\mu)^3}{N\cdot(SD)^3} \tag{11}$$

$$\text{SE} = -k\sum_{i=1}^{n}P(x_i)\log_2(P(x_i)) \tag{12}$$

where N is the segment length, x_i is the ith audio data sample in a segment, n is the total number of samples that make up an audio file, μ is the mean of audio samples, $P(x_i)$ is the probability of sample x_i and k is a positive constant.

2.3 Feature Selection

A lot of inherent redundancy in the features retrieved from DWT frequently has a detrimental impact on the model's performance. Additionally, the length of the training period substantially increases as the feature dimension grows. To avoid the above limitations, we thus employ the PSO, a feature selection algorithm, to minimize the feature dimension [9]. PSO algorithm treats each distinct feature subset as a particle, and then it optimizes an objective function that measures the effectiveness of the chosen subset. The algorithm initializes the position and velocity of each particle at random within the solution space. The position and velocity of each particle are then updated in each iteration using:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{13}$$

$$V_i^{t+1} = wV_i^t + c1\cdot r1\cdot\left(P_i^t - X_i^t\right) + c2\cdot r2\cdot\left(G^t - X_i^t\right) \tag{14}$$

where X_i^t is the position of the ith particle at iteration t, V_i^t is the velocity of the ith particle at iteration t, and w is a variable that regulates the impact of the particle's current velocity on its next velocity such that $w \in [0, 1]$, $c1$ and $c2$ are the cognitive and social parameters which regulate the impact of a particle's personal best and the global best on its velocity, $r1$, and $r2$ are random constants such that $r1, r2 \in [0, 1]$, P_i^t represents the personal best position of the ith particle at iteration t and G^t represents the global best position of the population at iteration t. After updating the positions, each particle's objective function value is evaluated, and its personal best and the global best are updated accordingly. The algorithm ends when the minimal value of the objective function or the maximum number of iterations is attained. We use the NiaPy library (https://niapy.org/en/stable/) to implement PSO on the extracted DWT features of variable length (DWT-150, DWT-250, DWT-350, and DWT-500). The initial population size is set at 50, and the number of iterations is set to 100. The fitness of each solution is evaluated using the k-nearest neighbor (k-NN) algorithm using a continuous tenfold cross-validation process.

2.4 Network Architecture

In this research study, a new deep-CNNs model architecture is proposed to identify the predominant instruments in polyphonic music, as shown in Fig. 3. The inputs to the network are fed through two different pathways: one for MFCCs and Mel-spectrograms, and the other for DWT-based statistical features. The former employs a succession of convolutional blocks, whereas the latter uses a dense layer. We employ five convolutional blocks in total with convolutional layer filters rising from 16 to 256 by a factor of 2. Each convolutional block has two convolutional layers, a layer for batch normalization, a layer for maximum pooling, and a dropout layer with a dropout rate of 0.25. The batch normalization layer offers regularization while the dropout layer prevents overfitting. Each convolutional layer uses a 3×3 kernel of stride 1 with equal padding and ReLU as the activation function [30]. The max-pooling layer, in contrast, uses a 2×2 kernel of stride of 2. However, the final convolutional block lacks the max pooling layer for preserving the dimensionality of the input to the subsequent layer. To flatten the output of the last convolutional block, a global average pooling (GAP) layer is added after the dropout layer. The first path results in a 256-feature dimension output across a global average pooling (GAP) layer, whereas the second path results in a 64-feature dimension output across a dense layer. These two outputs are then concatenated to provide an output feature dimension of 320 ($256 + 64$). After that, the concatenated output of feature dimension 320 is passed over two dense layers. A dropout layer with a dropout rate of 0.5 follows the first dense layer, which has 512 units, and a batch normalization and a dropout layer with a dropout rate of 0.5 follow the second dense layer, which has 256 units. ReLU acts as the activation function for these two fully connected layers. Finally, the prediction output is produced by the dense output layer with 11 units using SoftMax as the activation function. Table 2 depicts the input dimensions and parameter values in each layer for the proposed network architecture.

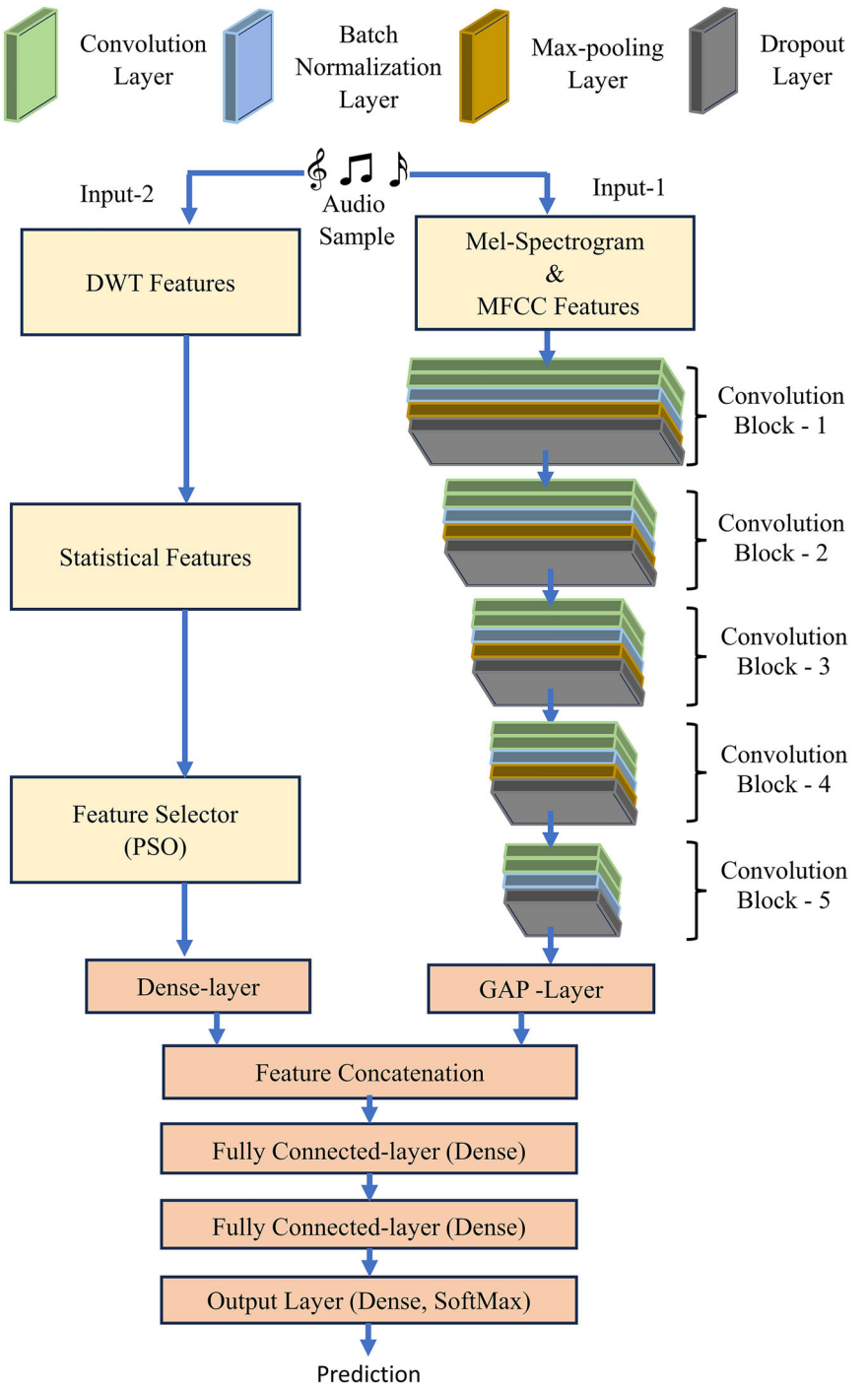


Fig. 3 Schematic of the proposed deep-CNNs model architecture

Table 2 Proposed deep-CNNs network architecture description

Input 1		Input 2	
Input dimension	Description	Input dimension	Description
44 × 148 × 1	Fusion of Mel-spectrogram and MFCC	250	DWT features
44 × 148 × 16	3 × 3 convolution, 16 filters	64	Dense
44 × 148 × 16	3 × 3 convolution, 16 filters		
44 × 148 × 16	Batch normalization		
22 × 74 × 16	2 × 2 max-pooling		
22 × 74 × 16	Dropout (0.25)		
22 × 74 × 32	3 × 3 convolution, 32 filters		
22 × 74 × 32	3 × 3 convolution, 32 filters		
22 × 74 × 32	Batch normalization		
11 × 37 × 32	2 × 2 max-pooling		
11 × 37 × 32	Dropout (0.25)		
11 × 37 × 64	3 × 3 convolution, 64 filters		
11 × 37 × 64	3 × 3 convolution, 64 filters		
11 × 37 × 64	Batch normalization		
5 × 18 × 64	2 × 2 max-pooling		
5 × 18 × 64	Dropout (0.25)		
5 × 18 × 128	3 × 3 convolution, 128 filters		
5 × 18 × 128	3 × 3 convolution, 128 filters		
5 × 18 × 128	Batch normalization		
2 × 9 × 128	2 × 2 max-pooling		
2 × 9 × 128	Dropout (0.25)		
2 × 9 × 256	3 × 3 convolution, 256 filters		
2 × 9 × 256	3 × 3 convolution, 256 filters		
2 × 9 × 256	Batch normalization		
2 × 9 × 256	Dropout (0.25)		
256	Global average pooling		
Concatenation of input 1 and input 2			
Input dimension		Description	
320		Concatenation	
512		Dense	
512		Dropout (0.5)	
256		Dense	
256		Batch normalization	
256		Dropout (0.5)	
11		Dense, SoftMax	

2.5 Training Configuration

We train the proposed network like [17] by utilizing categorical cross-entropy as a loss function and Adam as an optimizer. Additionally, the ReduceLROnPlateau call-back from Keras (https://keras.io/api/callbacks/reduce_lr_on_plateau/) is used to reduce the learning rate by a predetermined amount if a desired metric doesn't improve after a set number of epochs. For this, an initial learning rate of 0.01 is used, and if the validation loss does not decrease continuously for 7 consecutive epochs, the learning rate is reduced by a factor of 0.5. The minimum learning rate is set at 0.00005, while the batch size is set at 128. Since the training data audio files are each 3.0 s long, we experiment with a variety of analysis window sizes of 0.5 s, 1.0 s, 1.5 s, and 3.0 s and obtain 1.0 s as the ideal window size, like [17], regardless of the identification threshold 0.55. The original 3.0 s training data audio files are segmented without overlapping using an optimal window size of 1.0 s and applied the same label to each segmented chunk. The validation set is developed using 15% of the training dataset, and Keras' early stopping call-back is utilized to halt training if validation loss does not decrease for 20 epochs. Using the Glorot uniform initializer, the weights are initialized for the dense and convolutional layers, respectively [14].

3 System Evaluation

This section provides an overview of the system evaluation by initially describing the IRMAS dataset. The testing configuration and performance evaluation are then discussed.

3.1 IRMAS Dataset

The IRMAS dataset [5] contains stereo musical recordings with a 44.1 kHz sample rate. The dataset contains multiple classes of instruments, and in each instrument class, the audio files comprise music from a variety of production styles and performers. The audio files are annotated with predominant instruments present and are meant to train a classifier for the recognition of predominant instruments such as Organ (org), Clarinet (cla), Trumpet (tru), Cello (cel), Acoustic guitar (acg), Violin (vio), Piano (pia), Flute (flu), Electric guitar (elg), Saxophone (sax), and Human voice (voi). The IRMAS dataset has already been divided into distinct train and test datasets. The training dataset consists of 6705 stereo audio recordings, each of duration 3 s, that were taken from more than 2000 different recordings. These audio recordings are single-labeled with a single predominant instrument. The testing dataset consists of 4917 stereo audio recordings with a varied duration between 5 and 20 s. These audio recordings are multi-labeled and cover 1-5 instrument labels per sample. Both the training and testing datasets on the IRMAS dataset have highly uneven instrument distributions. To conduct fair comparisons with [17], 15% of the training dataset is utilized for developing a validation set. The test dataset is split into two halves, the development set, and the pure test set with no correlation between them. The development set is

used to identify the optimal model hyperparameters, while the pure test set is used for model evaluation.

3.2 Testing Configuration

The IMRAS test dataset consists of audio snippets of various lengths. As a result, we split the test dataset samples into segments of 1.0 s each to manage these variable-length inputs, much like [17]. The proposed CNN model is trained on 1.0 s segmented audio files of the train dataset and the SoftMax probabilities of individual instruments in each segment, for each file in the test dataset, are then predicted using the trained model. Then, for each distinct audio file, we compute a class-wise average of these probabilities across all the segments. We next divide the class-wise aggregate probabilities by the highest probability found among the segments to normalize it. The final forecast is then made using a threshold value. The instruments that have an aggregated probability higher than the selected threshold value are regarded as the predominant instruments. For choosing the ideal threshold, a range of threshold values between 0.2 and 0.8 is experimented, and an optimal threshold value of 0.55 is selected, as obtained the same described in [17].

3.3 Performance Evaluation

Using straightforward metrics like accuracy to evaluate performance may lead to inaccurate results since the IRMAS dataset comprises an uneven number of instances for each instrument class and the individual audio file contains a variable number of annotations. Therefore, the proposed CNN model performance is assessed using advanced metrics like the micro and macro average-based $F1$-score [5, 10, 11, 17, 53] as follows:

$$P_{\text{macro}} = \frac{1}{N}\sum_{n=1}^{N}\frac{tp_n}{tp_n + fp_n} \tag{15}$$

$$P_{\text{micro}} = \frac{\sum_{n=1}^{N} tp_n}{\sum_{n=1}^{N}(tp_n + fp_n)} \tag{16}$$

$$R_{\text{macro}} = \frac{1}{N}\sum_{n=1}^{N}\frac{tp_n}{tp_n + fn_n} \tag{17}$$

$$R_{\text{micro}} = \frac{\sum_{n=1}^{N} tp_n}{\sum_{n=1}^{N}(tp_n + fn_n)} \tag{18}$$

$$F1_{\text{macro}} = \frac{2P_{\text{macro}}R_{\text{macro}}}{P_{\text{macro}} + R_{\text{macro}}} \tag{19}$$

$$F1_{\text{micro}} = \frac{2P_{\text{micro}}R_{\text{micro}}}{P_{\text{micro}} + R_{\text{micro}}} \tag{20}$$

where N is the total number of classes, n is the class index, and tp_n, fp_n, and fn_n are the respective numbers of true positives, false positives, and false negatives for a specific class with index n.

4 Experimental Results and Discussion

4.1 Ablation Study for Wavelet and Coefficients-Level Selection

In this work, we conducted an ablation study to select the appropriate wavelet or wavelets from the list of wavelets shown in Table 1 and the level of signal decomposition to compute the appropriate number of DWT coefficients without taking the PSO into account to improve the performance of the proposed model for the task of predominant instrument recognition in polyphonic music. The model performance was examined while considering distinct wavelet types with different levels of signal decomposition, and the results are shown in Tables 3, 4, 5, 6, 7, 8, and 9. From the obtained investigation results, it is apparent that when five levels of signal decomposition are considered, the model performs better for each type of wavelet. So, we resumed our investigation into the model's performance, combining multiple wavelet types at random with five levels of signal decomposition, and the findings are displayed in Table 10. According to the findings of the experimental studies on ablation presented in Tables 3, 4, 5, 6, 7, 8, 9, and 10, it is recommended that these seven wavelet types be considered collectively to enhance the model performance while extracting six DWT coefficients: cD1, cD2, cD3, cD4, cD5, and cA5 spanning five levels of signal decomposition, as shown in Table 11. The maximum predicted values obtained through experimentation have been highlighted in bold in the Tables 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, and 13.

4.2 Proposed Model Performance Analysis

Our proposed CNN model architecture differs from the benchmark Han's CNN model [17] in a few key aspects. A five-convolutional-block design is used rather than the four-convolutional-block architecture adopted in [17]. The proposed architecture also includes batch normalization layers and the ReduceLROnPlateau call-back function. Additionally, rather than using the global max pooling layer, we utilize the global average pooling (GAP) layer. Experiments are conducted to show the efficacy of the proposed CNN model architecture. We train the model with early stopping for 150 epochs [35]. It is illustrated that changing the model architecture and using a multiple-feature fusion of Mel-spectrogram, MFCC, and statistical features extracted from DWT with PSO, a feature selection algorithm for removing irrelevant features, can improve the network performance. Figures 4 and 5 display the visual representation of Mel-spectrogram, MFCC, and DWT coefficients cD1, cD2, cD3, cD4, cD5, and cA5 up to level five considering one mother wavelet from seven distinct wavelet types for 1.0 s normalized audio clip featuring Cello as the lead instrument. A new CNN model architecture is proposed with Mel-spectrogram, MFCC features as input to deep-CNN

(Proposed-CNN) which is the modified version of the benchmark Han's CNN model, without considering the DWT features. Then, using PSO as a feature selector, the DWT features of varied dimensions, including DWT-150, DWT-250, DWT-350, and DWT-500, are employed as the second input features to the proposed CNN architecture. We consider Han's CNN model [17], which received a score of 0.619 and 0.513 for micro and macro $F1$ measures, respectively, as a benchmark for the model performance comparison. Table 12 summarizes the overall recognition performance of each proposed model in terms of model type, input feature dimension, and micro–macro measurement metrics (scores) in comparison to Han's CNN model. Further, Table 12 shows that all our proposed models: CNN, CNN+DWT (150), CNN+DWT (250), CNN+DWT (350), and CNN+DWT (500) -perform better than Han's CNN model. Additionally, it is inferred that the proposed-CNN+DWT (250) model performs the best among all the models, outperforming Han's CNN model [17] the most, achieving $F1$ measures for micro and macro as 0.695 and 0.631, which are 12.28% and 23.0% greater than Han's CNN model outcomes. The bar chart in Fig. 6 illustrates the same.

4.3 Instrument-Wise Performance Analysis

In this part, we analyze the instrument-wise performance of our proposed models, CNN, CNN+DWT (150), CNN+DWT (250), CNN+DWT (350), and CNN+DWT (500), in comparison to Han's CNN model as a benchmark, using micro and macro precision, recall, and $F1$ measures, as shown in Table 13. Table 13 shows that there is some variation in the recognition performance on the IRMAS dataset based on the type of instrument. The lowest scores of all the instruments across all the models are found for the cello and clarinet. This is primarily because both classes have much fewer data samples than the other instrument classes. However, despite the limited data available on flute class, it offers good recognition performance. This is explained by the fact that a flute has a very distinctive spectral pattern. As a result, the model can clearly distinguish the flute class from the other classes, leading to a successful performance evaluation for the flute class. Additionally, out of all the models, the voice class has the greatest $F1$ score. This is explained by the fact that the human voice produces recognizable inharmonic rhythms that make it easy to distinguish from other instruments. The voice class also has the highest data samples on the IRMAS dataset, which can be another factor. Other instruments perform only moderately. Table 13 clearly shows that our proposed CNN+DWT (250) model outperforms the benchmark Han et al. CNN model [17] for identifying the predominant instrument in polyphonic music. The bar chart in Fig. 7 serves as an illustration of the same.

Table 3 Performance investigation of CNN model with Coiflets wavelet based on micro and macro $F1$-metrics without optimal feature selection algorithm

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
1.	CNN + Coiflets	Mel-spectrogram + MFCCs	6	210	0.730	0.635	**0.680**	0.630	0.637	**0.607**
2.	CNN + Coiflets	Mel-spectrogram + MFCCs	5	175	0.708	0.632	0.668	0.607	0.660	0.598
3.	CNN + Coiflets	Mel-spectrogram + MFCCs	4	140	0.731	0.632	0.678	0.635	0.640	0.600
4.	CNN + Coiflets	Mel-spectrogram + MFCCs	3	105	0.713	0.631	0.670	0.613	0.631	0.595
5.	CNN + Coiflets	Mel-spectrogram + MFCCs	2	70	0.740	0.627	0.679	0.642	0.616	0.604

Table 4 Performance investigation of CNN model with Daubechies wavelet based on micro and macro $F1$-metrics without optimal feature selection algorithm

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
1.	CNN + Daubechies	Mel-spectrogram + MFCCs	6	420	0.732	0.626	**0.675**	0.627	0.620	**0.597**
2.	CNN + Daubechies	Mel-spectrogram + MFCCs	5	350	0.724	0.622	0.668	0.624	0.613	0.568
3.	CNN + Daubechies	Mel-spectrogram + MFCCs	4	280	0.730	0.623	0.672	0.625	0.618	0.586
4.	CNN + Daubechies	Mel-spectrogram + MFCCs	3	210	0.726	0.619	0.670	0.623	0.614	0.579
5.	CNN + Daubechies	Mel-spectrogram + MFCCs	2	140	0.728	0.625	0.673	0.624	0.626	0.591

Table 5 Performance investigation of CNN model with discrete Meyer wavelet based on micro and macro $F1$-metrics without optimal feature selection algorithm

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
1.	CNN + Discrete Meyer	Mel-spectrogram + MFCCs	6	42	0.744	0.624	**0.679**	0.649	0.629	**0.608**
2.	CNN + Discrete Meyer	Mel-spectrogram + MFCCs	5	35	0.714	0.616	0.661	0.619	0.622	0.588
3.	CNN + Discrete Meyer	Mel-spectrogram + MFCCs	4	28	0.748	0.619	0.678	0.641	0.624	0.608
4.	CNN + Discrete Meyer	Mel-spectrogram + MFCCs	3	21	0.744	0.616	0.674	0.649	0.617	0.606
5.	CNN + Discrete Meyer	Mel-spectrogram + MFCCs	2	14	0.740	0.627	0.678	0.646	0.625	0.610

Table 6 Performance investigation of CNN model with Haar wavelet based on micro and macro $F1$-metrics without optimal feature selection algorithm

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
1.	CNN + Haar	Mel-spectrogram + MFCCs	6	42	0.739	0.630	**0.680**	0.639	0.637	**0.610**
2.	CNN + Haar	Mel-spectrogram + MFCCs	5	35	0.734	0.611	0.667	0.634	0.610	0.589
3.	CNN + Haar	Mel-spectrogram + MFCCs	4	28	0.719	0.637	0.675	0.617	0.638	0.597
4.	CNN + Haar	Mel-spectrogram + MFCCs	3	21	0.722	0.621	0.669	0.636	0.618	0.592
5.	CNN + Haar	Mel-spectrogram + MFCCs	2	14	0.735	0.629	0.678	0.634	0.625	0.608

Table 7 Performance investigation of CNN model with Symlets wavelet based on micro and macro $F1$-metrics without optimal feature selection algorithm

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
1.	CNN + Symlets	Mel-spectrogram + MFCCs	6	294	0.736	0.630	**0.680**	0.632	0.626	**0.604**
2.	CNN + Symlets	Mel-spectrogram + MFCCs	5	245	0.703	0.639	0.669	0.609	0.645	0.595
3.	CNN + Symlets	Mel-spectrogram + MFCCs	4	196	0.720	0.638	0.676	0.625	0.642	0.605
4.	CNN + Symlets	Mel-spectrogram + MFCCs	3	147	0.726	0.623	0.671	0.633	0.620	0.601
5.	CNN + Symlets	Mel-spectrogram + MFCCs	2	98	0.724	0.639	0.679	0.618	0.615	0.593

Table 8 Performance investigation of CNN model with reverse biorthogonal wavelet based on micro and macro $F1$-metrics without optimal feature selection algorithm

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
1.	CNN + reverse biorthogonal	Mel-spectrogram + MFCCs	6	588	0.722	0.636	**0.676**	0.624	0.644	**0.607**
2.	CNN + reverse biorthogonal	Mel-spectrogram + MFCCs	5	490	0.710	0.629	0.667	0.607	0.620	0.582
3.	CNN + reverse biorthogonal	Mel-spectrogram + MFCCs	4	392	0.727	0.628	0.674	0.640	0.642	0.605
4.	CNN + reverse biorthogonal	Mel-spectrogram + MFCCs	3	294	0.723	0.621	0.668	0.630	0.627	0.599
5.	CNN + reverse biorthogonal	Mel-spectrogram + MFCCs	2	196	0.720	0.637	0.675	0.622	0.638	0.606

Table 9 Performance investigation of CNN model with biorthogonal wavelet based on micro and macro $F1$-metrics without optimal feature selection algorithm

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
1.	CNN + biorthogonal	Mel-spectrogram + MFCCs	6	588	0.733	0.631	**0.678**	0.630	0.632	**0.606**
2.	CNN + biorthogonal	Mel-spectrogram + MFCCs	5	490	0.708	0.631	0.667	0.612	0.624	0.580
3.	CNN + biorthogonal	Mel-spectrogram + MFCCs	4	392	0.723	0.635	0.676	0.621	0.624	0.596
4.	CNN + biorthogonal	Mel-spectrogram + MFCCs	3	294	0.719	0.624	0.668	0.616	0.603	0.585
5.	CNN + biorthogonal	Mel-spectrogram + MFCCs	2	196	0.722	0.638	0.678	0.628	0.639	0.601

Table 10 Performance investigation of CNN model with multiple wavelets based on micro and macro $F1$-metrics without optimal feature selection algorithm

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
1.	CNN + Coiflets +Daubechies	Mel-spectrogram + MFCCs	6	630	0.728	0.625	0.673	0.626	0.621	0.597
2.	CNN + Coiflets +Daubechies + Discrete Meyer	Mel-spectrogram + MFCCs	6	672	0.720	0.639	0.677	0.618	0.635	0.602
3.	CNN + Coiflets + Daubechies + Discrete Meyer + Haar	Mel-spectrogram + MFCCs	6	714	0.719	0.628	0.670	0.615	0.634	0.596

Table 10 continued

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
4.	CNN +Coiflets + Daubechies + Discrete Meyer + Haar + Symlets	Mel-spectrogram + MFCCs	6	1008	0.710	0.621	0.663	0.615	0.616	0.581
5.	CNN + Coiflets + Daubechies + Discrete Meyer + Haar + Symlets + reverse biorthogonal	Mel-spectrogram + MFCCs	6	1596	0.697	0.630	0.662	0.594	0.627	0.578

Table 10 *Continued.*

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
6.	CNN+ seven wavelet types	Mel-spectrogram + MFCCs	6	2184	**0.730**	**0.639**	**0.681**	**0.638**	**0.638**	**0.611**

Table 11 Performance investigation of CNN model with a combination of seven wavelet types based on micro and macro $F1$-metrics without optimal feature selection algorithm

Model information					Metrics					
Sl. no.	Model	Input-1 features	Wavelet coefficients used	Total feature dimension for input-2	Micro $F1$			Macro $F1$		
					P	R	$F1$	P	R	$F1$
1.	CNN + seven wavelet types	Mel-spectrogram + MFCCs	6	2184	**0.730**	**0.639**	**0.681**	**0.638**	**0.638**	**0.611**
2.	CNN + seven wavelet types	Mel-spectrogram + MFCCs	5	1820	0.715	0.619	0.664	0.609	0.598	0.575
3.	CNN + seven wavelet types	Mel-spectrogram + MFCCs	4	1456	0.704	0.630	0.665	0.602	0.605	0.573
4.	CNN + seven wavelet types	Mel-spectrogram + MFCCs	3	1092	0.729	0.622	0.671	0.631	0.635	0.601
5.	CNN + seven wavelet types	Mel-spectrogram + MFCCs	2	728	0.711	0.619	0.662	0.622	0.623	0.586

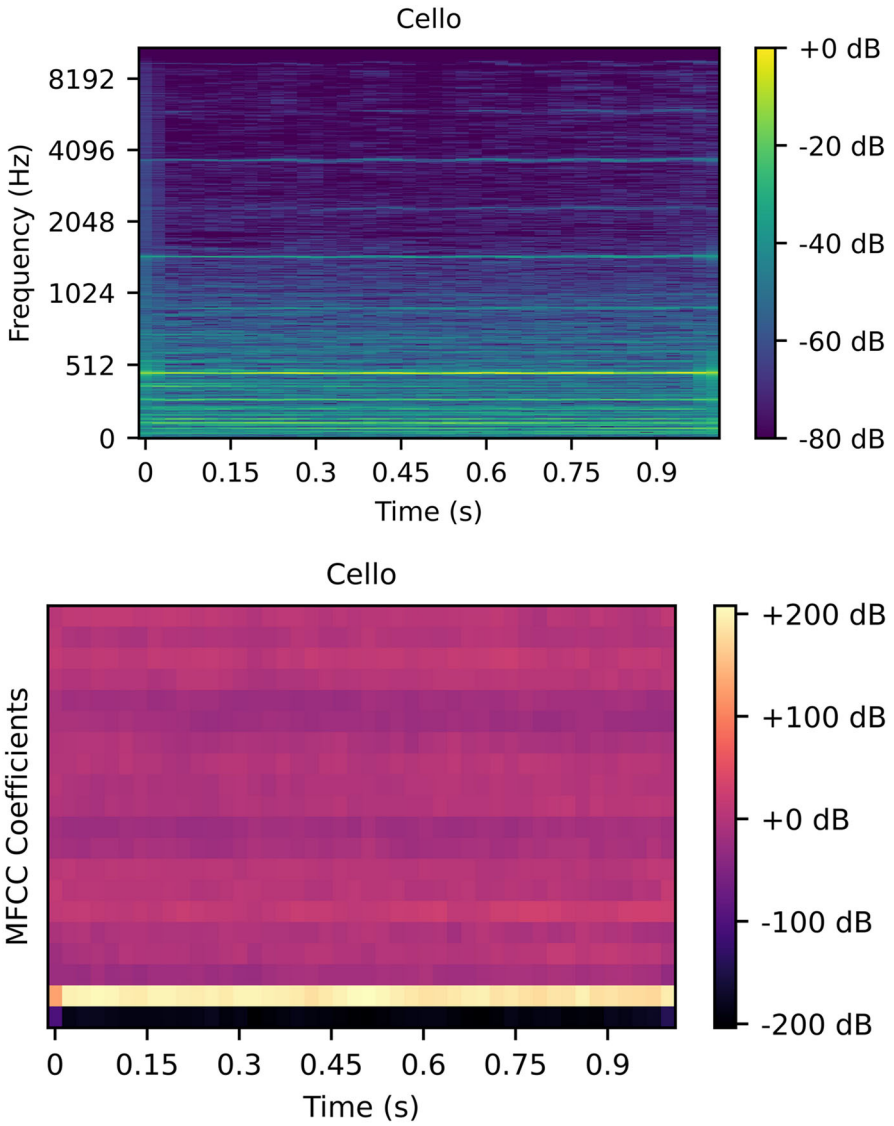
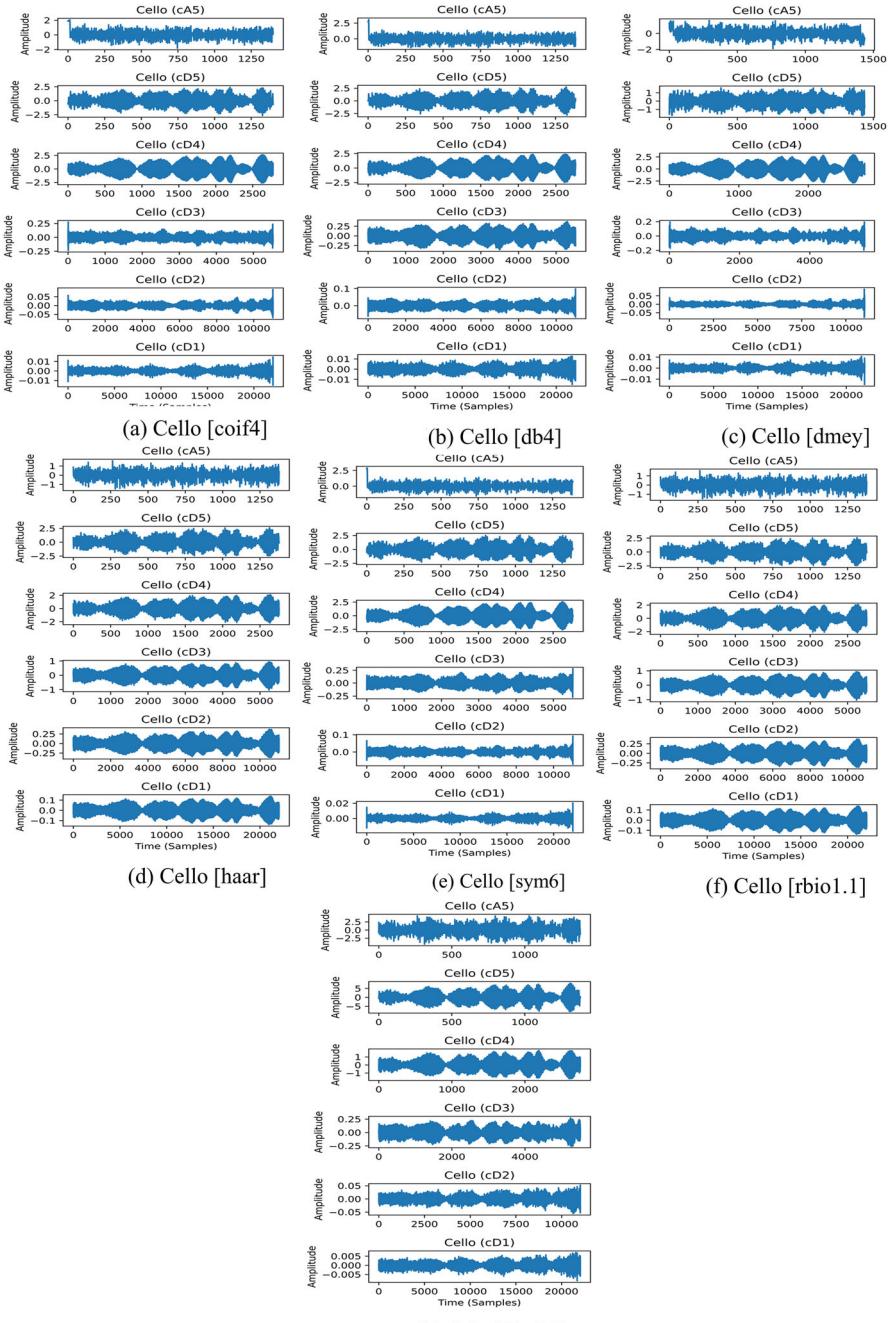


Fig. 4 Visualizations of Mel-spectrogram (top figure) and MFCC (bottom figure) for one-second normalized audio clip featuring Cello as the lead instrument

4.4 Comparison to Existing Model Algorithms

Table 14 compares the overall performance of several existing model algorithms with our proposed model algorithm CNN+DWT (250) for the task of identifying the predominant instrument in polyphonic music on the IRMAS dataset. Bosch et al. [5] used the framewise mean and variance statistics of typical hand-crafted timbral audio features to train their proposed SVM model algorithm on a flexible audio source separation framework (FASST) and reported 0.50 and 0.43 for micro and macro $F1$



(a) Cello [coif4] (b) Cello [db4] (c) Cello [dmey]

(d) Cello [haar] (e) Cello [sym6] (f) Cello [rbio1.1]

(g) Cello [bior3.1]

Fig. 5 Visualizations of six DWT coefficients (cD1, cD2, cD3, cD4, cD5, and cA5) taken from a single wavelet from each distinct wavelet type for one-second normalized audio clip featuring Cello as the lead instrument

Table 12 Performance comparison of various proposed CNN models with Han-CNN model based on micro and macro $F1$-metrics with optimal feature selection algorithm

Model information						Metrics					
Sl. no.	Model	Input-1 features	Input-2 features	Total feature dimension for input-2	Used feature dimension for input-2	Micro $F1$			Macro $F1$		
						P	R	$F1$	P	R	$F1$
1.	Han-CNN	Mel-spectrogram	–	–	–	0.668	0.575	0.619	0.548	0.530	0.513
2.	Proposed-CNN	Mel-spectrogram + MFCCs	–	–	–	0.723	0.585	0.646	0.631	0.587	0.577
3.	Proposed-CNN + DWT (150)	Mel-spectrogram + MFCCs	DWT Features	2184	150	0.735	0.636	0.682	0.632	0.629	0.607
4.	Proposed-CNN + DWT (250)	Mel-spectrogram + MFCCs	DWT Features	2184	250	**0.761**	**0.640**	**0.695**	**0.661**	**0.646**	**0.631**
5.	Proposed-CNN + DWT (350)	Mel-spectrogram + MFCCs	DWT Features	2184	350	0.738	0.631	0.680	0.647	0.639	0.614
6.	Proposed-CNN + DWT (500)	Mel-spectrogram + MFCCs	DWT Features	2184	500	0.640	0.615	0.627	0.553	0.600	0.546

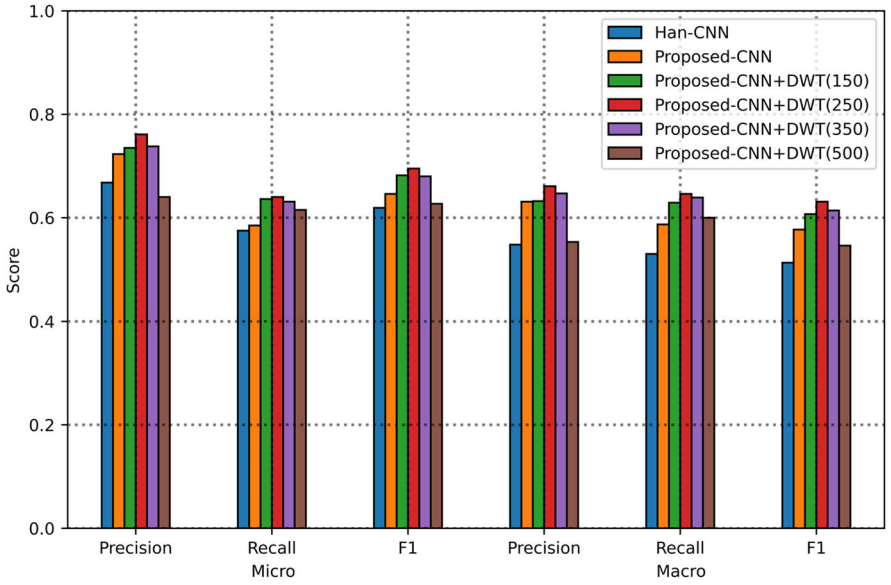


Fig. 6 Comparison of performance between the benchmark Han's CNN model architecture (Han-CNN) [17] and the proposed CNN model architectures: CNN, CNN+DWT (150), CNN+DWT (250), CNN+DWT (350), and CNN+DWT (500) based on micro and macro scores

measures, respectively. The benchmark Han et al. model [17] addressed a deep-CNN model algorithm for the predominant instrument recognition in polyphonic music with Mel-spectrogram feature as input to their model and reported improvements in both micro and macro $F1$ measures as 0.619 and 0.513, respectively. Pons et al. [34] adopted Han's CNN model [17] with a novel algorithm design that successfully captures the required audio timbre information and reported 0.589 and 0.516 for micro and macro $F1$ measures, respectively. Yu et al. [53] proposed a deep-CNN model with an auxiliary classification algorithm to make their model learn the varied instrument classes through a multitask learning approach and reported 0.685 and 0.597 for micro and macro $F1$ measures, respectively. Raghunath et al. [39] explored one transformer-based algorithm on an ensemble of tempogram, modgd-gram, and Mel-spectrogram visual representations for the detection of several dominant instruments in polyphonic music. Their model algorithm obtained 0.66 and 0.62 for micro and macro $F1$ measures, respectively. After experimenting with several fusion algorithms, Lekshmi et al. [24] addressed a late fusion algorithm that received 0.69 and 0.62 for micro and macro $F1$ measures, respectively. We proposed a new deep-CNNs model architecture, CNN+DWT (250), which employs a feature fusion of Mel-spectrogram, MFCC, and statistical features extracted from DWT as model inputs. The experimental outcomes show that our proposed model algorithm surpasses all other model algorithms, achieving 0.695 and 0.631 for micro and macro $F1$ measures, respectively, for the recognition of the predominant instrument in polyphonic music on the IRMAS dataset.

Table 13 Instrument-wise performance comparison of various proposed CNN models with Han-CNN model based on precision (P), recall (R), and $F1$-metrics

Instrument class	Han-CNN			Proposed-CNN			Proposed-CNN + DWT(150)			Proposed-CNN + DWT(250)			Proposed-CNN + DWT(350)			Proposed-CNN + DWT(500)		
	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$
Organ	0.45	0.46	0.45	0.77	0.35	0.49	0.72	0.62	0.67	0.80	0.55	0.65	0.74	0.62	0.67	0.67	0.61	0.64
Clarinet	0.11	0.65	0.18	0.33	0.43	0.38	0.23	0.43	0.30	0.33	0.57	0.42	0.38	0.57	0.46	0.10	0.50	0.17
Trumpet	0.47	0.42	0.44	0.48	0.77	0.59	0.55	0.69	0.61	0.61	0.71	0.66	0.46	0.68	0.55	0.36	0.62	0.46
Cello	0.55	0.55	0.55	0.35	0.52	0.42	0.26	0.57	0.36	0.38	0.54	0.45	0.35	0.57	0.43	0.30	0.52	0.38
Acoustic guitar	0.84	0.63	0.72	0.73	0.62	0.67	0.80	0.64	0.71	0.71	0.69	0.70	0.71	0.66	0.69	0.51	0.67	0.58
Violin	0.41	0.57	0.48	0.62	0.48	0.54	0.61	0.53	0.57	0.63	0.48	0.55	0.73	0.40	0.52	0.48	0.48	0.48
Piano	0.76	0.61	0.67	0.83	0.36	0.51	0.92	0.40	0.56	0.87	0.39	0.54	0.85	0.39	0.54	0.83	0.41	0.55
Flute	0.33	0.61	0.43	0.47	0.70	0.56	0.55	0.75	0.63	0.52	0.83	0.64	0.46	0.79	0.58	0.63	0.58	0.60
Electric guitar	0.69	0.69	0.69	0.86	0.44	0.59	0.79	0.61	0.69	0.85	0.62	0.72	0.90	0.55	0.68	0.77	0.61	0.68
Saxophone	0.62	0.61	0.61	0.58	0.85	0.69	0.58	0.79	0.67	0.58	0.82	0.68	0.58	0.88	0.70	0.47	0.79	0.59
Voice	0.94	0.78	0.85	0.91	0.93	0.92	0.95	0.89	0.92	0.98	0.90	0.94	0.97	0.91	0.94	0.95	0.82	0.88
Micro	0.64	0.64	**0.64**	0.72	0.58	**0.65**	0.73	0.64	**0.68**	0.76	0.64	**0.70**	0.74	0.63	**0.68**	0.64	0.61	**0.63**
Macro	0.56	0.60	**0.55**	0.63	0.59	**0.58**	0.63	0.63	**0.61**	0.66	0.65	**0.63**	0.65	0.64	**0.61**	0.55	0.60	**0.55**

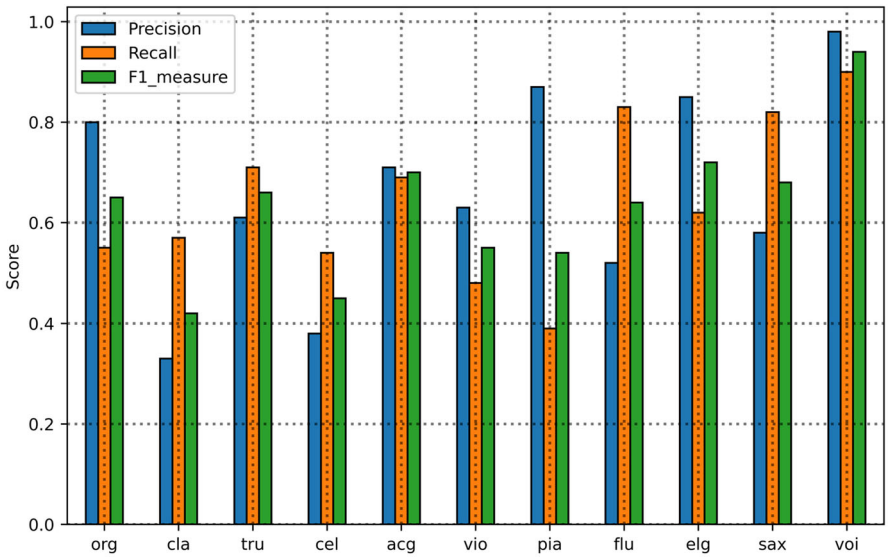


Fig. 7 Instrument-wise performance analysis for the proposed model CNN+DWT (250)

Table 14 Comparison of proposed model performance with various existing model algorithms on IRMAS dataset

Sl. no.	Model algorithm	Micro $F1$	Macro $F1$
1	Bosh et al. [5]	0.50	0.43
2	Han et al. [17]	0.619	0.513
3	Pons et al. [34]	0.589	0.516
4	Yu et al. [53]	0.685	0.597
5	Reghunath et al. [39]	0.66	0.62
6	Lekshmi et al. [24]	0.69	0.62
7	**Proposed-CNN + DWT (250)**	**0.695**	**0.631**

Performance of the proposed model is emphasized

5 Conclusion

In this article, we proposed a new multi-input deep-CNNs model architecture for the recognition of predominant musical instruments in polyphonic music. A fusion of Mel-spectrogram and MFCC features was used as the first input, and a concatenation of statistical features extracted from decomposed signals obtained through DWT was used as the second input to the proposed deep-CNNs model. Using PSO, a feature selection algorithm, the feature dimensionality was reduced by excluding the irrelevant features. All our proposed models were experimentally evaluated on the IRMAS dataset. In experimental work, the fixed-length, single-labeled train data was used for model training, whereas the variable-length, multi-labeled test data was utilized for model evaluation. By using the final proposed model CNN+DWT (250), we were able

to outperform the benchmark Han et al. CNN model architecture [17] by 12.28% and 23.0%, reaching the $F1$ measures for micro and macro as 0.695 and 0.631, respectively. Therefore, it can be hypothesized that using DWT-based features along with perceptually informed features, like Mel-spectrogram and MFCC, as input to deep-CNNs will result in a more performance-effective representation for the task of predominant instrument recognition in polyphonic music than doing so with just perceptually informed features. However, we think that integrating transformer models [37] with our proposed CNN+DWT (250) model would lead to better outcomes for this task.

References

1. A. al-Qerem, F. Kharbat, S. Nashwan, S. Ashraf, K. Blaou, General model for best feature extraction of EEG using discrete wavelet transform wavelet family and differential evolution. Int. J. Distrib. Sens. Netw. **16**, 1–21 (2020). https://doi.org/10.1177/1550147720911009
2. K. Alsharabi, Y.B. Salamah, A.M. Abdurraqeeb, M. Aljalal, F.A. Alturki, EEG signal processing for Alzheimer's disorders using discrete wavelet transform and machine learning approaches. IEEE Access **10**, 89781–89797 (2022). https://doi.org/10.1109/access.2022.3198988
3. J.J. Aucouturier, Sounds like teen spirit: Computational insights into the grounding of everyday musical terms, in *Language, Evolution and the Brain, Book Chapter-2* (City University of Hong Kong Press, 2009), pp. 35–64
4. E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, A. Klapuri, Automatic music transcription: challenges and future directions. J. Intell. Inf. Syst. **41**(3), 407–434 (2013). https://doi.org/10.1007/s10844-013-0258-3
5. J.J. Bosch, J. Janer, F. Fuhrmann, P. Herrera, A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals, in *Proceedings, International Society for Music Information Retrieval Conference (ISMIR 2012)* (2012), pp. 559–564. https://doi.org/10.5281/zenodo.1416075
6. L. Debnath, J.-P. Antoine, Wavelet transforms and their applications. Phys. Today **56**(4), 68–68 (2003). https://doi.org/10.1063/1.1580056
7. J.D. Deng, C. Simmermacher, S. Cranefield, A study on feature analysis for musical instrument classification. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **38**(2), 429–438 (2008). https://doi.org/10.1109/tsmcb.2007.913394
8. Z. Duan, B. Pardo, L. Daudet, A novel Cepstral representation for timbre modeling of sound sources in polyphonic mixtures, in *Proceedings, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)* (2014), pp. 7495–7499. https://doi.org/10.1109/icassp.2014.6855057
9. R.C. Eberhart, Y. Shi, Particle swarm optimization: development, applications and resources, in *Proceedings, IEEE Conference on Evolutionary Computation, (IEEE Cat. No.01TH8546), ICEC*, vol. 1 (2001), pp. 81–86. https://doi.org/10.1109/cec.2001.934374
10. M.R. Every, Discriminating between pitched sources in music audio. IEEE Trans. Audio Speech Lang. Process. **16**(2), 267–277 (2008). https://doi.org/10.1109/tasl.2007.908128
11. F. Fuhrmann, P. Herrera, Polyphonic instrument recognition for exploring semantic similarities in music, in *Proceedings, 13th International Conference on Digital Audio Effects (DAFx-10)* (2010), pp. 1–8. http://mtg.upf.edu/files/publications/ffuhrmann_dafx10_final_0.pdf
12. D. Ghosal, M.H. Kolekar, Music genre recognition using deep neural networks and transfer learning, in *Proceedings, Interspeech* (2018), pp. 2087–2091. https://doi.org/10.21437/interspeech.2018-2045
13. D. Giannoulis, A. Klapuri, Musical instrument recognition in polyphonic audio using missing feature approach. IEEE Trans. Audio Speech Lang. Process. **21**(9), 1805–1817 (2013). https://doi.org/10.1109/tasl.2013.2248720
14. X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in *Proceedings, 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 9, Chia Laguna Resort, Sardinia, Italy (2010), pp. 249–256. https://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

15. M. Goto, H. Hashiguchi, T. Nishimura, R. Oka, RWC music database: popular, classical, and jazz music database, in *Proceedings, 3rd International Conference on Music Information Retrieval (ISMIR)* (2002), pp. 287–288. https://www.researchgate.net/publication/220723431
16. S. Gururani, C. Summers, A. Lerch, Instrument activity detection in polyphonic music using deep neural networks, in *Proceedings, International Society for Music Information Retrieval Conference*, Paris, France (2018), pp. 569–576. https://www.researchgate.net/publication/332621784
17. Y. Han, J. Kim, K. Lee, Deep convolutional neural networks for predominant instrument recognition in polyphonic music. IEEE/ACM Trans. Audio. Speech Lang. Process. **25**(1), 208–221 (2016). https://doi.org/10.1109/taslp.2016.2632307
18. K.K. Hasan, U.K. Ngah, M.F.M. Salleh, Multilevel decomposition discrete wavelet transform for hardware image compression architectures applications, in *Proceedings, IEEE International Conference on Control System, Computing and Engineering*, Penang, Malaysia (2013), pp. 315–320. https://doi.org/10.1109/iccsce.2013.6719981
19. T. Heittola, A. Klapuri, T. Virtanen, Musical instrument recognition in polyphonic audio using source-filter model for sound separation, in *Proceedings, International Society for Music Information Retrieval Conference (ISMIR)* (2009), pp. 327–332. https://www.researchgate.net/publication/220723588
20. J. Huang, Y. Dong, J. Liu, C. Dong, H. Wang, Sports audio segmentation and classification, in *Proceedings, International Conference on Network Infrastructure and Digital Content (IC-NIDC ?09)* (IEEE, Beijing, China, 2009), pp. 379–383. https://doi.org/10.1109/icnidc.2009.5360872
21. R.T. Irene, C. Borrelli, M. Zanoni, M. Buccoli, A. Sarti, Automatic playlist generation using convolutional neural networks and recurrent neural networks, in *Proceedings, European Signal Processing Conference (EUSIPCO)* (IEEE, 2019), pp. 1–5. https://doi.org/10.23919/eusipco.2019.8903002
22. T. Kitahara, M. Goto, K. Komatani, T. Ogata, H.G. Okuno, Instrument identification in polyphonic music: feature weighting to minimize influence of sound overlaps. J. Appl. Signal Process. (EURASIP) **2007**, 155–155 (2007). https://doi.org/10.1155/2007/51979
23. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature **521**(7553), 436–444 (2015). https://doi.org/10.1038/nature14539
24. C.R. Lekshmi, R. Rajeev, Multiple predominant instruments recognition in polyphonic music using spectro/modgd-gram fusion. Circuits Syst. Signal Process. **42**(6), 3464–3484 (2023). https://doi.org/10.1007/s00034-022-02278-y
25. P. Li, J. Qian, T. Wang, Automatic instrument recognition in polyphonic music using convolutional neural networks (2015), pp. 1–5. https://doi.org/10.48550/arXiv.1511.05520. arXiv:1511.05520
26. P. Li, Z. Chen, L.T. Yang, Q. Zhang, M.J. Deen, Deep convolutional computation model for feature learning on big data in Internet of Things. IEEE Trans. Ind. Inf. **14**(2), 790–798 (2018). https://doi.org/10.1109/tii.2017.2739340
27. Y. Luo, N. Mesgarani, Conv-tasnet: surpassing ideal time-frequency magnitude masking for speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(8), 1256–1266 (2019). https://doi.org/10.1109/taslp.2019.2915167
28. E. Magosso, M. Ursino, A. Zaniboni, E. Gardella, A wavelet-based energetic approach for the analysis of biomedical signals: application to the electroencephalogram and electro-oculogram. Appl. Math. Comput. **207**(1), 42–62 (2009). https://doi.org/10.1016/j.amc.2007.10.069
29. B. McFee, C. Raffel, D. Liang, D.P.W. Ellis, M. McVicar, E. Battenberg, O. Nieto, Librosa: audio and music signal analysis in Python, in *Proceedings, 14th Python in Science Conference (SCIPY 2015)*, vol. 8 (2015), pp. 18–25. https://doi.org/10.25080/majora-7b98e3ed-003
30. V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in *Proceedings, 27th International Conference on Machine Learning*, Haifa, Israel (2010), pp. 807–814. https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf
31. T.-L. Nguyen, S. Kavuri, M. Lee, A multimodal convolutional neuro-fuzzy network for emotional understanding of movie clips. Neural Netw. **118**, 208–219 (2019). https://doi.org/10.1016/j.neunet.2019.06.010
32. [Online]. Available: http://theremin.music.uiowa.edu/MIS.html
33. F.J. Opolko, J. Wapnick, Mcgill University master samples. Montreal, QC, Canada: McGill University, Faculty of Music (1987). https://www.worldcat.org/title/mums-mcgill-university-master-samples/oclc/17946083
34. J. Pons, O. Slizovskaia, R. Gong, E. Gomez, X. Serra, Timbre analysis of music audio signals with convolutional neural networks, in *Proceedings, 25th European Signal Processing Conference* (IEEE, 2017), pp. 2744–2748. https://doi.org/10.23919/eusipco.2017.8081710

35. L. Prechelt, Early stopping—but when?, in *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*, vol. 7700, ed. by G.B. Orr, K.R. Muller (Springer, Berlin, 2012), pp.53–67. https://doi.org/10.1007/978-3-642-35289-8_5
36. H. Purwins, B. Li, T. Virtanen, J. Schluter, S.-Y. Chang, T. Sainath, Deep learning for audio signal processing. IEEE J. Sel. Top. Signal process **13**(2), 206–219 (2019). https://doi.org/10.1109/jstsp.2019.2908700
37. L. Qiu, S. Li, Y. Sung, DBTMPE: deep bidirectional transformers-based masked predictive encoder approach for music genre classification. Mathematics **9**(5), 1–17 (2021). https://doi.org/10.3390/math9050530
38. L.R. Rabiner, R.W. Schafer, *Theory and Applications of Digital Speech Processing* (Prentice Hall Press, Hoboken, 2010)
39. L.C. Reghunath, R. Rajan, Transformer-based ensemble method for multiple predominant instruments recognition in polyphonic music. EURASIP J. Audio Speech Music Process. **2022**(1), 1–14 (2022). https://doi.org/10.1186/s13636-022-00245-8
40. A. Sano, W. Chen, D. Lopez-Martinez, S. Taylor, R.W. Picard, Multimodal ambulatory sleep detection using LSTM recurrent neural networks. IEEE J. Biomed. Health Inform. **23**(4), 1607–1617 (2019). https://doi.org/10.1109/jbhi.2018.2867619
41. K. Schulze-Forster, K.G. Richard, L. Kelley, C.S.J. Doire, R. Badeau, Unsupervised music source separation using differentiable parametric source models. IEEE/ACM Trans. Audio Speech Lang. Process. **31**, 1276–1289 (2023). https://doi.org/10.1109/taslp.2023.3252272
42. M. Sharma, R.B. Pachori, U.R. Acharya, A new approach to characterize epileptic seizures using analytic time-frequency flexible wavelet transform and fractal dimension. Pattern Recogn. Lett. **94**, 172–179 (2017). https://doi.org/10.1016/j.patrec.2017.03.023
43. L. Shi, Y. Zhang, J. Zhang, Lung sound recognition method based on wavelet feature enhancement and time-frequency synchronous modeling. IEEE J. Biomed. Health Inform. **27**(1), 308–318 (2023). https://doi.org/10.1109/jbhi.2022.3210996
44. D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, M.D. Plumbley, Detection and classification of acoustic scenes and events. IEEE Trans. Multimed. **17**(10), 1733–1746 (2015). https://doi.org/10.1109/tmm.2015.2428998
45. M. Sukhavasi, S. Adapa, Music theme recognition using CNN and self-attention (2019). https://doi.org/10.48550/arXiv.1911.07041, arXiv preprint arXiv:1911.07041
46. T. Tuncer, S. Dogan, A. Subasi, Surface EMG signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition. Biomed. Signal Process. Control **58**, 1–12 (2020). https://doi.org/10.1016/j.bspc.2020.101872
47. T. Tuncer, S. Dogan, A. Subasi, EEG-based driving fatigue detection using multilevel feature extraction and iterative hybrid feature selection. Biomed. Signal Process. Control **68**, 1–11 (2021). https://doi.org/10.1016/j.bspc.2021.102591
48. S.P. Vaidya, Fingerprint-based robust medical image watermarking in hybrid transform. Vis. Comput. **39**, 2245–2260 (2022). https://doi.org/10.1007/s00371-022-02406-4
49. C.-Y. Wang, J.C. Wang, A. Santoso, C.C. Chiang, C.H. Wu, Sound event recognition using auditory-receptive-field binary pattern and hierarchical-diving deep belief network. IEEE/ACM Trans. Audio Speech Lang. Process. **26**(8), 1336–1351 (2018). https://doi.org/10.1109/taslp.2017.2738443
50. Wikipedia contributors. Mel-frequency cepstrum—Wikipedia, the free encyclopedia (2019). https://en.wikipedia.org/w/index.php?title=Mel-frequency_cepstrum&oldid=917928298
51. J. Wu, E. Vincent, S.A. Raczynski, T. Nishimoto, N. Ono, S. Sagayama, Polyphonic pitch estimation and instrument identification by joint modeling of sustained and attack sounds. IEEE J. Sel. Top. Signal Process. **5**(6), 1124–1132 (2011). https://doi.org/10.1109/jstsp.2011.2158064
52. X. Wu, C.-W. Ngo, Q. Li, Threading and auto documenting news videos: a promising solution to rapidly browse news topics. IEEE Signal Process. Mag. **23**(2), 59–68 (2006). https://doi.org/10.1109/msp.2006.1621449
53. D. Yu, H. Duan, J. Fang, B. Zeng, Predominant instrument recognition based on deep neural network with auxiliary classification. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 852–861 (2020). https://doi.org/10.1109/taslp.2020.2971419
54. N. Zermi, A. Khaldi, M.R. Kafi, F. Kahlessenane, S. Euschi, Robust SVD-based schemes for medical image watermarking. Microprocess. Microsyst. **84**, 1–12 (2021). https://doi.org/10.1016/j.micpro.2021.104134