applied sciences

MDPI

Article

Detecting Selected Instruments in the Sound Signal

Daniel Kostrzewa, Paweł Szwajnoch, Robert Brzeski and Dariusz Mrozek *

Department of Applied Informatics, Silesian University of Technology, 44-100 Gliwice, Poland; daniel.kostrzewa@polsl.pl (D.K.); robert.brzeski@polsl.pl (R.B.)
* Correspondence: dariusz.mrozek@polsl.pl

Abstract: Detecting instruments in a music signal is often used in database indexing, song annotation, and creating applications for musicians and music producers. Therefore, effective methods that automatically solve this issue need to be created. In this paper, the mentioned task is solved using mel-frequency cepstral coefficients (*MFCC*) and various architectures of artificial neural networks. The authors' contribution to the development of automatic instrument detection covers the methods used, particularly the neural network architectures and the voting committees created. All these methods were evaluated, and the results are presented and discussed in the paper. The proposed automatic instrument detection methods show that the best classification quality was obtained for an extensive model, which is the so-called committee of voting classifiers.

Keywords: convolutional neural network; music information retrieval; audio features; sound analysis; Mel-Frequency Cepstral Coefficients—*MFCC*; recognizing musical instruments; classifier committee; Medley-solos-DB; artificial neural network

1. Introduction

Detecting instruments in a signal is often used for indexing databases, annotating songs, and creating applications for musicians and music producers [1]. Detecting and classifying musical instruments in a signal is a non-trivial issue. This is due to the fact that any item that is the source of sound phenomena can be considered a musical instrument. Another important factor complicating the problem is the multitude of types of instruments. Moreover, the sound characteristics of two seemingly different types of instruments can be very similar. It may depend on the manner of articulation or even the degree of tuning [2].

The work aims to create methods for discovering which musical instrument appears in a given sound signal. Depending on one's predispositions and experience in listening to music, a person is able to indicate what instruments were used in a given piece. These predispositions and the musical experience influence the precision with which one can detect the instruments heard. The created methods should, therefore, work similarly to those of an experienced listener in a way that, after the learning process with a selected set of values representing/describing the sound, they will recognize instruments in the musical signal with sufficiently high efficiency. Instrument recognition itself is based primarily on the timbre and tone of the sound.

Sound timbre is a phenomenon that allows a person to distinguish what type of instrument the sound comes from, recognize speech, ambient sounds, etc. The sound coming from musical instruments is, in fact, the result of overlapping many simple vibrations with different frequencies and intensities, thus creating a complex, multidimensional phenomenon. The parameter values of these individual component tones determine the timbre of the sound. Determining sound timbre involves analyzing the acoustic spectrum, which can be obtained using the Fourier transform.

The term timbre may be related to the resemblance to the sound of a musical instrument (e.g., clarinet or cello), the image of temperature (e.g., warm or cold), shape (e.g., round or flat), or color (e.g., light or dark color) [2,3].

check for updates

Citation: Kostrzewa, D.; Szwajnoch, P.; Brzeski, R.; Mrozek, D. Detecting Selected Instruments in the Sound Signal. *Appl. Sci.* **2024**, *14*, 6330. https://doi.org/10.3390/app14146330

Academic Editor: Rocco Zaccagnino

Received: 5 June 2024
Revised: 13 July 2024
Accepted: 18 July 2024
Published: 20 July 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

In the described system, as in other systems related to recognizing sound components and analyzing sound signals, the key element is the appropriate extraction of selected features from the signal [4,5]. Building a model that detects instruments in an audio signal requires multi-step transformations to extract relevant information from the signal. The most popular and widely used method for mapping a parametric acoustic signal is the Mel-Frequency Cepstral Coefficients (*MFCC*). They were initially used in echo detection in seismic waves [6], but they are also used in speech detection problems [7]. *MFCC* parameters are also used to determine the genre and timbre of a musical piece [8]. The effectiveness of *MFCC* has also been confirmed in terms of music recommendations [9–11].

In this work, the extraction of sound components was also achieved by using *MFCC* coefficients, which are a mathematical representation of how the human ear perceives music. The role of classifiers was played by artificial dense neural networks, convolutional neural networks, and derivative models—the voting committees based on the above-mentioned ones.

There are two main ways to generate the final result of the classification of the created ensemble. One of them is to combine the outputs of individual base models with another dense layer (or layers), which adjusts its weights to the final generation of the classification result in the process of training the entire created assembly. The second method is to attach a voting system to the basic models to determine the classification result based on the outputs of the basic models. Such a voting system does not adjust its weights while training the entire ensemble but generates the result in accordance with the voting rules imposed on it.

The remainder of the paper is as follows: Section 1 introduces the undertaken issue. Section 2 presents the related works and highlights the contribution. Section 3 provides an overview of automatic instrument detection methods, describes the used database, and details of the built system. Section 4 describes the method of conducting and evaluating experiments. The conducted experiments and received results are presented in Sections 5 and 6. The paper is concluded in Section 7, where the future work is also outlined.

2. Related Work

The popularity and spread of machine learning, artificial intelligence, and similar solutions also have an impact on the field of acoustic signal analysis and processing [12–16]. Machine learning methods, in particular, artificial neural network (*ANN*), are an alternative to classic methods such as the k-nearest neighbor or the random trees algorithms. Their effectiveness mainly contributed to improving the quality of speech recognition applications and automatic music recommendation systems [17]. Compared to other deep learning techniques, especially convolutional neural networks (*CNN*) are considered extremely effective in the context of audio processing [18–21]. Apart from those mentioned, other applications of the *CNN* network include transcription, voice detection, chord recognition, beat detection, defining time signatures, frequency response analysis, and articulation recognition.

Convolutional networks [22–24] are also widely used in automatic instrument recognition systems. In [22], research was carried out consisting of carrying out the constant Q transform (*CQT*) of the signal, thus obtaining the input matrix for the convolutional network, which consisted of an input layer, two convolutional layers, and two dense layers. Moreover, the Rectified Linear Unit (*ReLU*) [25] was used as an activation function. The ReLU function works in the way that if the input x is less than 0, the output is equal to 0; if the input x is greater than 0, the output is equal to the input. The training set (Medley-solos-db, which is described in Section 3.1) consisted of 158 min of recordings of eight different instruments: clarinet, electric guitar, flute, piano, tenor saxophone, trumpet, violin, and a female voice, which in this context can be treated as a stringed instrument. The test set contained tracks with a total length of 208 min. These sets were unbalanced, so the recordings of some of the instruments constituted only a small percentage of the total set. Ten different configurations were tested, which signaled the general trend that increasing

the number of network parameters led to better network performance. The maximum efficiency of the algorithm that was achieved was 74% correct decisions.

In [26], the authors developed a deep learning architecture that is relatively small (in terms of a number of parameters). The main idea was to create an autoencoder to discover a set of embedded representations of the instrument's sound and then provide them to the specialized prototype layer. The mean accuracy obtained for the model based on the Medley-solos-db dataset is 67.3%.

Dubey et al. presented a relatively simple architecture that consists of a few steps [27]. Firstly, they generated Mel-spectrograms and a set of *MFCC* values, and then (due to a noticeable class imbalance in the Medley-solos-db dataset), the SMOTE algorithm was performed. The outcome of SMOTE was transmitted to the convolutional network for classification. The authors obtained stunning accuracy of over 99%. However, they provided the numerical results only for the training and validation subsets, so the real test accuracy is still unknown.

The actual state of the art is [28]. The authors developed a very sophisticated multi-level deep architecture. They are using it for creating a kind of foundation model for different music-oriented tasks, which is further prepared inter alia for instrument recognition tasks. The final accuracy for Medley-solos-db is 76.1%.

Other works on instrument detection in sound signals can also be found [29–31]. However, these are works carried out on different datasets. They differ in the number of different instruments, the number of samples, the degree of balance, the sound of selected instruments, the length of a single recording, and the method of recording (e.g., MIDI—containing individual notes, wav—containing the actual sound). Also, the purpose of these studies is often different.

Contribution

Several existing works in sound analysis and other domains show that using an ensemble of classifiers [32] leads to the improvement of classification efficiency [33–36]. This motivated the authors to apply this idea in the domain of instrument detection in two different approaches, namely through the concatenation of light classifiers within one complex neural classifier committee architecture and through a committee of voting classifiers [37–39].

The contribution of the paper consists of 3 elements. The first one is the creation of different architectures and testing of different parameters for the dense and convolutional neural network with the aim of instrument detection. The entire network architecture improvement path is described in Section 5. The second one is the creation and use of voting committees for automatic instrument detection. The third one is a comparison of the voting committees to the neural classifier committee, created from the same base models.

3. Data Pre-Processing

This section describes the database used and preliminary data preparation.

3.1. Dataset

The choice of the dataset was determined by several factors. Firstly, the criteria for assessing the built model included, among others, a comparative analysis of the results obtained and those presented in other scientific works. Therefore, the dataset should be publicly available to find works based on it. Moreover, the set should be large enough to avoid the model adapting to a specific, narrow set of data. Taking into account the field under consideration, the collection should contain recordings from various instruments, preferably many types.

Therefore, the current work uses the Medley-solos-DB [22,40] dataset, which has also been used in other scientific works described in the Section 2. This collection contains recordings of the following instruments (Table 1): clarinet, electric guitar, flute, piano,

tenor saxophone, trumpet, violin, female voice (which, in this context, can be treated as a stringed instrument).

Table 1. Number of recordings for each instrument.

Instrument	Number of Recordings—Samples
clarinet	1311
electric guitar	1854
female voice	1744
flute	3555
piano	6032
tenor saxophone	477
trumpet	627
violin	5971
in total	21,571

As can be seen, the set includes representatives of woodwind instruments (clarinet, flute, tenor saxophone), brass instruments (trumpet), string instruments (violin), hammer string instruments (piano), plucked string instruments (guitar), and female vocals. Moreover, the mentioned instruments have different sounds, even those belonging to the same group. Therefore, it can be concluded that the set used is well-diversified.

The sound of a live instrument is recorded in the form of *wav* files, which is a recording of not only individual notes but also the transitions between them.

The collection contains 21,571 audio recordings and is divided into three subsets by default:

- training (5841 samples),
- validation (3493 samples),
- test (12,237 samples).

The collection is not balanced—the number of samples for each instrument differs. The number of recordings of the tenor saxophone and trumpet is significantly smaller than for other instruments. The number of recordings of clarinet, electric guitar, and female voice is also significantly smaller compared to flute, piano, and violin.

Each recording is assigned an appropriate identifier and labels indicating the type of sample and the recorded instrument. The frequency of each recording is 22,050 Hz, and the length is 2972 milliseconds, which gives over 65,500 discrete values per recording.

3.2. Considered Features

The popularity of *MFCC* coefficients results from the multitude of possibilities they offer. First of all, they allow the extraction of a lot of different information from the signal, such as the timbre of the voice, the way the instrument is played, the type of instrument, or the musical genre. Therefore, this method is well suited for speech recognition systems, instruments, and other parameters related to music and sound. In the context of the topic of this work, it is important that they are a mathematical representation of the actual reception of sound by the listener. The disadvantage of the described algorithm is its high level of complexity—the path to obtaining the final matrix from the input signal is multi-stage and requires some experience in signal analysis. Moreover, *MFCC* coefficients are characterized by low resistance to noise [4].

Mel-cepstral parameters are great for extracting the necessary information from a sound signal, so instruments can be classified based on them. The choice of model and classifier remains crucial. While implementing this work, we decided to use artificial neural networks.

One of the basic phases of using artificial neural networks is data pre-processing. In this case, this means converting the audio file into a matrix of *MFCC* coefficients. An arti-

ficial neural network has a permanent structure, so to function properly, it must be ensured that the dimensions of the training, validation, and test data samples are the same. The use of the dataset (Section 3.1) ensures that the sound signals have the same waveform length. Otherwise, a procedure that would equalize the duration of all recordings should be implemented, preferably without losing valuable information. Due to the fact that the set used has recordings lasting 2972 milliseconds and the sampling frequency is 22,050 Hz, assuming the default and recommended value of the *hop length* parameter is 512, it is possible to calculate the width of the matrix taken at the network input. (Equations (1) and (2))

$$width\,of\ \ the\ \ matrix = \frac{sample\,rate \cdot duration}{hop\,length} \tag{1}$$

$$\frac{22{,}050 \cdot 2.972}{512} \approx 128 \tag{2}$$

The height of the mentioned matrix is equal to 13, i.e., the number of *MFCC* parameters extracted from the signal, which constitute a set representing frequency power in different ranges, and in this work, the full frequency spectrum is divided into 13 *MFCC*.

In summary, the input data fed to the first layer of the network have dimensions of 128×13.

4. Research Methodology

The initial process enabling conducting the research was the loading and appropriate processing of *wav* files and their gradual transformation until obtaining the *MFCC* coefficients, which serve as the model's input data.

The type of model, the optimal configuration of its parameters, and the degree of complexity were the essence of the tests performed. These elements are described in more detail in Section 5. Then, the model's decisions (musical instrument recognition results obtained on a given model) were subject to effectiveness assessment.

The model evaluation was based on several metrics:

- *Accuracy,* defining the ratio of the number of test samples that were correctly assigned to a given instrument to the number of all test samples.
- *Sensitivity—true positive rate*, that is a value showing what percentage of recordings of a specific instrument was correctly recognized in relation to the number of all recordings of only this instrument in the test set. This value is averaged for all instruments.
- *Precision—positive predictive value*, which expresses how many of the examples marked as a specific instrument turned out actually to be that instrument. This value is averaged for all instruments.
- *F1-score*, a measure of the harmonic mean (Equation (3)) of precision and sensitivity. The closer this value is to one, the better it indicates the model's performance. In the ideal case, when it takes the value 1, the analyzed model shows perfect sensitivity and precision.

$$F1\text{-}score = \frac{2 \cdot precision \cdot sensitivity}{precision + sensitivity} \tag{3}$$

- The value of the *loss function* of the neural network for each learning epoch, which should decrease throughout the entire learning process, after which this value should be appropriately low.
- A *confusion matrix*, which is very suitable for analyzing results in the context of multiclass problems such as the one discussed here. The matrix shows how many of the solutions proposed by the network turned out to coincide with real values, how many were misclassified, and how. Values represent the number of samples or the share (sometimes percentage) in a given row compared to all samples in a given class.

The neural network base models used in the work are rather light. This is due to the current choice of path to improve the results obtained. On the one hand, it is possible to expand the base model with the expectation of improving the results. On the other

hand, it is possible to use an ensemble containing relatively simpler base models. If the computational load is not taken into account, it would also be possible to build ensembles containing complex neural network structures. The current research focuses on creating an ensemble containing relatively light base models. Nevertheless, the basic model will be optimized through changes to its architecture.

5. Used Architectures and Experimental Results

This section presents the basic architectures of dense and convolutional neural networks, as well as conducted experiments and received results.

For the learning process of the used neural networks, the categorical cross-entropy was chosen as the loss function, and the optimization method was set to the *Adam* algorithm. The learning rate of the algorithm was set to 0.0001 (instead of the default value of 0.001), and the remaining parameters had default values ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-7}$). The network was trained for 50 epochs with a fixed batch size of 32.

5.1. Dense Neural Networks

The first type of model constructed during the research was a dense neural network, the simplified diagram of which is shown in Figure 1. The created *MFCCs* input data, which is matrix 128 × 13 at the beginning, needs to be flattened into one-dimensional data, which gives a vector of 1664 values (128 × 13). Such a vector of input data is processed by four consecutive layers, for which the weights of individual neurons will be set during the learning process. The last layer uses the activation function *softmax*, which causes the values of all outputs to sum to one. In practice, each output neuron symbolizes one of eight instruments, and their values indicate the probability of a given instrument appearing in the recording, determined by the model. The final decision takes the form of the index of the neuron with the highest value, i.e., the highest probability.

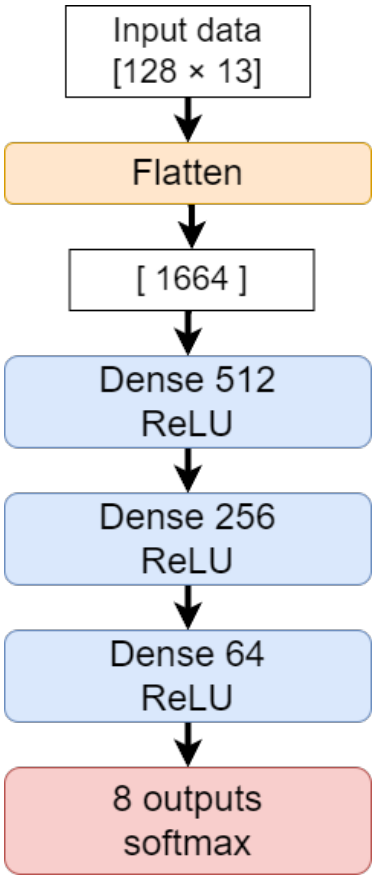

Figure 1. Dense network diagram.

This network architecture is marked as Model ANN1. The trained model correctly assigned 63% of the samples in the test set. This result is relatively satisfactory, but during training, the model very quickly achieved high accuracy and low error for the training data, but it failed to achieve similar results in relation to the validation data. Such dependence is a typical symptom of *overfitting*.

One way to counteract overfitting is the so-called *drop-out* [41]. Therefore, drop-out layers were added in the second experiment, and the *drop-out rate* value was selected through testing. Too small values caused the overfitting phenomenon, and too high values caused the network not to develop any patterns. The assessment of whether *overfitting* was eliminated was based on the analysis of the change in accuracy and value of the loss function during training and validation. The results are described in Table 2.

Table 2. The impact of the *drop-out* value on the efficiency of a dense network.

Drop-Out Rate	Overfitting	Accuracy	Sensitivity	Precision	F1-Score
0.05	Yes	0.62	0.60	0.61	0.60
0.1	No	0.65	**0.62**	**0.64**	**0.63**
0.2	No	0.60	0.60	0.61	0.60
0.3	No	0.62	0.57	0.59	0.58
0.4	No	**0.66**	0.50	0.49	0.49
0.5	No *	0.48	0.12	0.01	0.02
0.6	No *	0.22	0.12	0.01	0.02

For each evaluation parameter, the best result obtained is presented in bold. * When the value of *drop-out rate* was too high, the network weights became somewhat random because the network was not able to develop appropriate patterns. So this architecture did not even work for the training data.

The optimal value turned out to be a drop-out rate of 0.1, for which the accuracy of the network with respect to the test data was 65%, and F1-score was 63%. This network architecture is marked as Model ANN2 in Table 3.

Table 3. Efficiency of various dense neural networks.

Model	Normalization	Accuracy	Sensitivity	Precision	F1-Score
ANN1	No	0.63	0.61	0.61	0.61
ANN2	No	0.65	0.62	**0.64**	0.63
ANN3	Yes	0.65	**0.67**	0.63	**0.65**
ANN4	Yes	**0.68**	**0.67**	**0.64**	**0.65**

For each evaluation parameter, the best result obtained is presented in bold.

At this point, it is necessary to consider how the overall accuracy is distributed among individual classes, i.e., instruments. The confusion matrix is included in Figure 2. Analyzing this matrix, we can see that clarinet samples are incorrectly recognized, mainly as piano and violin; electric guitar samples are 90% correctly recognized; female voice samples are correctly recognized in 84%; flute samples are only in 34% correctly recognized, but 22% and 39% of samples are incorrectly recognized, mainly as piano and violin; piano samples are 99% correctly recognized; tenor saxophone samples are incorrectly recognized mainly as electric guitar but also as piano and violin; trumpet samples are incorrectly recognized mainly as violin; violin samples are in 88% correctly recognized. In summary, we can say that the model rarely indicates clarinet, saxophone, and trumpet. The highest effectiveness was recorded for singing, electric guitar, piano, and violin. The model shows good sensitivity for these classes, but only for female singing can good precision be noted. The reasons for this state were seen in the so-called *dying ReLU*.

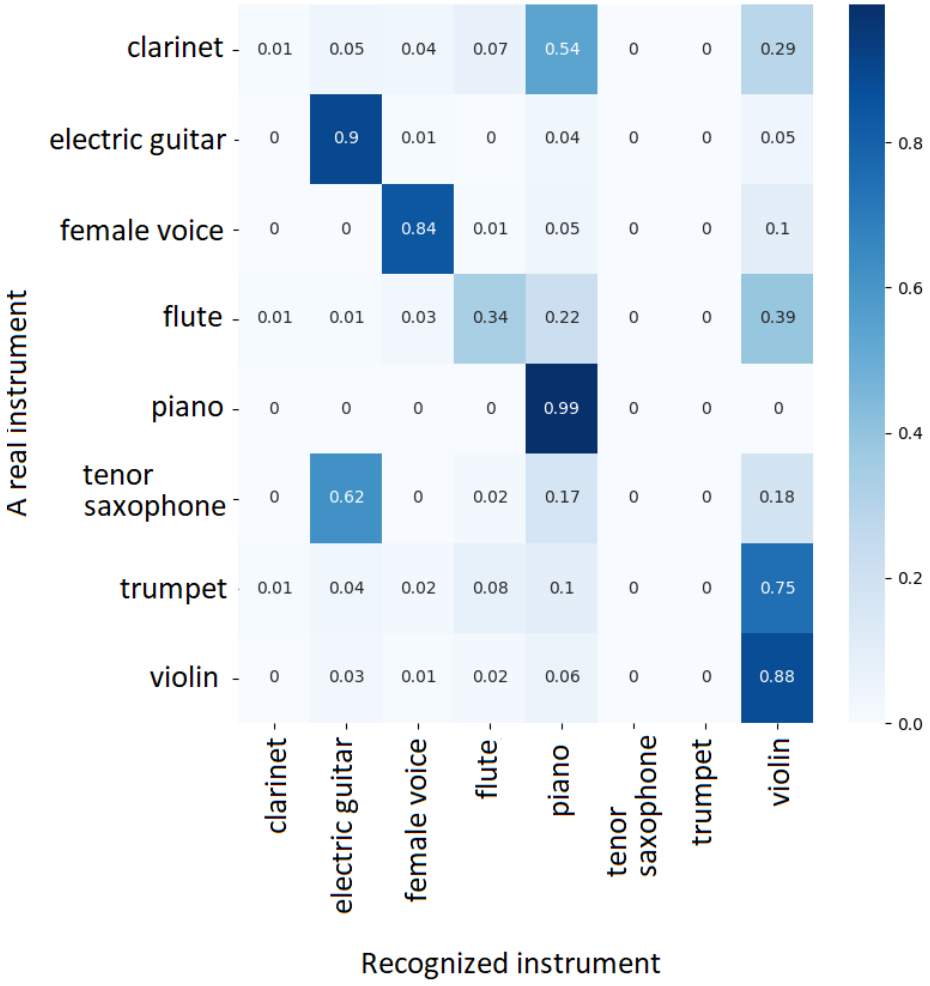

Figure 2. Confusion matrix (dense network, ANN1).

Dying *ReLU* appears when the sum of inputs for a large number of neurons has negative values. Considering its characteristics, the *ReLU* turns off such neurons by resetting their outputs. This results in the loss of valuable information encoded in negative values. The *MFCC* matrices obtained from the analyzed instruments may indeed contain negative values. Therefore, the *dying ReLU* problem may indeed concern the built model. One solution to this problem is to minimize or completely exclude negative values. This can be obtained, among others, by normalizing the data to the range 0–1 using the *min-max* method. Such normalization was performed separately for the *MFCC* coefficients of all rows, taking into account only the training set.

The previously described ANN2 Model was retrained this time based on the normalized set and was marked as the ANN3 Model in Table 3. The accuracy obtained was 65%, which is the same as in the previous case, but this time, it was spread over more instruments, indicating better model precision. The network maintained its tendency to recognize guitar, vocals, piano, and violin well and also improved its performance for saxophone, clarinet, and especially trumpet.

Another technique that is effective in solving the dying *ReLU* problem is to use the *LeakyReLU* [25] activation function. The test was performed again on a network with such an activation function and marked as Model ANN4 in the Table 3. The accuracy obtained was 68%, and the confusion matrix indicates improved performance—particularly for the violin.

Table 3 summarizes the quality of models ANN1 to ANN4.

5.2. Convolutional Neural Networks

In the next phase of experiments, research was carried out on the effectiveness of convolutional networks. Although mainly used in image processing, convolutional networks also work well in the audio field. The purpose of the tests was to check whether the *CNN* network would also be effective in the analyzed problem.

The convolutional networks used for testing were based on the architecture presented in Figure 3. The created *MFCCs* matrix 128 × 13 is an input data of evaluated convolutional neural networks. Such a matrix is processed by consecutive layers of three convolutional blocks, and then the data are flattened into one dimension so that it can be fed to the dense layers. The last layer contains 8 neurons according to the number of recognized instruments, and, as in the previous model, the sofmax activation function is used.

The first convolutional network that was built was marked as Model CNN1. Compared to the Figure 3 architecture, Model CNN1 does not have drop-out layers.

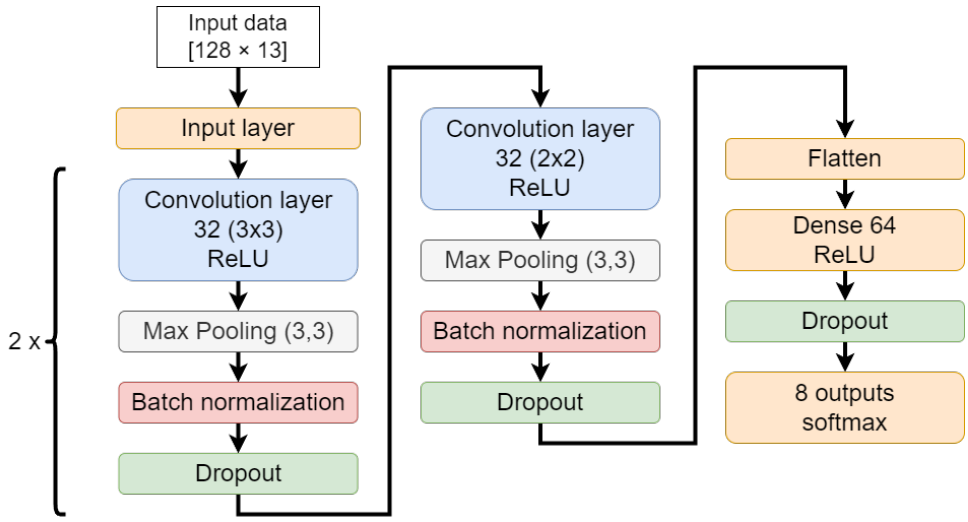

Figure 3. Diagram of a convolutional network.

It is worth paying attention to the *batch normalization*. This is a technique that allows to speed up the calculation process and improves the quality of the network. Knowing that data normalization brought the expected results in dense networks, it was decided to check whether it would be similar in the case of *CNN* networks using this technique. Unlike the normalization performed in the previous cases, *batch normalization* does not involve transforming the raw input data but normalizes the signals sent between the network layers. The model was trained using parameters with values similar to those used in the case of dense networks. The network with the given configuration was able to achieve an accuracy of 66%. The initial configuration of the *CNN* network, as in the case of dense networks, is characterized by overfitting.

The overfitting phenomenon was again partially eliminated by implementing *drop-out* (Figure 3). This model with the *drop-out rate* parameter set to 0.1 is marked as Model CNN2. However, it should be noted that this technique proved to be more effective for dense networks. The effectiveness of the network increased to 68%, and although precision for some instruments actually increased slightly, there was a significant drop in sensitivity for the clarinet.

The next set of tests was performed for different values of the drop-out parameter. The Table 4 confirms a relationship analogous to that occurring in dense networks: too high a *drop-out rate* value causes deterioration of results, although this time, the drop in effectiveness is not so drastic. Moreover, it should be noted that the optimal value of this parameter for *CNN* networks is slightly larger than for dense networks (0.2 and 0.1, respectively). Therefore, the network with a drop-out of 0.2 was marked as Model CNN3, for which the accuracy is 69%.

Table 4. The impact of the *drop-out* value on the efficiency of the convolutional network.

Drop-Out Rate	Accuracy	Sensitivity	Precision	F1-Score
0.1	0.68	**0.63**	0.58	0.60
0.2	**0.69**	0.61	**0.64**	**0.62**
0.3	0.65	0.59	0.59	0.59
0.4	0.61	0.51	0.57	0.54
0.5	0.56	0.47	0.53	0.50

For each evaluation parameter, the best result obtained is presented in bold.

Guided by the experience gained from previous tests, it was decided to use normalization to reduce overfitting and improve the results. There are already normalizing elements in the network structure, but they operate based on individual batches of data, not the entire set. The test was to see whether normalizing the entire batch along with *batch normalization* would improve the results or if it would lose some information by scaling too frequently.

This network architecture with a drop-out rate equal to 0.1 is marked as Model CNN4 in Table 5. The network again made 68% accurate predictions, overfitting was slightly reduced, and based on the confusion matrix, it can be concluded that it managed to modestly increase sensitivity in relation to the clarinet and saxophone (by about 20%).

A network analogous to CNN4 was also tested, but this time with a dropout of 0.2. This network architecture is marked as Model CNN5 in Table 5. The network made 64% of its predictions accurate.

The next step was to check whether, as in the previous tests, the *LeakyReLU* activation function would improve the results. The network was built according to the diagram in Figure 3, replacing the activation function *ReLU* with *LeakyReLU*, and the training process was started based on normalized input data. This network architecture is marked as Model CNN6 in Table 5. The obtained efficiency was 63%, and the confusion matrix indicates that the distribution of accuracy among individual instruments has not changed.

Table 5 summarizes the performance of various *CNN* network configurations.

Table 5. Efficiency of various convolutional neural networks.

Model	Norm. *	Drop-Out	Accuracy	Sensitivity	Precision	F1-Score
CNN1	No	0.1	0.66	0.66	0.62	0.64
CNN2	No	0.1	0.68	0.63	0.58	0.60
CNN3	No	0.2	**0.69**	0.61	**0.64**	0.62
CNN4	Yes	0.1	0.68	**0.67**	0.63	**0.65**
CNN5	Yes	0.2	0.64	0.63	0.58	0.60
CNN6	Yes	0.1	0.63	0.64	0.63	0.63

For each evaluation parameter, the best result obtained is presented in bold. * Normalization of input data.

Analyzing Table 5, it can be concluded that the CNN4 model has the highest F1-score efficiency. This network achieved an accuracy of 68% and a F1-score of 65%. The confusion matrix associated with the results of this model is similar to all other models tested to date. The networks achieve the best results for five classes: piano, singing, guitar, trumpet, and violin. The remaining three, i.e., clarinet, saxophone, and flute, are correctly assigned much less often—the accuracy for these instruments does not exceed 50%. The clarinet and flute are usually confused with the piano. The model more often defines these instruments as pianos than labels them correctly. However, the saxophone is accurately classified the least frequently for most of the tested models. The vast majority of saxophone recordings are labeled as electric guitars.

Similar results were achieved by the CNN3 model. It has the highest accuracy of all models at 69%, but the F1-score is 3% lower than the CNN4 Model.

Basically, the results achieved are satisfactory. Most well-configured models produce results in the range of 0.6–0.7 for both accuracy and F1-score. Taking into account only the group of five instruments mentioned above, the overall efficiency of the models would be much higher. Therefore, in order to significantly improve the current results, the way of classifying the clarinet, flute, and saxophone should be corrected.

6. Classifiers Committee

In order to obtain better results, it was decided to create the classifier committee. It is a set of models, in this case, neural networks, combined into one larger model. In ideal conditions, such a model takes advantage of the most effective properties of each of the base models, thus achieving a better result than each of the component models would achieve working alone.

6.1. Neural Classifier Committee

Initially, we decided to connect three *CNN* networks marked in the Table 5 with numbers 2, 3, and 4. The diagram of the constructed committee is shown in Figure 4. Three previously trained models (each with 8 outputs) were connected using a special *concatenate* layer. Once the committee is built, the possibility of training previously trained models is excluded. Later, the entire structure is trained for 100 epochs based on the same training and validation data that were used to calibrate all base models. The exclusion mentioned above of component models from training, as well as the process of additional training of the output layers (after concatenation), is intended to tune the final layers of the model to the signals sent to them from higher-order components in order to optimize the final generation of the classification result of 8 instruments.

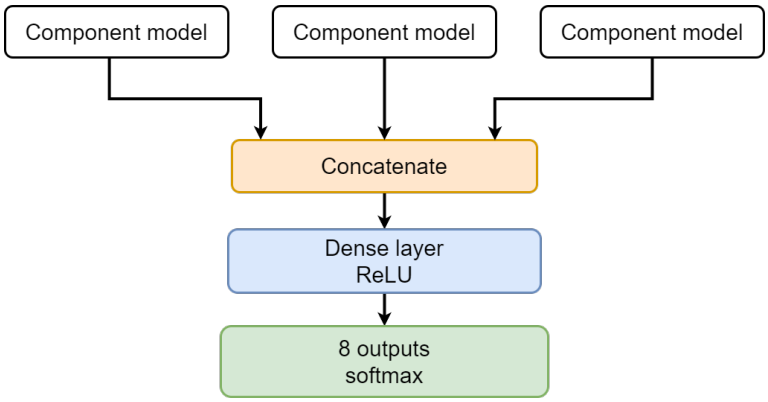

Figure 4. Model diagram—classifier committee.

Although, in theory, the described structure seems to have great potential, the results obtained thanks to the creation of the classifier committee are not impressive. We achieved 63% correct classifications, a sensitivity of 0.60, a precision of 0.58, and an F1-score of 0.59. Moreover, by analyzing the obtained confusion matrix, we could conclude that the problem regarding three inaccurately recognized instruments (clarinet, flute, and saxophone) had not been eliminated. Despite everything, the results still maintain a certain satisfactory level.

6.2. Committee of Voting Classifiers

In the further part of the tests, the model was rebuilt into the committee of voting classifiers. Its operating principle is simpler than the previous model. A few classifiers are first trained and then placed in a committee, where they analyze each subsequent sample of the test set and vote for the class to which the sample is to be assigned. In the event of a tie, when the number of votes cast for several classes is equal, the class chosen is the one voted for by the model that is the most confident in its decision, i.e., with the highest value

at the output of the neuron of the last layer of the network. The diagram of the committee based on three classifiers is shown in Figure 5.

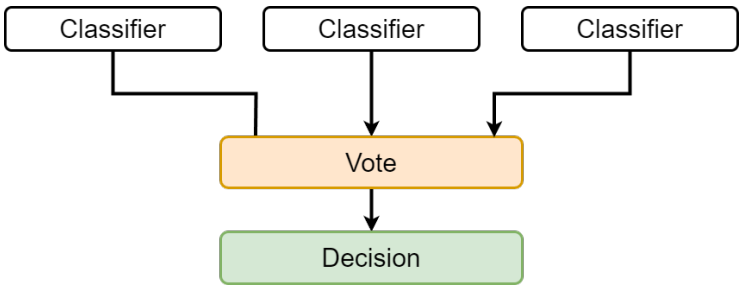

Figure 5. Diagram of the voting classifiers committee.

During the first tests, a committee was configured consisting of the *CNN* networks marked in Table 5 as CNN2, CNN3, and CNN4. Later, the committee was expanded to include one dense network marked in Table 3 as ANN4. The last voting committee model additionally included the ANN3 dense network.

The result of the tests performed is placed in Table 6. As can be seen, for the first time during the research, an accuracy of 70% was achieved. The values of other parameters are also at a good level, especially with regard to sensitivity. There is a noticeable tendency for some parameters to improve as the committee expands. However, in this case, these differences are small.

Table 6. Results of voting classifier committees.

Number of Models	Accuracy	Sensitivity	Precision	F1-Score
3	**0.70**	0.65	0.59	0.62
4	**0.70**	0.67	0.59	0.63
5	**0.70**	**0.68**	**0.60**	**0.64**

For each evaluation parameter, the best result obtained is presented in bold.

7. Conclusions and Future Work

The results obtained during the research can be considered satisfactory. What is good about the research is that as more and more advanced models were configured, the effectiveness often improved. Table 7 summarizes the performance of the different types of models tested. Figure 6 shows a comparison of accuracy parameter values for different methods.

Table 7. Results of different classifiers.

Model	Accuracy	Sensitivity	Precision	F1-Score
Dense network	0.68	0.67	0.64	0.65
CNN	0.68	0.67	0.63	0.65
Classifiers Committee	0.63	0.60	0.58	0.59
Voting Classifiers *	0.70	0.68	0.60	0.64

* Committee of Voting Classifiers.

As can be seen, the highest accuracy was achieved by using a committee of voting classifiers, but both dense networks and convolutional networks turned out to be good tools for recognizing musical instruments. It is worth noting that compared to the work [26] discussed in Section 2, the obtained accuracies of dense and convolutional networks are better by about 1%. However, when comparing the result obtained by the committee of voting classifiers to the work of [26], the improvement achieved is 3%. Compared to the

work [22], the accuracy is lower by 4%, and to [28] by about 6%. This may be due to the advanced sound pre-processing process based on the *CQT* transform, the use of Shepard sounds, and building very sophisticated deep architecture.

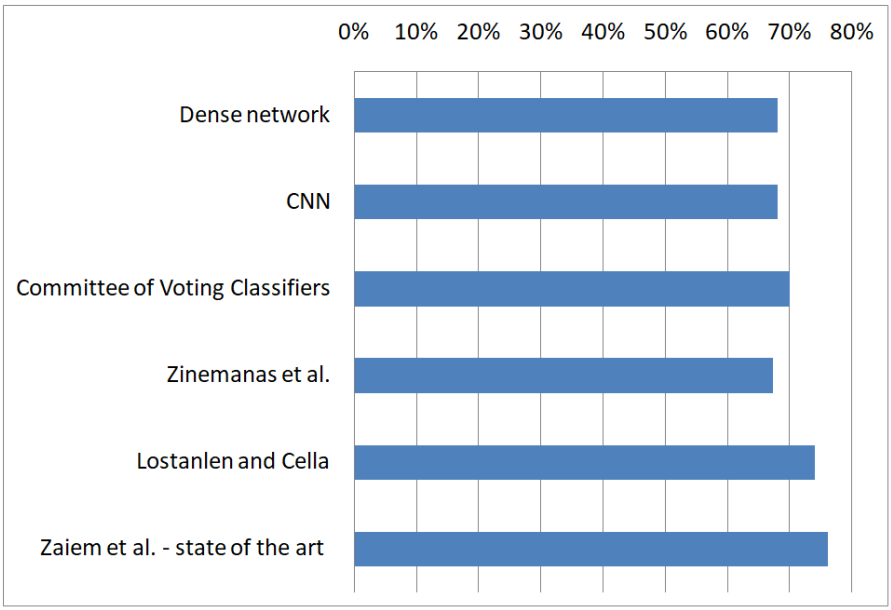

Figure 6. Comparison of accuracy parameter values for different methods (Zinemanas et al. [26], Lostanlen and Cella [22], Zaiem et al. [28]).

The sound of the selected instruments may influence the quality of instrument detection, so that instruments with a similar sound may be confused with each other—incorrectly classified. The analysis of the confusion matrix (Figure 2) shows just such a situation of confusing one instrument with one or two others with a similar sound characteristic. The imbalanced dataset also impacts the quality of instrument detection—instruments represented by a smaller number of recordings may be less well recognized due to insufficient learning of the appropriate classification by the model. A comparison of Table 1 'Number of recordings for each instrument' and the confusion matrix (Figure 2) shows just such a situation. Instruments with the largest number of recordings are classified with the best accuracy, and those with the smallest number are classified with the worst accuracy. The created model on some instruments (electric guitar, female voice, piano, violin) works much better than the average, which was significantly reduced by the results for those instruments for which there were a small number of recordings. For these four best classifiable instruments, we can see (confusion matrix—Figure 2) that 90% of electric guitar recordings, 84% of female voice recordings, 99% of piano recordings, and 88% of violin recordings were classified correctly.

Using an ensemble of classifiers not only improves the final result but also each basic classifier can be trained and improved/changed independently of the others. In this way, it is possible to improve the model by modifying its individual components.

Possible further development of work in the researched field could involve the use of a complex implementation technique in the work [22] consisting of transforming input signals using Shepard sounds. Another direction could be the introduction of an advanced algorithm for high-quality data augmentation, for example, the *random erasing* method proposed in the work [42], where its high effectiveness for convolutional neural networks was proven. Another way to improve the result would be to use more advanced neural network architectures. When creating voting committees, it would be possible to both increase the number of classifiers participating in voting and use more sophisticated voting methods.

The field of automatic instrument detection is a relatively new field of research. However, our research allows us to draw preliminary positive conclusions, and the solution has great potential for development.

Author Contributions: Conceptualization, D.K. and P.S.; methodology, D.K.; software, P.S.; validation, D.K., P.S. and R.B.; data curation, P.S.; writing—original draft preparation, D.K. and R.B.; writing—review and editing, D.K., R.B. and D.M.; supervision, D.K.; funding acquisition, D.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the ReActive Too project that has received funding from the European Union's Horizon 2020 Research, Innovation, and Staff Exchange Programme under the Marie Skłodowska-Curie Action (Grant Agreement No 871163), partially by a pro-quality grant for highly scored publications or issued patents (grant No 02/100/RGJ23/0026), Statutory Research funds of Department of Applied Informatics, Silesian University of Technology, Gliwice, Poland (grant No 02/100/BK_24/0035). Scientific work published as part of an international project co-financed by the program of the Polish Minister of Science and Higher Education entitled "PMW" in the years 2021–2025 (contract no. 5169/H2020/2020/2).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset Medley-solos-DB used in this research is available on: https://zenodo.org/record/3464194#.YXKfDLozZH4 (accessed on 4 June 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Eronen, A. *Automatic Musical Instrument Recognition*; Tampere University of Technology: Tampere, Finland, 2001.
2. Drobner, M. *Instrumentoznawstwo i Akustyka*; Polskie Wydawnictwo Muzyczne: Cracow, Poland, 1985.
3. Brożek, A. Filozofia nowej muzyki. *Semin. Sci.* **2011**, *1*, 10–20.
4. Davies, S.B.; Mermelstein, P. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Trans. Acoust. Speech Signal Process.* **1980**, *28*, 357–366. [CrossRef]
5. Han, W.; Chan, C.F.; Choy, C.S.; Pun, K.P. An Efficient MFCC Extraction Method in Speech Recognition. In Proceedings of the 2006 IEEE International Symposium on Circuits and Systems (ISCAS), Kos, Greece, 21–24 May 2006.
6. Bogert, B.P.; Ossanna, J. Computer Experimentation on Echo Detection, Using the Cepstrum and Pseudoautocovariance. *J. Acoust. Soc. Am.* **1966**, *39*, 1258–1259. [CrossRef]
7. Stern, R.M.; Acero, A. *Acoustical Pre-Processing for Robust Speech Recognition;* Technical Report; Carnegie-Mellon University Pittsburgh PA School of Computer Science: Pittsburgh, PA, USA, 1989.
8. Tzanetakis, G.; Cook, P. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.* **2002**, *10*, 293–302. [CrossRef]
9. Van den Oord, A.; Dieleman, S.; Schrauwen, B. Deep content-based music recommendation. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2643–2651.
10. Kostrzewa, D.; Ciszynski, M.; Brzeski, R. Evolvable hybrid ensembles for musical genre classification. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Boston, MA, USA, 9–13 July 2022; pp. 252–255.
11. Kostrzewa, D.; Mazur, W.; Brzeski, R. Wide Ensembles of Neural Networks in Music Genre Classification. In Proceedings of the Computational Science–ICCS 2022: 22nd International Conference, London, UK, 21–23 June 2022; Proceedings, Part II; Springer: Cham, Switzerland, 2022; pp. 64–71.
12. Sachdeva, N.; Gupta, K.; Pudi, V. Attentive neural architecture incorporating song features for music recommendation. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2–7 October 2018; pp. 417–421.
13. Aswale, S.P.; Shrivastava, P.C.; Bhagat, R.; Joshi, V.B.; Shende, S.M. Multilingual Indian Musical Type Classification. In Proceedings of the International Conference on VLSI, Communication and Signal processing, Prayagraj, India, 14–16 October 2022; Springer: Singapore, 2022; pp. 419–430.
14. Choudhury, N.; Deka, D.; Sarmah, S.; Sarma, P. Music Genre Classification Using Convolutional Neural Network. In Proceedings of the 2023 4th International Conference on Computing and Communication Systems (I3CS), Shillong, India, 16–18 March 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5.
15. Le Thuy, D.T.; Van Loan, T.; Thanh, C.B.; Cuong, N.H. Music Genre Classification Using DenseNet and Data Augmentation. *Comput. Syst. Sci. Eng.* **2023**, *47*, 657. [CrossRef]
16. Xu, Z.; Feng, Y.; Song, S.; Xu, Y.; Wang, R.; Zhang, L.; Liu, J. Research on Music Genre Classification Based on Residual Network. In Proceedings of the International Conference on Mobile Computing, Applications, and Services, Messina, Italy, 17–18 November 2022; Springer: Cham, Switzerland, 2022; pp. 209–223.

17. Kostrzewa, D.; Chrobak, J.; Brzeski, R. Attributes Relevance in Content-Based Music Recommendation System. *Appl. Sci.* **2024**, *14*, 855. [CrossRef]
18. Pons, J.; Serra, X. Randomly weighted cnns for (music) audio classification. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 336–340.
19. Vall, A.; Dorfer, M.; Eghbal-Zadeh, H.; Schedl, M.; Burjorjee, K.; Widmer, G. Feature-combination hybrid recommender systems for automated music playlist continuation. *User Model. User-Adapt. Interact.* **2019**, *29*, 527–572. [CrossRef]
20. Zhang, Y. Music recommendation system and recommendation model based on convolutional neural network. *Mob. Inf. Syst.* **2022**, *2022*, 3387598. [CrossRef]
21. Elbir, A.; Aydin, N. Music genre classification and music recommendation by using deep learning. *Electron. Lett.* **2020**, *56*, 627–629. [CrossRef]
22. Lostanlen, V.; Cella, C.E. Deep convolutional networks on the pitch spiral for music instrument recognition. In Proceedings of the 17th ISMIR Conference, New York, NY, USA, 7–11 August 2016.
23. Lagrange, M.; Gontier, F. Bandwidth extension of musical audio signals with no side information using dilated convolutional neural networks. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: Piscataway, NJ, USA, 2020.
24. Vincent, L.; Andén, J.; Lagrange, M. Extended playing techniques: The next milestone in musical instrument recognition. In Proceedings of the 5th International Conference on Digital Libraries for Musicology, Paris, France, 28 September 2018.
25. Bai, Y. RELU-function and derived function review. In Proceedings of the SHS Web of Conferences, Dali, China, 13–15 May 2022; EDP Sciences: Jules, France, 2022; Volume 144, p. 02006.
26. Zinemanas, P.; Rocamora, M.; Miron, M.; Font, F.; Serra, X. An interpretable deep learning model for automatic sound classification. *Electronics* **2021**, *10*, 850. [CrossRef]
27. Dubey, S.S.; Hanamshet, V.V.; Patil, M.D.; Dhongade, D.V. Music Instrument Recognition Using Deep Learning. In Proceedings of the 2023 6th International Conference on Advances in Science and Technology (ICAST), Mumbai, India, 8–9 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 63–68.
28. Zaiem, S.; Parcollet, T.; Essid, S.; Heba, A. Pretext tasks selection for multitask self-supervised audio representation learning. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 1439–1453. [CrossRef]
29. Blaszke, M.; Kostek, B. Musical instrument identification using deep learning approach. *Sensors* **2022**, *22*, 3033. [CrossRef]
30. Han, Y.; Kim, J.; Lee, K. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *25*, 208–221. [CrossRef]
31. Avramidis, K.; Kratimenos, A.; Garoufis, C.; Zlatintsi, A.; Maragos, P. Deep convolutional and recurrent networks for polyphonic instrument classification from monophonic raw audio waveforms. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 3010–3014.
32. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [CrossRef]
33. Zhang, L.; Lim, C.P.; Yu, Y.; Jiang, M. Sound classification using evolving ensemble models and Particle Swarm Optimization. *Appl. Soft Comput.* **2022**, *116*, 108322. [CrossRef]
34. Mohammed, A.; Kora, R. An effective ensemble deep learning framework for text classification. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 8825–8837. [CrossRef]
35. Abro, A.A. Vote-based: Ensemble approach. *Sak. Univ. J. Sci.* **2021**, *25*, 858–866. [CrossRef]
36. Nanni, L.; Maguolo, G.; Brahnam, S.; Paci, M. An ensemble of convolutional neural networks for audio classification. *Appl. Sci.* **2021**, *11*, 5796. [CrossRef]
37. Onan, A.; Korukoğlu, S.; Bulut, H. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Syst. Appl.* **2016**, *62*, 1–16. [CrossRef]
38. Nanni, L.; Costa, Y.M.; Lumini, A.; Kim, M.Y.; Baek, S.R. Combining visual and acoustic features for music genre classification. *Expert Syst. Appl.* **2016**, *45*, 108–117. [CrossRef]
39. Bahuleyan, H. Music genre classification using machine learning techniques. *arXiv* **2018**, arXiv:1804.01149.
40. Lostanlen, V.; Cella, C.E.; Bittner, R.; Essid, S. *Medley-solos-DB: A Crosscollection Dataset for Musical Instrument Recognition*; Zenodo: Meyrin, Switzerland, 2018.
41. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
42. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random Erasing Data Augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.