Indonesian Journal of Electrical Engineering and Computer Science
Vol. 43, No. 1, July 2026, pp. 114~126
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v43.i1.pp114-126

A hybrid CNN-autoencoder-SVM/XGBoost model for polyphonic orchestral instrument classification

Kelvin Wyeth[1], Iman Herwidiana Kartowisastro[1,2]
[1]Department of Computer Science, BINUS Graduate Program, Bina Nusantara University, Jakarta, Indonesia
[2]Department of Computer Engineering, Faculty of Engineering, Bina Nusantara University, Jakarta, Indonesia

Article Info

Article history:

Received Mar 20, 2026
Revised Jun 22, 2026
Accepted Jun 29, 2026

Keywords:

Autoencoder
CNN information retrieval
Music
Orchestral instrument classification
polyphonic audio
SVM
XGBoost

ABSTRACT

Polyphonic orchestral recordings pose significant challenges in music information retrieval (MIR) due to their overlapping frequency ranges and timbral similarities among instrument families, which complicate multi-label instrument classification. Prior studies have explored the integration of convolutional neural networks (CNN)-based feature extraction with classical machine learning (ML) classifiers, often on monophonic or simpler datasets like IRMAS. But the integration of deep learning (DL) feature extraction, Autoencoder (AE)-based dimensionality reduction, and ML classifiers for polyphonic orchestral instrument recognition remains underexplored. This study proposes a hybrid framework utilizing a pre-trained Inception V3 CNN for feature extraction from mel-spectrograms, followed by an optional 50% dimensionality reduction via AE, and finally, classification with support vector machines (SVM) or extreme gradient boosting (XGBoost). Experiments were run on two polyphonic datasets, OpenMIC-2018 and Orchset. The results demonstrate that non-AE configurations generally outperform AE variants. These results extend prior studies such using polyphonic datasets. The results highlight the practical value of hybrid CNN-ML pipelines and the trade-offs of feature compression in MIR.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Kelvin Wyeth
Computer Science Department, Master of Computer Science, Bina Nusantara University
Jakarta, Indonesia
Email: kelvin.w500@gmail.com

1. INTRODUCTION

Machine learning (ML) has played a vital role in audio analysis for decades. During this time, music information retrieval (MIR) and audio signal processing have evolved into crucial research areas that bridge artificial intelligence (AI) with the understanding of music. Through ML, information such as genre, mood, and instrument activity can be automatically extracted from audio signals [1]-[4]. Among these, the task of automatic musical instrument recognition (AIR) has gained increasing attention. Identifying which instruments are present in a musical excerpt supports applications such as music indexing, content-based retrieval, music recommendation systems, and intelligent music production [5].

There are several challenges in musical instrument recognition. Instrument recognition on monophonic or monotimbral recordings has achieved strong performance in previous studies. However, instrument recognition in polyphonic music, where multiple instruments play simultaneously, remains highly challenging, even for predominant-instrument recognition [6]. Orchestral performances and recordings typically feature instruments of different families such as strings, woodwinds, brass, and percussion [7]. As a result, instrument boundaries become less distinguishable than in monophonic recordings, making reliable multi-label instrument recognition substantially more challenging [2], [5].

Early studies in the field of MIR relied heavily on traditional ML algorithms to classify instruments, such as support vector machines (SVM) and k-Nearest Neighbors (kNN), which were trained on handcrafted features such as mel-frequency cepstral coefficients (MFCC) or Sonograms. Those techniques achieved high classification accuracy on monophonic or clean datasets, like IRMAS [1]. The emergence of deep learning (DL), particularly convolutional neural networks (CNNs), has significantly changed how musical information is processed. CNNs allow direct learning from spectrograms or even raw audio [8]. Pre-trained CNNs, such as VGGish, ResNet, and Inception V3, initially used in image classification, are now used in audio analysis. Since audio spectrograms share structural characteristics with images, CNNs have been shown to learn discriminative features from spectrogram-based audio representations [9]. Several studies have also explored hybrid approaches that combine DL to extract optimal audio features, which are then classified with traditional ML classifiers. Such combinations have demonstrated promising performance in several studies [2]. Another important development is the use of autoencoders (AEs) in feature reduction. AEs transform high-dimensional representations into compact latent features, reducing redundancy while preserving important information from the original feature representation [10], [11].

Those previous studies have reported promising results using CNN-based architectures and hybrid CNN-ML frameworks. Although AEs have been widely used for dimensionality reduction, it remains unclear whether compressing CNN-derived feature representations preserves sufficient discriminative information for multi-label polyphonic instrument recognition. Furthermore, existing studies often focus on monophonic recordings, single-label classification, or general music classification tasks rather than polyphonic orchestral environments where multiple instrument families overlap simultaneously. Consequently, the effectiveness of combining CNN feature extraction, AE-based dimensionality reduction, and ML classifiers for multi-label orchestral instrument recognition remains underexplored.

To address this gap, the study proposes a hybrid framework for identifying orchestral instruments in polyphonic music. This framework uses a pre-trained Inception V3 CNN for feature extraction, a fully connected AE for feature compression, and two ML classifiers, which are SVM and extreme gradient boosting (XGBoost), for multi-label classification. This framework aims to combine the representational strength of DL with the classification capabilities of traditional ML algorithms.

The research seeks to answer the following question: Can AE-based dimensionality reduction improve the effectiveness of CNN-derived features for multi-label orchestral instrument recognition using ML classifiers? To investigate this, we evaluate the model with two open datasets. The two datasets were selected to represent different recording conditions, with OpenMIC-2018 containing diverse real-world music recordings and Orchset focusing on polyphonic orchestral excerpts. Comparisons are made between configurations that use AE-based feature reduction and those that do not to assess the effect of feature reduction on classification performance.

The main contributions of this study are threefold. First, this study systematically evaluates a hybrid CNN-AE-ML framework for multi-label orchestral instrument recognition in polyphonic music containing overlapping instrument families. Second, the study evaluates the effect of AE-based dimensionality reduction on the performance of multi-label classification of orchestral instruments. Third, comparing and evaluating OpenMIC-2018 and Orchset datasets provides additional insight into the strengths and limitations of compressed and non-compressed deep feature representations in complex orchestral MIR scenarios.

2. THEORETICAL BASIS

Early studies in MIR primarily relied on handcrafted audio features such as MFCCs, Chroma, and Spectral Contrast, which were subsequently classified using traditional ML algorithms such as SVMs, kNN, and Random Forests [1], [12], [13]. The emergence of DL, particularly CNNs and AEs, enabled hierarchical feature learning directly from audio representations, leading to substantial improvements in audio classification performance [14], [15].

With the emergence of DL, CNN-based models made it possible to extract features directly from time-frequency representations such as spectrograms and Mel-spectrograms. Taenzer *et al.* [8] state that a commonality among DL approaches for AIR is summarized in three identical phases: pre-processing, embedding, and classification. The pre-processing phase involves transforming the music waveform input into a simplified signal representation, like a spectrogram. The embedding phase then takes the pre-processing output and converts it into a feature representation for classification. The classification phase will then detect and classify the instruments present in the feature representation. These models can learn timbral and temporal relationships more effectively than hand-crafted feature methods, leading to significant accuracy improvements in instrument recognition, genre classification, and music tagging tasks.

Chen *et al.* [16] investigated the interpretability of Convolutional Neural Networks for musical instrument recognition by analyzing feature-map activations derived from different spectrogram representations, including log-Mel, MFCC, and STFT. Their study demonstrated how variations in

spectrogram type influence kernel responses and class separability, offering insight into how CNNs learn timbral and spectral distinctions among instruments. This work emphasizes the importance of input representation design, which supports the use of Mel-spectrograms as a perceptually meaningful feature domain in an audio-based classification task. Additionally, Wang [17] presented a DL approach for music performance evaluation that relied on spectral audio representations, further illustrating the applicability of spectral audio representations in music-related ML tasks. Recent research has also demonstrated the effectiveness of transfer learning in MIR. Pre-trained CNN architectures originally designed for image classification, such as VGGish, ResNet, and GoogleNet (Inception V3), have been successfully adapted for audio analysis. Tsalera *et al.* [9] compared multiple pre-trained CNNs for audio classification and found that both GoogleNet and VGGish performed competitively when used as frozen feature extractors on small-scale audio datasets.

The aforementioned studies demonstrate that spectrogram-based representations and transfer-learning strategies enable CNNs to learn discriminative timbral and spectral patterns from musical audio. However, as it was stated before, most existing evaluations focus on monophonic recordings, predominant-instrument recognition, or general audio-classification tasks, leaving questions regarding the effectiveness of transfer-learned CNN representations in polyphonic orchestral environments where multiple instrument families overlap simultaneously.

Several studies have explored combining CNNs with classical ML classifiers to balance performance and computational efficiency. Prabavathy *et al.* [1] explored the use of SVM and k-Nearest Neighbors (kNN) classifiers trained on MFCC and Sonogram features, reporting an accuracy of 99.29% for monophonic recordings. The authors suggested that future work should explore DL approaches. Building on this foundation, the same authors later incorporated GoogleNet as a deep feature extractor before applying SVM and kNN classification on the IRMAS dataset, achieving a slightly higher accuracy of 99.37%, and the authors further suggested the use of pre-trained CNN architectures and a broader range of instrument classes in future studies [2]. Li *et al.* [18] developed a multi-scale deep feature fusion approach for birdsong classification that combines convolutional feature extractors with traditional ML classifiers. Their experiments demonstrate that integrating hierarchical CNN features with tree-based and kernel-based models can improve generalization across diverse acoustic conditions. Similarly, Ashraf *et al.* [19] introduced a hybrid CNN-RNN variant model for music classification, capturing both spectral and temporal characteristics of musical signals. Their findings demonstrated that combining convolutional and recurrent layers improves classification accuracy over pure CNN baselines, indicating the advantages of hybrid feature learning.

In another study, Gan *et al.* [20] proposed a hybrid pipeline for music genre classification that integrates variational mode decomposition (VMD) and an improved whale optimization algorithm (IWOA) for feature selection, followed by an XGBoost classifier for final prediction. Their method achieved superior accuracy compared to conventional neural networks, demonstrating that gradient-boosted decision-tree ensembles such as XGBoost can effectively complement feature-engineering approaches. These findings support the use of XGBoost as a classifier within hybrid deep-feature learning pipelines. Liu *et al.* [3] proposed an instrument-recognition framework that fuses multiple audio features, which include MFCCs, chroma, and spectral contrast, and classifies them with XGBoost. Their results show that ensemble gradient-boosting models outperform several deep-learning baselines on structured, tabular feature representations. The studies suggest that XGBoost can serve as an effective downstream classifier for structured feature representations and may therefore be suitable for use within hybrid deep-feature learning pipelines.

Collectively, these studies suggest that combining deep feature extraction with traditional ML classifiers can provide strong classification performance while potentially reducing training complexity compared with fully end-to-end DL models. Nevertheless, most studies evaluate hybrid architectures on monophonic recordings, genre classification tasks, or non-orchestral audio datasets, limiting understanding of their effectiveness for polyphonic orchestral instrument recognition.

Besides DL and ML combinations, researchers have increasingly focused on dimensionality reduction to improve model generalization and reduce computational demands. AEs, which are unsupervised neural networks designed to learn compact latent representations, have become a popular choice for this purpose. Berahmand *et al.* [10] provide a detailed survey of AE architectures, emphasizing their ability to enhance feature efficiency and minimize overfitting in high-dimensional data. Pasini *et al.* [21] introduced Music2Latent, a consistency AE framework designed for compact audio representation learning. Their model enforces latent-space consistency between input and reconstruction, enabling efficient audio compression without significant loss of perceptual quality. The study highlights the ability of AEs to capture salient spectral-temporal features in musical data, reinforcing their suitability for dimensionality reduction in hybrid learning architectures. In another AE-based study, Nellas *et al.* [11] demonstrated that Convolutional AEs could achieve good classification results, even with substantially compressed feature sets. AEs have been applied to MIR problems, such as music genre classification and audio denoising [22]. A related study by

Kumar *et al.* [23] uses spectrogram and chroma representations and puts them into a stacked AE before using a softmax classifier for classification. It substantially improves accuracy over conventional approaches.

Xu *et al.* [24] conducted a comparative evaluation of several multi-label classification methods on real-world datasets. Their findings highlight the importance of classifier selection in multi-label learning scenarios, which is relevant to polyphonic instrument recognition where multiple instrument labels may occur simultaneously.

Other recent studies have continued to explore DL architectures for musical instrument detection. Kostrzewa *et al.* [25] conducted experiments to detect musical instruments present in an audio signal. Their study used MFCC features and compared several neural network architectures, including a dense neural network, a CNN, a neural classifier committee, and voting classifiers. The experiments were carried out on the Medley-solos-DB dataset, which contains over 20,000 solo recordings of individual instruments. The study achieved good results on monophonic recordings of solo instruments. However, recognizing instruments in polyphonic orchestral recordings remains challenging, motivating the hybrid CNN-AE-Classifier approach proposed in this study.

Sechet *et al.* [26] proposed a hierarchical DL framework for detecting minority instruments in polyphonic orchestral recordings. Their model introduces specialized sub-classifiers to address class imbalance, which is prevalent in datasets where dominant instruments such as strings overshadow less-represented ones like brass or woodwinds. The study achieved improved recognition of minority classes through hierarchical feature learning and adaptive weighting strategies. The research shows the importance of addressing class imbalance when evaluating polyphonic orchestral instrument recognition systems, particularly when minority instrument families are underrepresented.

In summary, previous studies demonstrate the effectiveness of CNN-based feature extraction, AE-based dimensionality reduction, and ML classifiers when applied individually or within hybrid frameworks. These methods were selected because they provide complementary strengths: CNNs enable hierarchical feature extraction, AEs reduce feature redundancy through latent-space compression, SVMs perform effectively in high-dimensional feature spaces, and XGBoost offers robust ensemble-based classification capabilities.

However, limited evidence exists regarding whether AE-compressed CNN embeddings improve multi-label instrument recognition in polyphonic orchestral recordings, particularly across datasets with differing acoustic characteristics. Consequently, the practical benefits and limitations of dimensionality reduction within hybrid CNN-ML architectures remain insufficiently understood. This gap motivates the proposed investigation using OpenMIC-2018 and Orchset as complementary evaluation datasets.

3. METHOD

3.1. Overview of the proposed system

The proposed system is a CNN-AE-ML architecture. The proposed system consists of five stages: audio preprocessing and Mel-spectrogram generation, feature extraction with a pre-trained CNN (Inception v3), dimensionality reduction with an AE, classification using SVM and XGBoost, and finally, performance evaluation using standard metrics, namely Subset Accuracy, F1-Score (Micro and Macro), ROC-AUC (Micro and Macro), and mean average precision (mAP). They are standard metrics for comparing multi-label methods across various domains [27]. Additionally, confusion matrices were used to gauge the performance of the classifier by comparing the predicted labels to the actual labels [28]. F1-Score bar plots were used to visually compare F1-Scores for each label. The research framework can be seen in Figure 1.

3.2. Datasets

The proposed system was evaluated using two publicly available polyphonic audio datasets: OpenMIC-2018 and Orchset, both accessible through the Zenodo repository. The Orchset dataset was developed by Bosch *et al.* [29] for the study of dominant melodic instrument identification in orchestral recordings. It contains 64 excerpts taken from classical music made by famed composers, such as Beethoven, Dvořák, and Haydn. The annotations classify the dominant instrument groups into three groups: strings, winds, and brass. Because of its emphasis on polyphonic orchestral textures and realistic ensemble timbres, Orchset provides a strong benchmark for evaluating classification models under controlled and musically authentic conditions.

Meanwhile, the OpenMIC-2018 dataset, developed by Humphrey *et al.* [30], comprises approximately 20,000 short audio clips in .ogg format. Each clip, taken from music from various genres, is annotated for the presence of one or more instruments, covering a broad range of genres and recording environments. With 20 instrument labels, OpenMIC-2018 is widely regarded in the MIR community as a standard benchmark for multi-label instrument recognition. But, it must go through a filtering process, in

which only instruments corresponding to orchestral string, wind, and brass families were retained for this study. Together, these two datasets enable a comprehensive evaluation of the proposed system.

For all classification experiments, the dataset was divided into training and testing subsets using an 80:20 split. Hyperparameter optimization was performed exclusively on the training subset using GridSearchCV with cross-validation, while the held-out testing subset was used only for final performance evaluation. A fixed random seed of 42 was used throughout dataset splitting and model training to ensure reproducibility.

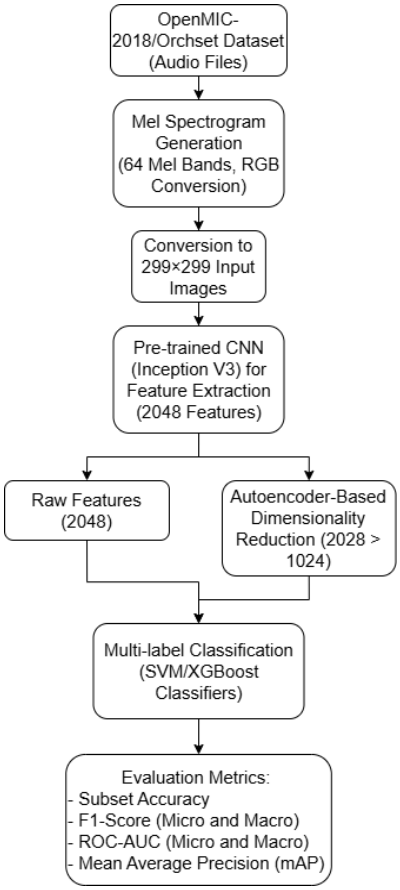

Figure 1. Research framework of the proposed hybrid CNN-AE-classifier model for orchestral instrument classification using the orchset and OpenMIC-2018 datasets

3.3. Implementation

The implementation follows a pipeline consisting of four main stages: audio preprocessing, feature extraction using a pre-trained Inception V3 CNN, feature compression using a fully connected AE, and classification using either SVM or XGBoost. The experiment was executed in a Python virtual environment, utilizing dedicated Jupyter Notebooks for each dataset during the preprocessing and feature extraction stages to maintain data separation. In the first stage, audio signals from both datasets are transformed into Mel-spectrogram representations. Mel-spectrograms were generated using the Librosa library, which internally applies a short-time fourier transform (STFT) followed by a Mel filter bank projection to convert frequency components into a perceptually relevant scale as shown in Figures 2 and 3. Two separate notebooks were used for this stage, one for each dataset.

In the second stage, the features of the mel-spectrograms are extracted with a pre-trained Inception V3 CNN model. The output consists of 2048-dimensional feature vectors representing the spectral and temporal characteristics of each Mel-spectrogram. During this step, the OpenMIC-2018 dataset is filtered. Only orchestral instruments are kept and labelled with strings, winds, and brass. The filtering ensures consistency with Orchset's labelling. Similarly, two separate notebooks were used for this stage, one for each dataset.

In the third stage, the extracted features pass through an AE. The AE compresses the 2048-dimensional vectors into 1024-dimensional latent representations. The AE is trained separately for each dataset using the Adam optimizer with mean squared error (MSE) loss, providing a reduced yet discriminative feature space for classification.

The fourth and final stage is classification with either SVM or XGBoost. Each classifier is trained and optimized via GridSearchCV to determine the best hyperparameters. Two configurations are evaluated: CNN+Classifier, which uses CNN features, and CNN+AE+Classifier, which uses AE-compressed features. Classification results are evaluated with multi-label metrics, such as F1-score, Subset Accuracy, mAP, and ROC-AUC, along with confusion matrices and F1-score bar plots for qualitative comparison.

3.3.1. Audio preprocessing into mel-spectrograms

All audio files from both datasets are first converted to mono and resampled to 16 kHz to maintain a consistent sampling rate. The waveforms are then transformed into Mel-spectrograms using the Python library, Librosa. This transformation captures patterns of frequency and time that are perceptually meaningful to human hearing. The Mel-spectrograms were generated using 64 Mel bands, a maximum frequency of 8 kHz, and a hop length of 256 samples. The resulting spectrograms were converted to the decibel scale using Librosa's power_to_db function. The resulting Mel-spectrograms are stored as NumPy (.npy) arrays to simplify loading and computation in the next stages.

To meet the input requirements of the pre-trained CNN model, the Inception V3, each spectrogram is reshaped into a three-channel RGB image and resized to 299×299 pixels. These steps ensure uniformity between datasets and help the CNN focus on meaningful frequency relationships rather than irrelevant variations in loudness or scale.

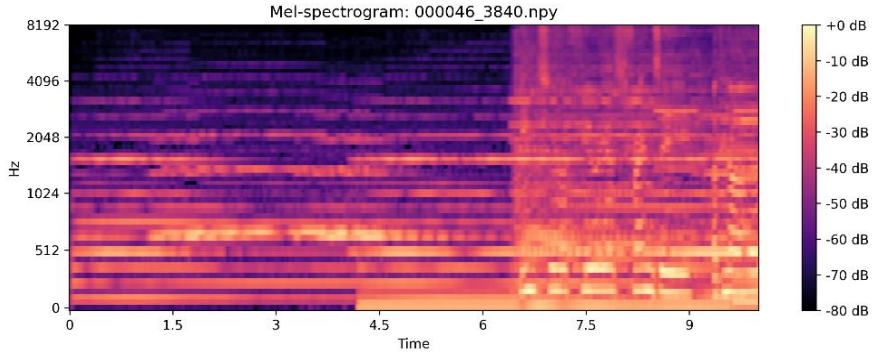

Figure 2. Mel-Spectrogram sample from OpenMIC-2018

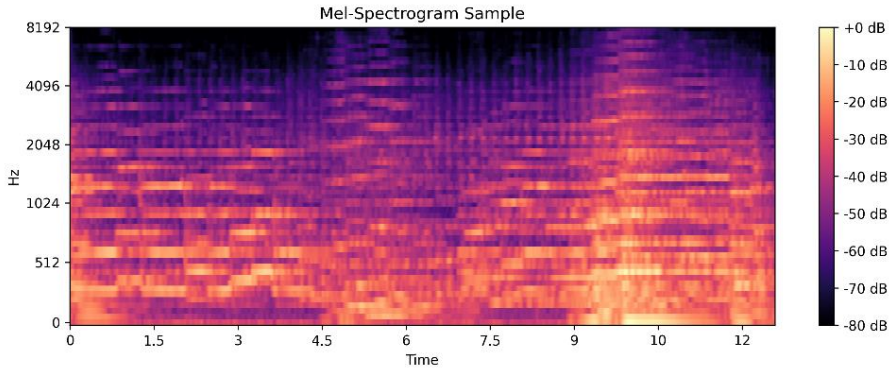

Figure 3. Mel-spectrogram sample from orchset

3.3.2. OpenMIC-2018 Filtering

The OpenMIC-2018 dataset requires an additional filtering stage so that it matches the orchestral taxonomy used in Orchset. The original dataset includes 20 instrument classes, many of which are not typically found in an orchestra. During the filtering process, instruments corresponding to the three target

orchestral families were retained. Violin, cello, and bass were grouped as strings; flute and clarinet were grouped as winds; trumpet and trombone were grouped as brass. Instrument classes that did not belong to the target orchestral families were excluded from the experiment.

After the OpenMIC-2018 filtering process, 4,418 clips remain from the original 20,000. Each sample is labeled according to one or more of the orchestral families, following a multi-label format that parallels the structure of the Orchset dataset. This alignment makes cross-dataset evaluation possible under similar class definitions and recording conditions. Despite that, the resulting label distribution remained imbalanced across instrument families. String instruments appeared more frequently than brass and wind instruments. The original class distribution was retained to preserve the datasets' realistic characteristics and to evaluate model performance under naturally imbalanced conditions. Class imbalance was partially addressed through hyperparameter optimization of the SVM classifier, which included evaluation of different class-weight configurations within GridSearchCV later on. The class-weight parameter allows greater emphasis to be assigned to underrepresented classes during training. No additional resampling techniques, such as oversampling or synthetic sample generation, were applied. The label distributions for each dataset by the end of the process are shown in Figures 4 and 5.

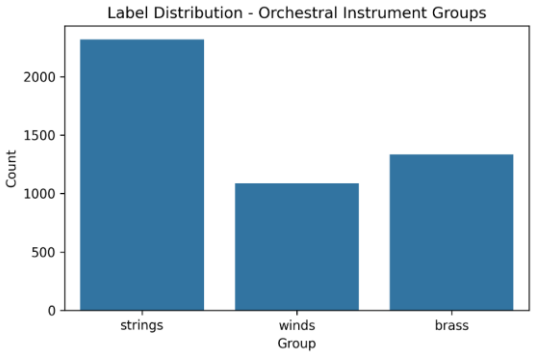

Figure 4. Label distribution bar plot for OpenMIC-2018

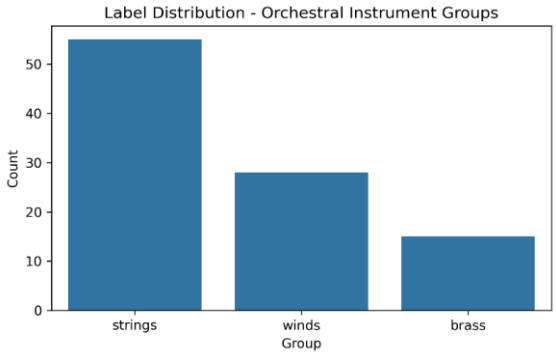

Figure 5. Label distribution bar plot for orchset

3.3.3. Feature Extraction with CNN (Inception V3)

Feature extraction is performed using Inception V3 pre-trained on ImageNet. In this study, the network is used as a fixed feature extractor, meaning that no fine-tuning is performed and all pre-trained weights remain unchanged. The final classification layer is removed and replaced with an identity layer, allowing activations from the penultimate layer to be extracted as 2048-dimensional feature vectors. These vectors serve as compact representations of the Mel-spectrograms and are subsequently used for dimensionality reduction and classification. The pre-trained Inception V3 model contained 25,112,264 parameters. The feature extractor was pre-trained and used only for forward inference. No fine-tuning of Inception V3 was performed. This reduces computational cost compared to training the network from scratch while retaining the benefits of transfer learning.

3.3.4. Dimensionality Reduction with AE

To reduce feature redundancy and computational complexity, a fully connected AE was implemented in PyTorch. The AE consists of an encoder and a decoder network. The encoder compresses the 2048-dimensional CNN feature vector into a 1024-dimensional latent representation through an intermediate hidden layer containing 1536 neurons. The decoder mirrors this structure by reconstructing the latent representation back to the original feature dimension. The latent dimension was set to 1024, corresponding to a 50% reduction from the original 2048-dimensional CNN feature space. This configuration was selected as a practical compromise intended to reduce feature dimensionality while retaining sufficient information for downstream classification. More aggressive compression may increase information loss, whereas smaller reductions provide limited computational benefits. AEs were selected because they learn nonlinear feature representations, while conventional dimensionality-reduction methods such as PCA, clustering-based reduction, or wavelet-based approaches may not preserve complex relationships within high-dimensional CNN embeddings.

ReLU activation functions are applied after each hidden layer. The model is trained for 100 epochs using the Adam optimizer with a learning rate of 0.001 (1-e3). MSE is used as the reconstruction loss

function, allowing the network to learn latent representations that preserve important information from the original CNN features while reducing dimensionality by 50%.

The proposed AE contains 9,443,328 trainable parameters. Training and validation losses are monitored throughout the training process to evaluate convergence and reconstruction performance. The resulting 1024-dimensional latent features are subsequently used as inputs for the SVM and XGBoost classifiers. Although the proposed framework introduces additional computational overhead through AE training, the CNN feature extractor remains frozen and is used only for inference, making the overall computational burden lower than that of end-to-end CNN fine-tuning. A fully connected AE was selected because the input consists of 2048-dimensional feature vectors extracted by Inception V3 rather than spatial feature maps, making dense layers a natural choice for representation learning.

3.3.5. Classification with SVM and XGBoost

The compressed features produced by the AE are subsequently used as inputs for two ML classifiers: support vector machine (SVM) and XGBoost. For SVM, a one-vs-rest (OvR) strategy is employed to support multi-label classification. Hyperparameter optimization is performed using GridSearchCV with three-fold cross-validation. The search space includes the regularization parameter (C), kernel type (linear and RBF), kernel coefficient (gamma), and class-weight settings. For XGBoost, multi-label classification is implemented using a MultiOutputClassifier wrapper. Hyperparameters are optimized using GridSearchCV with three-fold cross-validation. The search includes the number of estimators, tree depth, learning rate, subsampling ratio, column sampling ratio, and regularization parameters. GPU acceleration is utilized during training to reduce computational time. Two experimental configurations are evaluated. The first uses the original 2048-dimensional CNN features directly (CNN + Classifier), while the second uses the 1024-dimensional AE-compressed features (CNN + AE + Classifier). This results in four model variants: CNN + SVM, CNN + AE + SVM, CNN + XGBoost, and CNN + AE + XGBoost.

Model performance is evaluated using Subset Accuracy, mAP, Micro F1-Score, Macro F1-Score, Micro ROC-AUC, and Macro ROC-AUC. Confusion matrices and per-class F1-score visualizations are additionally used to analyze classification behavior across instrument families. To assess result stability, all experiments were repeated three times under identical configurations. Because fixed random seeds and deterministic preprocessing procedures were used, all runs produced identical evaluation results. Consequently, standard deviations are not reported.

4. RESULTS AND DISCUSSION

Figures 6 and 7 show the convergence behavior of the AE during training. For both datasets, the reconstruction loss decreases steadily for both training and validation sets, indicating that the AE successfully minimizes reconstruction error and learns stable latent representations. No severe divergence between training and validation loss was observed for OpenMIC-2018, indicating stable AE training and reasonable generalization. However, the Orchset had a widening gap between training and validation loss after approximately 20 epochs, which would indicate mild overfitting.

The performance comparison in the classification phase, for each model on each dataset, OpenMIC-2018 and Orchset, can be seen in Tables 1 and 2, with the best models and the best metric scores for each dataset highlighted in bold. Once again, the comparison is between four models, which are CNN + SVM, CNN + AE + SVM, CNN + XGBoost, and CNN + AE + XGBoost. The metrics used were Subset Accuracy, mAP, macro and micro F1-score, and macro and micro ROC-AUC. The results were as follows:

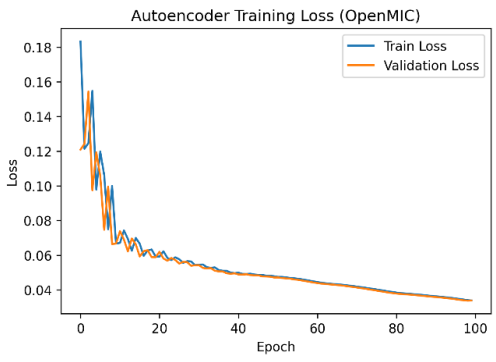

Figure 6. AE training and validation loss for OpenMIC-2018

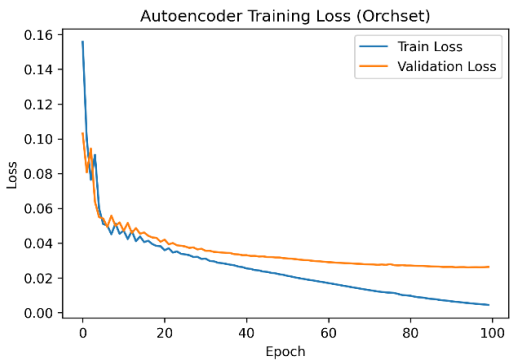

Figure 7. AE training and validation loss for orchset

Table 1. Model performance comparison on the OpenMIC-2018 dataset

Model	Subset Accuracy	mAP	F1-Macro	F1-Micro	ROC-AUC (macro)	ROC-AUC (micro)
CNN + AE + SVM	0.467	0.518	0.637	0.654	0.727	0.726
CNN + SVM	**0.562**	**0.558**	**0.667**	**0.685**	**0.749**	**0.755**
CNN + AE + XGBoost	0.520	0.500	0.570	0.624	0.677	0.714
CNN + XGBoost	0.536	0.538	0.619	0.664	0.708	0.742

Table 2. Model performance comparison on the orchset dataset

Model	Subset Accuracy	mAP	F1-Macro	F1-Micro	ROC-AUC (macro)	ROC-AUC (micro)
CNN + AE + SVM	0.231	0.483	**0.583**	0.650	0.568	0.655
CNN + SVM	0.385	0.499	0.510	0.684	0.543	0.701
CNN + AE + XGBoost	0.385	0.481	0.511	0.718	0.552	0.730
CNN + XGBoost	**0.539**	**0.506**	0.529	**0.757**	**0.591**	**0.775**

As detailed in Table 1, the CNN + SVM model achieved the best overall performance across all evaluation metrics for the OpenMIC-2018 dataset. The original 2048-dimensional CNN features consistently outperformed the AE-compressed features on both datasets, suggesting that the compressed representation discarded information useful for instrument discrimination. The confusion matrices and F1-score plots are visible in Figures 8 and 9.

The confusion matrices in Figure 8 reveal that strings Figure 8(a) were identified more reliably than the other instrument families, achieving the largest number of true positive predictions. In contrast, winds, as shown in Figure 8(b), produced the lowest F1-score, suggesting greater overlap with other instrument classes in the CNN feature space. Brass, as shown in Figure 8(c), achieved moderate performance, with relatively high recall but lower precision, indicating that the classifier tended to over-predict the presence of brass instruments. The per-class F1-score analysis shown in Figure 9 shows that strings (0) achieved the highest classification performance with 0.75, followed by brass (3) with 0.65 and winds (2) with 0.60.

For the Orchset dataset, the CNN + XGBoost model achieved the best overall performance across most metrics, including Subset Accuracy and ROC-AUC, as seen from the results in Table 2. Although CNN + AE + SVM achieved the highest F1-Macro score, its performance on other metrics was comparatively lower. Overall, the original CNN features provided stronger discriminative information than the AE-compressed representations, just like OpenMIC-2018.

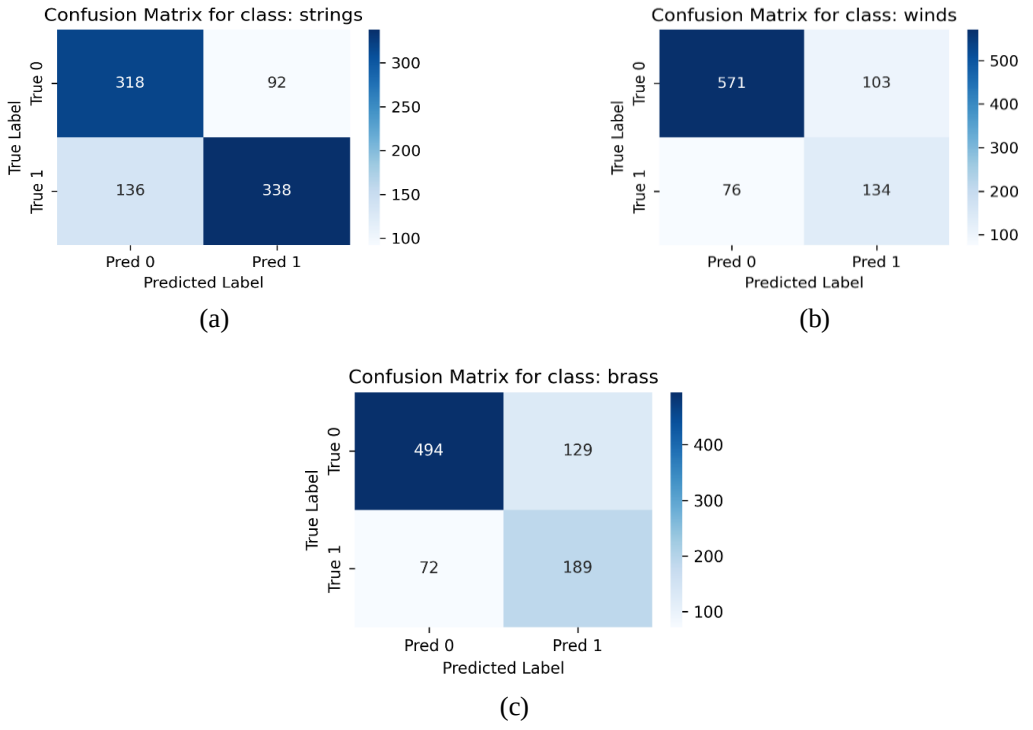

Figure 8. Confusion Matrices for the CNN + SVM Model on OpenMIC-2018 for classes: (a) strings, (b) winds, and (c) brass

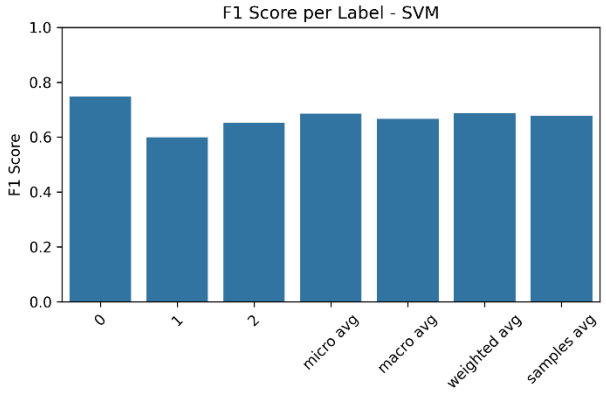

Figure 9. F1-Score Bar Plot for the CNN + SVM Model on OpenMIC-2018

As seen in the confusion matrices in Figure 10, the classifier predicted the string class for nearly all samples, as shown in Figure 10(a), resulting in perfect recall but lower precision due to false positives. Winds, as shown in Figure 10(b), achieved a more balanced trade-off between precision and recall. In contrast, the classifier failed to identify any brass samples as shown in Figure 10(c), producing zero true positives. Consequently, the Macro F1-score of 0.529 was considerably lower than the Micro F1-score of 0.757. The per-class F1-score analysis shown in Figure 11 reveals substantial variation among instrument families. Strings achieved the highest F1-score with 0.82, followed by winds with 0.77, whereas brass obtained an F1-score of 0.00. The Orchset dataset contains a relatively small number of samples, which may increase the sensitivity of performance metrics to individual classification errors and limit the generalizability of the reported results.

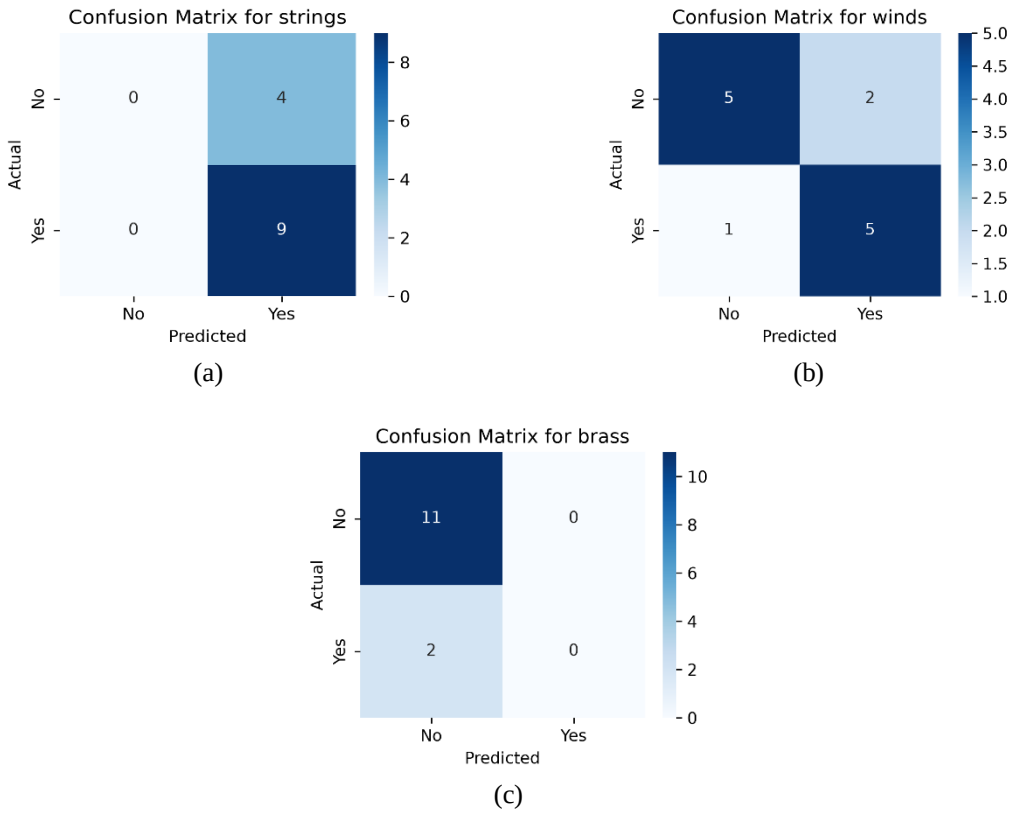

Figure 10. Confusion Matrices for the CNN + XGBoost model on orchset for classes: (a) strings, (b) winds, and (c) brass

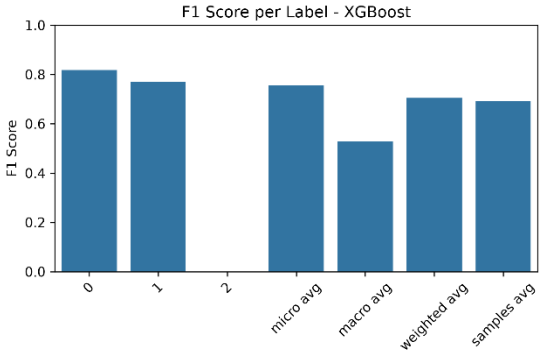

Figure 11. F1-Score bar plot for the CNN + XGBoost model on orchset

Across both datasets, classifiers trained on the original 2048-dimensional CNN features achieved better overall performance than those trained on the 1024-dimensional AE representations. The performance decrease after compression suggests that some discriminative information was lost during dimensionality reduction. Previous studies by Berahmand *et al.* [10], Nellas *et al.* [11], and Kumar *et al.* [23] showed that AEs can reduce dimensionality while maintaining classification performance. In the present study, however, the compressed representations were less effective for polyphonic orchestral recordings, where preserving fine-grained timbral information appears to be critical for instrument discrimination.

The substantial gap between macro and micro F1-scores and ROC-AUC values highlights the impact of class imbalance, with models performing better on dominant classes such as strings than on underrepresented classes such as brass. This effect is particularly evident in Orchset, where the classifier failed to identify any brass samples, which suggests that brass-related feature representations may overlap with those of other instrument families, making discrimination more difficult under the selected classification threshold. This result aligns with the observations of Sechet *et al.* [26], who developed hierarchical approaches specifically to address minority instrument detection in polyphonic orchestral recordings. Similar difficulties arising from spectral and timbral overlap have also been reported by Krause and Muller [5], Szeliga *et al.* [6], and Prabavathy *et al.* [2]. Because no dedicated class-imbalance mitigation strategy was applied, the performance degradation observed for the brass class may partly reflect the skewed label distribution of the dataset.

The summary of the results of both OpenMIC-2018 and Orchset builds upon and extends previous hybrid approaches in the MIR literature. Achieved high accuracies, up to 99.37%, using GoogleNet with SVM and kNN on the IRMAS dataset, primarily for monophonic or less complex recordings [1], [2]. This study, however, achieved substantially lower performance on OpenMIC-2018 and Orchset. This difference is expected because polyphonic orchestral recordings contain overlapping timbral and spectral information, making instrument recognition considerably more difficult. Similarly, Liu *et al.* [3] showed the effectiveness of XGBoost with fused handcrafted features (MFCCs, chroma, spectral contrast). The aforementioned direct numerical comparison should be interpreted with caution because the previous studies used monophonic datasets such as IRMAS or Medley-solos-DB, whereas OpenMIC-2018 and Orchset contain polyphonic recordings with overlapping instruments.

In conclusion, AE-based feature compression does not necessarily improve classification performance for polyphonic orchestral instrument recognition, highlighting the trade-off between dimensionality reduction and discriminative feature preservation in hybrid CNN–ML pipelines.

5. CONCLUSION AND FUTURE WORK

The results showed that models trained on the original CNN feature representations outperformed those trained on AE-compressed representations across both datasets. The best performance on OpenMIC-2018 was achieved by CNN + SVM, while the best performance on Orchset was achieved by CNN + XGBoost. Under the selected AE architecture and 50% compression ratio, dimensionality reduction did not improve multi-label orchestral instrument recognition. Although feature dimensionality was reduced successfully, the compressed representations generally produced lower classification performance than the original CNN features, suggesting that some discriminative information was lost during compression. These findings highlight the trade-off between reducing feature dimensionality and preserving discriminative information for downstream classification.

Several directions may be explored in future work by using larger and more diverse datasets to improve generalizability. Additional computational resources may support experimentation with deeper architectures, larger datasets, and more extensive hyperparameter optimization strategies. Incorporating attention-based architectures, such as CNN-RNN hybrids or Transformer-based models, may further improve the modeling of temporal dependencies in polyphonic music. Investigating alternative dimensionality-reduction techniques, different latent-space dimensions, and class-imbalance mitigation strategies may also provide further insight into improving the performance of multi-label orchestral instrument recognition.

FUNDING INFORMATION

We declare that no funding was involved in the study written in this paper.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Kelvin Wyeth	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Iman H. Kartowisastro				✓	✓		✓			✓	✓	✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

DATA AVAILABILITY

The data used and analyzed in this study are openly available in Zenodo, at https://zenodo.org/records/1289786 for Orchset. [29] and https://zenodo.org/records/1432913 for OpenMIC-2018 [30].

REFERENCES

[1] S. Prabavathy, V. Rathikarani, and P. Dhanalakshmi, "Classification of musical instruments using SVM and KNN," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 7, pp. 1186–1190, May 2020, doi: 10.35940/ijitee.G5836.059720.

[2] S. Prabavathy, V. Rathikarani, and P. Dhanalakshmi, "Musical instrument sound classification using GoogleNet with SVM and kNN model," in *Second International Conference on Image Processing and Capsule Networks*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 230–240, doi: 10.1007/978-3-030-84760-9_21.

[3] Y. Liu, Y. Yin, Q. Zhu, and W. Cui, "Musical instrument recognition by XGBoost combining feature fusion," Jun. 2022. [Online]. Available: https://github.com/Jay-Codeman/Musical-Instrument-Recognition-by-XGBoost.

[4] A. Lerch and P. Knees, "Machine learning applied to music/audio signal processing," *Electronics (Switzerland)*, vol. 10, no. 24, Dec. 2021, doi: 10.3390/electronics10243077.

[5] M. Krause and M. Muller, "Hierarchical classification for instrument activity detection in orchestral music recordings," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 2567–2578, 2023, doi: 10.1109/TASLP.2023.3291506.

[6] D. Szeliga, P. Tarasiuk, B. Stasiak, and P. S. Szczepaniak, "Musical instrument recognition with a convolutional neural network and staged training," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 2493–2502, doi: 10.1016/j.procs.2022.09.307.

[7] A. Ratliff, "What Is An Orchestra?," AudioLover. Accessed: Jul. 17, 2025, [Online]. Available: https://audiolover.com/production-technology/orchestra/what-is-an-orchestra/

[8] M. Taenzer, S. I. Mimilakis, and J. Abeßer, "Deep learning-based music instrument recognition: exploring learned feature representations," in *Proc. of the 15th International Symposium on CMMR*, 2021.

[9] E. Tsalera, A. Papadakis, and M. Samarakou, "Comparison of pre-trained CNNs for audio classification using transfer learning," *Journal of Sensor and Actuator Networks*, vol. 10, no. 4, Dec. 2021, doi: 10.3390/jsan10040072.

[10] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and their applications in machine learning: a survey," *Artificial Intelligence Review*, vol. 57, no. 2, Feb. 2024, doi: 10.1007/s10462-023-10662-6.

[11] I. A. Nellas, S. K. Tasoulis, V. P. Plagianakos, and S. V. Georgakopoulos, "Supervised dimensionality reduction and image classification utilizing convolutional autoencoders," Aug. 2022, [Online]. Available: http://arxiv.org/abs/2208.12152

[12] S. Rajesh and N. J. Nalini, "Musical instrument emotion recognition using deep recurrent neural network," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 16–25. doi: 10.1016/j.procs.2020.03.178.
[13] S. K. Mahanta, A. F. U. R. Khilji, and P. Pakray, "Deep neural network for musical instrument recognition using MFCCs," May 2021. Accessed: Jan. 22, 2025. [Online]. Available: https://arxiv.org/pdf/2105.00933v1
[14] N. A. Saputra, L. S. Riza, A. Setiawan, and I. Hamidah, "A systematic review for classification and selection of deep learning methods," *Decision Analytics Journal*, vol. 12, Sep. 2024, doi: 10.1016/j.dajour.2024.100489.
[15] M. Amini, "Machine learning and deep learning: a review of methods and applications," *World Information Technology and Engineering Journal*, vol. 10, no. 07, pp. 3897–3904, 2023, [Online]. Available: https://www.researchgate.net/publication/371011515
[16] R. Chen, A. Ghobakhlou, and A. Narayanan, "Interpreting CNN models for musical instrument recognition using multi-spectrogram heatmap analysis: a preliminary study," *Front. Artif. Intell.*, vol. 7, 2024, doi: 10.3389/frai.2024.1499913.
[17] Y. Wang, "Evaluation of national music performance by integrating attention mechanism and music theory rules," *Journal of Advances in Information Technology*, Sep. 2025, doi: 10.37965/jait.2025.0842.
[18] W. Li *et al.*, "Multi-scale deep feature fusion with machine learning classifier for birdsong classification," *Applied Sciences (Switzerland)*, vol. 15, no. 4, Feb. 2025, doi: 10.3390/app15041885.
[19] M. Ashraf *et al.*, "A hybrid CNN and RNN variant model for music classification," *Applied Sciences (Switzerland)*, vol. 13, no. 3, Feb. 2023, doi: 10.3390/app13031476.
[20] R. Gan, T. Huang, J. Shao, and F. Wang, "Music genre classification based on VMD-IWOA-XGBOOST," *Mathematics*, vol. 12, no. 10, May 2024, doi: 10.3390/math12101549.
[21] M. Pasini, S. Lattner, and G. Fazekas, "Music2Latent: consistency autoencoders for latent audio compression," Aug. 2024, [Online]. Available: http://arxiv.org/abs/2408.06500
[22] L. Qiu, S. Li, and Y. Sung, "3D-DCDAE: Unsupervised music latent representations learning method based on a deep 3D convolutional denoising autoencoder for music genre classification," *Mathematics*, vol. 9, no. 18, Sep. 2021, doi: 10.3390/math9182274.
[23] A. Kumar, S. S. Solanki, and M. Chandra, "Stacked auto-encoders based visual features for speech/music classification," *Expert Systems with Applications*, vol. 208, no. 118041, Dec. 2022, doi: 10.1016/j.eswa.2022.118041.
[24] S. Xu, Y. Zhang, X. An, and S. Pi, "Performance evaluation of seven multi-label classification methods on real-world patent and publication datasets," *JDIS Journal of Data*, vol. 9, no. 2, pp. 81–103, Feb. 2024, doi: 10.2478/jdis-2024.
[25] D. Kostrzewa, P. Szwajnoch, R. Brzeski, and D. Mrozek, "Detecting selected instruments in the sound signal," *Applied Sciences*, vol. 14, no. 14, p. 6330, Jul. 2024, doi: 10.3390/app14146330.
[26] D. Sechet, F. Bugiotti, M. Kowalski, E. D'hérouville, and F. Langiewicz, "A hierarchical deep learning approach for minority instrument detection," in *Proceedings of the 27th International Conference on Digital Audio Effects (DAFx24)*, Surrey, Sep. 2024, pp. 3–7.
[27] J. Bogatinovski, L. Todorovski, S. Džeroski, and D. Kocev, "Comprehensive comparative study of multi-label classification methods," *Expert Syst. Appl.*, vol. 203, Oct. 2022, doi: 10.1016/j.eswa.2022.117215.
[28] G. Ayu, V. M. Giri, and L. Radhitya, "Musical instrument classification using audio features and convolutional neural network," *Journal of Applied Informatics and Computing (JAIC)*, vol. 8, no. 1, p. 226, 2024, Accessed: Jan. 22, 2025. [Online]. Available: https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/8058
[29] J. J. Bosch, R. Marxer, and E. Gómez, "Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music," *Journal of New Music Research*, vol. 45, no. 2, pp. 101–177, May 2016, doi: 10.1080/09298215.2016.1182191#.V1bZDvmLTcs.
[30] E. J. Humphrey, S. Durand, and B. McFee, "OpenMIC-2018: an open dataset for multiple instrument recognition," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, Sep. 2018, pp. 438–444. [Online]. Available: http://freemusicarchive.org/

BIOGRAPHIES OF AUTHORS

Kelvin Wyeth has a bachelor's degree in computer science from his studies at Pelita Harapan University from 2019 to 2023. He has been continuing his studies for a master's degree in computer science at Bina Nusantara University, Jakarta, Indonesia, since 2024. His research interests include computer vision, signal processing, pattern recognition, machine learning, and deep learning. He can be contacted at email: kelvin.w500@gmail.com.

Iman H. Kartowisastro has been lecturing and doing research at Bina Nusantara University, Jakarta, Indonesia, since 1991. He graduated from Trisakti University, Jakarta, Indonesia, in 1986 with a major in Electronics and Telecommunication. He then obtained his M.Sc. in Information Engineering and Ph.D. in Robotics Control in 1987 and 1991, respectively, both from City University of London, U.K. His research interests are in Computer Vision, Computational Intelligence, and Signal Processing. Apart from being active in lecturing and conducting research, since 2021, he has also been deeply involved in higher education institutions and study program quality evaluation as a Secretary of the Accreditation Council, the National Accreditation Agency for Higher Education in Indonesia. He can be contacted at the email: ihkartowisastro@binus.ac.id