Dynamic Feature Learning with Involution and Convolution for Predominant Instrument Recognition in Polyphonic Music

Lekshmi C. R.[1*] and Jishnu Teja Dandamudi[2]

[1]School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore, 641112, Tamil Nadu, India.
[2]School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore, 641112, Tamil Nadu, India.

*Corresponding author(s). E-mail(s): cr_lekshmi@cb.amrita.edu;
Contributing authors: djishnuteja2006@gmail.com;

Abstract

This study presents a novel approach to identifying the predominant instrument in polyphonic music. By combining the strengths of Convolutional Neural Networks (CNN) and Involutional Neural Networks (INN) through an ensemble method, our approach achieves state-of-the-art performance while reducing computational complexity. Unlike traditional methods that rely on sliding window and aggregation strategies, our approach directly learns to recognize individual instruments from variable-length polyphonic audio. The proposed ensemble model, using soft voting, effectively uses the global frequency patterns captured by CNN and the dynamic localized features extracted by INN. Evaluations on the IRMAS dataset demonstrate that our proposed Ensemble CI model achieves a 3.33% and 8% improvement in micro and macro F1 scores, respectively, over the state-of-the-art Han model. Furthermore, our CNN-based model requires only 641k trainable parameters, while the Involution-based model reduces complexity to just 7k parameters, compared to the 1,446k parameters required by the Han model.

Keywords: Polyphonic music, Convolution, Involution, Ensemble, Soft voting

1 Introduction

Music Information Retrieval (MIR) has become a key research area, driven by the rapid expansion of digital music libraries and the growing demand for effective music organization and analysis. Among its diverse tasks, identifying predominant instruments in polyphonic music remains particularly challenging. Polyphonic music consists of overlapping sounds, making it difficult to isolate and distinguish individual instruments. Traditional approaches, such as sliding window analysis and aggregation strategies, are computationally intensive and require complex post-processing, limiting their scalability and real-time applicability.

Convolutional Neural Networks (CNNs) have emerged as a cornerstone of deep learning, transforming fields such as vision, speech, and audio analysis. Initially introduced with LeNet-5 [20] for handwritten digit recognition, CNNs have since evolved significantly. Milestones such as AlexNet [21], which leveraged deeper architectures, ReLU activations, and dropout, revolutionized image classification. Subsequent advances, such as VGGNet [16] and ResNet [15], addressed challenges like vanishing gradients and optimized network depth. In MIR, CNNs are widely adopted for their ability to extract spatial and temporal hierarchies from data. By transforming audio signals into time-frequency representations like spectrograms, CNNs excel at identifying local and global frequency patterns, enabling accurate recognition of overlapping instrument sounds. However, their computational complexity and large parameter requirements can hinder efficiency, particularly in real-time or resource-constrained scenarios.

Involution Neural Networks (INNs), introduced by [18], offer a compelling alternative to traditional convolutional methods. They replace fixed kernels with dynamic, position-specific kernels that adapt spatially across the input while remaining channel-agnostic. This approach enables efficient spatial modeling with fewer parameters and lower computational costs. Additionally, INNs have been integrated into existing deep learning architectures, such as ResNet[15] and MobileNet[19], to enhance spatial representation capabilities without significantly increasing computational costs. The ability of INNs to generalize self-attention mechanisms with spatial modeling makes them particularly valuable for tasks requiring adaptive spatial processing [18]. They offer an efficient alternative to traditional CNNs, especially in applications where real-time processing and resource efficiency are crucial. In the context of polyphonic music analysis, INNs excel at capturing localized patterns, handling overlapping sounds, and accommodating diverse timbres. By learning adaptive spatial relationships, INNs eliminate the need for handcrafted features or extensive post-processing, making them a robust and efficient choice for predominant instrument recognition.

The integration of CNNs and INNs presents a unique opportunity to capitalize on their complementary strengths. Ensemble techniques, such as soft voting, further enhance the robustness and generalization of MIR systems[26]. By aggregating predictions from multiple models, ensemble methods mitigate

individual model weaknesses and improve overall performance. This approach is particularly valuable for polyphonic instrument recognition, where the complexity of overlapping sounds demands highly adaptive and efficient modeling strategies [26].

The major contributions of this work are:

- A novel hybrid deep learning framework for the proposed task combining CNNs and INNs, where CNNs capture global frequency structures. INNs adaptively extract localized spatial features, improving instrument recognition.
- An ensemble learning strategy leveraging soft voting, enabling robust recognition by combining CNN and INN predictions to enhance generalization.
- A computationally efficient approach that eliminates the need for extensive post-processing, making it well-suited for real-time applications in music analysis.
- Unlike state-of-the-art (SOTA) methods, we do not rely on sliding window analysis or aggregation strategies. Instead, our approach directly processes variable-length polyphonic audio, reducing computational complexity while improving efficiency and accuracy.

By integrating CNNs and INNs into a unified ensemble model, the proposed method effectively enhances instrument recognition accuracy while reducing computational overhead, making it a scalable solution for automated music retrieval and classification.

2 Related work

The recognition of predominant instruments in polyphonic music has seen significant advancements over the years. Kitahara *et al.*[1] introduced a fusion model utilizing spectral, temporal, and modulation features with principal component analysis (PCA) to enhance classification accuracy. Building on this, Fuhrmann *et al.* [2] used support vector machines (SVMs) with features extracted from musical audio signals, and Bosch *et al.* [3] incorporated source separation as a preprocessing step to refine feature extraction and recognition accuracy.

Han *et al.* [4] utilized a Mel-spectrogram-CNN approach with aggregation over sliding windows, while Pons *et al.*[5] optimized this method for better timbral capture. Gururani *et al.* [6] applied a deep neural network (DNN) with temporal max-pooling for instrument detection, while Yu *et al.* [10] introduced multitask learning with auxiliary classification for improved category recognition. Gomez *et al.* [11] explored source separation and transfer learning as preprocessing steps, which improved performance in smaller datasets. Additionally, Soraghan *et al.* [12] used the Hilbert-Huang Transform (HHT) with CNN, and Kratimenos *et al.* [13] trained VGG-like CNN classifiers on augmented versions for the proposed task Lekshmi *et al.* [17] used Mel-spectrogram and phase-based modgdgram representations with data

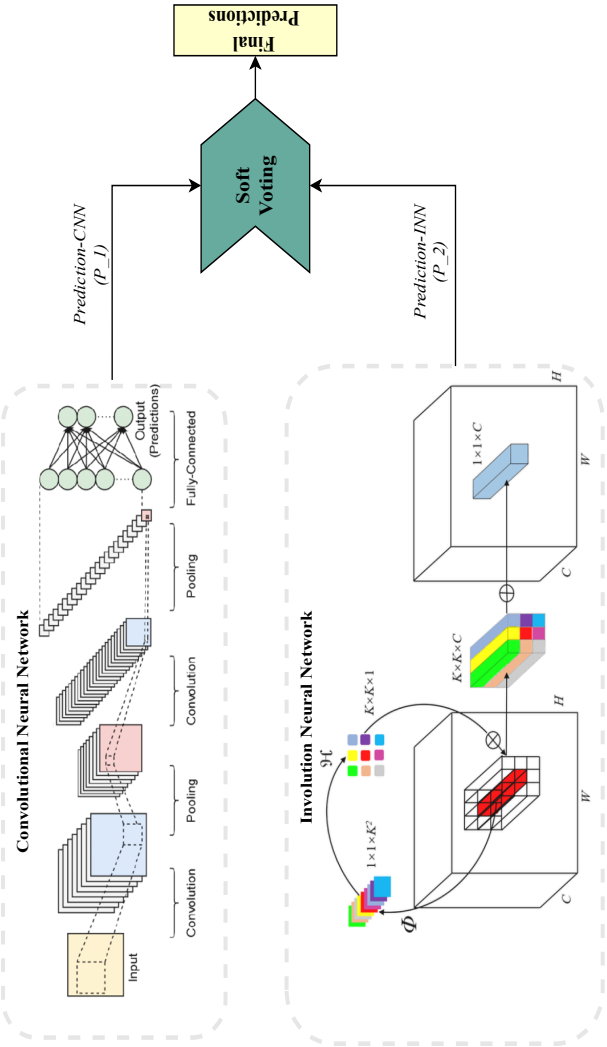

Fig. 1 Block diagram of the proposed method of predominant instrument recognition, where the CNN extracts global frequency features to generate prediction probabilities (P_1), the INN captures localized spatial dependencies to produce probabilities (P_2), and the final class probability is computed as the average of P_1 and P_2 to enhance predominant instrument recognition accuracy.

augmentation using WaveGAN for the proposed task. They also experimented with different transformer architectures with data augmentation for vision tasks using an ensemble of Mel-spectrogram, tempogram, and modgdgram [26]. Recent studies have demonstrated the effectiveness of neural networks in capturing complex nonlinear patterns [27–29] and the robustness of Gaussian process regression in modeling uncertainty [30, 31], reinforcing our approach in predominant instrument recognition, where intricate spectral-temporal dependencies and overlapping frequency regions require advanced learning techniques for accurate classification.

Despite advancements in predominant instrument recognition, existing methods rely on sliding window analysis and aggregation, adding computational overhead and post-processing demands [4, 5]. Deep learning approaches [12, 13, 26] often suffer from high complexity and limited generalization to variable-length audio. Additionally, preprocessing techniques like source separation [3, 11] enhance performance but increase computational costs. Our proposed hybrid framework overcomes these limitations by integrating CNNs and INNs for efficient spatial modeling without extensive post-processing. Unlike SOTA methods, it directly processes variable-length polyphonic audio, reducing computational complexity while maintaining better generalization and robustness through an ensemble strategy.

The outline of the rest of the paper is as follows. Section 3 explains the feature extraction. Section 4 explains the system description. Performance evaluation is explained in Section 5 followed by the analysis of results in Section 6. Finally, the paper is concluded in Section 7.

3 Feature extraction

A mel-spectrogram effectively transforms audio signals from the time domain to the frequency domain, showing how energy is distributed across frequencies over time. Its design aligns with human auditory perception, emphasizing lower frequencies and compressing higher ones, which enhances feature extraction for tasks like speech recognition and music classification. Additionally, mel-spectrograms reduce dimensionality through a mel filter bank, maintaining essential information while being robust to noise, thus improving model performance in real-world scenarios. Their ability to retain both temporal and spectral details makes them suitable for various applications, including sound event detection [17],[4]. Given their effectiveness and historical success, mel-spectrograms are widely used in audio processing, yielding strong results in complex tasks.

In our analysis, we chose 224 mel filter banks, utilized an 8192-point FFT, applied a Hanning window of 2205 samples (about 50 ms),[4], and set a hop length of 441 samples (roughly 10 ms). This setup provides a comprehensive frequency representation while ensuring adequate temporal resolution, making it well-suited for audio signal processing.

Table 1 Model Architecture Summary (Involution Neural Networks)

Layer (type)	Output Shape	Param #
InputLayer	(None, 32, 32, 3)	0
Involution (inv 1)	(None, 32, 32, 3)	26
ReLU (re lu 4)	(None, 32, 32, 3)	0
MaxPooling2D	(None, 10, 10, 3)	0
Involution (inv 2)	(None, 10, 10, 3)	26
ReLU (re lu 6)	(None, 10, 10, 3)	0
MaxPooling2D	(None, 3, 3, 3)	0
Involution (inv 3)	(None, 3, 3, 3)	26
ReLU (re lu 8)	(None, 3, 3, 3)	0
Flatten	(None, 27)	0
Dense (dense 2)	(None, 64)	1,792
Dense (dense 3)	(None, 11)	715
Total Parameters		**7,745**

4 System Description

The block diagram of the proposed method of predominant instrument recognition is illustrated in Figure 1. The experiment progresses in two phases, feature extraction at the front end and classification at the back end. At the back end, we experimented with CNN, INN, and an ensemble using a soft voting approach. This system employs CNN to capture global features associated with frequency patterns in polyphonic music, while INN focuses on spatial adaptability by dynamically generating position-specific kernels to extract localized features for differentiating overlapping instruments. The ensemble approach combines both models using soft voting, which aggregates the outputs from CNN and INN, enhancing the overall recognition performance. This hybrid model improves the accuracy of identifying multiple overlapping instruments, making it effective for polyphonic music instrument recognition, particularly in cases where spectral overlap presents challenges. The performance of the proposed method is compared with state-of-the-art Han's model [4].

4.1 Convolutional Neural Network

The CNN is employed to capture global features from the spectrogram. CNNs use a series of convolutional layers that apply learnable filters to the input, extracting important patterns such as harmonic structures, pitch contours, and frequency relationships. These patterns are essential for recognizing instruments, especially in polyphonic music, where instruments overlap in time and frequency [17]. The convolution operation for each neuron j in the CNN is expressed as [20]:

$$y_j = f\left(\sum_{i=1}^{n} x_i \cdot w_{ij} + b_j\right) \tag{1}$$

where:

- y_j is the output of neuron j,

Table 2 Model Architecture Summary of Convolutional Neural Networks

Layer (type)	Output Shape	Param #
Input	(None, 32, 32, 3)	0
Conv2D (256 filters)	(None, 32, 32, 256)	7,168
BatchNormalization	(None, 32, 32, 256)	512
MaxPooling2D	(None, 16, 16, 256)	0
Dropout	(None, 16, 16, 256)	0
Conv2D (128 filters)	(None, 16, 16, 128)	295,040
BatchNormalization	(None, 16, 16, 128)	256
MaxPooling2D	(None, 8, 8, 128)	0
Dropout	(None, 8, 8, 128)	0
Conv2D (64 filters)	(None, 8, 8, 64)	73,792
BatchNormalization	(None, 8, 8, 64)	128
MaxPooling2D	(None, 4, 4, 64)	0
Dropout	(None, 4, 4, 64)	0
Flatten	(None, 1024)	0
Dense (256 units)	(None, 256)	262,400
Dropout	(None, 256)	0
Dense (11 units)	(None, 11)	2,827
Total Parameters		**641,123**

- x_i represents the input features from the spectrogram,
- w_{ij} is the weight of the connection between the input feature and neuron j,
- b_j is the bias term,
- f is the activation function [20].

The convolution operation applied to an input feature map X with a kernel W of size $K \times K$ is formally defined as:

$$Y(i,j,c) = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} \sum_{d=1}^{D} W(m,n,d,c) \cdot X(i+m, j+n, d) + b_c \tag{2}$$

where:

- $Y(i,j,c)$ is the output feature map at spatial location (i,j) for channel c,
- $X(i+m, j+n, d)$ represents the input feature at position $(i+m, j+n)$ in channel d,
- $W(m,n,d,c)$ is the convolutional filter applied to the receptive field,
- b_c is the bias term for channel c,
- K is the kernel size, and D is the input channel depth.

4.2 Involution Neural Network

The INN is used to overcome the spatial rigidity of traditional convolution by dynamically generating position-specific kernels for each spatial location [18]. This adaptability is crucial for polyphonic music, where overlapping instruments create complex harmonic structures. INN enables the model to focus

on local, fine-grained features essential for distinguishing between overlapping instruments.

Kernel Generation:

For each spatial position (i, j) in the input, INN generates a unique kernel K_{ij} using a function ϕ parameterized by θ [14],[18]:

$$K_{ij} = \phi(x_{ij};\ \theta) \tag{3}$$

where:

- x_{ij} is the input at position (i, j),
- ϕ is the function that generates the kernel based on the input features,
- θ represents the parameters of the function.

Kernel Application:

After generating the kernel, it is applied to aggregate features from the neighboring spatial positions within a local region $R(i, j)$. The output at position (i, j) is computed as [14]:

$$y_{ij} = \sum_{(p,q)\in R(i,j)} K_{ij}(p, q) \cdot x_{pq} \tag{4}$$

where:

- y_{ij} is the output feature at position (i, j),
- $K_{ij}(p, q)$ is the kernel at position (p, q),
- x_{pq} is the input feature at position (p, q) within the local region $R(i, j)$.

4.3 Ensemble-CI

To enhance the system's accuracy, an ensemble model is employed, combining the predictions of both the CNN and INN models. The ensemble uses **soft voting** [23],[24], where the predicted probabilities from each model are aggregated. This approach allows the system to take advantage of the complementary strengths of both models.

Each model (CNN and INN) produces a probability distribution for each class. Let the probability distribution for class C_k predicted by the i-th model be $P_i(C_k)$. The soft-voting ensemble computes the final probability for each class C_k by averaging the probabilities from both models [25]:

$$P_{\text{final}}(C_k) = \frac{1}{M} \sum_{i=1}^{M} P_i(C_k) \tag{5}$$

where:

- M is the number of models in the ensemble (in this case, $M = 2$, corresponding to CNN and INN),

- $P_i(C_k)$ is the predicted probability for class C_k from the i-th model.

After calculating the final probabilities, the class with the highest averaged probability is selected as the final predicted class [25]:

$$\hat{y} = \arg\max_{C_k} P_{\text{final}}(C_k) \tag{6}$$

Hard voting or majority voting is not a suitable method for instrument recognition from polyphonic music [4],[26]. The soft voting mechanism helps mitigate the biases or weaknesses of individual models, leading to more robust and accurate predictions, especially in the complex task of polyphonic music instrument recognition.

5 Performance Evaluation

5.1 Dataset and Its Challenges

Our study utilizes the IRMAS dataset [2, 3], a comprehensive collection of musical audio excerpts, to investigate the automatic identification of predominant instruments in music. It comprises 6,705 training samples, each featuring a 3-second excerpt from a distinct recording, and 2,874 testing samples with varying durations between 5 and 20 seconds. The dataset includes 11 pitched instruments, carefully selected and annotated to facilitate the development of robust instrument classification models. Initially, we considered only single-labeled audio files during the testing phase. However, in later experiments, we expanded our evaluation to include the entire set of 2,874 polyphonic test files with variable lengths.

The sliding window approach presents several limitations, including increased computational complexity, inconsistent training and testing paradigms, loss of contextual information, and an inability to capture long-range dependencies and inter-instrument relationships. In contrast, our proposed method directly processes full-length test audio files by transforming them into Mel-spectrograms, which are then analyzed using a convolutional and involution network. This enables the effective capture of complex instrument interactions and contextual information, ultimately enhancing instrument recognition accuracy.

5.2 Evaluation and Experimental Setup

To evaluate our instrument recognition system, we calculated precision, recall, and F1 scores. Given the class imbalance across the 11 instrument categories, we computed both micro and macro averages—micro averages provided an overall performance measure influenced by dominant classes, while macro averages ensured equal weighting across all instrument categories. This dual evaluation method facilitated a more comprehensive analysis of system performance.

Table 3 Pseudocode of the Proposed CNN-INN Ensemble Model

Step 1: Feature Extraction using CNN
1.1 Initialize CNN parameters: W, b
1.2 For each convolutional layer l:
Apply convolution operation: $y_j^{(l)} = f\left(\sum_{i=1}^{n} x_i^{(l)} \cdot W_{ij}^{(l)} + b_j^{(l)}\right)$
Normalize using batch normalization: $\hat{y}_j^{(l)} = \frac{y_j^{(l)} - \mu_j}{\sigma_j + \epsilon}$
Apply max pooling: $y^{(l+1)} = \max_{p,q \in R} \hat{y}_{pq}^{(l)}$
Implement dropout regularization.
1.3 Flatten feature maps into vector F_{CNN}.
1.4 Pass through fully connected layers: $z = W_{\text{FC}} F_{\text{CNN}} + b_{\text{FC}}$
1.5 Compute classification probabilities using softmax: $P_{\text{CNN}}(C_k) = \frac{e^{z_k}}{\sum_j e^{z_j}}$
Step 2: Feature Extraction using INN
2.1 For each spatial position (i, j), generate dynamic kernel: $K_{ij} = \phi(X_{ij};\, \theta)$
2.2 Apply the kernel to extract local features: $y_{ij} = \sum_{(p,q) \in R(i,j)} K_{ij}(p, q) \cdot X_{pq}$
2.3 Use activation and pooling mechanisms.
2.4 Flatten feature maps into vector F_{INN}.
2.5 Process through fully connected layers: $z' = W_{\text{FC}} F_{\text{INN}} + b_{\text{FC}}$
2.6 Compute probabilities via softmax: $P_{\text{INN}}(C_k) = \frac{e^{z'_k}}{\sum_j e^{z'_j}}$
Step 3: Soft Voting Ensemble
3.1 Compute final class probabilities by averaging: $P_{\text{final}}(C_k) = \frac{1}{2}\left(P_{\text{CNN}}(C_k) + P_{\text{INN}}(C_k)\right)$
3.2 Determine the predicted class: $\hat{y} = \arg\max_{C_k} P_{\text{final}}(C_k)$
Step 4: Output the final predicted instrument class $\hat{y}$

The experiment was conducted in three phases: CNN-based, INN-based, and ensemble soft voting-based approaches. We utilized the IRMAS dataset, consisting of 1,305 polyphonic audio files across 11 single-labeled instrument classes, reserving 20% of the training data for validation. Additionally, we experimented with variable-length polyphonic test files, totaling 2,874 files, to further validate the generalization of our approach.

For benchmarking, we compared our models with Han's method [4], which employed a sliding window technique for short-time analysis and class-wise averaging of sigmoid outputs. To maintain consistency, we re-implemented Han's model with a 1-second slice length. Our models were trained on Google Colab for 200 epochs using the Adam optimizer and categorical cross-entropy loss function.

To assess the effectiveness of handcrafted feature extraction, we followed the methodology in [17, 22] and experimented with traditional machine learning models such as deep neural networks (DNN) and support vector machines (SVMs). The extracted features included Mel-frequency cepstral coefficients (MFCC-13), spectral centroid, spectral bandwidth, root mean square energy, spectral roll-off, and chroma short-time Fourier transform (STFT). These features were computed using the Librosa framework, and the machine learning models were evaluated using the same experimental setup.

6 Results and Analysis

The results of the single predominant experiment are summarized in Table 5. The proposed Ensemble CI achieves micro and macro F1 scores of 0.76 and 0.69, respectively, significantly surpassing the state-of-the-art Han model [4], which reports micro and macro F1 scores of 0.61 and 0.54. This corresponds to improvements of 24.59% and 27.77% in micro and macro F1 scores, respectively, over the baseline. The CNN and INN models demonstrate significant improvements over Han's model[4]. CNN excels in classes like Flute and Piano, with F1 scores of 0.73 and 0.81, surpassing Han's 0.54 and 0.71. INN further enhances performance in challenging categories, achieving F1-scores of 0.84 for Electric Guitar and 0.72 for Organ, compared to Han's 0.59 and 0.33. Both CNN and INN achieved an F1-score of 0.79 for the Acoustic Guitar, outperforming Han's 0.57. While Ensemble-CI leads in overall metrics, CNN and INN effectively capture complex patterns in polyphonic music.

In aggregate metrics, Ensemble-CI achieves a micro-average F1-score of 0.76, followed by INN (0.75) and CNN (0.74), all outperforming Han's 0.61. The macro-average F1-score of Ensemble-CI (0.69) surpasses CNN (0.64), INN (0.65), and Han (0.54), reflecting balanced performance across classes. Weighted average F1-scores also favor Ensemble-CI at 0.77, compared to CNN and INN (0.75 each) and Han (0.61). These results highlight the robustness of the proposed models, particularly Ensemble-CI, in managing class imbalances and delivering consistent recognition across diverse instruments. The ensemble model combines the predictions of these architectures, leveraging their strengths. By aggregating diverse predictions, ensemble learning reduces the impact of errors from individual models and captures complementary features more effectively.

Table 4 Precision (P), Recall (R), and F1 Score for All the Experiments

SL.No	Class	Han Model			CNN			INN			Ensemble-CI		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	Cello	0.76	0.68	0.72	0.91	0.42	0.57	0.96	0.34	0.50	0.97	0.38	0.54
2	Clarinet	0.34	0.84	0.48	0.67	0.64	0.65	0.75	0.60	0.67	0.94	0.68	0.79
3	Flute	0.55	0.54	0.54	0.71	0.76	0.73	0.75	0.69	0.72	0.70	0.65	0.68
4	Acoustic Guitar	0.62	0.53	0.57	0.81	0.75	0.78	0.81	0.77	0.79	0.83	0.77	0.80
5	Electric Guitar	0.70	0.51	0.59	0.77	0.83	0.80	0.81	0.87	0.84	0.80	0.87	0.84
6	Organ	0.25	0.49	0.33	0.60	0.73	0.66	0.71	0.73	0.72	0.82	0.73	0.77
7	Piano	0.74	0.69	0.71	0.80	0.82	0.81	0.77	0.83	0.80	0.78	0.85	0.81
8	Saxophone	0.14	0.60	0.23	0.15	0.70	0.25	0.14	0.60	0.23	0.17	0.70	0.27
9	Trumpet	0.52	0.79	0.63	0.25	0.50	0.33	0.50	0.43	0.46	0.60	0.64	0.62
10	Violin	0.39	0.62	0.48	0.56	0.82	0.66	0.52	0.82	0.63	0.57	0.85	0.69
11	Voice	0.69	0.58	0.63	0.84	0.66	0.74	0.80	0.70	0.75	0.81	0.73	0.77
Micro Avg		0.65	0.60	0.61	0.74	0.74	0.74	0.75	0.75	0.75	0.76	0.76	0.76
Macro Avg		0.52	0.62	0.54	0.64	0.69	0.64	0.68	0.67	0.65	0.73	0.71	0.69
Weighted Avg		0.65	0.60	0.61	0.78	0.74	0.75	0.78	0.75	0.75	0.79	0.76	0.77

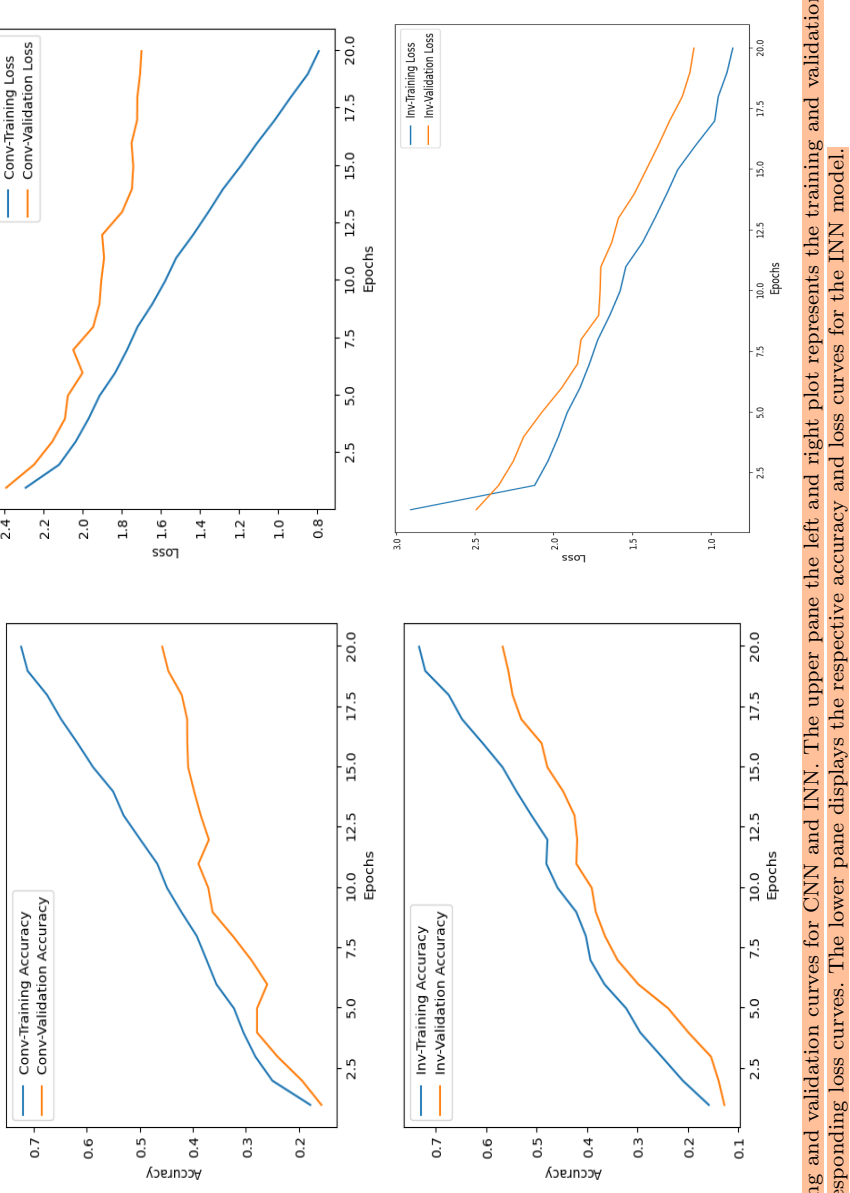

Fig. 2 Training and validation curves for CNN and INN. The upper pane the left and right plot represents the training and validation accuracy of CNN and corresponding loss curves. The lower pane displays the respective accuracy and loss curves for the INN model.

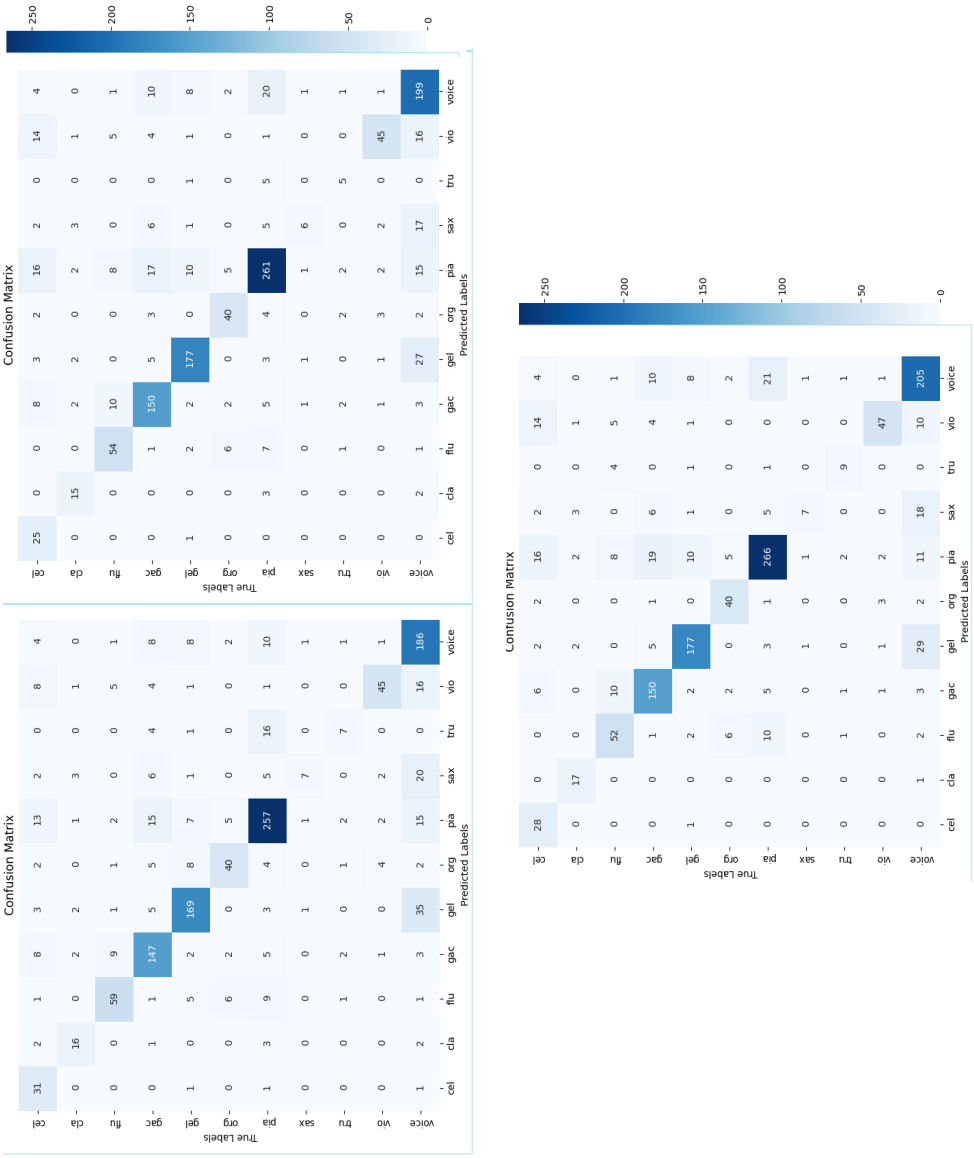

Fig. 3 Confusion matrices for CNN, INN, and Ensemble models, from left to right.

Table 5 Single Predominant Result Comparison on IRMAS dataset

Model	Params	Micro F1	Macro F1	Train Time (s)	GPU Infer Time (s)
Han *et al.* [4]	1446k	0.60	0.50	5.82	1.492
Proposed CNN	641k	0.74	0.64	5.11	1.349
Proposed INN	7k	**0.75**	**0.65**	2.46	1.363
Ensemble-CI	–	**0.76**	**0.69**	–	–

6.1 Instrument wise Performance

Figure 4 shows instrument -wise performance of all models. The proposed CNN demonstrates improved accuracy over Han's model for most instruments, notably achieving higher recall for Flute (0.76), Acoustic Guitar (0.75), and Piano (0.82). The INN further enhances accuracy for instruments like Electric Guitar (0.87) and Piano (0.83), while maintaining competitive performance across other classes. The Ensemble-CI model consistently outperforms the individual models, achieving the highest recall for instruments such as Piano (0.85), Violin (0.85), and Trumpet (0.64). However, certain instruments, like Saxophone and Cello, show relatively modest improvements across all models which can be attributed to the limited number of training and test samples available for these instruments [4]. Overall, the ensemble-based approach proves most effective in balancing accuracy across the diverse instrument classes, leveraging the strengths of both CNN and INN architectures.

6.2 Training and Validation Performance

The training and validation loss curves for the convolutional model, as shown in Figure 2, exhibit a steady decline, indicating effective learning. Training accuracy improves consistently, while validation accuracy rises initially and stabilizes in later epochs, suggesting convergence. A gap between training and validation performance in the final epochs indicates potential over-fitting or differences in data distribution. However, the stabilization of validation metrics suggests a balance between bias and variance, demonstrating the model's robustness for further testing. In contrast, as observed in Figure 2, the proposed INN shows a more rapid decrease in loss, reflecting quicker learning. The validation loss closely follows the training loss with minimal variation, suggesting strong generalization and reduced over-fitting. Likewise, the validation accuracy closely tracks the training accuracy, indicating consistent performance across both training and validation datasets.

Therefore, the ensemble of involution and convolution models offers several advantages, as reflected in the training curves. The involution model's faster loss reduction accelerates convergence, complementing the convolution model's steady progress. This synergy enhances training efficiency and generalization, with the involution model's minimal training-validation gap mitigating the slight over-fitting seen in convolution models. By combining the strengths of both methods—convolutions for capturing complex features and involutions for efficiently handling spatial patterns—the ensemble improves robustness, leading to better overall performance and stability across diverse datasets.

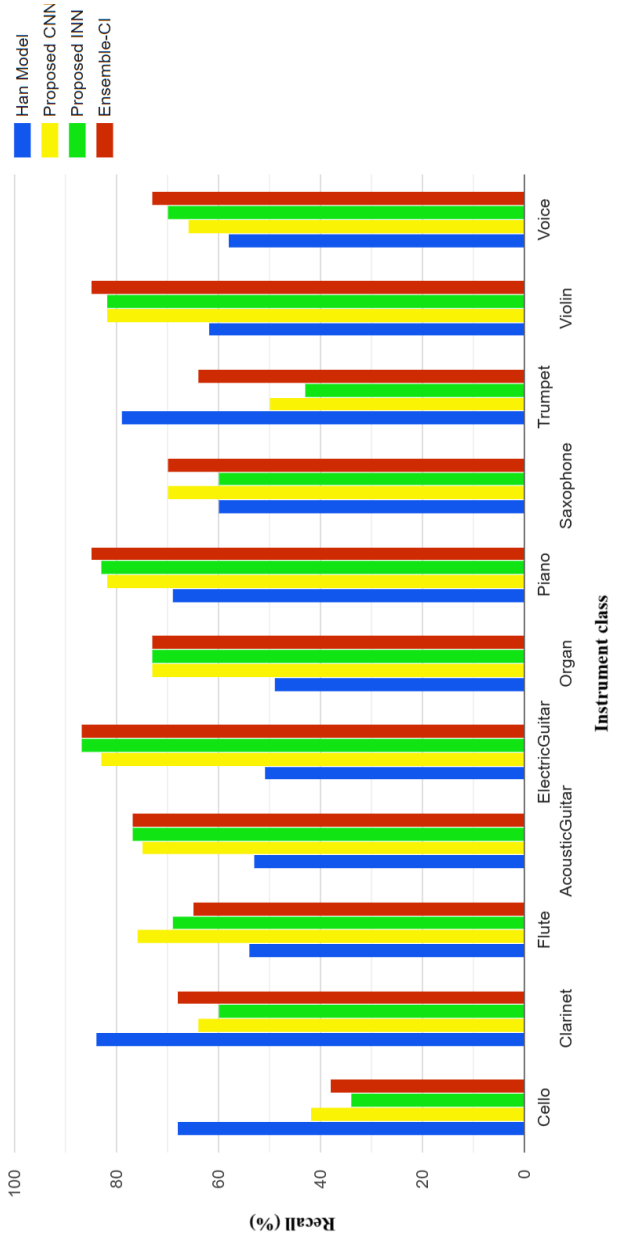

Fig. 4 Instrument-wise performance comparison

6.3 Trade-off Analysis Between Model Complexity, Computational Cost, and Performance

The trade-off between model complexity, computational cost, and processing time is evident from the table 5 when comparing the Han model [4] with our proposed approaches. The Han model, with 1.446 million parameters, has higher computational demands, resulting in a lower Micro F1 score of 0.60 and a Macro F1 score of 0.50. Additionally, its training time (5.82s) and GPU inference time (1.492s) are comparatively higher. Our proposed CNN model, with 641k parameters, improves performance (Micro F1: 0.74, Macro F1: 0.64) while slightly reducing training time (5.11s) and inference time (1.349s). The INN model, with only 7k parameters, further enhances accuracy (Micro F1: 0.75, Macro F1: 0.65) while significantly lowering training time (2.46s), making it computationally efficient. The Ensemble-CI method, which combines CNN and INN predictions, achieves the highest accuracy (Micro F1: 0.76, Macro F1: 0.69) but comes with increased computational complexity. While the Han model is more resource-intensive with lower performance, our models effectively balance accuracy and efficiency, demonstrating their suitability for music instrument recognition.

6.4 Analysis of Confusion Matrices for CNN, INN, and Ensemble Models

Figure 3 displays the confusion matrices for the CNN, INN, and Ensemble models. The CNN demonstrated decent performance but faced challenges with misclassifications in voice, piano, and electric guitar. The INN showed improved accuracy in these classes, addressing some of CNN's weaknesses. The Ensemble model leveraged the strengths of both CNN and INN, achieving the best overall accuracy. It corrected misclassifications in CNN while preserving the improved performance in INN, highlighting its ability to generalize effectively across all classes.

Table 6 Comparison of Different Models for Predominant Instrument Recognition

Sl.No	Model/#parameters	Micro F1	Macro F1
1	Bosch et al. [14]	0.50	0.43
2	Han et al./1446k [1]	0.60	0.50
3	Single-layer/62k [5]	0.56	0.48
4	Multi-layer/743k [5]	0.59	0.52
5	MTF-DNN [17]	0.32	0.28
6	MTF-SVM [22]	0.25	0.23
7	Proposed Convolution/671k	0.61	0.52
8	Proposed Involution/7k	0.62	0.54
9	Ensemble CI	0.62	0.54

6.5 Multiple Predominant Instrument Recognition

The IRMAS dataset includes variable-length testing files, many with multiple predominant instruments. Initially, we focused on single predominant instrument recognition but later extended our study to all 2,874 testing files for multiple instrument recognition and the results are tabulated in Table 6

Our networks were trained on fixed-length excerpts with single predominant instruments, while inference was performed on variable-length polyphonic files. Prior studies, such as Bosch *et al.* [3] and Han *et al.* [4], relied on handcrafted features with SVMs or CNN-based models using sliding window aggregation. Unlike these methods, our ensemble model, combining convolutional and involutional networks, directly estimates instruments from full-length Mel-spectrograms, achieving superior accuracy and robustness. We also experimented with Music Texture Feature (MTF)-based models but found they struggled to capture complex spectral-temporal dependencies in polyphonic music, particularly in overlapping frequency regions. In contrast, our model effectively learns hierarchical representations, leading to more accurate multiple predominant instrument recognition.

The number of trainable parameters plays a crucial role in determining the computational efficiency of deep learning models. The Han et al [4] model, which incorporates a sliding window analysis and aggregation strategy, requires 1,446k trainable parameters. Similarly, Pons et al.[5] proposed two models, a single-layer variant with 62k parameters and a multi-layer variant with 742k parameters, both of which follow the same aggregation approach as Han et al. In contrast, our proposed convolutional model eliminates the need for sliding window analysis and aggregation while achieving competitive performance with only 641k parameters. Furthermore, our involution-based model significantly reduces computational complexity, requiring just 7k parameters while maintaining effectiveness. This demonstrates that our approach not only enhances recognition accuracy but also optimizes computational efficiency, making it more suitable for real-time and resource-constrained applications.

7 Conclusion

This study introduces an ensemble approach combining CNNs and INNs to tackle the challenge of recognizing the predominant instrument in polyphonic music. The proposed models, especially the Ensemble-CI, show notable improvements in accuracy and efficiency, requiring fewer parameters compared to conventional methods. The synergy between CNN and INN enhances the model's ability to capture complex features while managing spatial patterns effectively, resulting in improved robustness and generalization. Future work can address data scarcity in predominant instrument recognition through data augmentation, transfer learning, and semi-supervised learning [32]. Additionally, real-time deployment of the ensemble model can be explored for music information retrieval and audio processing, ensuring low-latency and high-accuracy predictions.

Declarations

1. The authors declare that they have no competing interests.
2. The datasets analyzed during the current study are available at https://www.upf.edu/web/mtg/irmas
3. Lekshmi C. R. and Jishnu Teja Dandamudi jointly designed, implemented, and interpreted the computer simulations and prepared the manuscript.

References

[1] Kitahara, T., Goto, M., Komatani, K., Ogata, T., Okuno, H.G.: Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps. EURASIP Journal on Advances in Signal Processing 2007, 1–15 (2006)

[2] Fuhrmann, F., Herrera, P.: Polyphonic instrument recognition for exploring semantic similarities in music. In: Proc. of 13th Int. Conference on Digital Audio Effects DAFx10, pp. 1–8 (2010)

[3] Bosch, J.J., Janer, J., Fuhrmann, F., Herrera, P.: A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In Proc. of 13th Int. Society for Music Information Retrieval Conference, 552–564 (2012)

[4] Han, Y., Kim, J., Lee, K.: Deep convolutional neural networks for predominant instrument recognition in polyphonic music. IEEE/ACM Transactions on Audio, Speech, and Language Processing 25(1), 208–221 (2017)

[5] Pons, J., Slizovskaia, O., Gong, R., Gomez, E., Serra, X.: Timbre analysis of music audio signals with convolutional neural networks. In Proc. of 25th European Signal Processing Conference, 2744–2748 (2017). IEEE

[6] Gururani, S., Summers, C., Lerch, A.: Instrument activity detection in polyphonic music using deep neural networks. In Proc. of Int. Society for Music Information Retrieval Conference, 569–576 (2018)

[7] Duan, Z., Han, J., Pardo, B.: Multi-pitch streaming of harmonic sound mixtures. IEEE/ACM Transactions on Audio, Speech, and Language Processing 22(1), 138–150 (2013)

[8] Heittola, T., Klapuri, A., Virtanen, T.: Musical instrument recognition In polyphonic audio using source-filter model for sound separation. in Proc. of Int. Society of Music Information Retrieval Conference, 327–332 (2009)

[9] Li, P., Qian, J., Wang, T.: Automatic instrument recognition in polyphonic music using convolutional neural networks. arXiv:1511.05520, (2015)

[10] Yu, D., Duan, H., Fang, J., Zeng, B.: Predominant instrument recognition based on deep neural network with auxiliary classification. IEEE/ACM Transactions on Audio, Speech, and Language Processing 28, 852–861 (2020)

[11] Juan, G.C., Jakob, A., Cano, E.: Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning. in Proc. of Int. Society for Music Information Retrieval Conference, 577–584 (2018)

[12] Li, X., Wang, K., Soraghan, J., Ren, J.: Fusion of Hilbert-Huang transform and deep convolutional neural network for predominant musical instruments recognition. In Proc. of 9th Int. Conference on Artificial Intelligence in Music, Sound, Art, and Design, 80–89 (2020)

[13] Kratimenos, A., Avramidis, K., Garoufis, C., Zlatintsi, A., Maragos, P.: Augmentation methods on monophonic audio for instrument classification in polyphonic music. in Proc. of 28th European Signal Processing Conference, 156–160 (2021). IEEE

[14] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In Proceedings of the IEEE International Conference on computer vision, pages 764–773, 2017.

[15] K He Resnet, X Zhang, S Ren, and J Sun. Deep residual learning for image recognition. arXiv, 2015.

[16] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034, 2013.

[17] C R Lekshmi and Rajan Rajeev. Multiple predominant instruments recognition in polyphonic music using spectro/modgd-gram fusion. Circuits, Systems, and Signal Processing, 42(6):3464–3484, 2023.

[18] Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 12321–12330, 2021.

[19] Hang Song, YongHong Song, and YuanLin Zhang. Sca net: Sparse channel attention module for action recognition. In 2020 25th International Conference on Pattern Recognition (ICPR), pages 1189–1196. IEEE, 2021.

[20] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998 Nov;86(11):2278-324.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012;25(1097-1105):26.

[22] K. Racharla, V. Kumar, C. B. Jayant, A. Khairkar, and P. Harish, "Predominant musical instrument classification based on spectral features," in Proc. of 7th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 617–622, 2020

[23] Saloni Kumari, Deepika Kumar, and Mamta Mittal. An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. International Journal of Cognitive Computing in Engineering, 2:40–46, 2021

[24] Andrea Manconi, Giuliano Armano, Matteo Gnocchi, and Luciano Milanesi. A soft-voting ensemble classifier for detecting patients affected by covid-19. Applied Sciences, 12(15), 2022.

[25] Ury Naftaly, Nathan Intrator, and David Horn. Optimal ensemble averaging of neural networks. Network: Computation in Neural Systems, 8(3):283, 1997

[26] Lekshmi Chandrika Reghunath and Rajeev Rajan, "Transformer-based ensemble method for multiple predominant instruments recognition in polyphonic music," *EURASIP Journal on Audio, Speech, and Music Processing,* vol. 2022, no. 1, pp. 11, 2022.

[27] B Jin and X Xu. Carbon emission allowance price forecasting for China guangdong carbon emission exchange via the neural network. Global finance review. 2024; 6 (1): 3491, 2016.

[28] Bingzi Jin and Xiaojie Xu. Machine learning predictions of regional steel price indices for east china. Ironmaking & Steelmaking, page 03019233241254891, 2024.

[29] Bingzi Jin and Xiaojie Xu. Pre-owned housing price index forecasts using gaussian process regressions. Journal of Modelling in Management, 19(6):1927–1958, 2024.

[30] Bingzi Jin and Xiaojie Xu. Wholesale price forecasts of green grams using the neural network. Asian Journal of Economics and Banking, (ahead-of-print), 2024.

[31] Bingzi Jin, Xiaojie Xu, and Yun Zhang. Thermal coal futures trading volume predictions through the neural network. Journal of Modelling in Management, 20(2):585–619, 2025.

[32] Alzubaidi L, Bai J, Al-Sabaawi A, Santamaría J, Albahri AS, Al-Dabbagh BS, Fadhel MA, Manoufali M, Zhang J, Al-Timemy AH, Duan Y. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. Journal of Big Data. 2023 Apr 14;10(1):46.