Spatio-Channel Complementary Learning for Polyphonic Music Instrument Recognition

1st Lekshmi C. R.
Amrita School of Artificial Intelligence, Coimbatore
Amrita Vishwa Vidyapeetham, India
cr_lekshmi@cb.amrita.edu

2nd Jishnu Teja Dandamudi
Amrita School of Artificial Intelligence, Coimbatore
Amrita Vishwa Vidyapeetham, India
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 of the IRMAS dataset demonstrate improved recognition accuracy and efficiency, making our approach suitable for real-world music information retrieval applications.**

***Index Terms*—Polyphonic music, Convolution, Involution, Ensemble, Softvoting,**

I. 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, the identification of the predominant instruments in polyphonic music remains a particularly challenging task. Polyphonic music often consists of overlapping sounds, making it difficult to isolate and distinguish individual instruments [22]. 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 the cornerstone of deep learning, transforming fields such as vision, speech, and audio analysis. Initially introduced with LeNet-5 [11] for handwritten digit recognition, CNNs have since evolved significantly. Milestones such as AlexNet [8], which leveraged deeper architectures, ReLU activations, and dropout, revolutionized image classification. Subsequent advances such as VGGNet [25] and ResNet [23] addressed challenges such as 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 [9] [18], 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 [14], 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. In addition, INNs have been integrated into existing deep learning architectures, such as ResNet [23] and MobileNet [26], to enhance their spatial representation abilities without significant increases in computational cost. The ability of INNs to generalize self-attention mechanisms with spatial modeling makes them particularly valuable for tasks requiring adaptive spatial processing [14]. 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 hand-made features or extensive postprocessing, making them a robust and efficient choice for predominant instrument recognition.

The integration of CNNs and INNs offers a unique opportunity to capitalize on their complementary strengths. Ensemble techniques, such as soft voting, further enhance the robustness and generalization of MIR systems [21]. By aggregating predictions from multiple models, the 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 [21], [12], [20].

II. Related work

The recognition of predominant instruments in polyphonic music has seen significant advancements over the years. Kitahara *et al.* [6] 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.* [3] used support vector machines (SVMs) with features extracted from musical audio signals, and Bosch

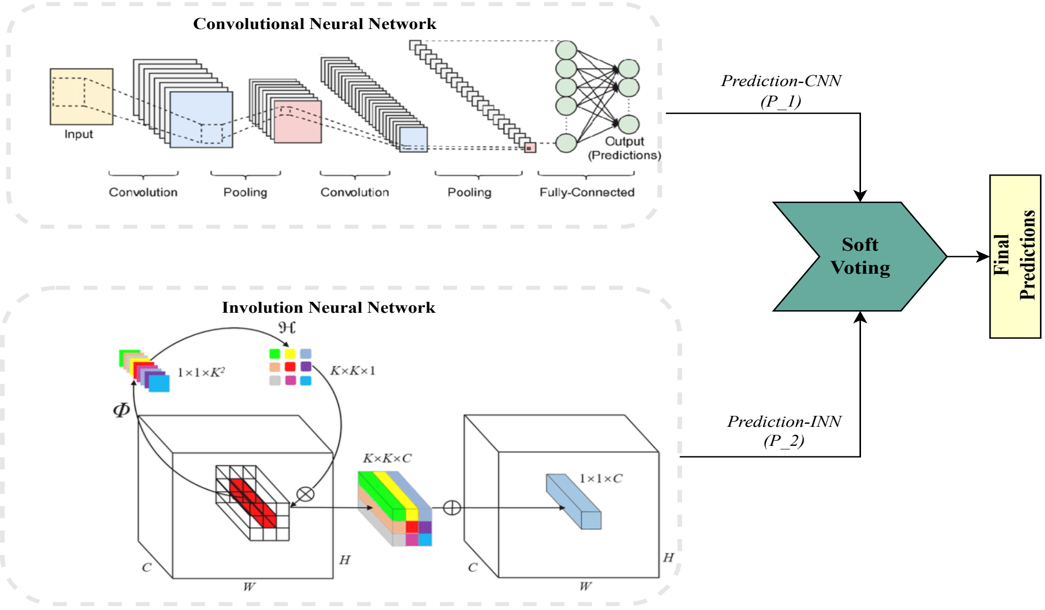

Fig. 1. Block diagram of proposed method of predominant instrument recognition

et al. [1] incorporated source separation as a preprocessing step to refine feature extraction and recognition accuracy.

Han *et al.* [5] utilized a Mel-spectrogram-CNN approach with aggregation over sliding windows, while Pons *et al.* [19] optimized this method for better timbral capture. Gururani *et al.* [24] applied a deep neural network (DNN) with temporal max-pooling for instrument detection, while Yu *et al.* [27] introduced multitask learning with auxiliary classification for improved category recognition. Gomez *et al.* [4] explored source separation and transfer learning as preprocessing steps, which improved performance in smaller datasets. Additionally, Soraghan *et al.* [15] used the Hilbert-Huang Transform (HHT) with CNN, and Kratimenos *et al.* [7] trained VGG-like CNN classifiers on augmented versions for the proposed task Lekshmi *et al.* [13], [12] used Mel-spectrogram and phase-based modgdgram representations with data 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 [21].

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.

III. 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 [13], [5]. 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), [5], 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.

IV. 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

TABLE I
MODEL ARCHITECTURE SUMMARY (INVOLUTION NEURAL NETWORKS)

Layer	**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**

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 [5].

A. *Convolution 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 [13].

The convolution operation for each neuron j in the CNN is expressed as [11]:

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

Where: y_j is the output of neuron j, 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 [11].

The proposed CNN model shown in Table II starts with an input layer (32x32x3) and passes through three convolutional layers with 256, 128, and 64 filters, respectively, each followed by batch normalization, max pooling, and dropout. The output is then flattened and passed through a dense layer with 256 units, followed by an output layer with 11 units and a softmax activation function for classification. The total number of parameters is 641,123.

B. *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 [14]. 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 [2], [14]:

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

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 [2]:

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

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 $R(i, j)$ [2].

The proposed INN model shown in Table I begins with an input layer (32x32x3) and passes through three involution layers with 3 filters each, followed by ReLU activations and max pooling. After flattening the output, it goes through two dense layers with 64 and 11 units, respectively, with the final layer using a softmax activation for classification. The total number of parameters in this model is 7,745. INN is especially useful for recognizing fine-grained, localized features, such as subtle variations in timbre or harmonic content, which are key in differentiating overlapping instruments in polyphonic music.

TABLE II
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**

TABLE III
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

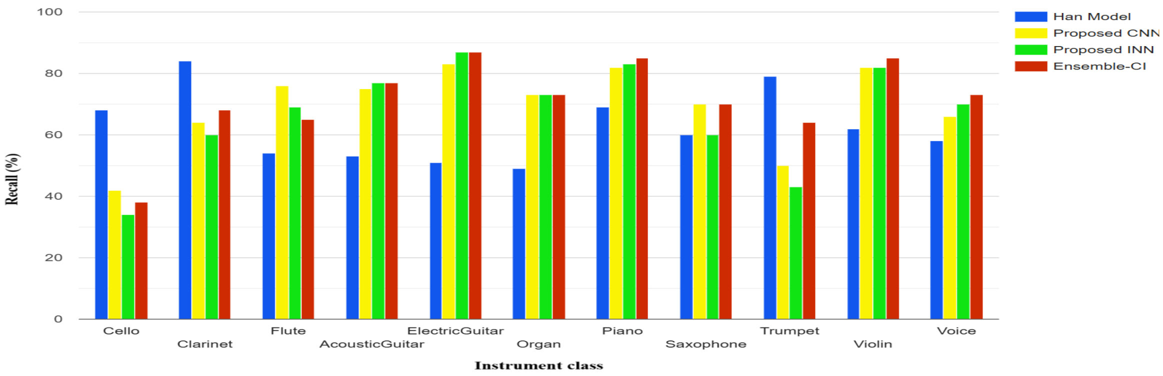

Fig. 2. Instrument-wise performance comparison

C. 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** [10], [16] 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 [17]:

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

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 [17].

After calculating the final probabilities, the class with the highest averaged probability is selected as the final predicted class [17]:

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

Hard voting or majority voting is not a suitable method for instrument recognition from polyphonic music [5], [21]. 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.

V. PERFORMANCE EVALUATION

A. Dataset and its challenges

Our study utilizes the IRMAS dataset [3], [1], 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 5-20 seconds. The dataset includes 11 pitched instruments, carefully selected and annotated to facilitate the development of robust instrument classification models. The dataset consists of testing audio samples with multiple predominant instruments as labels, we have considered all the audio files with a single predominant instrument (single label) during the testing phase. The sliding window approach is fraught with limitations, including elevated computational complexity, inconsistent training, and testing paradigms, loss of contextual information, and inability to capture long-range dependencies and inter-instrument relationships. In contrast, our proposed method leverages the entirety of test audio files, transforming them into Mel-spectrograms that are subsequently processed by a convolutional and involution network, enabling the effective capture of complex instrument interactions and contextual information, ultimately enhancing the accuracy of instrument recognition.

B. Evaluation and Experimental setup

To evaluate the performance of our instrument recognition system, we used a comprehensive approach by calculating precision, recall, and F1 scores. To address the class imbalance across the 11 instrument categories, we computed both micro and macro averages. Micro averages provided an overall performance measure, influenced more by the frequently occurring instrument classes. On the other hand, macro averages offered a balanced view by equally weighting the performance of all instrument categories. This dual evaluation method enabled a detailed analysis of the system's performance across different classes.

The experiment consisted of three phases: CNN-based, INN-based, and ensemble soft voting-based approaches. We utilized the IRMAS dataset, which contains 1305 polyphonic audio files across 11 single-labeled instrument classes, reserving 20% of the training data for validation. Our method was benchmarked against Han's model [5], which utilized a sliding window for short-time analysis and averaged sigmoid outputs class-wise. For consistency, we implemented Han's model [5] with a 1-second slice length. The proposed models were trained on Google Colab for 200 epochs, using the Adam optimizer and categorical cross-entropy loss function.

VI. Results and Analysis

The results of the experiment are summarized in Table IV. 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 [5], 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 [5]. 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.

A. Instrument wise Performance

Figure 2 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 [5]. 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.

B. Training and Validation Performance

The training and validation loss curves for the convolutional model, as shown in Figure 3, 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 3, 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.

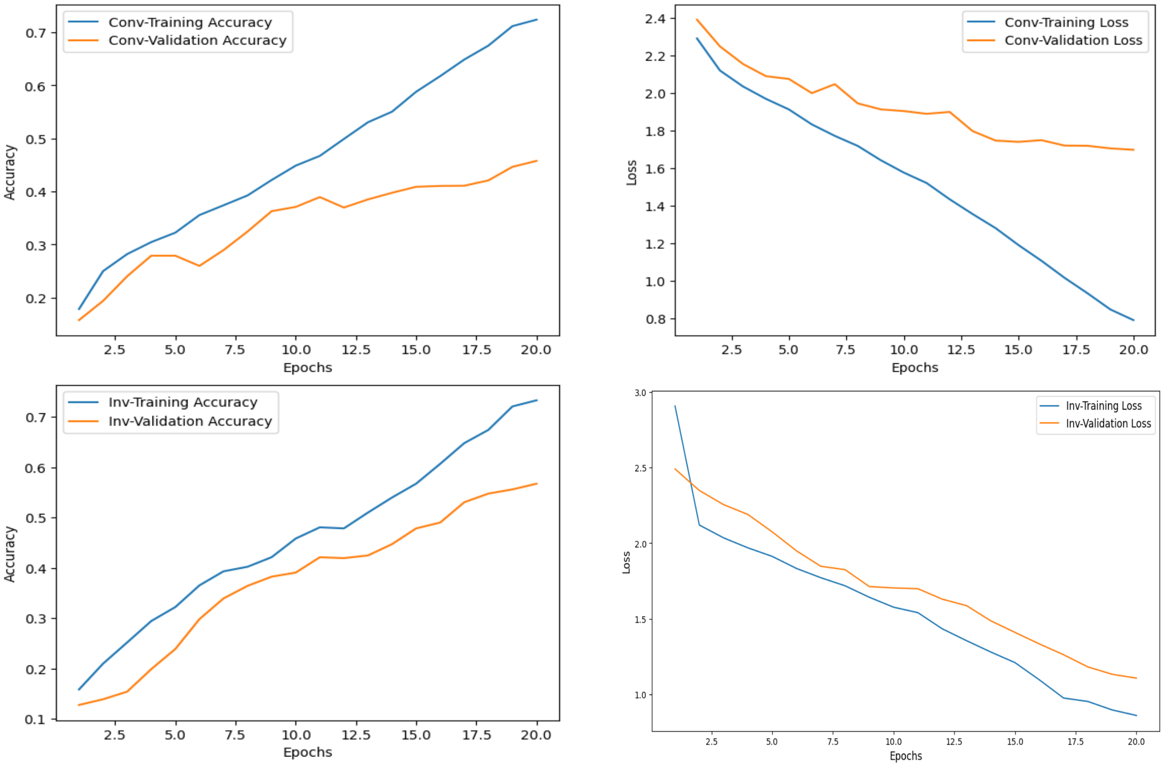

Fig. 3. Training and validation curves for CNN and INN

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.

C. *Comparison with state-of-the-art Han model [5]*

Han *et al.* [5] introduced a Mel-spectrogram CNN, which employs multiple convolutional layers followed by max pooling. To handle variable-length test data, they used sliding window analysis, aggregating multiple sigmoid outputs with thresholding. This method involves 1,446k trainable parameters. In contrast, the proposed CNN uses a shallower architecture with three convolutional layers and max pooling, incorporating batch normalization to prevent overfitting, resulting in only 641k parameters.

One of the main strengths of our approach is the use of an ensemble of models, rather than relying on a single model for predictions. By combining the CNN and INN — which has only 7k parameters — the ensemble enhances both generalization and accuracy. The CNN excels at capturing complex features, while the INN efficiently manages spatial patterns. The complementary learning between the two models leads to improved performance, providing more accurate predictions and better generalization. This ensemble approach leverages the strengths of both models, resulting in a more robust and efficient solution.

TABLE IV
PERFORMANCE COMPARISON ON IRMAS [3]DATASET.

Model	Parameters	Micro F1	Macro F1
Han *et al.* [5]	1446k	0.60	0.50
Proposed CNN	641k	0.74	0.64
Proposed INN	7k	**0.75**	**0.65**
Ensemble-CI		**0.76**	**0.69**

VII. 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. As future work, the real-time application of the ensemble model can be explored, especially in domains like music information retrieval and audio processing, where achieving low-latency and high-accuracy predictions is vital.

References

[1] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera. Acomparison of sound segregation techniques for predominant instrument recognition in musical audio signals. *in Proc. of 13th Int. Society for Music Information Retrieval Conference (ISMIR)*, 2012.

[2] 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.

[3] F. Fuhrmann and P. Herrera. Polyphonic instrument recognition for exploring semantic similarities in music. *in Proc. of 13th Int. Conf. on Digital Audio Effects DAFx10, Graz, Austria*, 14(1):1–8, 2010.

[4] Juan S Gómez, Jakob Abeßer, and Estefanía Cano. Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning. In *ISMIR*, pages 577–584, 2018.

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

[6] T Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps. *EURASIP Journal of Applied Signal Processing*, pages 155–175, 2007.

[7] A. Kratimenos, K Avramidis, C. Garoufis, Athanasia Zlatintsi, and Petros M. Augmentation methods on monophonic audio for instrument classification in polyphonic music. *in Proc.of 28th European Signal Processing Conference (EUSIPCO)*, pages 156–160, 2021.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012 alexnet. *Adv. Neural Inf. Process. Syst*, pages 1–9, 2012.

[9] Arun Kumar T.K., R. Vinayakumar, Sajith Variyar V.V., V. Sowmya, and K.P. Soman. Convolutional neural networks for fingerprint liveness detection system. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 243–246, 2019.

[10] 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.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] CR Lekshmi and Rajeev Rajan. Predominant instrument recognition in polyphonic music using convolutional recurrent neural networks. In *International Symposium on Computer Music Multidisciplinary Research*, pages 214–227. Springer, 2021.

[13] CR 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.

[14] 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.

[15] X. Li, K. Wang, J. Soraghan, and J. Ren. Fusion of hilbert-huang transform and deep convolutional neural network for predominant musical instruments recognition. *in Proc. of 9th Int. Conf. on Artificial Intelligence in Music, Sound, Art and Design*, 2020.

[16] 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.

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

[18] Mredulraj S. Pandianchery, V. Sowmya, E. A. Gopalakrishnan, Vinayakumar Ravi, and K. P. Soman. Centralized cnn–gru model by federated learning for covid-19 prediction in india. *IEEE Transactions on Computational Social Systems*, 11(1):1362–1371, 2024.

[19] J. Pons, O. Slizovskaia, R. Gong, Emilia Gómez, and X. Serra. Timbre analysis of music audio signals with convolutional neural networks. *in Proc. of 2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2744–2748, 2017.

[20] LC Reghunath and R Rajan. Attention-based predominant instruments recognition in polyphonic music. In *Proceedings of 18th Sound and Music Computing Conference (SMC)*, pages 199–206, 2021.

[21] 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*, 2022(1):11, 2022.

[22] Lekshmi Chandrika Reghunath and Rajeev Rajan. Predominant audio source separation in polyphonic music. *EURASIP Journal on Audio, Speech, and Music Processing*, 2023(1):49, 2023.

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

[24] G. Siddharth, C. Summers, and A. Lerch. Instrument activity detection in polyphonic music using deep neural networks. *in Proc. of Int. Society for Music Information Retrieval Conference (ISMIR)*, 2018.

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

[26] 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.

[27] Dongyan Yu, Huiping Duan, Jun Fang, and Bing Zeng. Predominant instrument recognition based on deep neural network with auxiliary classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:852–861, 2020.